Analysing routing protocols: four nodes topologies are sufficient

Véronique Cortier, Jan Degrieck, and Stéphanie Delaune

LSV, ENS Cachan & CNRS & INRIA Saclay Île-de-France, France

Thursday, January 19th, 2012

Mobile ad hoc networks

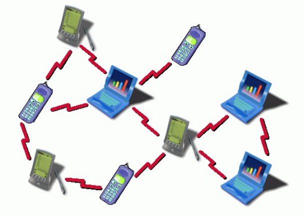
networks with no *a priori* infrastructure

$\longrightarrow$ nodes have to communicate to establish routes and allow the transfer of data from one node to another.

Mobile ad hoc networks

networks with no *a priori* infrastructure

$\longrightarrow$ nodes have to communicate to establish routes and allow the transfer of data from one node to another.



Some applications:

- military operations, and emergency disaster relief;
- self-organizing wireless sensor networks;
- vehicular ad hoc networks;
- wireless public access for dense urban areas.

Routing protocols

Routing protocols

Protocols series of rules describing how each participant should behave in order to achieve a common goal

Routing goal allowing distant nodes to communicate

Routing protocols

Routing protocols

Protocols series of rules describing how each participant should behave in order to achieve a common goal

Routing goal allowing distant nodes to communicate

Two main families:

- table routing protocols, *e.g.* AODV (1999):
  $\longrightarrow$ each node knows the following node on the route towards a destination. This information is stored in routing tables.
- source routing protocol, *e.g.* DSR (2001):
  $\longrightarrow$ the source node provides the entire route that the messages have to follow.

Routing protocols

Routing protocols

Protocols series of rules describing how each participant should behave in order to achieve a common goal

Routing goal allowing distant nodes to communicate

Two main families:

- table routing protocols, *e.g.* AODV (1999):
  $\longrightarrow$ each node knows the following node on the route towards a destination. This information is stored in routing tables.
- source routing protocol, *e.g.* DSR (2001):
  $\longrightarrow$ the source node provides the entire route that the messages have to follow.

Routing is fundamental service in any kind of networks

Secure versions of routing protocols

Goal: provide some guarantees even in an adversarial setting.

Examples: SAODV (2002), SRP applied on DSR (2002)

Secure versions of routing protocols

Goal: provide some guarantees even in an adversarial setting.

Examples: SAODV (2002), SRP applied on DSR (2002)

They rely on some security mechanisms:

- cryptographic primitives: *e.g.* signatures, encryptions, hash functions, MAC, . . .
- neighboorhood tests implemented using secure neighboorhood discovery protocols *e.g.* NDP protocol, SEND protocol, . . .

Secure versions of routing protocols

Goal: provide some guarantees even in an adversarial setting.

Examples: SAODV (2002), SRP applied on DSR (2002)

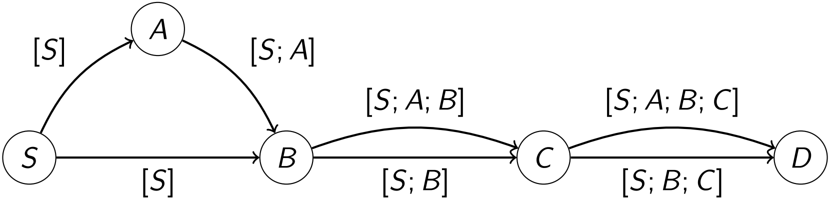
They rely on some security mechanisms:

- cryptographic primitives: *e.g.* signatures, encryptions, hash functions, MAC, ...
- neighboorhood tests implemented using secure neighboorhood discovery protocols *e.g.* NDP protocol, SEND protocol, ...

$\longrightarrow$ We will model those mechanisms in an abstract way.

