How to protect security protocols against active attackers

Véronique Cortier

Loria, CNRS

January 16th, 2007

Joint work with Bogdan Warinschi and Eugen Zălinescu

Some other results

1. Implementation of an Avispa module for computational security guarantees
2. Decidability of Key cycles

Implementation in Avispa [Heinrich Hördegen]

Goal : Use the soundness result in the case of asymmetric encryption and signatures (and also hashes in the ROM).

Theorem

If a protocol is secure in the formal model, it is secure in the computational one.
→ For trace properties (like authentication) and secrecy interpreted as indistinguishability

Implementation in Avispa [Heinrich Hördegen]

Goal : Use the soundness result in the case of asymmetric encryption and signatures (and also hashes in the ROM).

Problem :

	Our Formal Model	Simplified Formal Model (Avispa)
Encryption	$\{m\}'_{\mathsf{pk}(a)}$	$\{m\}_{\mathsf{pk}(a)}$
Signatures	$[m]_{\mathsf{pk}(a)}$	$\{m\}_{\mathsf{dk}(a)}$
hash	$h(M)$	$\{m\}_{\mathsf{pk}(h)}$

Implementation in Avispa [Heinrich Hördegen]

Goal : Use the soundness result in the case of asymmetric encryption and signatures (and also hashes in the ROM).

Problem :

	Our Formal Model	Simplified Formal Model (Avispa)
Encryption	$\{m\}'_{\mathsf{pk}(a)}$	$\{m\}_{\mathsf{pk}(a)}$
Signatures	$[m]_{\mathsf{pk}(a)}$	$\{m\}_{\mathsf{dk}(a)}$
hash	$h(M)$	$\{m\}_{\mathsf{pk}(h)}$

Proposition : P secure in the simplified model $\Rightarrow$ P secure in our model :

- for secrecy
- for authentication on atomic data (nonces, keys, ...)

Implementation in Avispa [Heinrich Hördegen]

Goal : Use the soundness result in the case of asymmetric encryption and signatures (and also hashes in the ROM).

Problem :

	Our Formal Model	Simplified Formal Model (Avispa)
Encryption	$\{m\}^{I}_{\mathsf{pk}(a)}$	$\{m\}_{\mathsf{pk}(a)}$
Signatures	$[m]_{\mathsf{pk}(a)}$	$\{m\}_{\mathsf{dk}(a)}$
hash	$h(M)$	$\{m\}_{\mathsf{pk}(h)}$

Proposition : P secure in the simplified model $\Rightarrow$ P secure in our model :

- for secrecy
- for authentication on atomic data (nonces, keys, ...)

$\rightarrow$ Implementation of a module in Avispa that checks whether the soundness result applies.

Deciding Key cycles [Eugen Zălinescu]

Ultimate goal : obtain a soundness result in the presence of symmetric encryption

One usually needs to forbid key cycles (Laud, Backes, Lakhnech, ...)

$$\{k_1\}_{k_2} \quad \{k_2\}_{k_3} \quad \cdots \quad \{k_n\}_{k_1}$$

Deciding Key cycles [Eugen Zălinescu]

Ultimate goal : obtain a soundness result in the presence of symmetric encryption

One usually needs to forbid key cycles (Laud, Backes, Lakhnech, ...)

$$\{k_1\}_{k_2} \quad \{k_2\}_{k_3} \quad \cdots \quad \{k_n\}_{k_1}$$

Theorem

Deciding whether a protocol can produce key cycles on honest keys, for a bounded number of sessions is NP-complete.

Deciding Key cycles [Eugen Zălinescu]

Ultimate goal : obtain a soundness result in the presence of symmetric encryption

One usually needs to forbid key cycles (Laud, Backes, Lakhnech, ...)

$$\{k_1\}_{k_2} \quad \{k_2\}_{k_3} \quad \cdots \quad \{k_n\}_{k_1}$$

Theorem

Deciding whether a protocol can produce key cycles on honest keys, for a bounded number of sessions is NP-complete.

Proof : Constraint system solving using transformation rules (Hubert Comon)

Outline of the talk

Building secure protocols

Instead of verifying existing protocols, we propose a construction to build secure protocols, against active adversary.

Given a protocol P (possibly with no cryptographic primitives)

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

We construct a protocol $\widetilde{P}$, adding nonces, asymmetric encryption and nonces, such that $\widetilde{P}$ is secure against active adversary, for an unbounded number of sessions.

