A Security Protocol Animator Tool for AVISPA

Yann Glouche[1] Thomas Genet[1]

[1]IRISA-INRIA, Rennes, France
Team LANDE

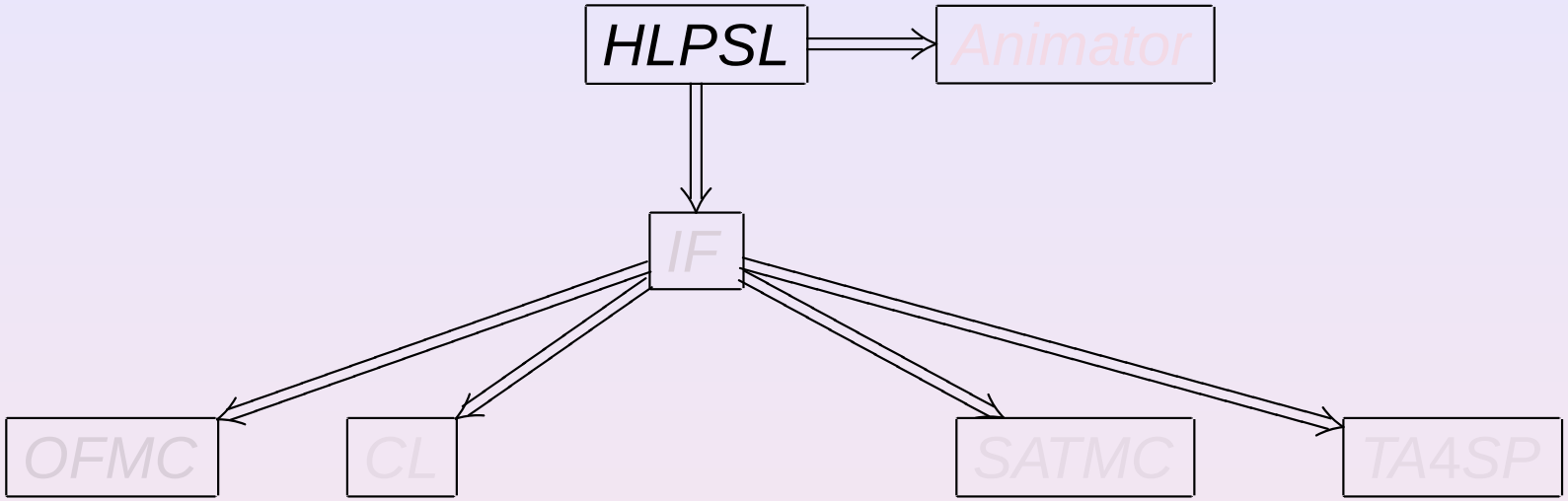
Artist2 Security Workshop, 18th May 2006

The Need for a protocol animator in the AVISPA System

Avispa project

- AVISPA is a verification tool for cryptographic protocols.
- High Level Protocol Specification Language (HLPSL).
- Ability to use different techniques on the same protocol specification.

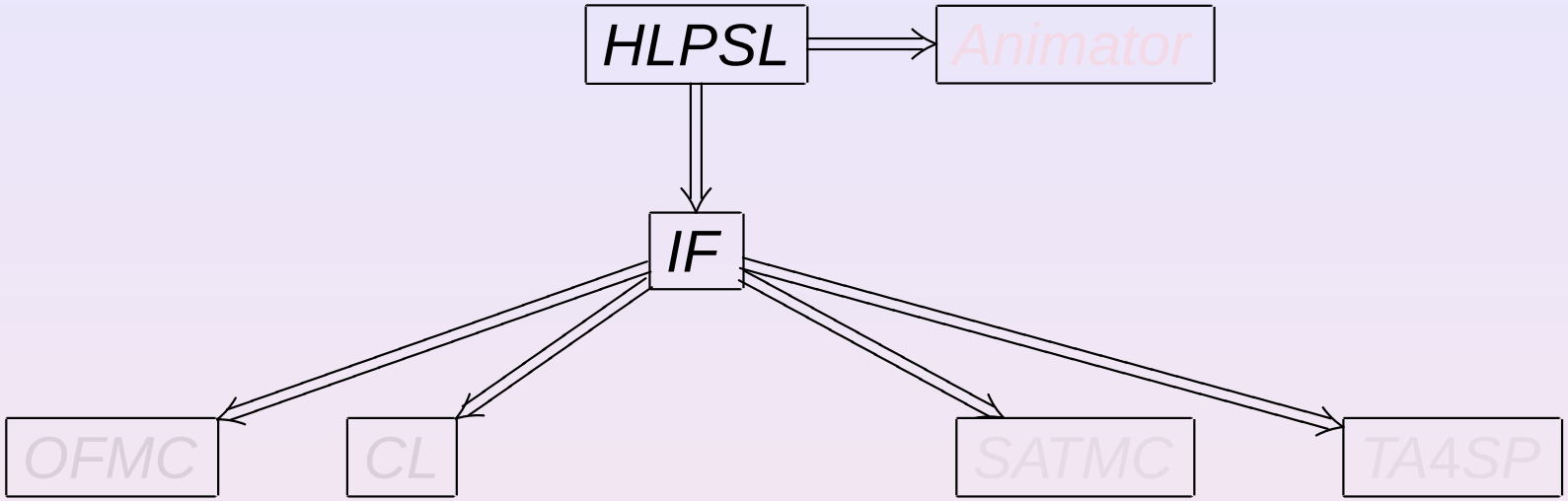
The Need for a protocol animator in the AVISPA System



Avispa tools

1. On-the-Fly Model-Checker (OFMC)
2. Constraint-Logic-based model-checker (CL)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata Automatic Approximations for the Analysis of Security Protocol (TA4SP)

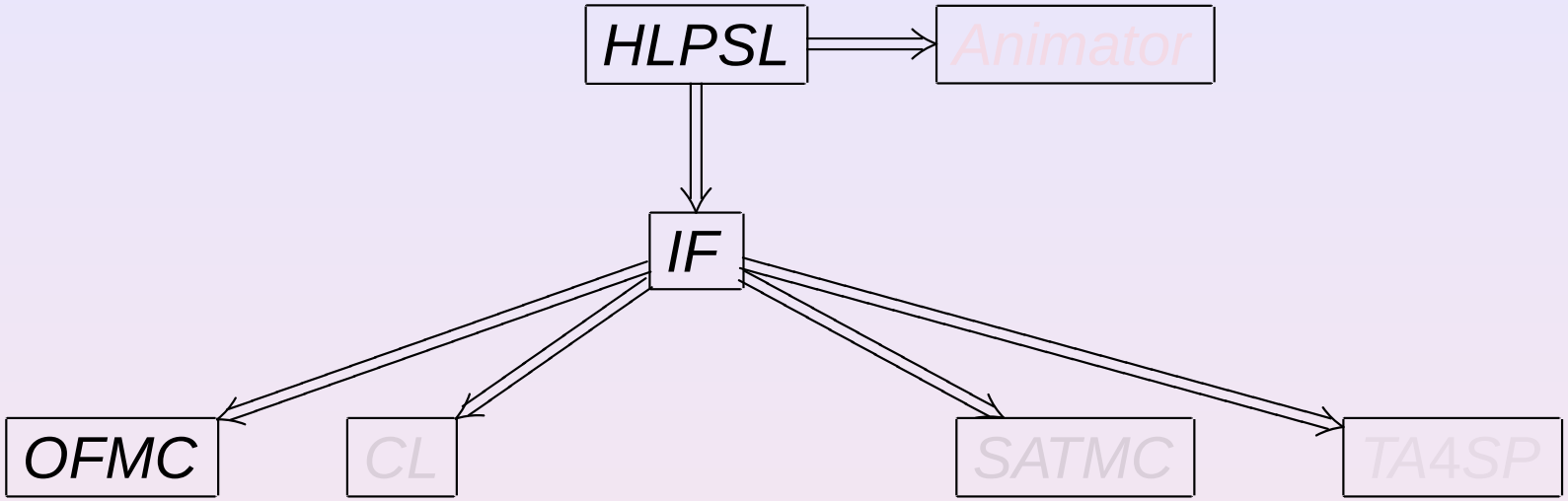
The Need for a protocol animator in the AVISPA System



Avispa tools

1. On-the-Fly Model-Checker (OFMC)
2. Constraint-Logic-based model-checker (CL)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata Automatic Approximations for the Analysis of Security Protocol (TA4SP)

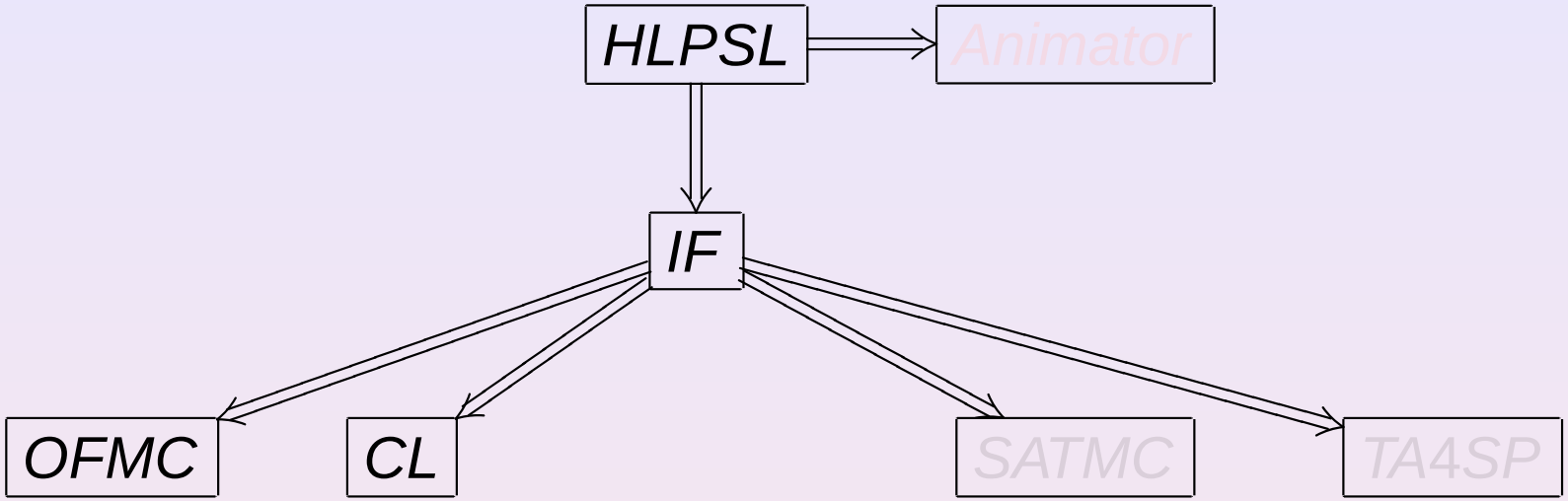
The Need for a protocol animator in the AVISPA System



