Towards Security Analyses of an Identity Federation Protocol

Marinella Petrocchi
IIT-CNR, Italy

joint work with

Maurice ter Beek, ISTI-CNR, Italy

Corrado Moiso, Telecom Italia, Italy

AICT 2007, Morne, Mauritius

Outline of the talk

- Rationale
- Identity federation protocols
- Network protocols for identity federation
- Modeling and Analysis
  - approach
  - specification language
  - formalization of the network protocol by Telecom Italia
  - analysis and results on a MITM attack
- Conclusions and future work

Rationale

PROTOCOLS

- Increasing interest in defining telecommunication protocols allowing an user to access all services belonging to the same (*circle of trust*), with (cross-domain) *single sign on*
- *Identity federation* process: federating an entity's identity and accessing services without explicitly presenting any credentials
- Reference: *Liberty Alliance*
  - consortium formed to define processes for federating identities
  - series of specifications use *Security Assertion Markup Language* (SAML)

Rationale (2)

SECURITY FEATURES THAT A FEDERATED IDENTITY PROCESS SHOULD GUARANTEE

- Limiting access to *authenticated* and *authorized* users.
- Preserving *privacy* of users:
  - w.r.t. sensitive user information (e.g., network addresses)
  - guarantee a user's identity without explicitly discovering it
  - possibly disclosing information related only to the service for which the access is requested (*e.g.,* destination preferences if the service is a travel agency)
- (Optional) Granting users *anonymous* access to services (*e.g.,* for temporary federations)

