Safely composing security protocols via tagging

Stéphanie Delaune

LSV, ENS Cachan & CNRS & INRIA project SECSI

March, 14, 2008

$\longrightarrow$ joint work with Véronique Cortier, Jérémie Delaitre, Myrto Arapinis and Steve Kremer

Context: cryptographic protocols

Paiement Internet

Cryptographic protocols

- small programs designed to secure communication (e.g. secrecy)
- use cryptographic primitives (e.g. encryption, signature, ...)

Context: cryptographic protocols

Paiement Internet

Cryptographic protocols

- small programs designed to secure communication (e.g. secrecy)
- use cryptographic primitives (e.g. encryption, signature, ...)

The network is unsecure!

Communications take place over a public network like the Internet.

Cryptographic protocols (symbolic approach)

Messages are abstracted by terms

- pairing $\langle \mathsf{m}_1, \mathsf{m}_2 \rangle$,
- symmetric $\mathsf{enc}(\mathsf{m}, \mathsf{k})$ and public key encryption $\mathsf{enca}(\mathsf{m}, \mathsf{pub}(\mathsf{A}))$,
- signature $\mathsf{sign}(\mathsf{m}, \mathsf{priv}(\mathsf{A}))$.

Cryptographic protocols (symbolic approach)

Messages are abstracted by terms

- pairing $\langle m_1, m_2 \rangle$,
- symmetric $\mathsf{enc}(m, k)$ and public key encryption $\mathsf{enca}(m, \mathsf{pub}(A))$,
- signature $\mathsf{sign}(m, \mathsf{priv}(A))$.

Presence of an idealized attacker

- may read, intercept and send messages,
- may build new messages following deduction rules (symbolic manipulation on terms).

Cryptographic protocols (symbolic approach)

Messages are abstracted by terms

- pairing $\langle m_1, m_2 \rangle$,
- symmetric $\text{enc}(m, k)$ and public key encryption $\text{enca}(m, \text{pub}(A))$,
- signature $\text{sign}(m, \text{priv}(A))$.

Presence of an idealized attacker

- may read, intercept and send messages,
- may build new messages following deduction rules (symbolic manipulation on terms).

Examples:

$$\frac{m \quad k}{\text{enc}(m, k)} \qquad \frac{\text{enc}(m, k) \quad k}{m} \qquad \frac{\text{enca}(m, \text{pub}(a)) \quad \text{priv}(a)}{m}$$

Motivations

Formal verification of security protocols

- Existing tools allow us to verify relatively small protocols and sometimes only for a bounded number of sessions
- Most often, we verify them in isolation
  $\longrightarrow$ this is not sufficient

Motivations

Formal verification of security protocols

- Existing tools allow us to verify relatively small protocols and sometimes only for a bounded number of sessions
- Most often, we verify them in isolation
  $\longrightarrow$ this is not sufficient

Example:

$$P_1 : \quad A \rightarrow B : \quad \mathrm{enca}(s, \mathrm{pub}(B))$$

Question: What about the secrecy of s?

Motivations

Formal verification of security protocols

- Existing tools allow us to verify relatively small protocols and sometimes only for a bounded number of sessions
- Most often, we verify them in isolation
  $\longrightarrow$ this is not sufficient

Example:

$P_1: \quad A \rightarrow B: \quad \text{enca}(s, \text{pub}(B))$

$$P_2: \quad \begin{array}{ll} A \rightarrow B: & \text{enca}(N_a, \text{pub}(B)) \\ B \rightarrow A: & N_a \end{array}$$

Question: What about the secrecy of s?

Motivations

Formal verification of security protocols

- Existing tools allow us to verify relatively small protocols and sometimes only for a bounded number of sessions
- Most often, we verify them in isolation
  $\longrightarrow$ this is not sufficient

Our goal

investigate sufficient conditions to ensure that protocols can be safely used in an environment where:

1. other sessions of the same protocol may be executed;
2. other sessions of another protocol may be executed as well.