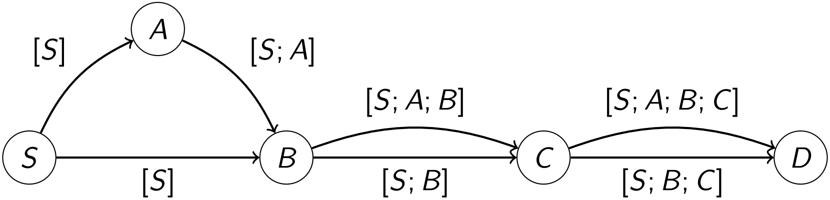
Example: SRP applied on DSR (1/2)

Request phase:



Example: SRP applied on DSR (1/2)

Request phase:



For security purposes:

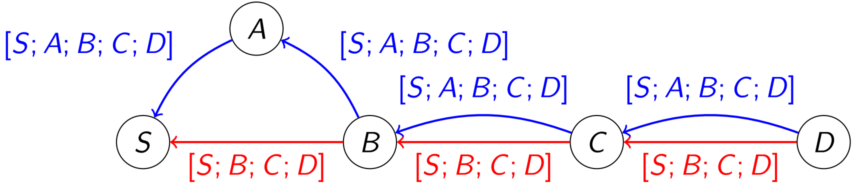
- the request contains in addition a mac built by the source:

$$mac(\langle req, S, D, id \rangle, shk(S, D))$$

- each intermediate node checks that the received request is locally correct before adding its name and relaying it over the network.

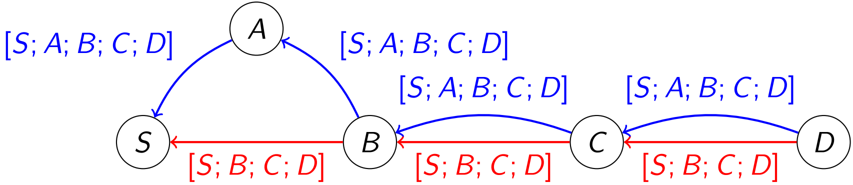
Example: SRP applied on DSR (2/2)

Reply phase:



Example: SRP applied on DSR (2/2)

Reply phase:



For security purposes:

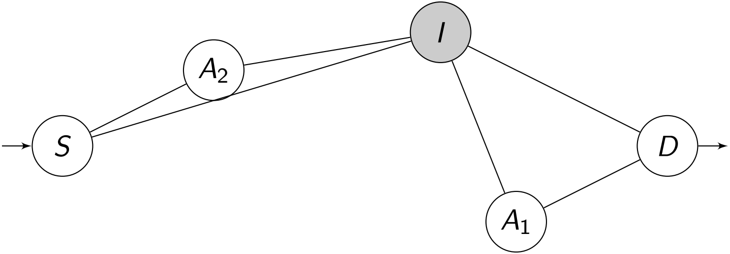
- the reply contains in addition a mac built by the destination:

$$mac(\langle rep, D, S, id, route \rangle, shk(S, D))$$

- each intermediate node checks that the received reply is locally correct before forwarding it to the next hop.

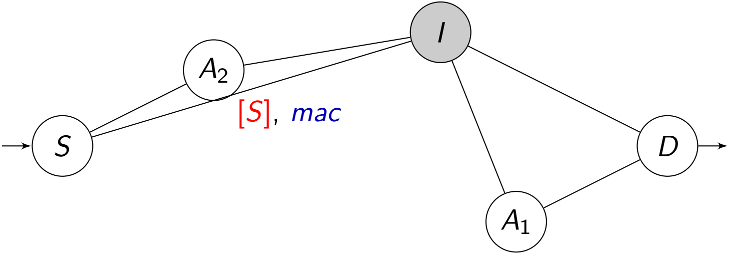
Attack on SRP applied on DSR

[Buttyán & Vajda, 2004]



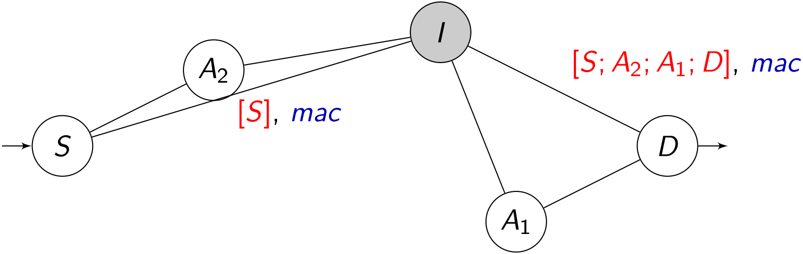
Attack on SRP applied on DSR

[Buttyán & Vajda, 2004]