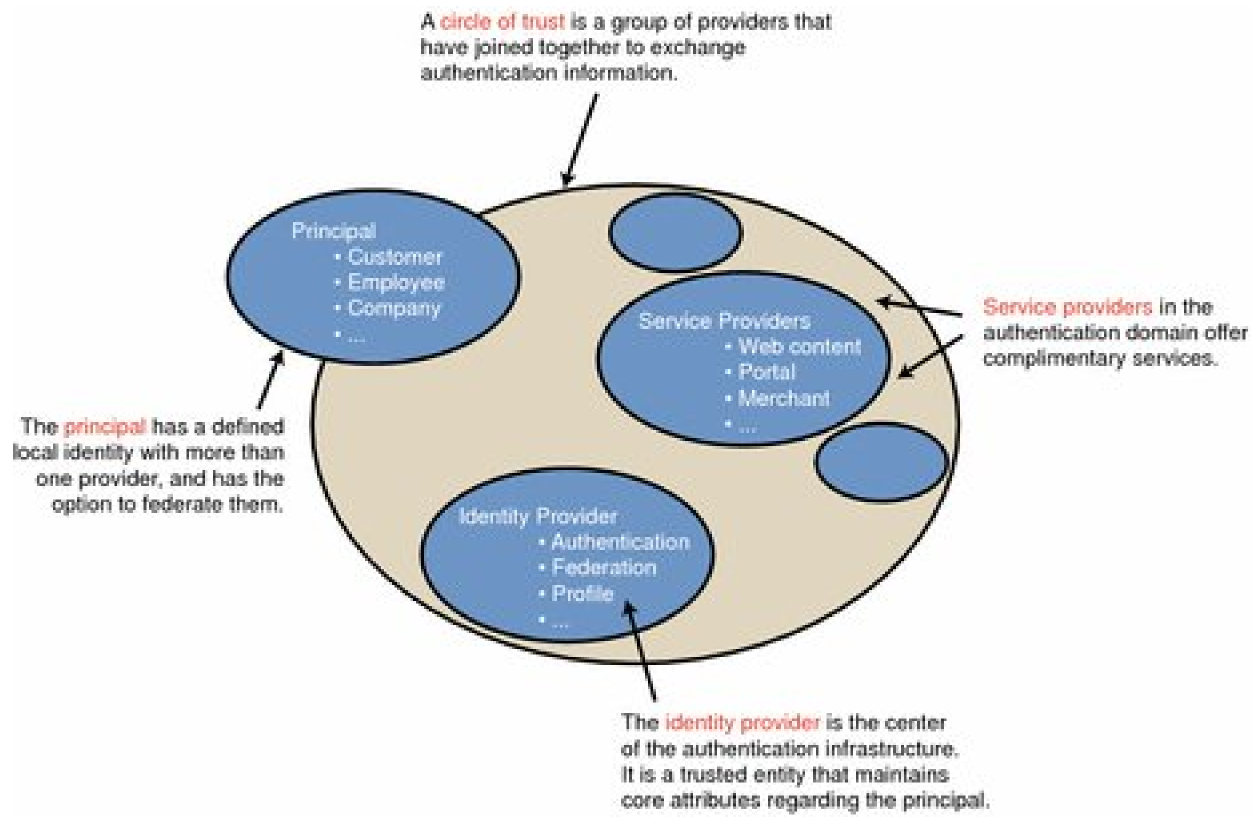
The goal

- Formal modeling and analysis of security protocols is an active branch of computer security
- successful techniques based on, e.g., process algebras, authentication logic, type systems have been applied
- we formally specify three users scenarios of a network protocol for identity federation proposed by Telecom Italia, by adding primitives for assure basic security properties
- we also model checking the specifications to test their correctness

Federating identities

- ABC Airlines and XYZ Car Rental Company decide to create a circle of trust.
- Mary has accounts on both ABC's and XYZ's Web sites.
- She logs into ABC's Web site. "You may share (or federate) your ABC online identity with members of our affinity group, which includes XYZ."
- Mary likes the idea, so she gives her permission.
- Mary goes to XYZ: "We see you're logged into the ABC Web site. Would you like to link your XYZ online identity with your ABC online identity?" OK!
- ...
- In the future, when she goes to either the ABC or XYZ site, she need only log into one and she's automatically logged into the other.

Federated Identity Architecture Example



Features

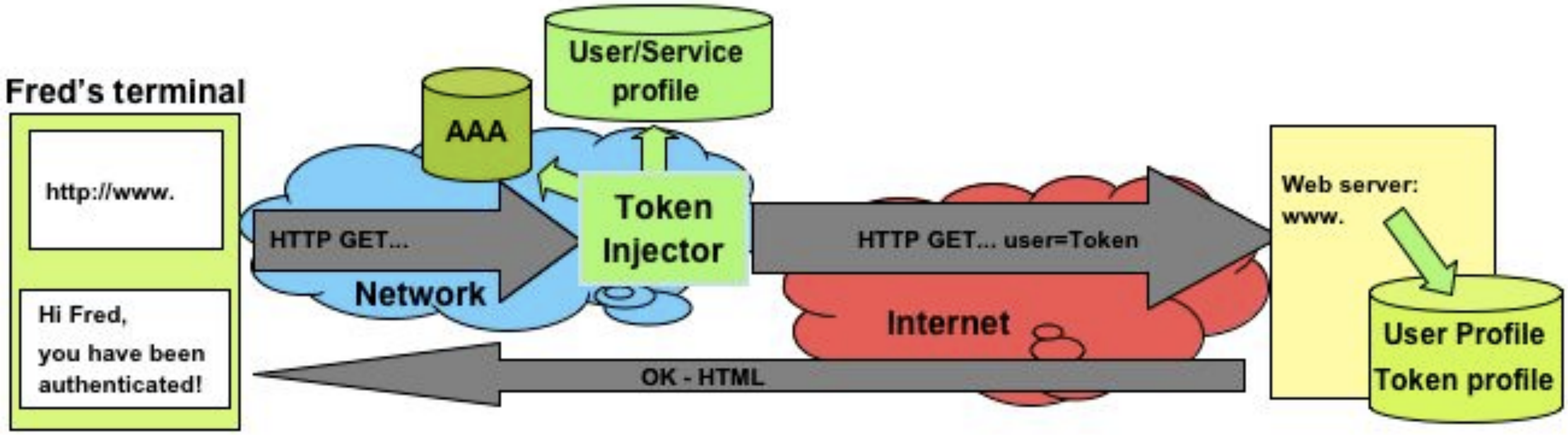
- Authentication is delegated to an *identity provider*, allowing *single sign on*
- A user token is a sequence of characters that identifies the user to each pair of parties in the circle of trust.
- User tokens are opaque, which indicates that a user handle as meaning only to the two parties that federate their users' identities.

