Relations between abstract and probabilistic models of cryptographic protocols

Véronique Cortier
Join work with Bogdan Warinschi

Loria, CNRS & INRIA project CASSIS, Nancy, France
Context: cryptographic protocols

- **Widely used**: web (SSH, SSL, ...), pay-per-view, electronic purse, mobile phone, ...

- **Should ensure**: confidentiality, authenticity, integrity, anonymity, ...

Workshop Nancy-Saarbrücken - October, 14th 2005
Context: cryptographic protocols

- **Widely used:** web (SSH, SSL, ...), pay-per-view, electronic purse, mobile phone, ...

- **Should ensure:** confidentiality, authenticity, integrity, anonymity, ...

- **Presence of an attacker**
  - may **read** every message sent on the net,
  - may **intercept and send** new messages.
Difficulties of the verification

Protocols are run in a hostile network.

The hostile network is modeled by an intruder that:

- intercepts all messages,
- synthesizes new messages, doing arbitrary computations
- sends fake messages.
Difficulties of the verification

Protocols are run in a hostile network.

The hostile network is modeled by an intruder that:

- intercepts all messages,
- synthesizes new messages, doing arbitrary computations,
- sends fake messages.

Protocol

- unbounded number of agents,
- unbounded number of sessions,
- unbounded depth of messages.
1. Formal approaches

2. Link between formal approaches and cryptanalysis
   (a) messages
   (b) adversary
   (c) security properties
   (d) main result
Formal approaches

- Messages are abstracted using terms. These terms are build over a fixed signature. E.g., $\Sigma = \{<>, \text{enc}, \text{dec}, \ldots\}$. 

This approach allows to detect any logical attack that does not rely on weaknesses of the encryption algorithm.
Formal approaches

- Messages are abstracted using terms. These terms are build over a fixed signature. E.g., \( \Sigma = \{<>, \text{enc}, \text{dec}, \ldots\} \).

- The attacker can do symbolic manipulations on terms.

\[
\begin{align*}
\neg I(x) \lor \neg I(y) & \lor I(\langle x, y \rangle) \quad (1) \\
\neg I(x) \lor \neg I(y) & \lor I(\{x\}_y) \quad (2) \\
\neg I(\{x\}_y) \lor \neg I(y) & \lor I(x) \quad (3)
\end{align*}
\]
Formal approaches

- Messages are abstracted using terms. These terms are build over a fixed signature. E.g., $\Sigma = \{<>, \text{enc}, \text{dec}, \ldots\}$.

- The attacker can do symbolic manipulations on terms.

\begin{align*}
- I(x) \lor -I(y) \lor I(\langle x, y \rangle) \\
- I(x) \lor -I(y) \lor I(\{x\}_y) \\
- I(\{x\}_y) \lor -I(y) \lor I(x)
\end{align*}

This approach allows to detect any logical attack that does not rely on weaknesses of the encryption algorithm.
Protocol description

For the protocol

\[
\begin{align*}
A \Rightarrow S : & \quad A, \{B, K_{ab}\}_{K_{as}} \\
S \Rightarrow B : & \quad \{N_s, A, K_{ab}\}_{K_{bs}}
\end{align*}
\]

The corresponding clauses are:

\[
I(\langle a, \{b, k(a, b)\}_{k(a,s)} \rangle) \\
-I(\langle a, \{b, x\}_{k(a,s)} \rangle) \lor I(\{n(s, b), a, x\}_{k(b,s)})
\]

(3) \quad (4)
Decidability and complexity results

- In general, secrecy preservation is undecidable.