Avispa tools

1. On-the-Fly Model-Checker (OFMC)
2. Constraint-Logic-based model-checker (CL)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata Automatic Approximations for the Analysis of Security Protocol (TA4SP)

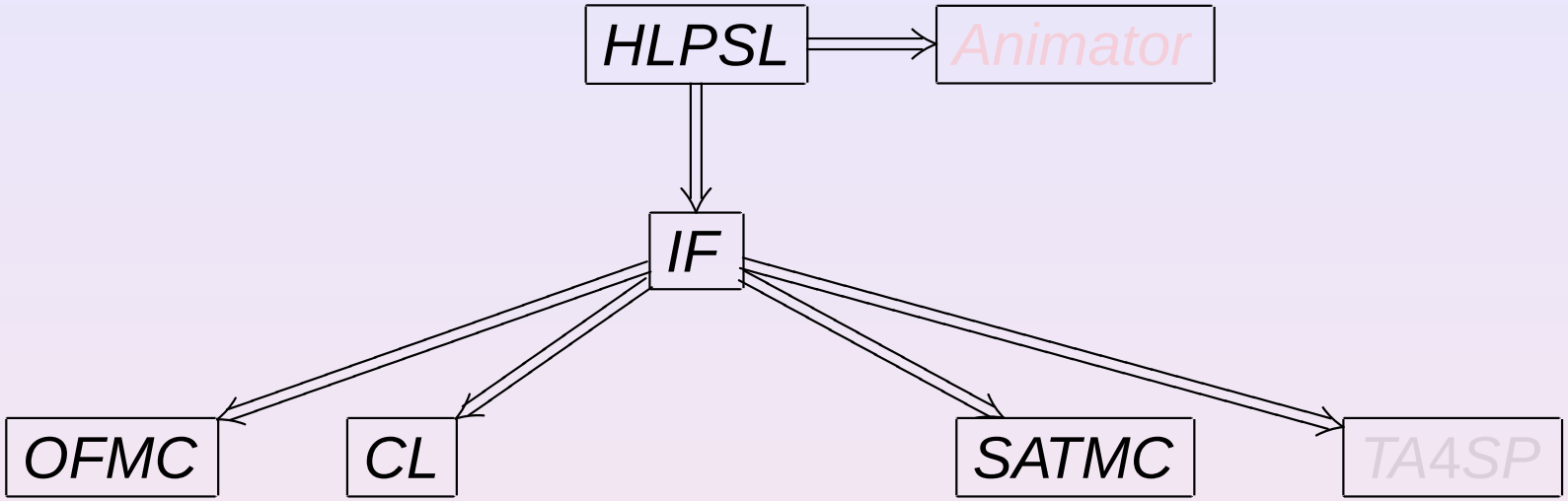
The Need for a protocol animator in the AVISPA System



Avispa tools

1. On-the-Fly Model-Checker (OFMC)
2. Constraint-Logic-based model-checker (CL)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata Automatic Approximations for the Analysis of Security Protocol (TA4SP)

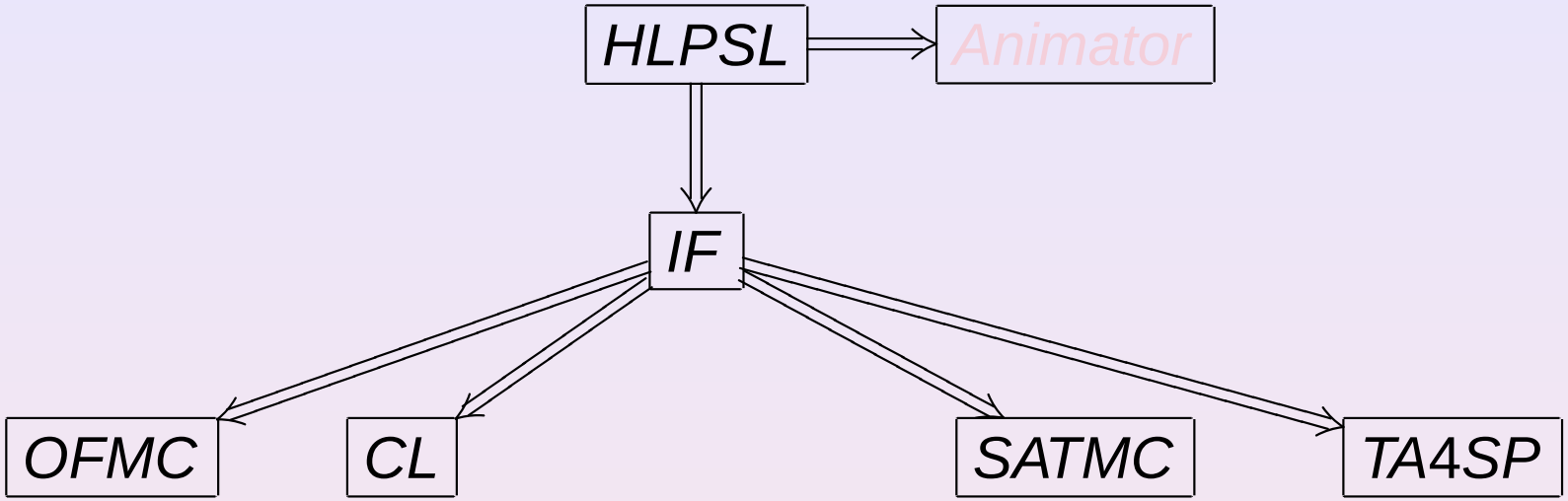
The Need for a protocol animator in the AVISPA System



Avispa tools

1. On-the-Fly Model-Checker (OFMC)
2. Constraint-Logic-based model-checker (CL)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata Automatic Approximations for the Analysis of Security Protocol (TA4SP)

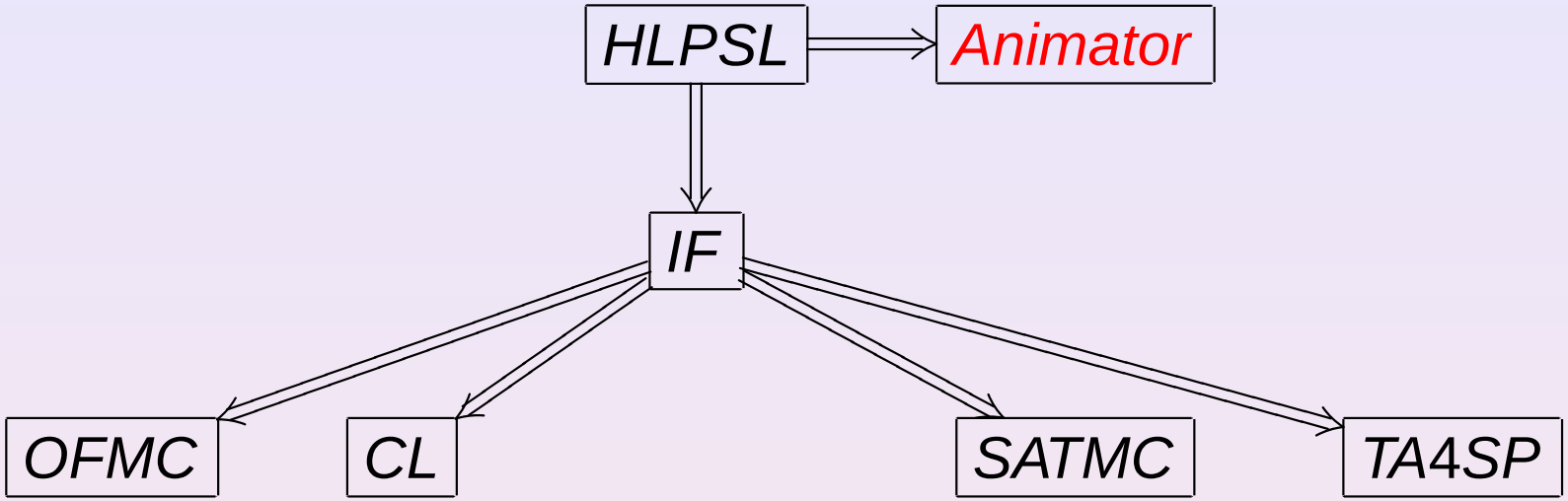
The Need for a protocol animator in the AVISPA System



Avispa tools

1. On-the-Fly Model-Checker (OFMC)
2. Constraint-Logic-based model-checker (CL)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata Automatic Approximations for the Analysis of Security Protocol (TA4SP)

The Need for a protocol animator in the AVISPA System



Avispa tools

1. On-the-Fly Model-Checker (OFMC)
2. Constraint-Logic-based model-checker (CL)
3. SAT-based Model-Checker (SATMC)
4. Tree Automata Automatic Approximations for the Analysis of Security Protocol (TA4SP)