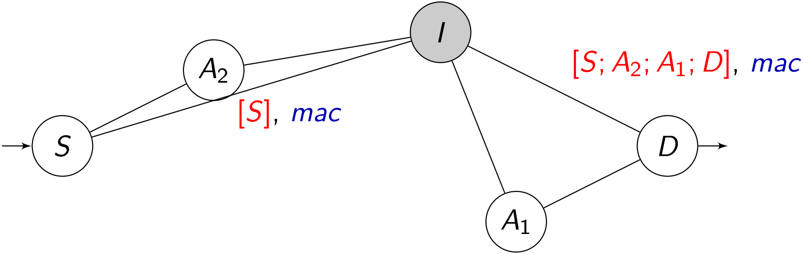
Attack on SRP applied on DSR

[Buttyán & Vajda, 2004]



Attack on SRP applied on DSR

[Buttyán & Vajda, 2004]



Reply phase:

1. D accepts the request and sends

$$rep, S, D, id, l_{\text{route}}, mac(\langle rep, S, D, id, l_{\text{route}}\rangle, shk(S, D))$$

2. I simply forwards this message to S.

Difficulties of the verification

Some automatic verification tools

- AVISPA platform [Armando *et al.*, 2005]
  $\longrightarrow$ state-of-the-art for bounded verification
- ProVerif [Blanchet *et al.*, 2001]
  $\longrightarrow$ quite flexible to analyse security properties and to deal with various cryptographic primitives

Difficulties of the verification

Some automatic verification tools

- AVISPA platform [Armando *et al.*, 2005]
  $\longrightarrow$ state-of-the-art for bounded verification
- ProVerif [Blanchet *et al.*, 2001]
  $\longrightarrow$ quite flexible to analyse security properties and to deal with various cryptographic primitives

Specificities of routing protocols

- **topology**: communication, the power of the attacker, security property, neigboorhood checks, ...
- an arbitrary number of agents can be involved in one session;
- they use lists and may perform some recursive operations.

Difficulties of the verification

Some automatic verification tools

- AVISPA platform [Armando *et al.*, 2005]
  $\longrightarrow$ state-of-the-art for bounded verification
- ProVerif [Blanchet *et al.*, 2001]
  $\longrightarrow$ quite flexible to analyse security properties and to deal with various cryptographic primitives

Specificities of routing protocols

- **topology**: communication, the power of the attacker, security property, neigboorhood checks, . . .
- an arbitrary number of agents can be involved in one session;
- they use lists and may perform some recursive operations.

None of the existing tools are well-suited to analyse routing protocols

Some related work

Case studies using some automatic tools

For instance, some case studies (*e.g.* ARAN, endairA) have been carried out using the AVISPA platform considering some arbitrary fixed topologies.

[Benetti *et al*, 2010]

General frameworks

Several frameworks have been proposed to model secure routing protocols.

e.g. [S. Nanz & C. Hanking, 2006] [G. Àcs, 2009]

Some related work

Case studies using some automatic tools

For instance, some case studies (*e.g.* ARAN, endairA) have been carried out using the AVISPA platform considering some arbitrary fixed topologies.
[Benetti *et al*, 2010]

General frameworks

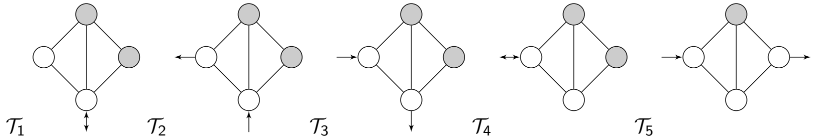
Several frameworks have been proposed to model secure routing protocols.
e.g. [S. Nanz & C. Hanking, 2006] [G. Àcs, 2009]

Decision procedures for a bounded number of sessions and arbitrary topologies, but no implementation exist. [Arnaud *et al.*, 2010]

Recently, a reduction result obtained by taking advantages of symmetries have been proposed. [Andel *et al*, 2011]
$\longrightarrow$ *However, the number of topologies is still infinite or really large even when considering a bounded number of nodes.*

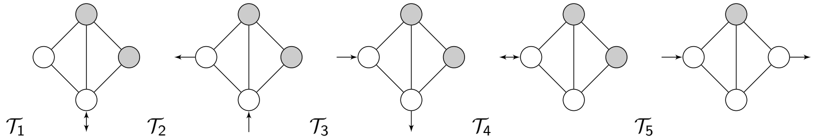
Our contributions

Reduction result: only 5 topologies are sufficient !