Transformation

P protocol with l participants.

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

Transformation

P protocol with I participants.

First Phase : broadcast of fresh nonces

$$\begin{array}{lcl} A_1 \rightarrow All & : & N_1 \\ A_2 \rightarrow All & : & N_2 \\ \vdots & & \\ A_I \rightarrow All & : & N_k \end{array}$$

Every participant obtain a sessionID $= A_1, A_2, \ldots, A_k, N_1, \ldots, N_k$.

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

Transformation

P protocol with l participants.

First Phase : broadcast of fresh nonces

$$\begin{array}{rcl} A_1 \rightarrow All & : & N_1 \\ A_2 \rightarrow All & : & N_2 \\ \vdots & & \\ A_l \rightarrow All & : & N_k \end{array}$$

Every participant obtain a sessionID $= A_1, A_2, \ldots, A_k, N_1, \ldots, N_k$.

Second Phase : Securing the initial protocol

$$\begin{array}{rcl} A_{i_1} \rightarrow A_{j_1} & : & m_1, [m_1, \text{sessionID}]_{\text{sk}(A_{i_1})} \\ A_{i_2} \rightarrow A_{j_2} & : & m_2, [m_2, \text{sessionID}]_{\text{sk}(A_{i_2})} \\ \vdots & & \\ A_{i_k} \rightarrow A_{j_k} & : & m_k, [m_k, \text{sessionID}]_{\text{sk}(A_{i_k})} \end{array}$$

Transformation

P protocol with I participants.

First Phase : broadcast of fresh nonces

$$\begin{array}{rcl} A_1 \rightarrow All & : & N_1 \\ A_2 \rightarrow All & : & N_2 \\ \vdots & & \\ A_I \rightarrow All & : & N_k \end{array}$$

Every participant obtain a sessionID $= A_1, A_2, \ldots, A_k, N_1, \ldots, N_k$.

Second Phase : Securing the initial protocol

$$\begin{array}{rcl} A_{i_1} \rightarrow A_{j_1} & : & \{m_1, [m_1, \text{sessionID}]_{\text{sk}(A_{i_1})}\}_{\text{pk}(A_{j_1})} \\ A_{i_2} \rightarrow A_{j_2} & : & \{m_2, [m_2, \text{sessionID}]_{\text{sk}(A_{i_2})}\}_{\text{pk}(A_{j_2})} \\ \vdots & & \\ A_{i_k} \rightarrow A_{j_k} & : & \{m_k, [m_k, \text{sessionID}]_{\text{sk}(A_{i_k})}\}_{\text{pk}(A_{j_k})} \end{array}$$

Choice of the model

Model developed by D. Micciancio and B. Warinschi [TCC'04]

- Choice not crucial for the result
- Rather general :
  - Unbounded number of sessions
  - many primitives : pairing, symmetric and asymmetric encryption, signatures, ...

Primitives

Primitives	Free algebra of terms
agents	A_i, a_i
nonces	$X^j_{A_i}, n(a_i, j, s)$
keys	$K^j_{A_i}, k(a_i, j, s)$
pairing	$\langle m_1, m_2 \rangle$
asymmetric encryption	$\{m\}_{\mathsf{ek}(a)}$
symmetric encryption	$\{\!\{m\}\!\}_k$
signature	$[m]_{\mathsf{sk}(a)}$

Protocol

Rules of the form $M_1 \longrightarrow M_2$

A protocol is a finite set of roles.

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

Intended behavior : given by a function *sender* $\mathcal{S}$ that returns for each role r at control point p the role-control point pair $(r', p') = \mathcal{S}(r, p)$ which should have emitted the expected message.