$\longrightarrow$ protocols may share identities and keys (e.g. public keys, long-term symmetric keys)

Outline of the talk

Outline of the talk

1. Introduction
2. Composition result I: "from one session to many"
3. Composition result II: "from one protocol to many"
4. Conclusion

Summary: "from one session to many"

Our goal:

compose different sessions from the same protocol

⟶ well-known fact: an attack may involve an arbitrary number of sessions

Summary: "from one session to many"

Our goal:

compose different sessions from the same protocol

$\longrightarrow$ well-known fact: an attack may involve an arbitrary number of sessions

Solution

- a transformation which maps a protocol P that is secure for a single session to a protocol $\overline{P}$ that is secure for an unbounded number of sessions.
- side-effect: an effective strategy to design secure protocols

Our transformation

Let P be a protocol with ℓ participants as given below:

$$
\begin{array}{rcl}
A_{i_1} \rightarrow A_{j_1} & : & m_1 \\
A_{i_2} \rightarrow A_{j_2} & : & m_2 \\
& \vdots & \\
A_{i_k} \rightarrow A_{j_k} & : & m_k
\end{array}
$$

Our transformation

The protocol $\overline{P}$ (with ℓ participants) is decribed below:

Initialisation phase: broadcast of fresh nonces

$$\begin{array}{rl} A_1 \rightarrow All : & A_1, N_1 \\ A_2 \rightarrow All : & A_2, N_2 \\ \vdots & \\ A_\ell \rightarrow All : & A_\ell, N_\ell \end{array}$$

Our transformation

The protocol $\overline{P}$ (with ℓ participants) is decribed below:

Initialisation phase: broadcast of fresh nonces

$$\begin{array}{rl} A_1 \rightarrow All: & A_1, N_1 \\ A_2 \rightarrow All: & A_2, N_2 \\ & \vdots \\ A_\ell \rightarrow All: & A_\ell, N_\ell \end{array}$$

Every particicpant obtain a tag $= \langle A_1, N_1, A_2, N_2, \ldots, A_\ell, N_\ell \rangle$

Our transformation

The protocol $\overline{P}$ (with ℓ participants) is decribed below:

Initialisation phase: broadcast of fresh nonces

$$\begin{array}{rl} A_1 \rightarrow All: & A_1, N_1 \\ A_2 \rightarrow All: & A_2, N_2 \\ \vdots & \\ A_\ell \rightarrow All: & A_\ell, N_\ell \end{array}$$

Every particicpant obtain a tag $= \langle A_1, N_1, A_2, N_2, \ldots, A_\ell, N_\ell \rangle$

Main phase:

$$\begin{array}{rl} A_{i_1} \rightarrow A_{j_1}: & \overline{m_1} \\ A_{i_2} \rightarrow A_{j_2}: & \overline{m_2} \\ \vdots & \\ A_{i_k} \rightarrow A_{j_k}: & \overline{m_k} \end{array}$$

where the function $\overline{m}$ is defined by:

$$\begin{cases} \overline{\langle u_1, u_2 \rangle} \rightarrow \langle \overline{u_1}, \overline{u_2} \rangle \\ \overline{f(u_1, u_2)} \rightarrow f(\langle \text{tag}, \overline{u_1} \rangle, \overline{u_2}) \quad \text{when } f \in \{\text{enc}, \text{enca}, \text{sign}\} \\ \overline{u} \rightarrow u \quad \text{otherwise} \end{cases}$$

Composition result "from one session to many"

Theorem

Let P be a protocol with no critical long-term keys in plaintext position.

If P preserves the secrecy of s for a single honest session of each role then $\overline{P}$ preserves the secrecy of s for an unbounded number of sessions.

- critical long-term keys do not appear in plaintext
  $\longrightarrow$ this can be easily checked on the finite specification of the protcol
  $\longrightarrow$ often satisfied since it is considered as a prudent practice
- single honest session of each role
  $\longrightarrow$ i.e. one an instance of each role (in general 2 or 3);
  $\longrightarrow$ participants engaged in this session are honest.

Composition result "from one session to many"

Theorem

Let P be a protocol with no critical long-term keys in plaintext position.

If P preserves the secrecy of s for a single honest session of each role then $\overline{P}$ preserves the secrecy of s for an unbounded number of sessions.

- critical long-term keys do not appear in plaintext
  $\longrightarrow$ this can be easily checked on the finite specification of the protcol
  $\longrightarrow$ often satisfied since it is considered as a prudent practice
- single honest session of each role
  $\longrightarrow$ i.e. one an instance of each role (in general 2 or 3);
  $\longrightarrow$ participants engaged in this session are honest.

Exemple: Needham-Schroeder public key protocol
$\longrightarrow$ the Lowe's famous man-in-the-middle attack is prevented

Related work

Computational models

Several compilers already exist in the area of cryptographic design, e.g.