Our contributions

Reduction result: only 5 topologies are sufficient !



$\longrightarrow$ very **general model** encompassing many families of routing protocols with recursive tests/operations, various cryptographic primitives, various kind of neighbourhood checks.

Case studies: We use the tool ProVerif to analyse the SRP/DSR and the SDMSR protocols.

Outline

Outline

Messages

Messages are represented by terms built on a sorted signature.
$\longrightarrow$ regarding the sort system, we consider a special sort agent that only contains names and variables

Messages

Messages are represented by terms built on a sorted signature.
$\longrightarrow$ regarding the sort system, we consider a special sort agent that only contains names and variables

Example (signature)

- mac : term $\times$ term $\to$ term,
- $\langle\rangle$: term $\times$ term $\to$ term,
- shk : agent $\times$ agent $\to$ term,
- $::$: agent $\times$ list $\to$ list,
- $\perp$: $\to$ list,
- req, rep : $\to$ term.

Messages

Messages are represented by terms built on a sorted signature.
$\longrightarrow$ regarding the sort system, we consider a special sort agent that only contains names and variables

Attacker is modeled using a deduction relation defined through an arbitrary inference system.

Messages

Messages are represented by terms built on a sorted signature.
$\longrightarrow$ regarding the sort system, we consider a special sort agent that only contains names and variables

Attacker is modeled using a deduction relation defined through an arbitrary inference system.

Example (inference system)

$$\frac{y_1 \quad y_2}{\langle y_1, y_2\rangle} \quad \frac{\langle y_1, y_2\rangle}{y_1} \quad \frac{\langle y_1, y_2\rangle}{y_2} \quad \frac{x \quad z}{x :: z} \quad \frac{x :: z}{x} \quad \frac{x :: z}{z} \quad \frac{y_1 \quad y_2}{mac(y_1, y_2)}$$

Messages

Messages are represented by terms built on a sorted signature.
$\longrightarrow$ regarding the sort system, we consider a special sort agent that only contains names and variables

Attacker is modeled using a deduction relation defined through an arbitrary inference system.

Operations on received terms are modeled using functions over terms, *i.e.* functions of the form: $\mathsf{f} : \mathsf{term} \times \ldots \times \mathsf{term} \to \mathsf{term}$

Messages

Messages are represented by terms built on a sorted signature.
$\longrightarrow$ regarding the sort system, we consider a special sort agent that only contains names and variables

Attacker is modeled using a deduction relation defined through an arbitrary inference system.

Operations on received terms are modeled using functions over terms, *i.e.* functions of the form: f : term $\times \ldots \times$ term $\rightarrow$ term

Example (function over terms)

- standard application of cryptographic operations:

$$(x, y, z) \mapsto mac(\langle x, y \rangle, z)$$

- various operations on lists, *e.g.* reversal, concatenation, ...
- recursive operations and recursive tests used in many routing protocols, *e.g.* SMNDP, Ariadne, endairA, ...

Routing protocols

Definition

A routing protocol is a set of parametrized ground processes.

Routing protocols

Definition

A routing protocol is a set of parametrized ground processes.

Processes P, Q, R:

$\mathrm{out}(\mathrm{f}(u_1, \ldots, u_n)).P$	emission
$\mathrm{in}(u).P$	reception
if Φ then P	conditional
$P \mid Q$	parallel composition
$!P$	replication
new $n.P$	fresh name generation

Definition

A routing protocol is a set of parametrized ground processes.

Processes P, Q, R:

$\mathrm{out}(\mathrm{f}(u_1, \ldots, u_n)).P$	emission
$\mathrm{in}(u).P$	reception
if Φ then P	conditional
$P \mid Q$	parallel composition
$!P$	replication
new $n.P$	fresh name generation

Formulas Φ, Φ_1, Φ_2:

$p(u_1, \ldots, u_n)$	literal with $p \in \mathcal{P}$
$\Phi_1 \wedge \Phi_2$	conjunction

Example (SRP/DSR)

The routing protocol SRP/DSR can be modeled using the following set of parametrized processes:

$$\{P_{\mathsf{src}}(x_S, x_D);\ P_{\mathsf{request}}(x_V);\ P_{\mathsf{reply}}(x_V);\ P_{\mathsf{dest}}(x_D)\}.$$