Example

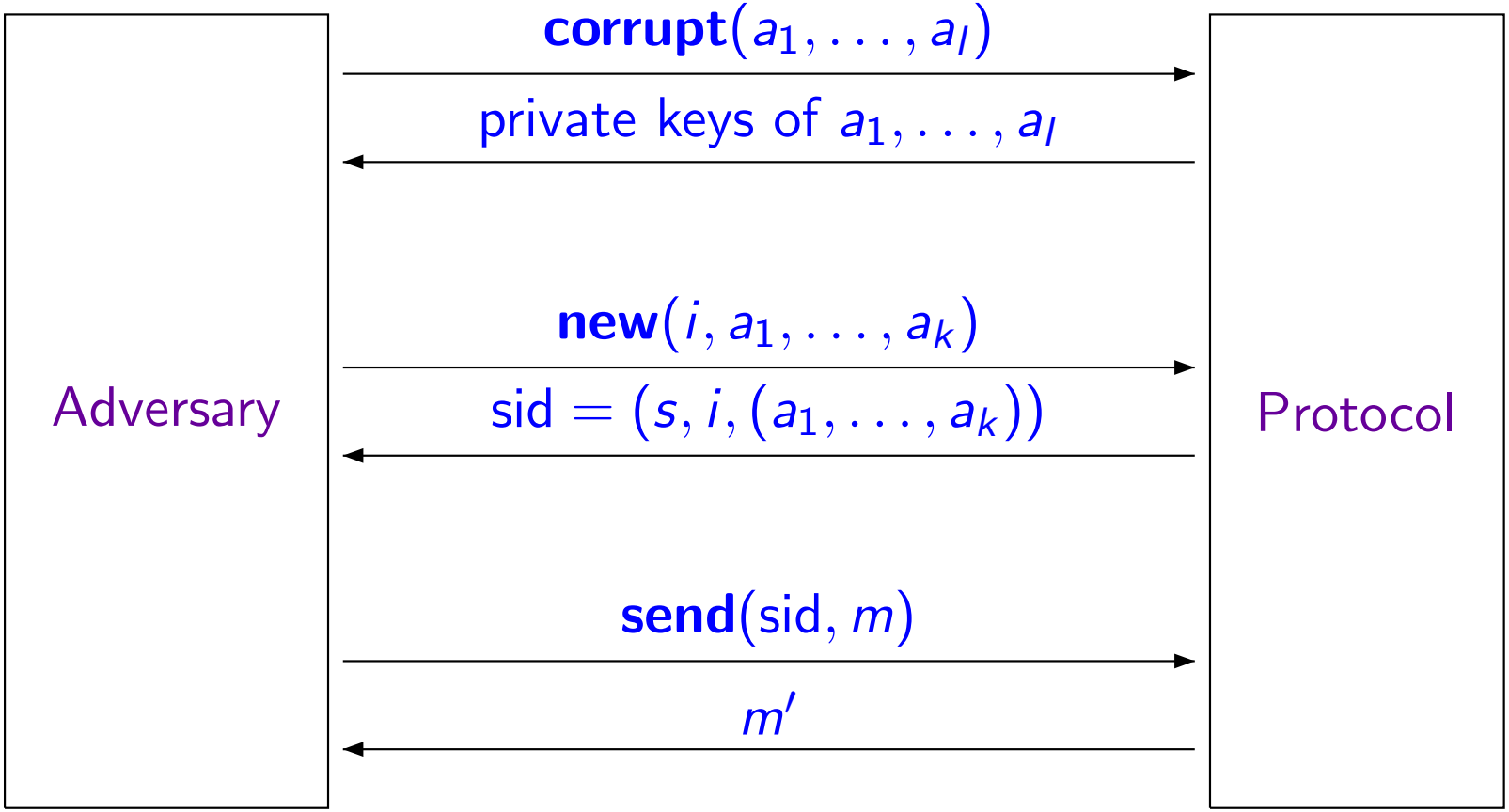
Needham-Schroeder-Lowe protocol

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

$$
\Pi(1) = \begin{array}{rcll} (\mathsf{init} & \to & \{X^1_{A_1}, A_1\}_{\mathsf{pk}(A_2)}) & \mathcal{S}(1,1) = (0,0) \\ (\{X^1_{A_1}, X^1_{A_2}, A_2\}_{\mathsf{pk}(A_1)} & \to & \{X^1_{A_2}\}_{\mathsf{pk}(A_2)}) & \mathcal{S}(1,2) = (2,1) \end{array}
$$

$$
\Pi(2) = \begin{array}{rcll} (\{X^1_{A_1}, A_1\}_{\mathsf{pk}(A_2)} & \to & \{X^1_{A_1}, X^1_{A_2}, A_2\}_{\mathsf{pk}(A_1)}) & \mathcal{S}(2,1) = (1,1) \\ (\{X^1_{A_2}\}_{\mathsf{pk}(A_2)} & \to & \mathsf{stop}) & \mathcal{S}(2,2) = (1,2) \end{array}
$$