- For a bounded number of sessions, secrecy is co-NP-complete [RusinowitchTuruani CSFW01]
  \[\rightarrow\] constraint solving

- For an unbounded number of sessions
  - for one-copy protocols, secrecy is DEXPTIME-complete [CortierComon RTA03] [SeildVerma LPAR04]
    \[\rightarrow\] tree automata, resolution theorem proving
  - for message-length bounded protocols, secrecy is DEXPTIME-complete [Durgin et al FMSP99] [Chevalier et al CSL03]
The Avispa Platform: www.avispa-project.org
Results

- over 80 protocols analyzed (selected by Siemens and discussed by the IETF) in few minutes or few seconds for most of them
- tools for both a bounded number of sessions (search for attacks) and an unbounded number of sessions (security proof)
- first tool that allows algebraic properties (XOR)
- new attacks have been discovered
- publicly available: web interface, download, protocol library, ...
- already used by 45 sites including several companies (France Telecom, Siemens, SAP,...)
1. Formal approaches

2. Link between formal approaches and cryptanalysis
   (a) messages
   (b) adversary
   (c) security properties
   (d) main result
Formal model: several abstractions

Messages are modeled by terms.

- \( \{m\}_k \): message \( m \) encrypted by \( k \)
- \( \langle m_1, m_2 \rangle \): pair of \( m_1 \) and \( m_2 \)
- ...

→ no collisions:

\[
\forall m, m', k, k' \quad \{m\}_k \neq \{m'\}_{k'}, \{\{m\}_k\}_k \neq m, \langle m, m' \rangle \neq \{m\}_k, \ldots
\]
Formal model: several abstractions

Messages are modeled by terms.

- \(\{m\}_k\): message \(m\) encrypted by \(k\)
- \(\langle m_1, m_2 \rangle\): pair of \(m_1\) and \(m_2\)
- ...

\[\forall m, m', k, k' \quad \{m\}_k \neq \{m'\}_{k'}, \{\{m\}_k\}_k \neq m, \langle m, m' \rangle \neq \{m\}_k, \ldots\]

Perfect encryption assumption:

Nothing can be learned from \(\{m\}_k\) except if \(k\) is known.

\[\rightarrow\] The intruder can perform only specific actions like pairing and encrypting messages or decrypting whenever he has the inverse key.
Example: message deduction

In the formal model \( \{\{n\}_k, \{n, n\}_k, \{n, n, n\}_k\} \not\models n \).

Does this imply \( \{0110, 10101101, 111001100101\} \not\models 1010 \) in the concrete model?
Adversary

Formal model:
- intercepts all messages,
- synthesizes messages, doing symbolic manipulations on terms,
- sends fake messages.

Concrete model:
- intercepts all messages,
- synthesizes messages, doing any polynomial computations,
- sends fake messages.
Adversary

**Formal model:**
- intercepts all messages,
- synthesizes messages, doing *symbolic manipulations on terms*
- sends fake messages.

**Concrete model:**
- intercepts all messages,
- synthesizes messages, doing *any polynomial computations*
- sends fake messages.
Secrecy Properties

Formal models: property on traces

A data $s$ is secret if the adversary cannot produce $s$.

\[
\mathcal{P} \cup \{-I(s)\} \not\models \bot
\]
Secrecy Properties

Formal models: property on traces

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

$$
\mathcal{P} \cup \{\neg I(s)\} \not\vdash \bot
$$

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 $\Pr[\text{Exp}^1 = 1] - \Pr[\text{Exp}^0 = 1]$ is a negligible function.
Comparison of the two main approaches:

<table>
<thead>
<tr>
<th></th>
<th>Formal approach</th>
<th>Concrete approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages</td>
<td>terms</td>
<td>bitstrings</td>
</tr>
<tr>
<td>Encryption</td>
<td>idealized</td>
<td>algorithm</td>
</tr>
<tr>
<td>Adversary</td>
<td>idealized</td>
<td>any polynomial algorithm</td>
</tr>
<tr>
<td>Proof</td>
<td>automatic</td>
<td>by hand, tedious and error-prone</td>
</tr>
</tbody>
</table>