The network protocol

proposed by Telecom Italia, [ICIN'06]

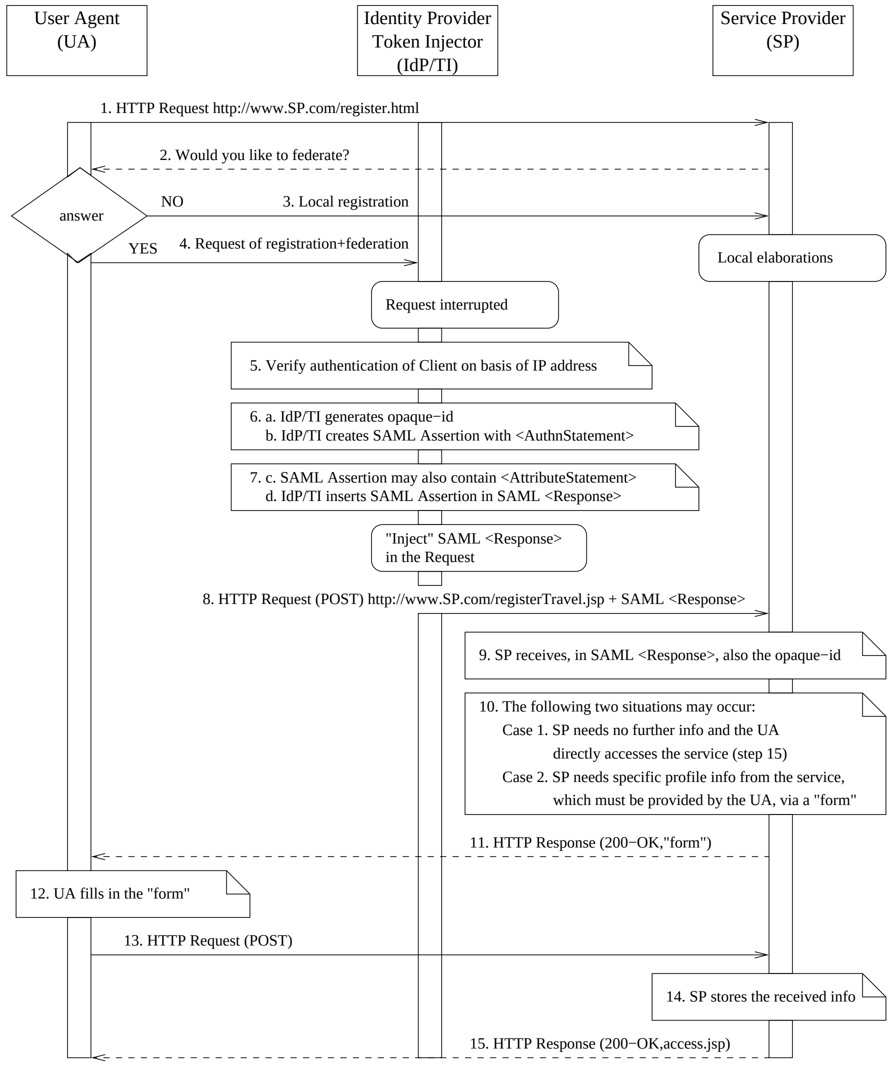
- is an identity federation protocol
- permits users to access services through different access networks (e.g., fixed and mobile)
- gives the network provider the role of the identity provider $\longrightarrow$ services will rely on the authentication information provided by the access network