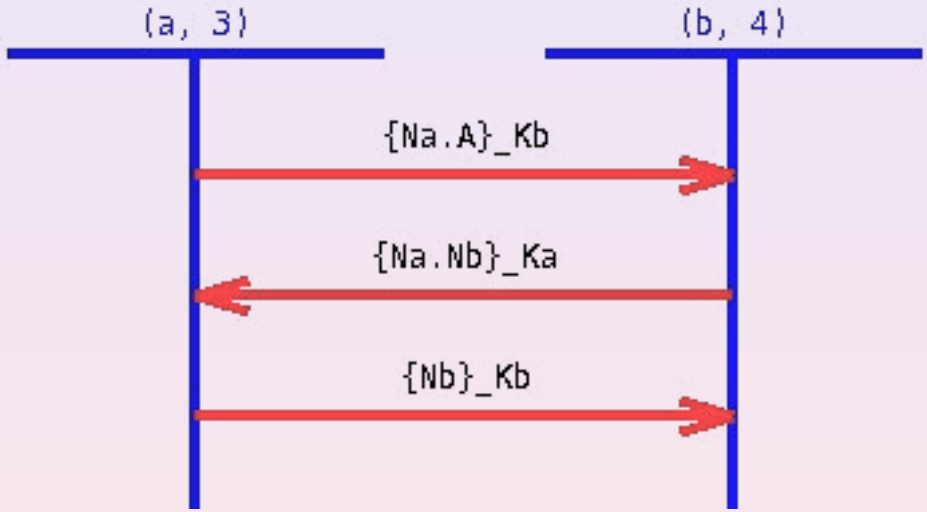
The Need for a protocol animator in the AVISPA System

```
role a(...)
 State=0 ∧ RCV(start)
  =|> State’:=1 ∧ Na’:=new()
      ∧ SND({Na’.A}_Kb)
 State=1 ∧ RCV(Na.Nb’_Ka)
  =|> State’:=2
       ∧ SND({Nb’}_Kb)
role b(...)
 State=0 ∧ RCV({Na’.A’}_Kb)
  =|> State’:=1 ∧ Nb’:=new()
      ∧ SND({Na’.Nb}_Ka)
  State=1 ∧ RCV({Nb}_Kb)
  =|> State’:=2
```

$$A \rightarrow B : \{Na, A\}_{Kb}$$
$$A \rightarrow B : \{Na, Nb\}_{Ka}$$
$$A \rightarrow B : \{Nb\}_{Kb}$$

The Need for a protocol animator in the AVISPA System

```
role a(...)
 State=0 ∧ RCV(start)
  =|> State’:=1 ∧ Na’:=new()
      ∧ SND({Na’.A}_Kb)
 State=1 ∧ RCV(Na.Nb’_Ka)
  =|> State’:=2
       ∧ SND({Nb’}_Kb)
role b(...)
 State=0 ∧ RCV({Na’.A’}_Kb)
  =|> State’:=1 ∧ Nb’:=new()
      ∧ SND({Na’.Nb}_Ka)
  State=1 ∧ RCV({Nb}_Kb)
  =|> State’:=2
```



The Need for a protocol animator

- ⇒ produce interactively MSC from an HLPSL specification.

The protocol animator

Protocol specification

- Protocol specifications in HLPSL are divided into roles.

Example (Protocol specification)

- Roles declaration

```
role a(A : agent...)
 State=0 ∧ RCV(start)
  =|> State’:=1 ∧ Na’:=new() ∧ SND({Na’.A}_Kb)
 State=1 ∧ RCV(Na.Nb’_Ka) =|> State’:=2 ∧ SND({Nb’}_Kb)
role b(B : agent...)
 State=0 ∧ RCV({Na’.A’}_Kb)
  =|> State’:=1 ∧ Nb’:=new() ∧ SND({Na’.Nb}_Ka)
  State=1 ∧ RCV({Nb}_Kb) =|> State’:=2
```

- Session declaration

```
role one_session(A, B : agent...)  composition
  a(A...)  ∧ b(B...)
```

- Scenario declaration

```
one_session(alice, bob...)∧one_session(charlie, dane...)
```

The protocol animator

Protocol specification

- Protocol specifications in HLPSL are divided into roles.

Example (Protocol specification)

- Roles declaration

```
role a(A : agent...)
 State=0 ∧ RCV(start)
  =|> State'=1 ∧ Na'=new() ∧ SND({Na'.A}_Kb)
 State=1 ∧ RCV(Na.Nb'_Ka) =|> State'=2 ∧ SND({Nb'}_Kb)
role b(B : agent...)
 State=0 ∧ RCV({Na'.A'}_Kb)
  =|> State'=1 ∧ Nb'=new() ∧ SND({Na'.Nb}_Ka)
  State=1 ∧ RCV({Nb}_Kb) =|> State'=2
```

- Session declaration

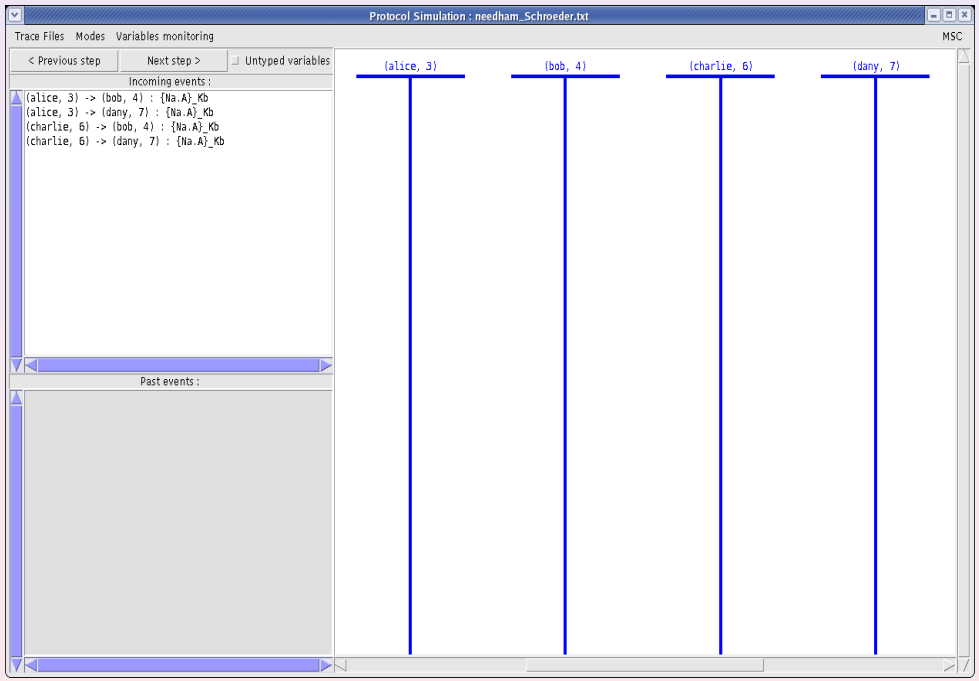
```
role one_session(A, B : agent...)  composition
  a(A...)  ∧ b(B...)
```

- Scenario declaration

```
one_session(alice, bob...)∧one_session(charlie, dane...)
```

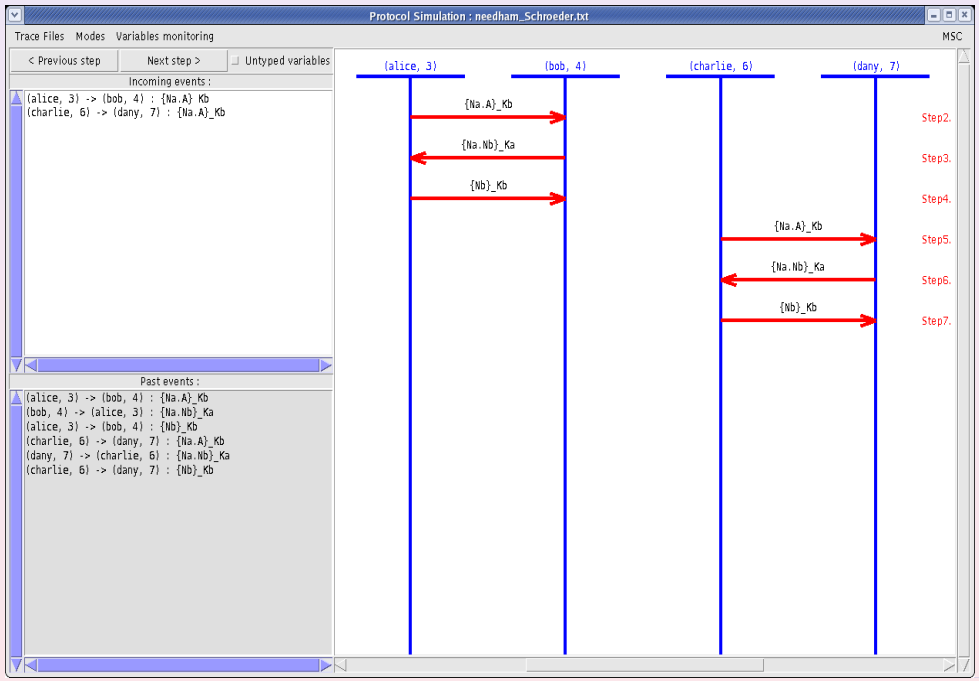
The protocol animator

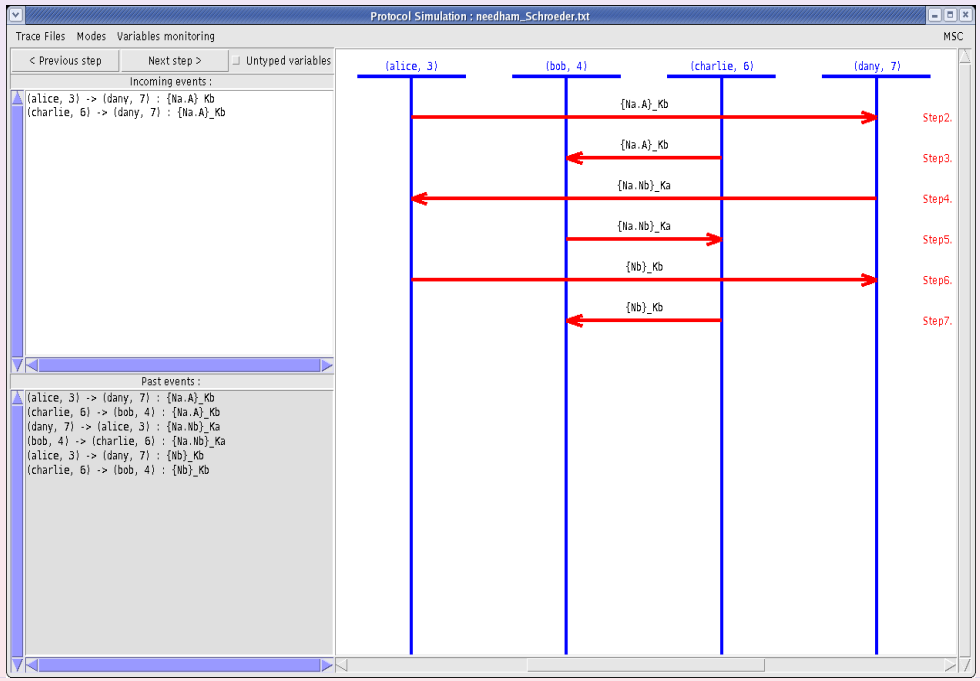
$$\texttt{one\_session(alice, bob...)} \wedge \texttt{one\_session(charlie, dane...)}$$