Link between the two approaches?
Main Result

The perfect public key encryption corresponds to the IND-CCA2 security notion

Theorem: [Micciancio-Warinschi TCC’04, Cortier-Warinschi ESOP’05]

- for protocols with public key encryption and signatures
- if a protocol is secure in the formal approach (proof given by a tool for example),
- if the public key encryption algorithm is IND-CCA2,
- if the signature scheme is existentially unforgeable,

then the protocol is secure in the concrete approach.
Formal Model

- agents $A_i, a_i$
- nonces (random numbers) $X^j_{A_i}, n(a_i, j, s)$
- pairing $\langle m_1, m_2 \rangle$
- asymmetric encryption $\{m\}_{ek(a)}$
- signature $[m]_{sk(a)}$
Formal Model

- agents $A_i, a_i$
- nonces (random numbers) $X^{j}_{A_i}, n(a_i, j, s)$
- pairing $\langle m_1, m_2 \rangle$
- asymmetric encryption $\{m\}_{ek(a)}$
- signature $[m]_{sk(a)}$

Rules of the form $M_1 \rightarrow M_2$

A protocol is a finite set of roles.

$$\text{Roles} = (M_1, M_2)(M_3, M_4) \cdots (M_k, M_{k+1})$$
Example

Needham-Schroeder-Lowe protocol

\[ A \rightarrow B : \{ N_a, A \}_{ek(B)} \]

\[ B \rightarrow A : \{ N_a, N_b, B \}_{ek(A)} \]

\[ A \rightarrow B : \{ N_b \}_{ek(B)} \]
Example

Needham-Schroeder-Lowe protocol

\[
\begin{align*}
A \rightarrow B : & \quad \{N_a, A\}_{ek(B)} \\
B \rightarrow A : & \quad \{N_a, N_b, B\}_{ek(A)} \\
A \rightarrow B : & \quad \{N_b\}_{ek(B)} \\
\end{align*}
\]

\[
\begin{align*}
\Pi(1) & = (\text{init} \rightarrow \{X^1_{A_1}, A_1\}_{ek(A_2)}), \\
& \quad (\{X^1_{A_1}, X^1_{A_2}, A_2\}_{ek(A_1)} \rightarrow \{X^1_{A_2}\}_{ek(A_2)}) \\
\Pi(2) & = (\{X^1_{A_1}, A_1\}_{ek(A_2)} \rightarrow \{X^1_{A_1}, X^1_{A_2}, A_2\}_{ek(A_1)}), \\
& \quad (\{X^1_{A_2}\}_{ek(A_2)} \rightarrow \text{stop})
\end{align*}
\]

Variables are local to each role and each session.
Adversary

\[ \text{corrupt}(a_1, \ldots, a_l) \]

private keys of \( a_1, \ldots, a_l \)

\[ \text{new}(i, a_1, \ldots, a_k) \]

\[ \text{sid} = (s, i, (a_1, \ldots, a_k)) \]

Protocol

\[ \text{send}(\text{sid}, m) \]

\( m' \)
Formal Intruder Deduction Rules

\[
\frac{S \vdash m_1 \quad S \vdash m_2}{S \vdash \langle m_1, m_2 \rangle}
\]

\[
\frac{S \vdash \langle m_1, m_2 \rangle}{S \vdash m_i} \quad i \in \{1, 2\}
\]
Formal Intruder Deduction Rules

\[
\begin{align*}
S \vdash m_1 & \quad S \vdash m_2 \\
\frac{}{S \vdash \langle m_1 , m_2 \rangle} & \\
S \vdash e_k(b) & \quad S \vdash m \\
\frac{}{S \vdash \{m\}_{ek(b)} } & \\
S \vdash \langle m_1 , m_2 \rangle & \quad i \in \{1, 2\} \\
\frac{}{S \vdash m_i} & \\
S \vdash \{m\}_{ek(b)} & \quad S \vdash dk(b) \\
\frac{}{S \vdash m} & \\
\end{align*}
\]
**Formal Intruder Deduction Rules**