Token injector mechanism

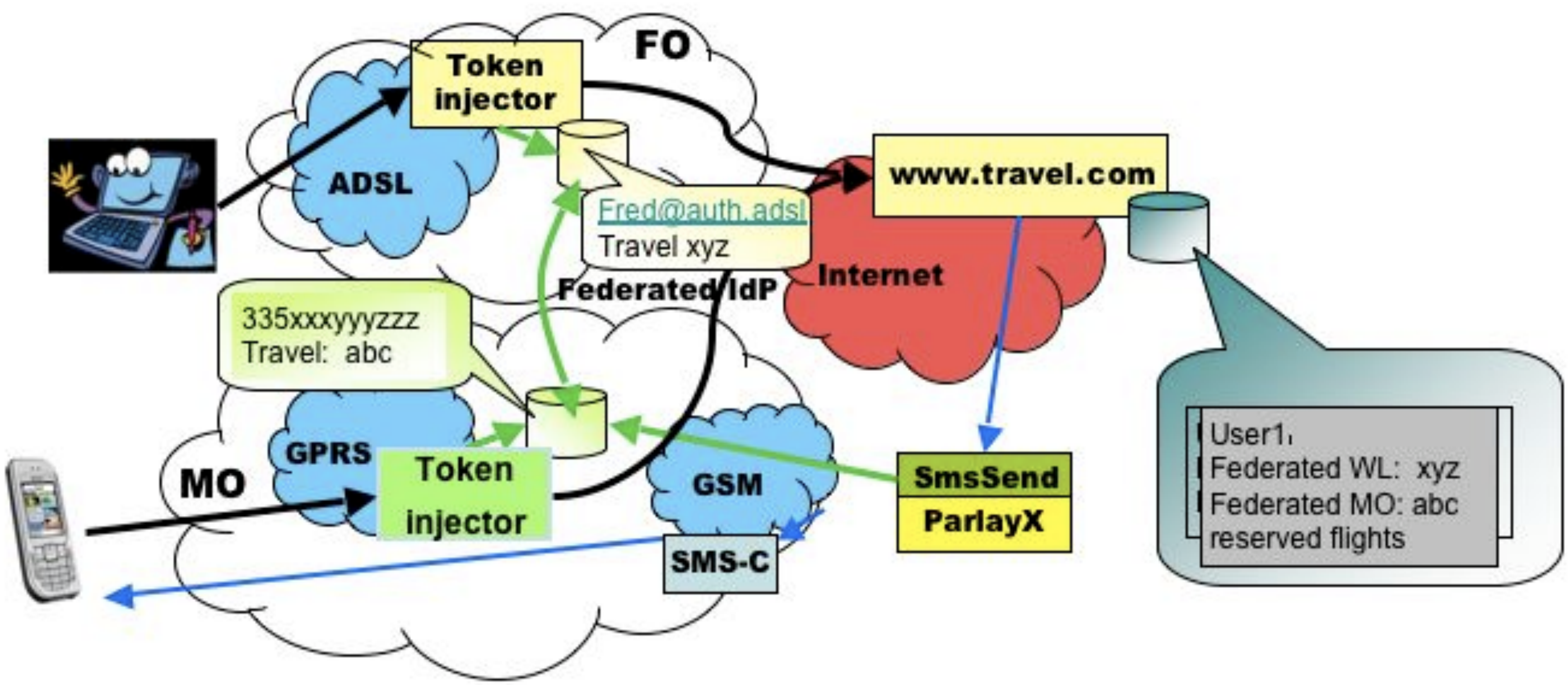


- intercepts http access requests
- (generate) and inject token
- forward to applications