The protocol animator

$$\texttt{one\_session(alice, bob...)} \wedge \texttt{one\_session(charlie, dane...)}$$





The protocol animator

The features of current version

- Full support of HLPSL
- Interactive construction of MSC guided by the user because of
  - non deterministic protocols
  - choices in interleaved sessions
- usual undo/redo in constructed MSCs
- MSCs import/export

The protocol animator

The features of current version

- Full support of HLPSL
- Interactive construction of MSC guided by the user because of
  - non deterministic protocols
  - choices in interleaved sessions
- usual undo/redo in constructed MSCs
- MSCs import/export

The protocol animator

The features of current version

- Full support of HLPSL
- Interactive construction of MSC guided by the user because of
  - non deterministic protocols
  - choices in interleaved sessions
- usual undo/redo in constructed MSCs
- MSCs import/export

The protocol animator

The features of current version

- Full support of HLPSL
- Interactive construction of MSC guided by the user because of
  - non deterministic protocols
  - choices in interleaved sessions
- usual undo/redo in constructed MSCs
- MSCs import/export

Experiments

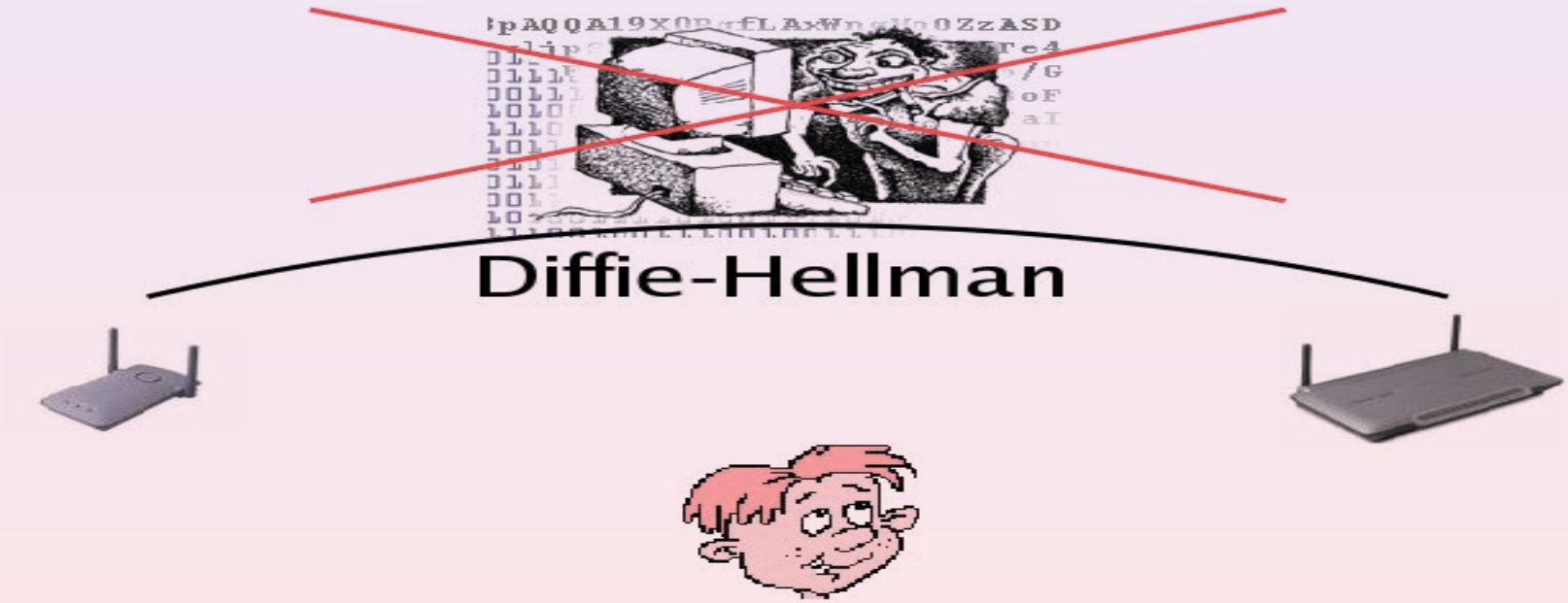
We have applied the animator to several protocols

- all the protocols of the AVISPA Library
- a new protocol developped by Thomson called User Supervised Device Pairing (USDP) for pairing two devices

Experiments

We have applied the animator to several protocols

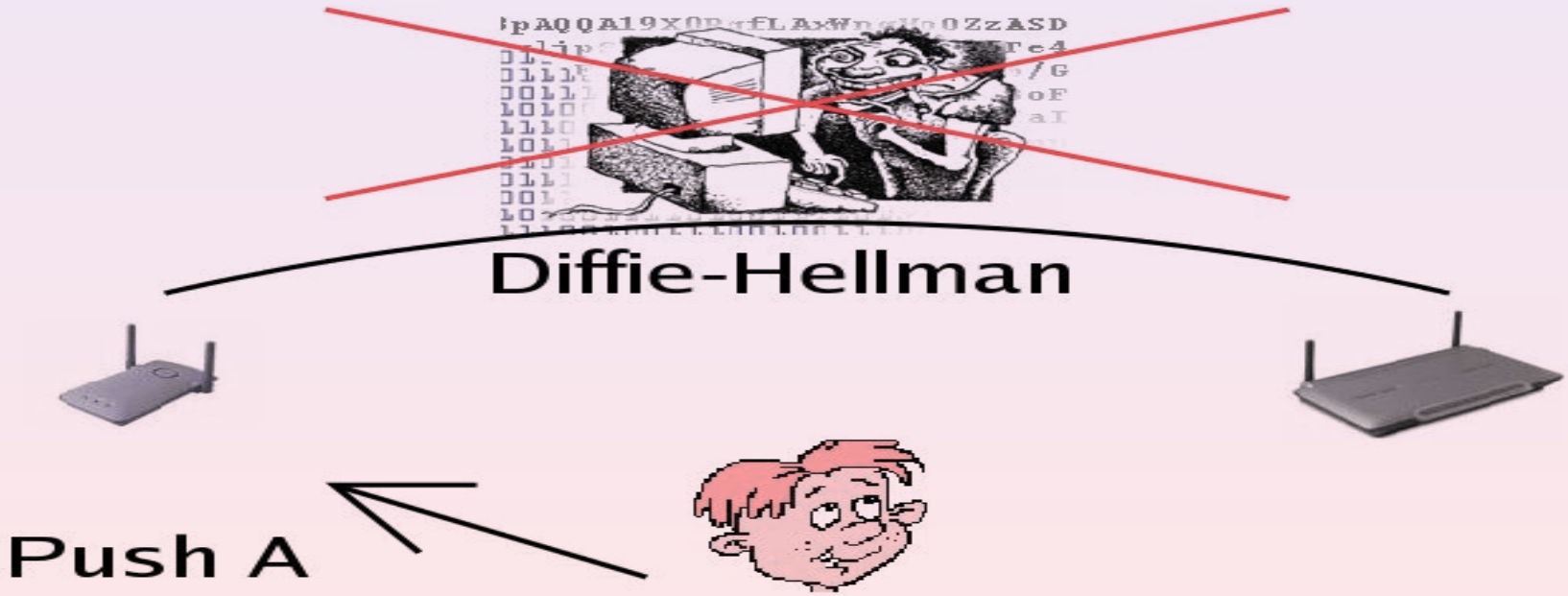
- all the protocols of the AVISPA Library
- a new protocol developped by Thomson called User Supervised Device Pairing (USDP) for pairing two devices



Experiments

We have applied the animator to several protocols

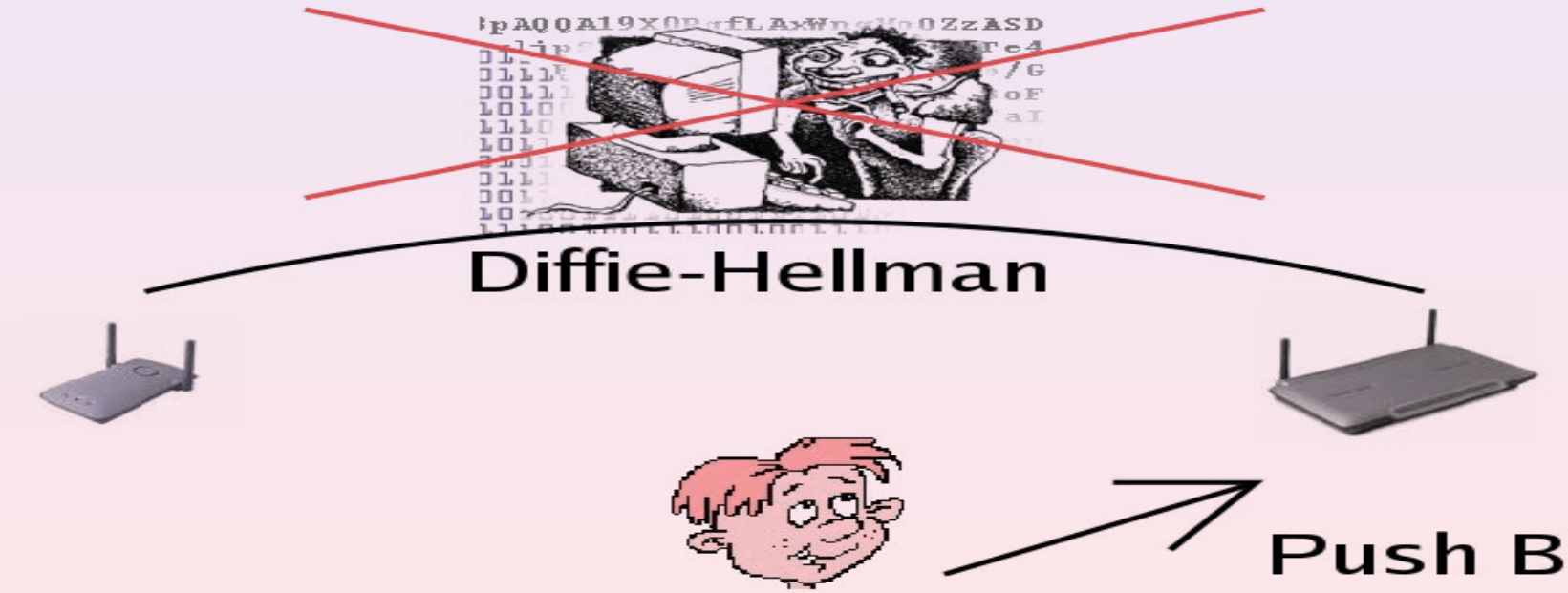
- all the protocols of the AVISPA Library
- a new protocol developped by Thomson called User Supervised Device Pairing (USDP) for pairing two devices