- *Scalable protocols for authenticated group key exchange* [Katz & Yung, 03]

Symbolic models

- *Synthesizing secure protocols* [Cortier *et al.*, 07]
- *How to guarantee secrecy for cryptographic protocols* [Beauquier & Gauche, 07]

$\longrightarrow$ the transformations make heavy use of cryptography

Outline of the talk

Summary: "from one protocol to many"

Our goal:

compose sessions coming from different protocols

Solution

we propose sufficient and rather tight conditions for a protocol to be safely used in an environment where other protocols may be executed as well;

Summary: "from one protocol to many"

Our goal:
compose sessions coming from different protocols

Solution
we propose sufficient and rather tight conditions for a protocol to be safely used in an environment where other protocols may be executed as well;

Example: (given in introduction)

$$P_1: \quad A \rightarrow B: \quad \mathrm{enca}(s, \mathrm{pub}(B))$$

$$P_2: \quad A \rightarrow B: \quad \mathrm{enca}(N_a, \mathrm{pub}(B)) \qquad B \rightarrow A: \quad N_a$$

$\longrightarrow$ protocols may share identities and keys (*e.g.* public keys, long-term symmetric keys)

Main condition - Tagging

Well-tagged protocol

Each protocol is given an identifier (*e.g.* the protocol's name). This identifier has to appear in any encrypted and signed message.

$\longrightarrow$ this tagging policy will avoid interaction between two differents protocols.

Main condition - Tagging

Well-tagged protocol

Each protocol is given an identifier (*e.g.* the protocol's name). This identifier has to appear in any encrypted and signed message.

$\longrightarrow$ this tagging policy will avoid interaction between two differents protocols.

Example: P_1 is 1-tagged whereas P_2 is 2-tagged

Protocol P_1

$$A \to B : \mathrm{enca}(\langle 1, s\rangle, \mathrm{pub}(B))$$

Protocol P_2

$$A \to B : \mathrm{enca}(\langle 2, N_a\rangle, \mathrm{pub}(B))$$
$$B \to A : N_a$$

Composition result "from one protocol to many"

Theorem

Let P_1 and P_2 be two well-tagged protocols such that

- *no critical long-term keys appear in plaintext position neither in P_1 nor in P_2,*
- *P_1 is α-tagged and P_2 is β-tagged with $\alpha \neq \beta$.*

If P_1 preserves the secrecy of s then $P_1 \mid P_2$ preserves the secrecy of s.

Composition result "from one protocol to many"

Theorem

Let P_1 and P_2 be two well-tagged protocols such that

- *no critical long-term keys appear in plaintext position neither in P_1 nor in P_2,*
- *P_1 is α-tagged and P_2 is β-tagged with $\alpha \neq \beta$.*

If P_1 preserves the secrecy of s then $P_1 \mid P_2$ preserves the secrecy of s.

Extensions that have been already done:

1. well-tagged condition can be relaxed: disjoint encryption is actually sufficient;
2. composition result holds for a class of security properties (secrecy, authentication, . . .)

Related work

The idea of adding an identifier is not novel:

Principle 10 in the prudent engineering paper

[Abadi & Needham, 95]

Related work

The idea of adding an identifier is not novel:

Principle 10 in the prudent engineering paper

[Abadi & Needham, 95]

There are also some formal results about this problem:

- *Protocol independence through disjoint encryption*

  [Guttman & Thayer, 00]

  ⟶ their condition has to hold on any valid execution of the protocol

- *Sufficient conditions for composing security protocols*

  [Andova *et al.*, 07]

  ⟶ they have to assume typing hypothesis, they can not deal with protocols with ciphertext forwarding

Outline of the talk

Conclusion

Two composition results

1. one that is useful to compose sessions coming from the same protocol
   $\longrightarrow$ this can be obtained with dynamic tags
2. one that can be used to compose protocols that satisfy disjoint encryption
   $\longrightarrow$ this can be obtained with static tags

$\longrightarrow$ to combine both results, use tag $= \langle id_\alpha, A_1, N_1, \ldots, A_\ell, N_\ell \rangle$.

Conclusion

Two composition results

1. one that is useful to compose sessions coming from the same protocol
   $\longrightarrow$ this can be obtained with dynamic tags
2. one that can be used to compose protocols that satisfy disjoint encryption
   $\longrightarrow$ this can be obtained with static tags

$\longrightarrow$ to combine both results, use tag $= \langle id_\alpha, A_1, N_1, \ldots, A_\ell, N_\ell \rangle$.

Future Work

- obtain a more fine-grained characterization of a decidable class (for an unbounded number of sessions and a class security properties)
- other kind of security properties (e.g. equivalence-based properties)
- other kind of composition (e.g. sequence)