Example (SRP/DSR)

The routing protocol SRP/DSR can be modeled using the following set of parametrized processes:

$$\{P_{\text{src}}(x_S, x_D);\ P_{\text{request}}(x_V);\ P_{\text{reply}}(x_V);\ P_{\text{dest}}(x_D)\}.$$

Source processes

$$P_{\text{src}}(x_S, x_D) =$$

Example (SRP/DSR)

The routing protocol SRP/DSR can be modeled using the following set of parametrized processes:

$$\{P_{\mathrm{src}}(x_S, x_D);\ P_{\mathrm{request}}(x_V);\ P_{\mathrm{reply}}(x_V);\ P_{\mathrm{dest}}(x_D)\}.$$

Source processes

$$P_{\mathrm{src}}(x_S, x_D) = \mathrm{new}\ id.$$

Example (SRP/DSR)

The routing protocol SRP/DSR can be modeled using the following set of parametrized processes:

$$\{P_{\mathrm{src}}(x_S, x_D);\; P_{\mathrm{request}}(x_V);\; P_{\mathrm{reply}}(x_V);\; P_{\mathrm{dest}}(x_D)\}.$$

Source processes

$$P_{\mathrm{src}}(x_S, x_D) = \mathrm{new}\ id.\ \mathrm{out}(u_1).$$

where

$$\left\{ u_1 = \langle req, x_S, x_D, id, [x_S], mac(\langle req, x_S, x_D, id\rangle, shk(x_S, x_D))\rangle \right.$$

Example (SRP/DSR)

The routing protocol SRP/DSR can be modeled using the following set of parametrized processes:

$$\{P_{\mathrm{src}}(x_S, x_D);\ P_{\mathrm{request}}(x_V);\ P_{\mathrm{reply}}(x_V);\ P_{\mathrm{dest}}(x_D)\}.$$

Source processes

$$P_{\mathrm{src}}(x_S, x_D) = \mathrm{new}\ id.\ \mathrm{out}(u_1).\ \mathrm{in}(u_2).$$

where

$$\begin{cases} u_1 = \langle req, x_S, x_D, id, [x_S], mac(\langle req, x_S, x_D, id\rangle, shk(x_S, x_D))\rangle \\ u_2 = \langle rep, x_D, x_S, id, x_L, mac(\langle rep, x_D, x_S, id, x_L\rangle, shk(x_S, x_D))\rangle \\ \end{cases}$$

$\longrightarrow$ at the end of the execution, x_L should contain a route from S to D.

Example (SRP/DSR)

The routing protocol SRP/DSR can be modeled using the following set of parametrized processes:

$$\{P_{\mathrm{src}}(x_S, x_D);\ P_{\mathrm{request}}(x_V);\ P_{\mathrm{reply}}(x_V);\ P_{\mathrm{dest}}(x_D)\}.$$

Source processes

$$P_{\mathrm{src}}(x_S, x_D) = \mathrm{new}\ id.\ \mathrm{out}(u_1).\ \mathrm{in}(u_2).\ \mathrm{if}\ \Phi_S\ \mathrm{then}\ 0$$

where

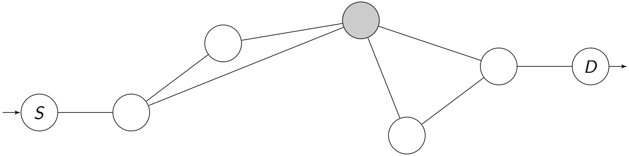
$$\begin{cases} u_1 = \langle req, x_S, x_D, id, [x_S], mac(\langle req, x_S, x_D, id\rangle, shk(x_S, x_D))\rangle \\ u_2 = \langle rep, x_D, x_S, id, x_L, mac(\langle rep, x_D, x_S, id, x_L\rangle, shk(x_S, x_D))\rangle \\ \Phi_S = \mathrm{checkl}(x_S, x_L) \wedge \mathrm{first}(x_S, x_L) \wedge \mathrm{last}(x_D, x_L) \end{cases}$$

$\longrightarrow$ at the end of the execution, x_L should contain a route from S to D.

Configuration and topology

A *topology* is given by a tuple $\mathcal{T} = (G, \mathcal{M}, S, D)$.