Network


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

Formal Intruder Deduction Rules

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

$$\frac{S\vdash \mathrm{pk}(b) \quad S\vdash m}{S\vdash\{m\}_{\mathrm{pk}(b)}} \qquad \frac{S\vdash\{m\}_{\mathrm{pk}(b)} \quad S\vdash \mathrm{dk}(b)}{S\vdash m}$$

$$\frac{S\vdash k \quad S\vdash m}{S\vdash\{\!|m|\!\}_k} \qquad \frac{S\vdash\{\!|m|\!\}_k \quad S\vdash k}{S\vdash m}$$

$$\frac{S\vdash \mathrm{sk}(b) \quad S\vdash m}{S\vdash[m]_{\mathrm{sk}(b)}} \qquad \frac{S\vdash[m]_{\mathrm{sk}(b)}}{S\vdash m}\,\text{(optional)}$$

Traces

Definition (Valid traces)

Any trace where the adversary only sends deducible messages

Corresponds to the usual modeling of traces of protocols against active adversaries.

Definition (Honest traces)

Valid traces where the adversary does not interfere in those sessions where all parties are honest (non corrupted)

Transforming protocols

Given a protocol P, we denote by $\widetilde{P}$ the transformed protocol.
Let $\mathsf{sessionID} = A_1, A_2, \ldots, A_k, N_1, \ldots, N_k$.

$$
\begin{array}{lcl}
A_i \rightarrow All & : & N_i \\
A_{i_1} \rightarrow A_{j_1} & : & \{m_1, [m_1, p_{i_1}, \mathsf{sessionID}]_{\mathsf{sk}'(A_{i_1})}\}_{\mathsf{pk}'(A_{j_1})} \\
A_{i_2} \rightarrow A_{j_2} & : & \{m_2, [m_2, p_{i_2}, \mathsf{sessionID}]_{\mathsf{sk}'(A_{i_2})}\}_{\mathsf{pk}'(A_{j_2})} \\
\vdots & & \\
A_{i_k} \rightarrow A_{j_k} & : & \{m_k, [m_k, p_{i_k}, \mathsf{sessionID}]_{\mathsf{sk}'(A_{i_k})}\}_{\mathsf{pk}'(A_{j_k})}
\end{array}
$$

- N_i fresh nonces, unused in P,
- $\mathsf{pk}'(A_i), \mathsf{sk}'(A_i)$ encryption and signing keys unused in P.

Correspondence properties between P and $\widetilde{P}$

Lemma

Let Π be a protocol and $\widetilde{\Pi}$ the corresponding transformed protocol. In $\widetilde{\Pi}$, any valid execution trace is an honest execution trace.

Consequences :

- It is sufficient to analyze honest execution traces.
- In practice, reasonably designed protocols should satisfies their security properties for honest execution traces.

Proof sketch

- Nonces of each agent ensure freshness
- Signature ensures authentication of the sender
- Encryption ensures that messages m_i cannot be learned and replayed in other (dishonest) sessions.
- Control points ensure that messages cannot be send at wrong steps.

Induction : Assume an agent A_i has started a session with honest agents $A_1, A_2, \ldots, A_k$ and some session id sessionID the nonces. If A receives a message of the form $\{m, [m, p, \mathsf{sessionID}]_{\mathsf{sk}'(B)}\}_{\mathsf{pk}'(A)}$ then it has been sent by the expected agent at the expected session.

Secrecy of honest values

Keys and nonces generated in honest sessions are secret.

Theorem

Let Π be a protocol and $\widetilde{\Pi}$ the corresponding transformed protocol. Let X a nonce or key variable of Π. We have

$$\widetilde{\Pi} \models \mathsf{Secrecy}(X)$$

In particular, keys and nonces generated in honest sessions cannot be learned using dishonest or partially dishonest sessions.

Authentication

Let $\Pi \models^{p,1} \mathrm{Auth}(A_i, A_j, X)$ holds when A_i authenticates A_j in the nonce or key variables X when only one single, honest sessions is executed.

Authentication

Let $\Pi \models^{p,1} \mathrm{Auth}(A_i, A_j, X)$ holds when A_i authenticates A_j in the nonce or key variables X when only one single, honest sessions is executed.

Theorem

Let Π be a protocol and $\widetilde{\Pi}$ the corresponding transformed protocol.

$$\Pi \models^{p,1} \mathrm{Auth}(A_i, A_j, X) \Rightarrow \widetilde{\Pi} \models \mathrm{Auth}(A_i, A_j, X)$$

for any identities A_i, A_j and for any key or nonce variable X of the protocol Π.