\[
\begin{align*}
S \vdash \langle m_1, m_2 \rangle & \quad S \vdash \langle m_1, m_2 \rangle & \quad S \vdash \langle m_1, m_2 \rangle & \quad i \in \{1, 2\} \\
S \vdash m_1 & \quad S \vdash m_2 & \quad S \vdash m_i & \quad i \in \mathbb{N} \\
S \vdash \{m\}_{ek(b)} & \quad S \vdash \{m\}_{ek(b)} & \quad \frac{S \vdash \{m\}_{ek(b)} \quad S \vdash dk(b)}{S \vdash m} & \quad S \vdash m_i \\
S \vdash sk(b) & \quad S \vdash \{m\}_{ek(b)} & \quad \frac{S \vdash \{m\}_{ek(b)} \quad S \vdash sk(b)}{S \vdash [m]_{sk(b)} \quad S \vdash m} & \quad i \in \mathbb{N} \\
S \vdash [m]_{sk(b)} & \quad S \vdash m & \quad S \vdash m
\end{align*}
\]
Hypotheses on the Implementation

- asymmetric encryption: IND-CCA2
  → 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

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

- signature: existentially unforgeable under chosen-message attack
  i.e. one can not produce a valid pair \( (m, \sigma) \)
Hypotheses on the Implementation

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

- signature: existentially unforgeable under chosen-message attack
  i.e. one cannot produce a valid pair \((m, \sigma)\)

- parsing:
  - each bit-string has a label which indicates its type (identity,
    nonce, key, signature, ...)
  - one can retrieve the (public) encryption key from an
    encrypted message.
  - one can retrieve the signed message from the signature
Trace properties

**State:** assignations of the local variables

**Trace:** sequence of states

**SymbTr:** Set of symbolic traces

**ConcTr:** Set of concrete traces
Trace properties

State: assignations of the local variables
Trace: sequence of states
SymbTr: Set of symbolic traces
ConcTr: Set of concrete traces

Trace properties

- A formal trace property is any subset $P^s \subseteq \text{SymbTr}$.  
- A concrete trace property is any subset $P^c \subseteq \text{ConcTr}$.  

Trace properties

State: assignments of the local variables
Trace: sequence of states
SymbTr: Set of symbolic traces
ConcTr: Set of concrete traces

Trace properties

- A formal trace property is any subset $P^s \subseteq \text{SymbTr}$.
- A concrete trace property is any subset $P^c \subseteq \text{ConcTr}$.

Satisfaction

- A protocol $\Pi$ satisfies symbolically $P^s$, denoted by $\Pi \models^s P^s$, if any valid formal trace $t^s \in P^s$.
- A protocol $\Pi$ satisfies concretely $P^c$, denoted by $\Pi \models^c P^c$, if any valid concrete trace $t^c \in P^c$, with overwhelming probability.
Soundness of trace properties

Theorem 1 :

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

Theorem 1:
Every concrete trace is the image of a valid formal trace, except with negligible probability.

Corollary 1:
Then $\Pi \models^s P^s$ implies $\Pi \models^c P^c$.

Applications: trace properties like authentication, ...
Soundness of trace properties

Theorem 1:

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

Corollary 1:

Then $\Pi \models^s P^s$ implies $\Pi \models^c P^c$.

Applications: trace properties like authentication, ...