Experiments

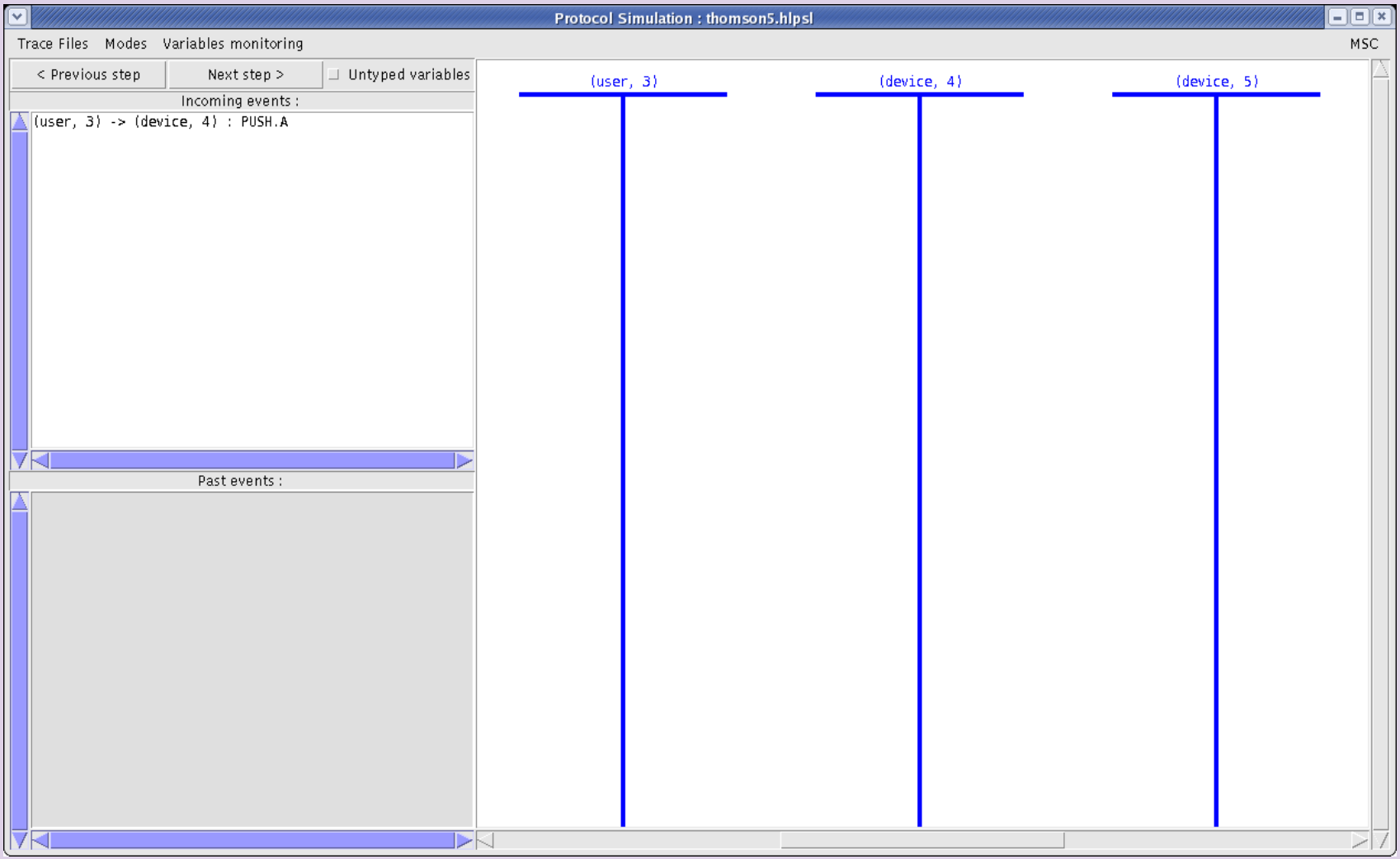
We have applied the animator to several protocols

- all the protocols of the AVISPA Library
- a new protocol developped by Thomson called User Supervised Device Pairing (USDP) for pairing two devices



