Verifying Security Protocols in Tamarin

Ralf Sasse

Institute of Information Security
ETH Zurich

Tamarin Day 5, v.1

Jan 29, 2016

Roadmap

1. **HISP**
2. **ARPKI**

Outline

1. **HISP**
2. **ARPKI**

HISP

See HISP slide set

Channels in Tamarin

- Usual communication via In/Out
- Channels with specific properties can be created by using Facts that sender writes to and receiver reads from – be very careful about their specification, or attacks may be missed
- Normal use has state Facts for each role, not shared Facts

Channels in Tamarin

- Usual communication via In/Out
- Channels with specific properties can be created by using Facts that sender writes to and receiver reads from – be very careful about their specification, or attacks may be missed
- Normal use has state Facts for each role, not shared Facts

- Secure (authentic and secret) channel
- Authentic channel
- Secret channel
- Fact name is irrelevant; systematic treatment possible, or ad-hoc

Secret Channels

As sender A, sending to B, on RHS of the rule:

$$\mathrm{SChan}(A, B, m)$$

On receiver B's side have a rule with the LHS:

$$\mathrm{SChan}(A, B, m)$$

Adversary can inject messages, so to pretend A sends m to B must add:

$$\mathrm{In}(A, B, m) \rightarrow SChan(A, B, m)$$

Only B can read it with a rule that has the fact on the LHS, but not authentic.

Authentic Channels

As sender A, sending to B, on RHS of the rule:

$$\mathrm{AChan}(A, B, m)$$

On receiver B's side have a rule with the LHS:

$$\mathrm{AChan}(A, B, m)$$

Adversary can eavesdrop messages, so must add:

$$AChan(A, B, m) \to \mathrm{Out}(A, B, m)$$

Only A can send it with a rule that has the fact on the RHS, but not secret.

Secure Channels

As sender A, sending to B, on RHS of the rule:

$$\mathrm{SAChan}(A, B, m)$$

On receiver B's side have a rule with the LHS:

$$\mathrm{SAChan}(A, B, m)$$

Only B can read it with a rule that has the fact on the LHS.

Adversary can neither inject messages, nor eavesdrop.

Outline

1. HISP

2. **ARPKI**

ARPKI

See ARPKI slide set

Conclusions

Now specify, and verify, your (own) protocols!