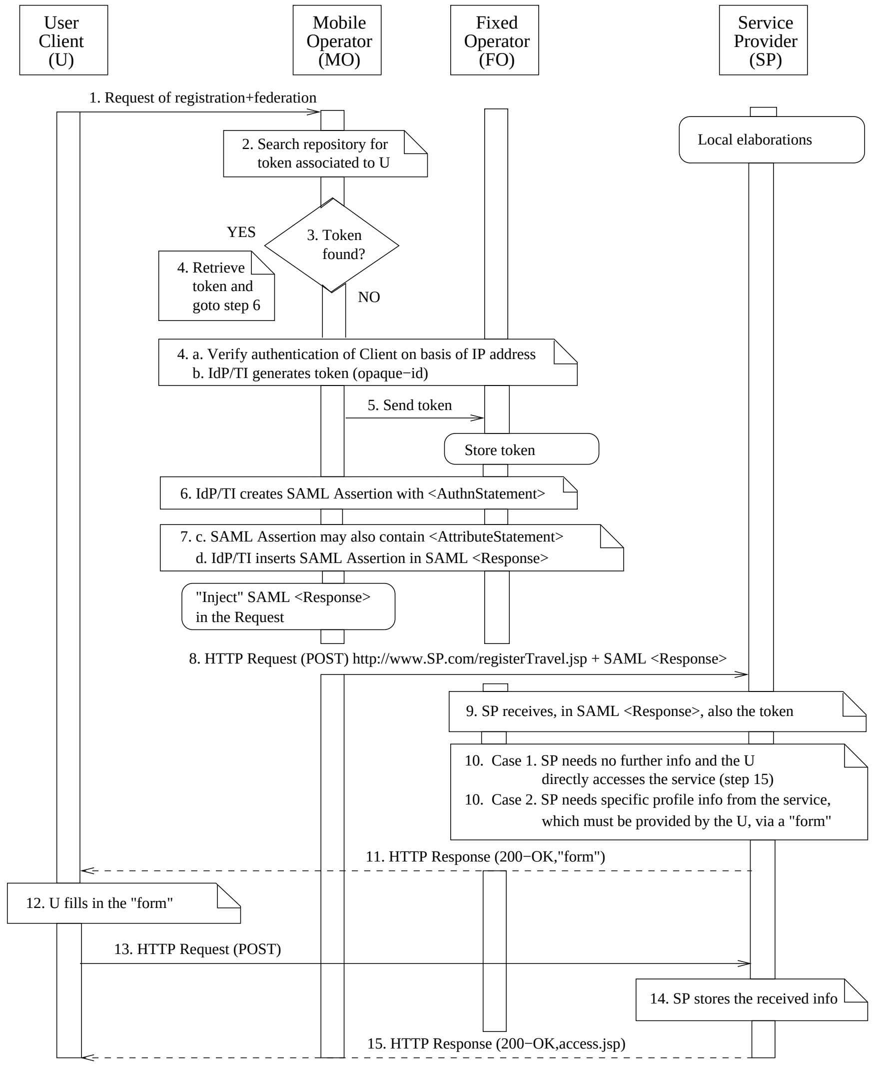
MSC for federated registration



Example: multiple access networks



MSC for multiple access networks



Analysis Approach

- We specify the protocol into the formal language Crypto-CCS
- We specify the property to be verified into a logic formula
- We add the *intruder* to the *honest* specification
  - its behavior is implicitly defined by the semantics of the language
- We check the property over the *intruder's knowledge*
  - intruder's knowledge $\longrightarrow$ the set of messages the intruders initially knows, plus what she receives as the computation goes on

Crypto-CCS

PROCESS ALGEBRA CCS + CRYPTOGRAPHIC PRIMITIVES

- Set of processes able to communicate via message passing
- Inference system models possible operation of messages

$$r = \frac{m_1 \quad \cdots \quad m_n}{m_0}$$

$S := S_1 \parallel S_2 \mid A$	*compound systems*
$A := \mathbf{0} \mid p.A \mid [m_1 \cdots m_n \vdash_r x]A; A_1$	*sequential agents*
$p := c!m \mid c?x$	*prefix constructs*

Informal semantics of Crypto-CCS

- $c!m$ send message;
- $c?x$ receive message;
- $\mathbf{0}$ does nothing;
- $p.A$ perform p and then behave as A;
- $[m_1 \cdots m_n \vdash_r x]A; A_1$ inference construct:
- $S_1 \,\|\, S_2$ parallel composition + synchronization

Example: $[m \quad pk_y^{-1} \vdash_{sign} x]A; \mathbf{0}$

A process that uses rule $sign$ to obtain a digitally signed message from plaintext m and private key pk_y^{-1} and then behaves as A, or otherwise does nothing.

An example inference system

for public key cryptography

$$\frac{x \quad y}{\textit{Pair}(x,y)}\ (\textit{pair}) \qquad \frac{x \quad pk_y^{-1}}{\{x\}_{pk_y^{-1}}}\ (\textit{sign}) \qquad \frac{x \quad \textit{KEY}}{\{x\}_{\textit{KEY}}}\ (\textit{enc})$$

$$\frac{\textit{Pair}(x,y)}{x}\ (\textit{1st}) \qquad \frac{\{x\}_{\textit{KEY}} \quad \textit{KEY}}{x}\ (\textit{dec})$$

$$\frac{\textit{Pair}(x,y)}{y}\ (\textit{2nd}) \qquad \frac{\{x\}_{pk_y^{-1}} \quad pk_y}{x}\ (\textit{ver}) \qquad \frac{x}{x}\ (\textit{check})$$