Why adding encryption ?

Compared to Katz& Young, we have added public encryption.

$$A \rightarrow B : \quad \{K_{ab}\}_{\mathrm{pk}(B)}, A$$
$$B \rightarrow A : \quad \{K_{ab}\}_{\mathrm{pk}(A)}$$

K_{ab} is secure in a honest, passive session but NOT in the active case

Why adding encryption ?

Compared to Katz& Young, we have added public encryption.

$$\begin{aligned} A \rightarrow B: &\quad N_a \\ B \rightarrow A: &\quad N_b \\ A \rightarrow B: &\quad [\{K_{ab}\}_{\mathsf{pk}(B)}, A, \mathsf{sessionID}]_{\mathsf{sk}(A)} \\ B \rightarrow A: &\quad [\{K_{ab}\}_{\mathsf{pk}(A)}, \mathsf{sessionID}]_{\mathsf{sk}(B)} \end{aligned}$$

K_{ab} is secure in a honest, passive session but NOT in the active case

Why adding encryption ?

Compared to Katz& Young, we have added public encryption.

$$\begin{aligned} A \rightarrow B: &\quad N_a \\ B \rightarrow A: &\quad N_b \\ A \rightarrow B: &\quad [\{K_{ab}\}_{\mathsf{pk}(B)}, A, \mathsf{sessionID}]_{\mathsf{sk}(A)} \\ B \rightarrow A: &\quad [\{K_{ab}\}_{\mathsf{pk}(A)}, \mathsf{sessionID}]_{\mathsf{sk}(B)} \end{aligned}$$

K_{ab} is secure in a honest, passive session but NOT in the active case

$$\begin{aligned} A \rightarrow C(B): &\quad [\{K_{ab}\}_{\mathsf{pk}(B)}, A, \mathsf{sessionID}_{a,b}]_{\mathsf{sk}(A)} \\ C \rightarrow B: &\quad [\{K_{ab}\}_{\mathsf{pk}(B)}, C, \mathsf{sessionID}_{c,b}]_{\mathsf{sk}(C)} \end{aligned}$$

Why adding encryption ?

Compared to Katz& Young, we have added public encryption.

$$\begin{aligned} A \rightarrow B: &\quad N_a \\ B \rightarrow A: &\quad N_b \\ A \rightarrow B: &\quad [\{K_{ab}\}_{\mathsf{pk}(B)}, A, \mathsf{sessionID}]_{\mathsf{sk}(A)} \\ B \rightarrow A: &\quad [\{K_{ab}\}_{\mathsf{pk}(A)}, \mathsf{sessionID}]_{\mathsf{sk}(B)} \end{aligned}$$

K_{ab} is secure in a honest, passive session but NOT in the active case

$$\begin{aligned} A \rightarrow C(B): &\quad [\{K_{ab}\}_{\mathsf{pk}(B)}, A, \mathsf{sessionID}_{a,b}]_{\mathsf{sk}(A)} \\ C \rightarrow B: &\quad [\{K_{ab}\}_{\mathsf{pk}(B)}, C, \mathsf{sessionID}_{c,b}]_{\mathsf{sk}(C)} \\ B \rightarrow C: &\quad [\{K_{ab}\}_{\mathsf{pk}(C)}, \mathsf{sessionID}_{c,b}]_{\mathsf{sk}(B)} \end{aligned}$$

Towards a compiler

- The transformation does not depend on the primitives nor on the security properties.
- Could be used to compile secure protocols.

Specification : an authentication protocol

$$\begin{aligned} A \rightarrow B : &\quad A \\ B \rightarrow A : &\quad B \end{aligned}$$

Transformed (compiled) :

$$\begin{aligned} A \rightarrow B : &\quad N_a \\ B \rightarrow A : &\quad N_b \\ A \rightarrow B : &\quad \{A, [A, 1, A, B, N_a, N_b]_{\mathsf{sk}(A)}\}_{\mathsf{pk}(B)} \\ B \rightarrow A : &\quad \{B, [B, 1, A, B, N_a, N_b]_{\mathsf{sk}(A)}\}_{\mathsf{pk}(A)} \end{aligned}$$

Future work

- Extend the transformation to arbitrary primitives.
  $\longrightarrow$ provided they do not interfere with encryption and signatures ?
- Extend the result to arbitrary security properties.
  $\longrightarrow$ deducibility/reduction result for honest sessions ?
- Design of a lighter transformation ?