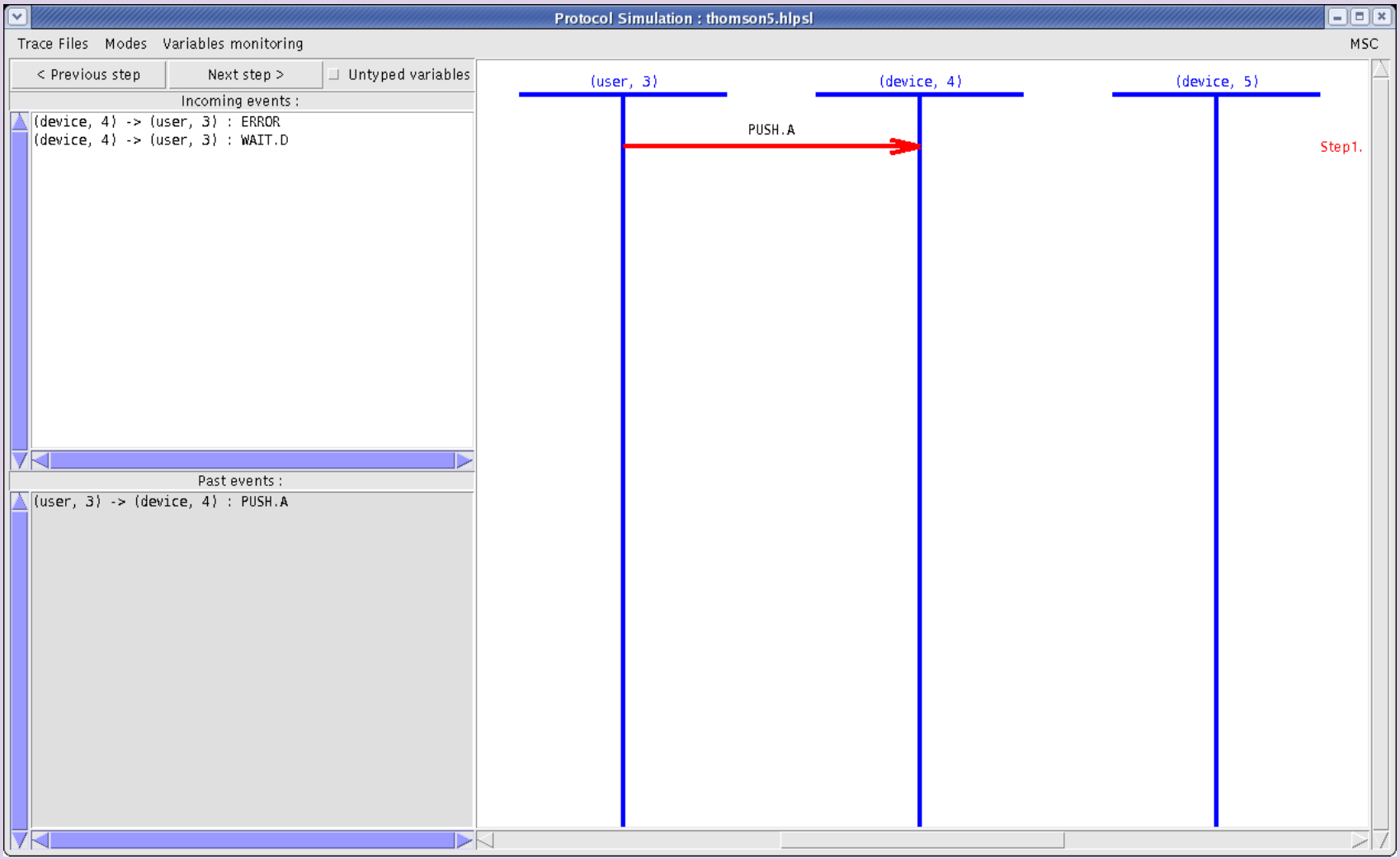
Experiments

An execution trace of Thomson's USDP protocol



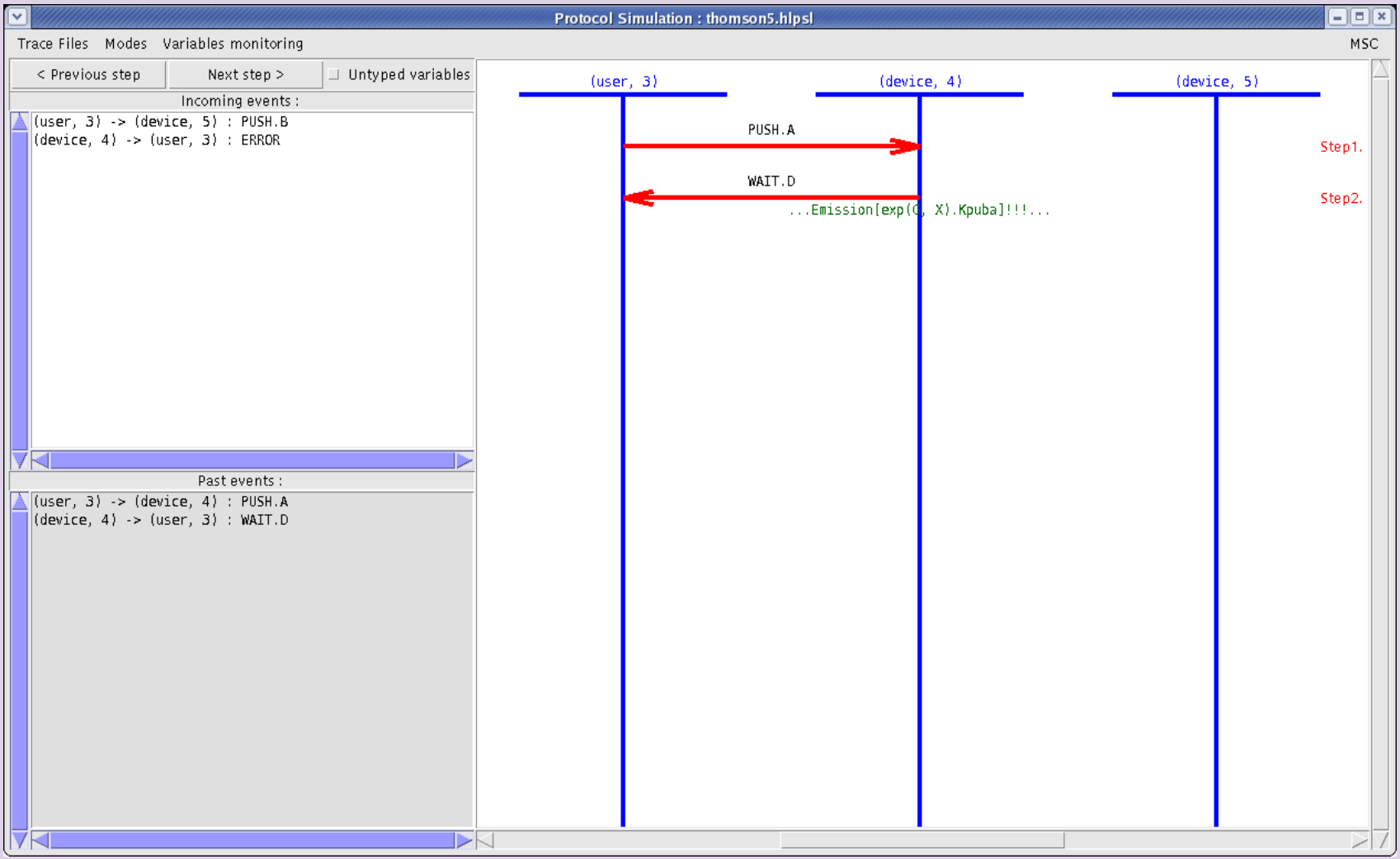
Experiments

An execution trace of Thomson's USDP protocol



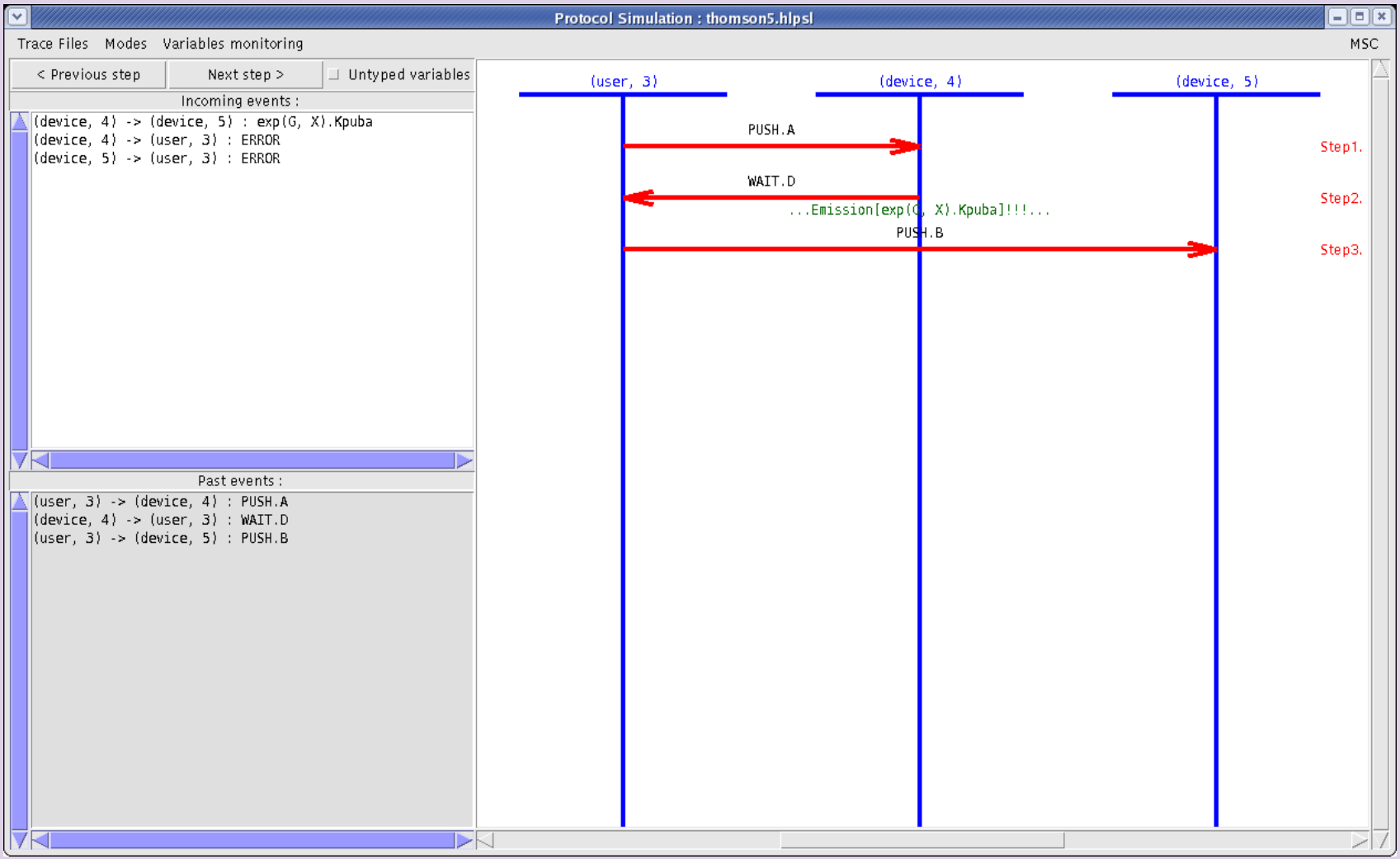
Experiments

An execution trace of Thomson's USDP protocol



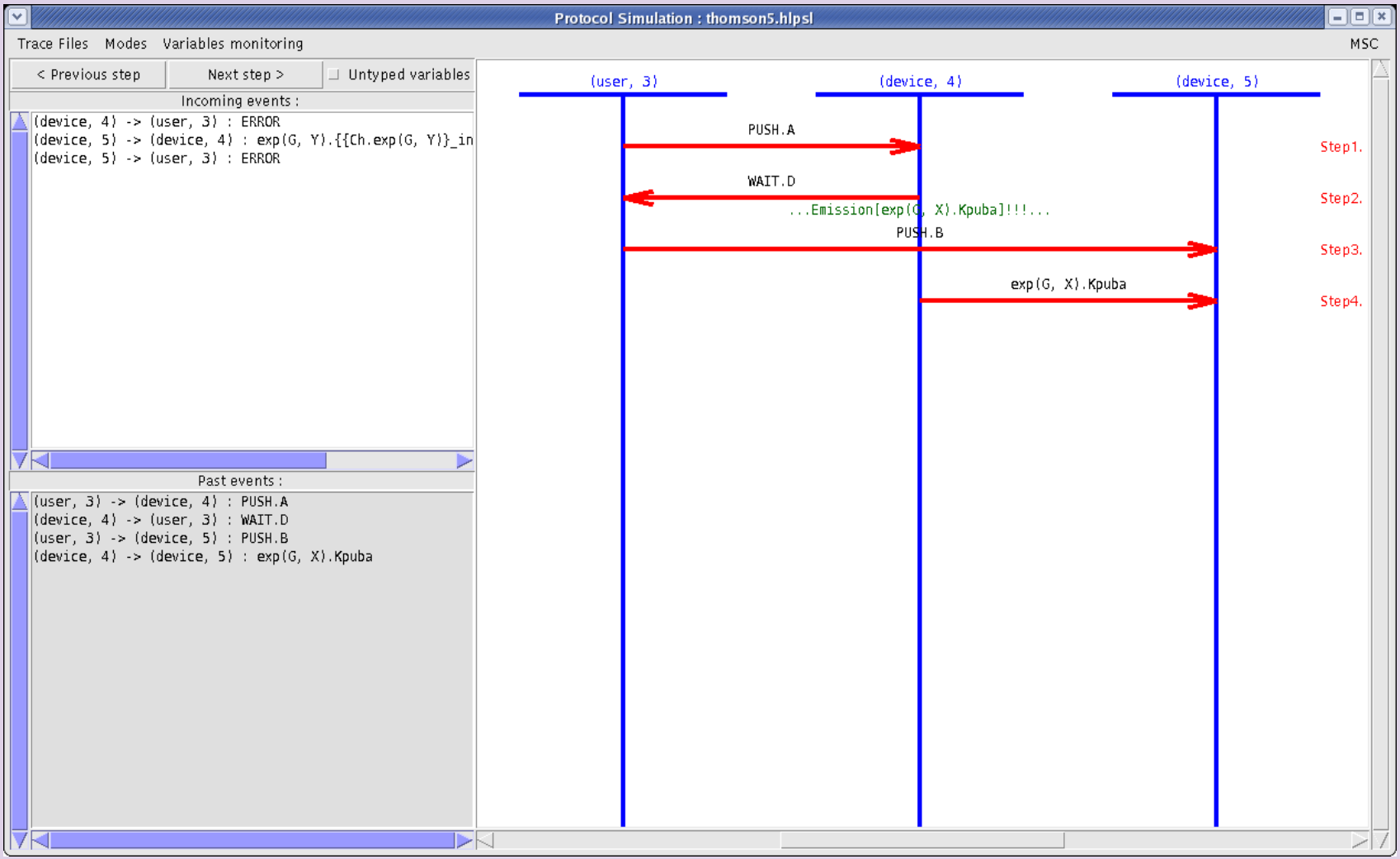
Experiments

An execution trace of Thomson's USDP protocol



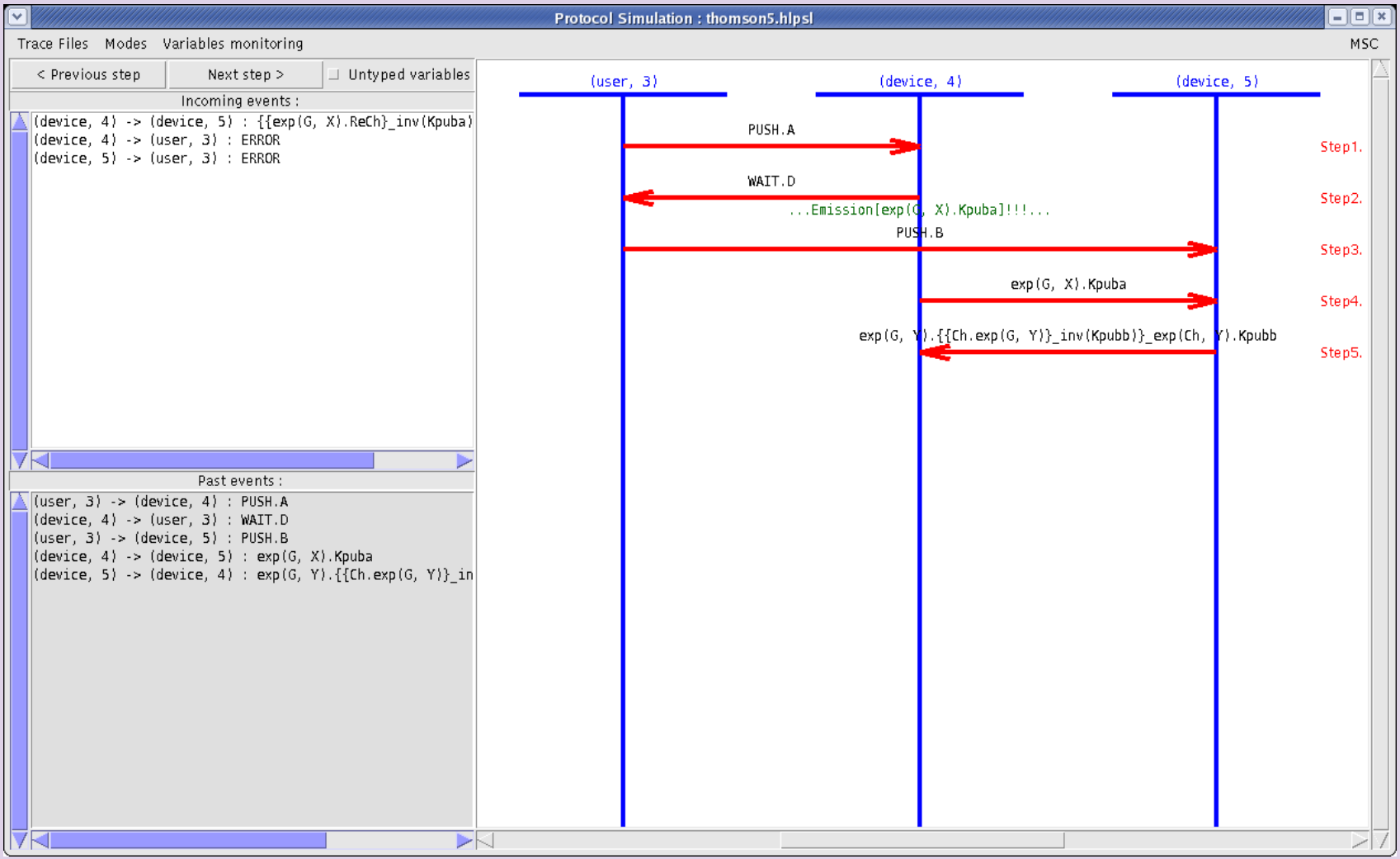
Experiments

An execution trace of Thomson's USDP protocol