Example

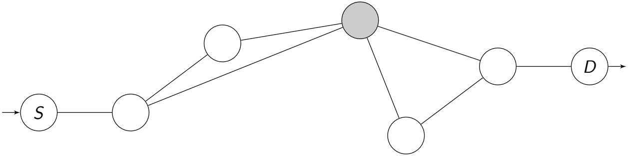


$\longrightarrow$ the attackers do not necessarily control the entire network.

Configuration and topology

A *topology* is given by a tuple $\mathcal{T} = (G, \mathcal{M}, S, D)$.

Example



$\longrightarrow$ the attackers do not necessarily control the entire network.

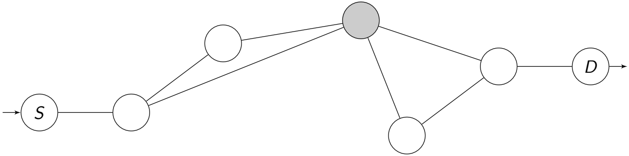
A *configuration* is a pair $(\mathcal{P}; \mathcal{I})$ where:

- $\mathcal{P}$ is a multiset of expressions of the form $\lfloor P \rfloor_A$;
- $\mathcal{I}$ is a set of terms representing the knowledge of the attackers.

Configuration and topology

A *topology* is given by a tuple $\mathcal{T} = (G, \mathcal{M}, S, D)$.

Example



$\longrightarrow$ the attackers do not necessarily control the entire network.

A *configuration* is a pair $(\mathcal{P}; \mathcal{I})$ where:

- $\mathcal{P}$ is a multiset of expressions of the form $\lfloor P \rfloor_A$;
- $\mathcal{I}$ is a set of terms representing the knowledge of the attackers.

$\longrightarrow$ the operational semantics is given by a transition system $\rightarrow_{\mathcal{T}}$
(only local communications are allowed)

What is an attack?

Security property

Intuitively, a valid route between S and D is a route that represents a path from S to D.

$\longrightarrow$ too strong (*e.g.* so-called wormhole and hidden channel attacks)

An admissible path is a path in which two consecutive nodes that are non-adjacent are both malicious.

What is an attack?

Security property

Intuitively, a valid route between S and D is a route that represents a path from S to D.

$\longrightarrow$ too strong (*e.g.* so-called wormhole and hidden channel attacks)

An admissible path is a path in which two consecutive nodes that are non-adjacent are both malicious.

What is an attack? $\longrightarrow$ an attack is modeled as a reachability property

Example: SRP/DSR protocol

$$P_0(x_S, x_D) = \text{new } id.\ \text{out}(u_1).\ \text{in}(u_2).\ \text{if } \Phi_S \text{ then } \text{out}(end(x_L))$$

Given a topology $\mathcal{T}$ and a configuration K,

$$K \text{ admits an attack in } \mathcal{T} \text{ if } K \rightarrow^*_{\mathcal{T}} (\lfloor \text{out}(end(l)).P \rfloor_A \cup \mathcal{P}; \mathcal{I})$$

where l is not an admissible path in $\mathcal{T}$.

Outline

Outline

Reduction results

Goal: allow one to analyse the security of a routing protocol considering only some specific and small topologies.

Reduction results

Goal: allow one to analyse the security of a routing protocol considering only some specific and small topologies.

We show that the existence of an attack is preserved
Step 1: when adding edges to the graph, yielding a quasi-complete topology;

$\longrightarrow$ protocols have to be completion-friendly, *i.e.*

$$[\![p(u_1, \ldots, u_k)]\!]_G = \text{true implies } [\![p(u_1, \ldots, u_k)]\!]_{G^+} = \text{true}$$

Reduction results

Goal: allow one to analyse the security of a routing protocol considering only some specific and small topologies.

We show that the existence of an attack is preserved
Step 1: when adding edges to the graph, yielding a quasi-complete topology;

$\longrightarrow$ protocols have to be completion-friendly, *i.e.*

$$[\![p(u_1, \ldots, u_k)]\!]_G = \text{true implies } [\![p(u_1, \ldots, u_k)]\!]_{G^+} = \text{true}$$

Step 2: when merging nodes that have the same neighbourhood and same honesty status, yielding a small graph.