Federated registration

$$\begin{array}{llcl} c_0 & \mathit{U} \mapsto \mathit{IdP} & : & r \\ c_1 & \mathit{IdP} \mapsto \mathit{SP} & : & \{r, \mathit{SAML\ assertion}\}_{K^{-1}_{\mathit{IdP}}} \\ c_2 & \mathit{SP} \mapsto \mathit{U} & : & \{ok/ko\}_{K^{-1}_{\mathit{SP}}} \end{array}$$

1. U asks IdP and SP to federate

2. r intercepted by $IdP \rightarrow$
   - authentication of U
   - token generation
   - assembling SAML assertion

3. SP grants/denies access to U

SAML Assertion

A SAML assertion declares “$Subj$ is authenticated”.

$\{$*Subj*, *Auth Stat*, *Attr Stat*$\}_{KEY}$ *encrypted SAML assertion*

Subj $\rightarrow$

- token id_U, univocally identifying U
- $AuthStat$ authentication statement
- $AttrStat$ list of user attribute + n_U^{IdP}, nonce to avoid replay attack

$\{r, SAML\}_{K_{IdP}^{-1}} \rightarrow$ signed by IdP for authenticity

Crypto-CCS specification - SP

$SP_0(0) \doteq$
$c_1?x_m.$ *receive SAML assertion + request*
$SP_1(x_m)$ *and go to next state*

$SP_1(x_m) \doteq$
$[x_m \quad k_{IdP} \vdash_{ver} x_p]$ *verify signature,*
$[x_p \vdash_{2nd} x_{enc}]$ *extract encryption,*
$[x_{enc} \quad KEY \vdash_{dec} x_{dec}]$ *decrypt,*
$[x_{dec} \vdash_{1st} x_{pair}]$ *extract pair: token + Auth Stat,*
$[x_{dec} \vdash_{2nd} x_{n_{IdP}}U]$ *extract nonce,*

$[x_{pair} \vdash 1st\ x_{id_U}]$	*extract token,*
$[x_{pair} \vdash 2nd\ x_{auth}]$	*extract Auth Stat,*
$[x_{auth} \vdash check\ x_{auth}]$	*test correctness Auth Stat,*
$[x_{n_U^{IdP}} \vdash check\ x_{n_U^{IdP}}]$	*test freshness nonce,*
$[x_{id_U}\quad x_{n_U^{IdP}} \vdash pair\ (x_{id_U}, x_{n_U^{IdP}})]$	*build pair to store,*
$c_S!(x_{id_U}, x_{n_U^{IdP}})$	*store token + nonce pair,*
$[access\quad k^{-1}_{SP} \vdash sign\ x_{sign}]$	*prepare signature to*
$c_2!x_{sign}.\mathbf{0}$	*grant access and stop*

Analysis of a Man-In-The-Middle Attack

Is it possible to intercept a conversation between IdP and SP, without awareness by IdP and SP?

Property: *"whenever SP concludes the network protocol apparently with IdP, it was indeed IdP that executed the protocol"*

We introduce two special actions in our Crypto-CCS specification: *commit*(*a*,*b*) and *run*(*b*,*a*).