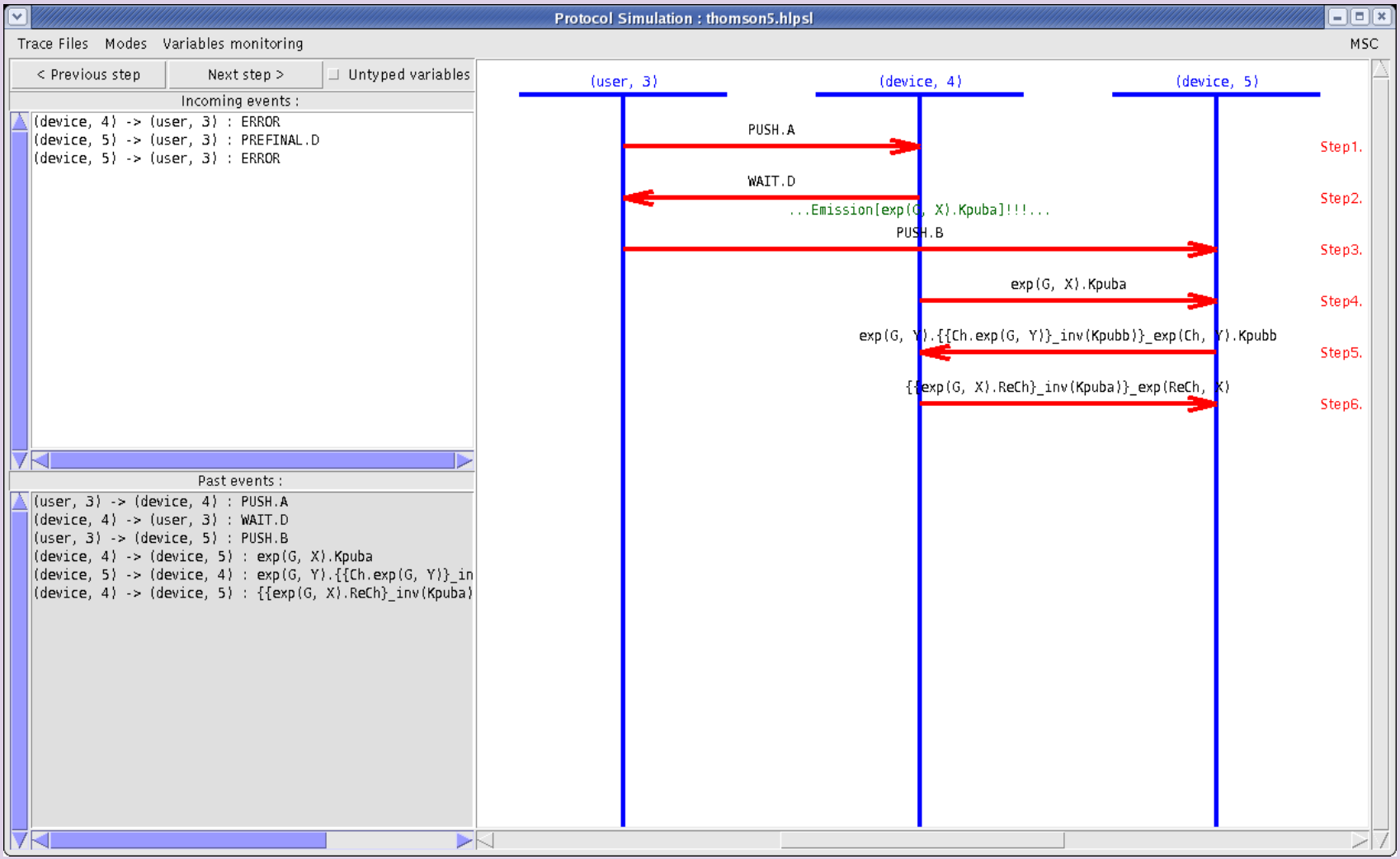
Experiments

An execution trace of Thomson's USDP protocol



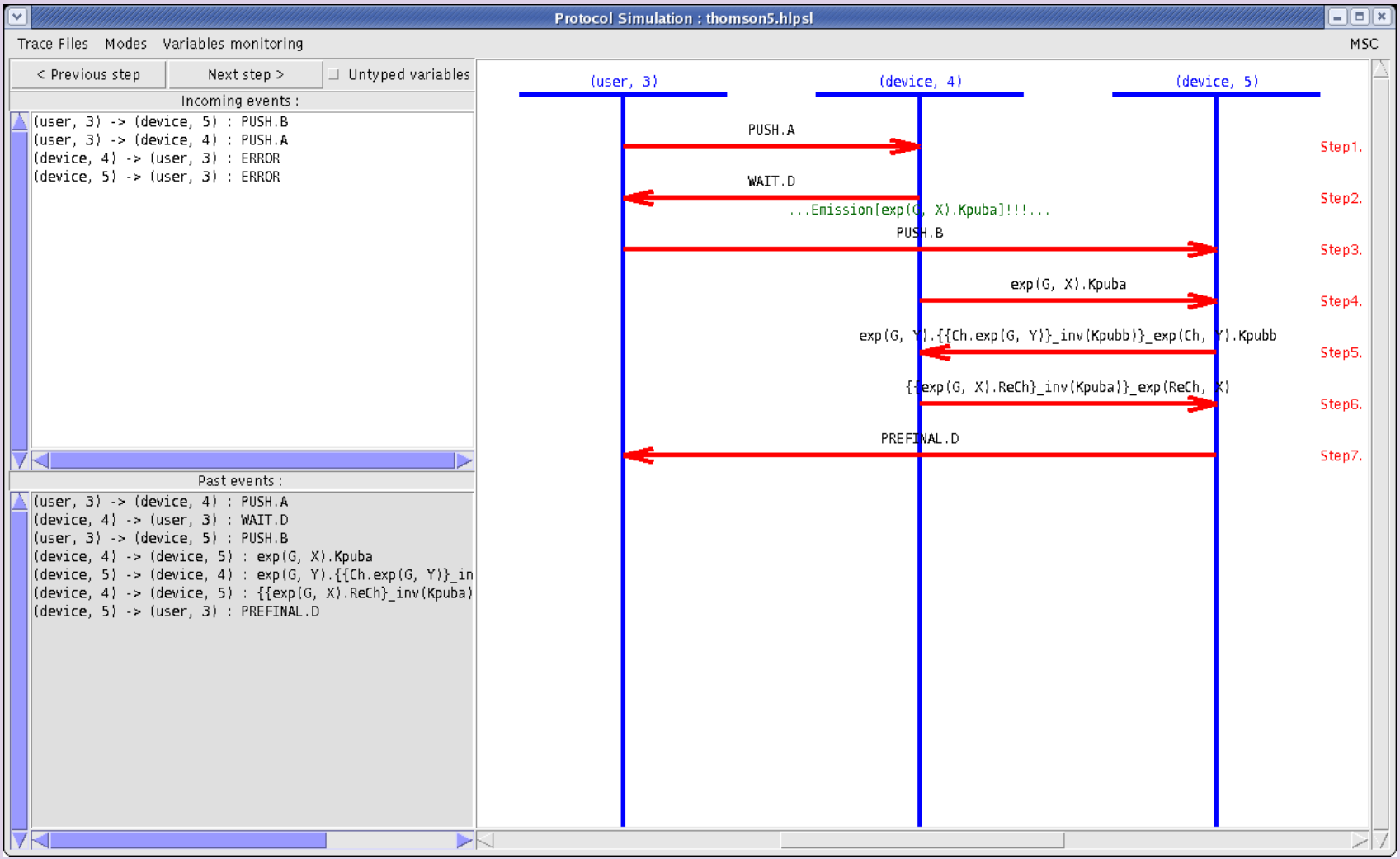
Experiments

An execution trace of Thomson's USDP protocol



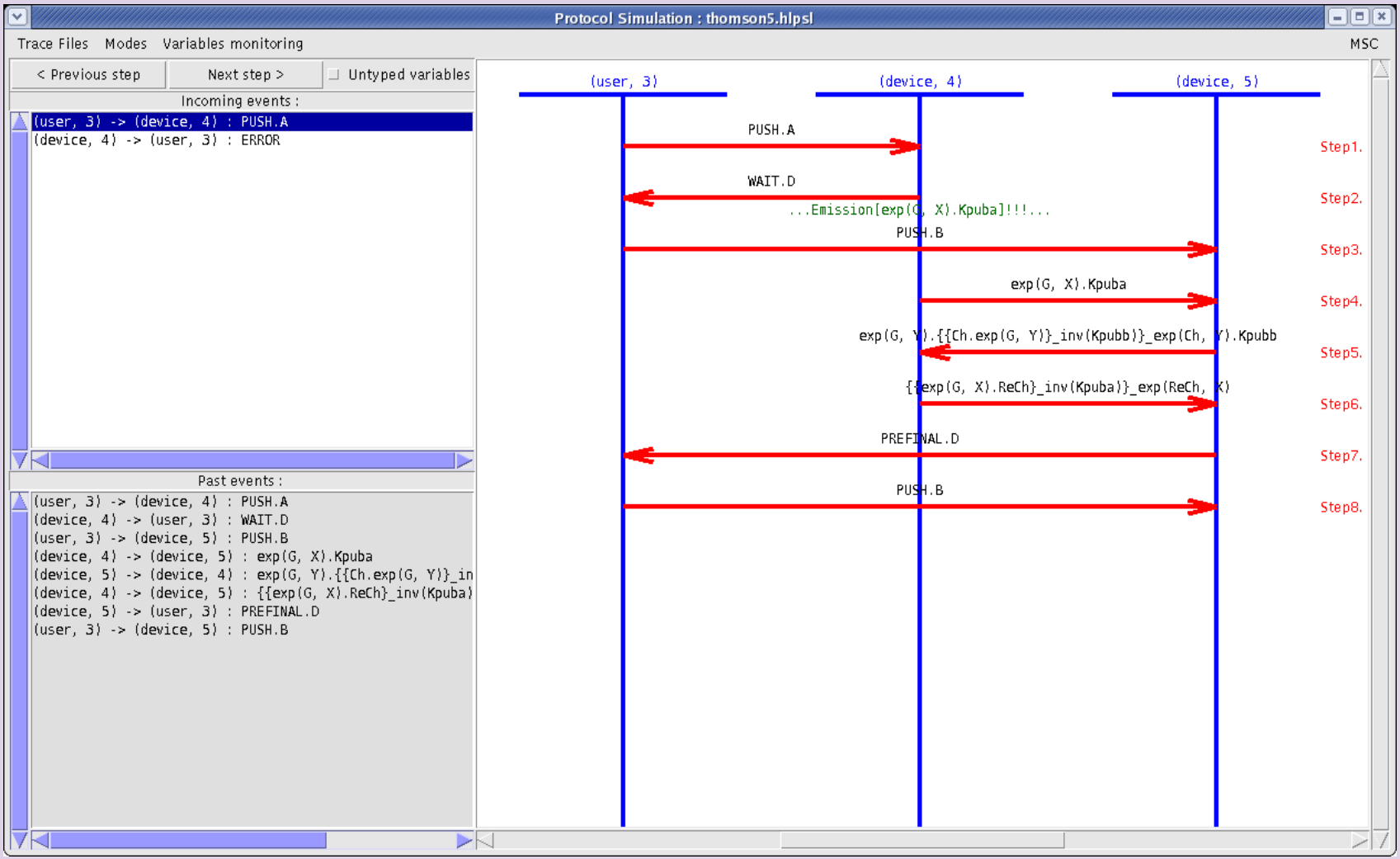
Experiments

An execution trace of Thomson's USDP protocol



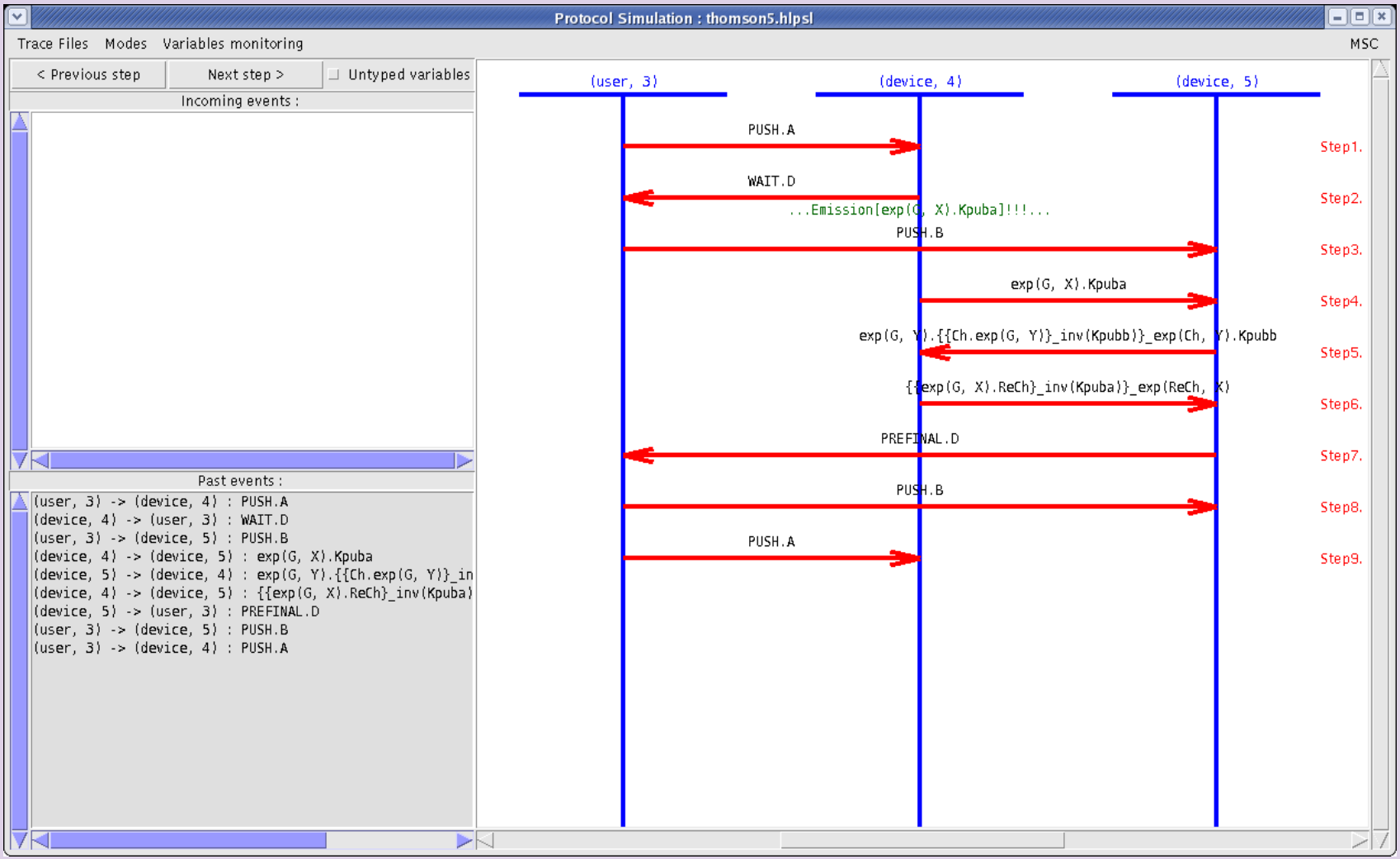
Experiments

An execution trace of Thomson's USDP protocol



Experiments

An execution trace of Thomson's USDP protocol



Futher Work

A correct treatment of mathematical functions

- This is not yet fully functional when messages include *exp*, *xor*.

Integration of a mode to replay interactively the attacks

- Execute an intruder role who receive, replay, and treat all messages sent by an agent.

Futher Work

A correct treatment of mathematical functions

- This is not yet fully functional when messages include *exp*, *xor*.

Integration of a mode to replay interactively the attacks

- Execute an intruder role who receive, replay, and treat all messages sent by an agent.