$\longrightarrow$ protocols have to be projection-friendly, *i.e.*

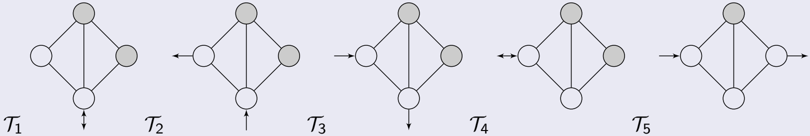
$[\![p(u_1, \ldots, u_k)]\!]_G = \text{true implies that } [\![p(u_1\rho, \ldots, u_k\rho)]\!]_{G\rho} = \text{true}$
$\mathsf{f}(u_1\rho, \ldots, u_k\rho) = \mathsf{f}(u_1, \ldots, u_k)\rho$.

Main result

Only five topologies are sufficient !

Theorem

Let $\mathcal{P}_{\text{routing}}$ be a routing protocol that is completion-friendly and projection-friendly. $\mathcal{P}_{\text{routing}}$ admits an attack if, and only if, it admits an attack for one of the topologies below:

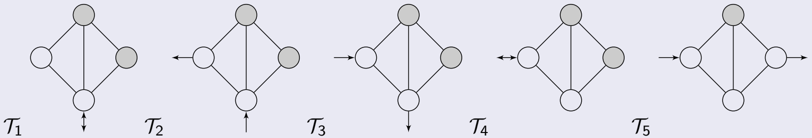


Main result

Only five topologies are sufficient !

Theorem

Let $\mathcal{P}_{\text{routing}}$ be a routing protocol that is completion-friendly and projection-friendly. $\mathcal{P}_{\text{routing}}$ admits an attack if, and only if, it admits an attack for one of the topologies below:

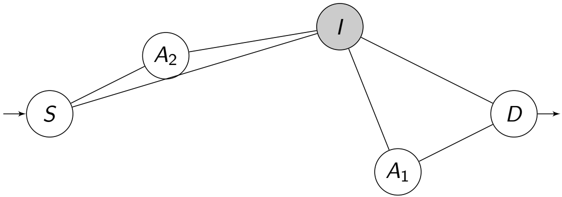


$\longrightarrow$ very **general model** encompassing many families of routing protocols with recursive tests/operations, various cryptographic primitives, various kind of neighbourhood checks.

Example SRP/DSR [Buttyán & Vajda, 2004]

Attack on the topology $\mathcal{T}$

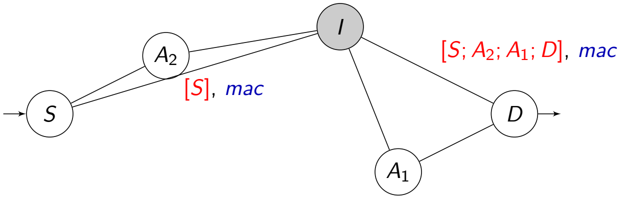
$\longrightarrow$ S accepts $[S; A_2; A_1; D]$



Example SRP/DSR [Buttyán & Vajda, 2004]

Attack on the topology $\mathcal{T}$

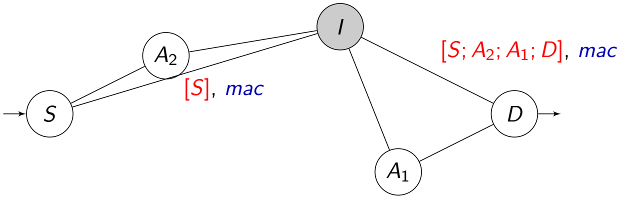
$\longrightarrow$ S accepts $[S; A_2; A_1; D]$



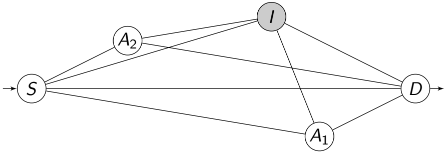
Example SRP/DSR [Buttyán & Vajda, 2004]