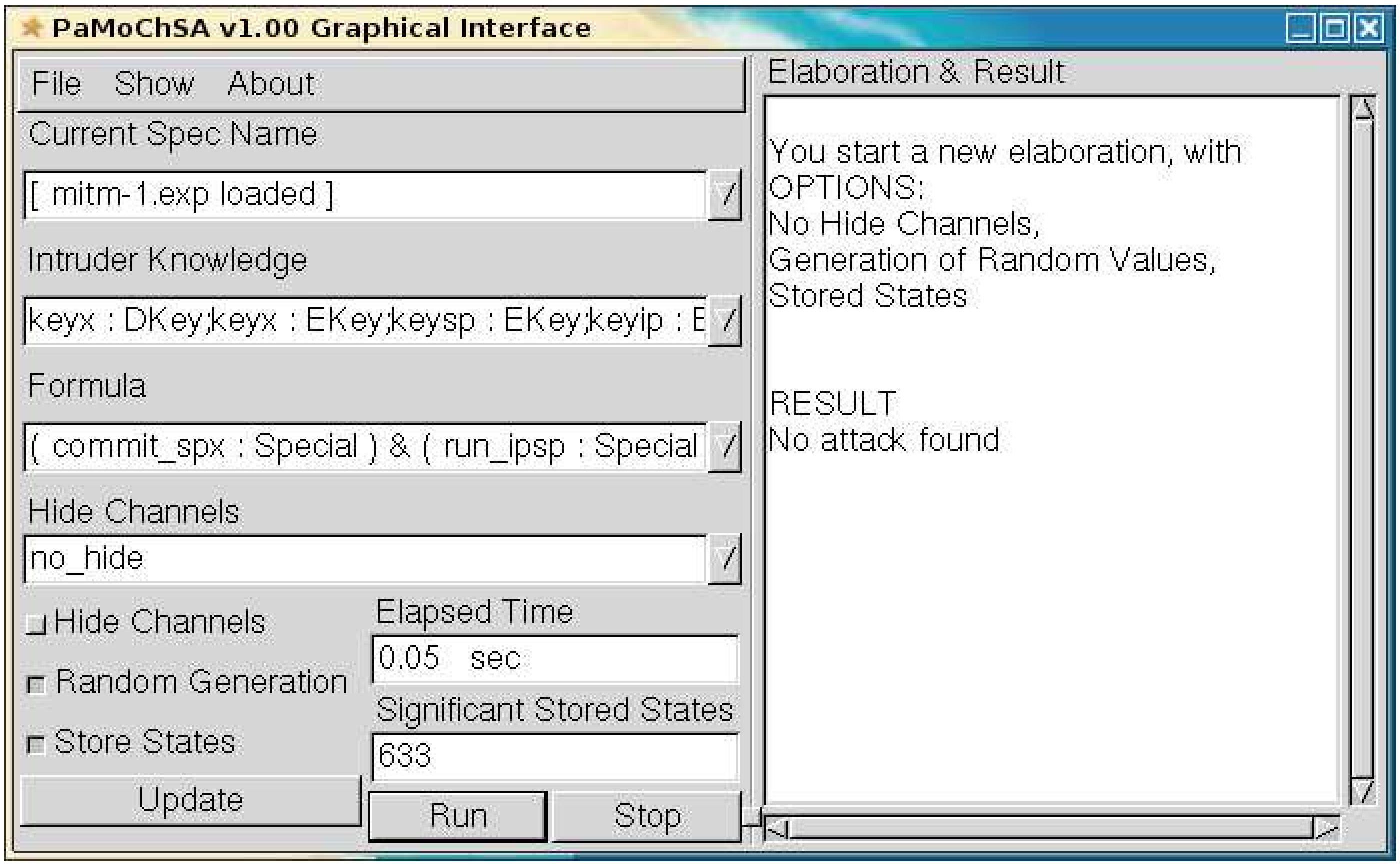
We ask the model checker if a computation exists s.t.

- *IdP* is convinced to have talked with *SP*, while in reality it was *SP* that has finished talking with X
- *SP* is convinced to have talked with *IdP*, while in reality it was *IdP* that has started talking with X

Input

- Specification file: `mitm-1.exp`
- Logic formula: ((*run*(*IdP,SP*) AND *commit*(*SP,X*)) OR
  ((*run*(*IdP,X*) AND *commit*(*SP,IdP*))
- Initial knowledge: $\{pk_X, pk_X^{-1}, pk_{IdP}, pk_{SP}\}$
- Result: **No attack found**

Screenshot of PaMoChSA's graphical interface



Conclusions

- a clear advantage of the use of formal methods in the design phase of a protocol is: *eventually arrive at a well-defined protocol that is guaranteed to satisfy certain desirable properties*
- result of initial analysis strengthens our confidence in the formal specifications we have specified.
- it leads us to believe that we correctly inserted digital signatures, encryption and nonces into the network protocol

Future Work

- we intend to extend the analysis by considering
  - more user scenarios;
  - more security properties (unsubscription, anonymity)
- accepted paper at YR-SOC 2007 on the case of the Federated Network Providers scenario
- deal with quantitative extensions of formal methods and tool (e.g., timed, probabilistic specification languages, stochastic model checkers)