Corollary 2:

If a protocol $\Pi$ preserves the secrecy of a nonce $N$ against an ideal adversary, then $\Pi(N \leftrightarrow n_0)$ and $\Pi(N \leftrightarrow n_1)$ are indistinguishable against any polynomial adversary.
Proof idea

**Key result:** every concrete trace is the image of a valid formal trace, except with negligible probability.

\[
\text{init}(1, a, b) \rightarrow \{a, n_a\}_K \quad \{n_a\}_K \text{non valid!}
\]

\[A : \text{init}(1, a, b) \quad m_1 \rightarrow \text{send}(m_2)\]
Proof idea

Key result: every concrete trace is the image of a valid formal trace, except with negligible probability.

\[
\text{init}(1, a, b) \rightarrow \{a, n_a\}_{K_b} \quad \{n_a\}_{K_b} \text{non valid!}
\]

\[
A : \quad \begin{array}{c}
\text{init}(1, a, b) \\
\uparrow
\end{array} \rightarrow \\
\begin{array}{c}
m_1 \\
\downarrow
\end{array} \rightarrow \\
\begin{array}{c}
\text{send}(m_2) \\
\uparrow
\end{array}
\]

Using the adversary \(A\), we build an adversary \(A'\) that breaks encryption.

\[
A' : \quad \langle\langle a, n_a^0\rangle, \langle a, n_a^1\rangle\rangle \rightarrow \begin{array}{c}
\text{encryption} \\
\text{oracle}
\end{array} \rightarrow \{a, n_a^\alpha\}_{K_b}
\]
Proof idea

Key result: every concrete trace is the image of a valid formal trace, except with negligible probability.

\[
\text{init}(1, a, b) \rightarrow \{a, n_a\} K_b \quad \text{\{\text{valid\}}}
\]

Using the adversary \( A \), we build an adversary \( A' \) that breaks encryption.

\[
A' : \quad (\langle a, n_a^0 \rangle, \langle a, n_a^1 \rangle) \rightarrow \text{encryption oracle} \rightarrow \{a, n_a^{\alpha}\} K_b
\]

\[
\rightarrow A \rightarrow \{n_a^{\alpha}\} K_b
\]
**Proof idea**

**Key result:** every concrete trace is the image of a valid formal trace, except with negligible probability.

\[
\begin{align*}
\text{init}(1, a, b) & \rightarrow \{a, n_a\} K_b & & \{n_a\} K_b \text{ non valid!} \\
\uparrow & \downarrow & & \uparrow \\
A : \text{init}(1, a, b) & \rightarrow m_1 & & \rightarrow \text{send}(m_2)
\end{align*}
\]

Using the adversary \(A\), we build an adversary \(A'\) that breaks encryption.

\[
\begin{align*}
A' : (\langle a, n_a^0 \rangle, \langle a, n_a^1 \rangle) & \rightarrow \text{encryption oracle} & \rightarrow \{a, n_a^\alpha\} K_b \\
\rightarrow A & \rightarrow \{n_a^\alpha\} K_b & \rightarrow \text{decryption oracle} & \rightarrow n_a^\alpha & \rightarrow \alpha
\end{align*}
\]
Related Work

Backes-Pfitzmann

- very general results: symmetric and asymmetric encryption, pairing, signatures, MACs.
- less abstracted model than classical Dolev-Yao models,
- automatic verification have to be developed specifically for this model.
Related Work

Backes-Pfizmann

- very general results: symmetric and asymmetric encryption, pairing, signatures, MACs.
- less abstracted model than classical Dolev-Yao models,
- automatic verification have to be developed specifically for this model.

Laud: symmetric encryption, bounded number of sessions

Datta-Derek-Mitchell-Shmatikov-Turuani: direct automatization of the proof in the concrete model
Conclusion

Applications:
Automatic computationally sound proof for secrecy and authentication properties, using for example the AVISPA platform (Cassis project)
Conclusion

Applications:
Automatic computationally sound proof for secrecy and authentication properties, using for example the AVISPA platform (Cassis project)

Extensions:

Cryptographic Primitives
- symmetric encryption, macs, hash, ...
- algebraic properties (exclusive or, modular exponentiation)

Properties
- secrecy of a key, of composed terms, of a vote, ...
- contract signature?

Cryptographic assumptions: what happens if the encryption scheme is IND-CPA for example?