Attack on the topology $\mathcal{T}$

$\longrightarrow$ S accepts $[S; A_2; A_1; D]$



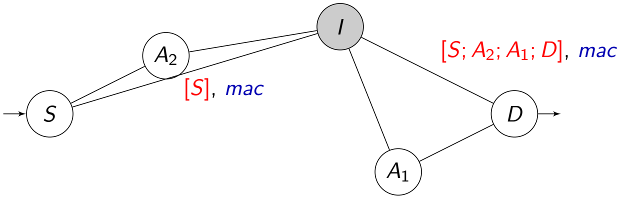
Step 1: Quasi-complete topology



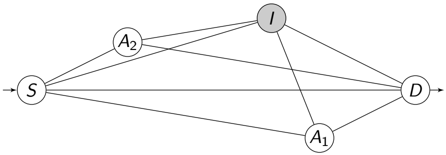
Example SRP/DSR [Buttyán & Vajda, 2004]

Attack on the topology $\mathcal{T}$

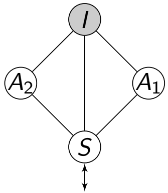
$\longrightarrow$ S accepts $[S; A_2; A_1; D]$



Step 1: Quasi-complete topology



Step 2: Reduced topology



Outline

ProVerif

Automated protocol verifier mainly developed by B. Blanchet.

`http://www.proverif.ens.fr/`

ProVerif

Automated protocol verifier mainly developed by B. Blanchet.

`http://www.proverif.ens.fr/`

Main features

- unbounded number of sessions;
- various cryptographic primitives modeled using rewriting rules and equations;
  $\longrightarrow$ not arbitrary functions over terms as we did
- an attacker who controls the entire network
  $\longrightarrow$ this is not a problem for the 5 topologies we have to analyse
- various security properties
  $\longrightarrow$ we can easily encode our security property but also neigbourhood checks by defining predicates using Horn clauses.

ProVerif

Automated protocol verifier mainly developed by B. Blanchet.

`http://www.proverif.ens.fr/`

Main features

- unbounded number of sessions;
- various cryptographic primitives modeled using rewriting rules and equations;
  $\longrightarrow$ not arbitrary functions over terms as we did
- an attacker who controls the entire network
  $\longrightarrow$ this is not a problem for the 5 topologies we have to analyse
- various security properties
  $\longrightarrow$ we can easily encode our security property but also neigbourhood checks by defining predicates using Horn clauses.

The tool may not terminate or give false attacks. It works well in practice.

Some case studies

Two case studies have been performed using ProVerif:

- SRP applied on DSR [Papadimitratos & Haas, 02]
- SDMSR that relies on signatures [Berton *et al.*, 06]

Some case studies

Two case studies have been performed using ProVerif:

- SRP applied on DSR [Papadimitratos & Haas, 02]
- SDMSR that relies on signatures [Berton *et al.*, 06]

Results

	SRP applied on DSR	SDMSR
$\mathcal{T}_1$	attack found	attack found
$\mathcal{T}_2$	attack found	attack found
$\mathcal{T}_3$	no attack found	no attack found
$\mathcal{T}_4$	no attack found	no attack found
$\mathcal{T}_5$	no attack found	no attack found

$\longrightarrow$ the running time of ProVerif was less than a few secondes.

Two case studies have been performed using ProVerif:

- SRP applied on DSR [Papadimitratos & Haas, 02]
- SDMSR that relies on signatures [Berton *et al.*, 06]

Results

	SRP applied on DSR	SDMSR
$\mathcal{T}_1$	attack found	attack found
$\mathcal{T}_2$	attack found	attack found
$\mathcal{T}_3$	no attack found	no attack found
$\mathcal{T}_4$	no attack found	no attack found
$\mathcal{T}_5$	no attack found	no attack found

$\longrightarrow$ the running time of ProVerif was less than a few secondes.

All the files for these experiments are available at:

`http://www.lsv.ens-cachan.fr/~delaune/RoutingProtocols.`

Outline

Conclusion

Our contribution

We have shown a simple reduction result that allows one to use standard verification tools for analysing routing protocols.

Conclusion

Our contribution

We have shown a simple reduction result that allows one to use standard verification tools for analysing routing protocols.

Some Perspectives:

- our model is very general but we only consider tests that are stable under projection of nodes names
  $\longrightarrow$ *e.g.* we can not handle disequality tests
- our work is limited to a single (crucial) property: the validity of the resulting route
  $\longrightarrow$ Which security properties are relevant for routing protocols?
- we do not model mobility during the execution of the protocol.
  $\longrightarrow$ What is the appropriate security property in this case?