Presentation of the ProVerif tool

Stéphanie Delaune

January 2018

ProVerif [Blanchet, 01]

ProVerif is a verifier for cryptographic protocols that may prove that a protocol is secure or exhibit attacks.

`http://proverif.inria.fr`

Advantages

- fully automatic, and quite efficient
- a rich process algebra: replication, else branches, ...
- handles many cryptographic primitives
- various security properties: secrecy, correspondences, equivalences

ProVerif [Blanchet, 01]

ProVerif is a verifier for cryptographic protocols that may prove that a protocol is secure or exhibit attacks.

`http://proverif.inria.fr`

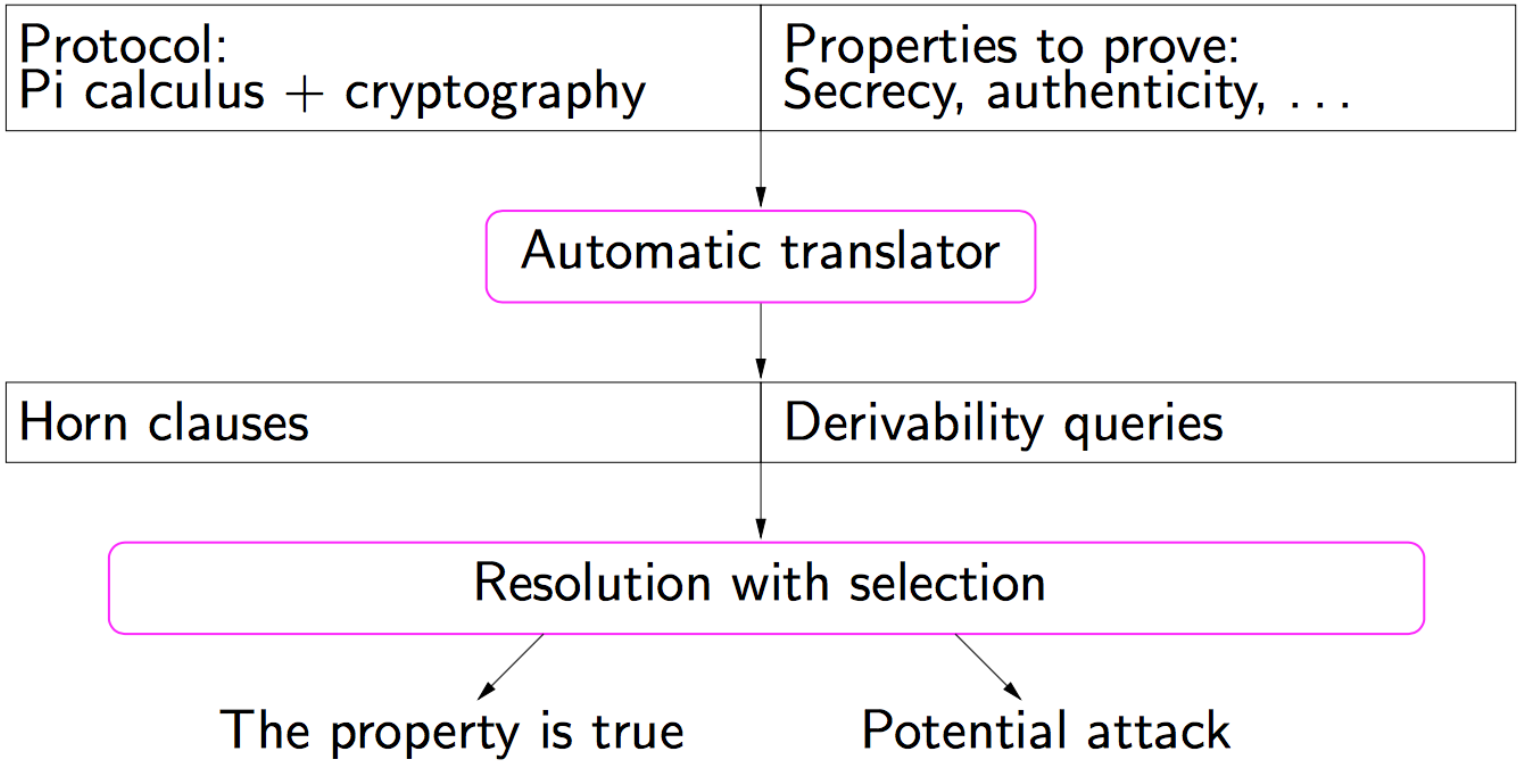
Advantages

- fully automatic, and quite efficient
- a rich process algebra: replication, else branches, ...
- handles many cryptographic primitives
- various security properties: secrecy, correspondences, equivalences

No miracle

- the tool can say "can not be proved";
- termination is not guaranteed

How does ProVerif work?



Some vocabulary

First order logic

Atoms	$P(t_1, \ldots, t_n)$ where t_i are terms, P is a predicate
Literals	$P(t_1, \ldots, t_n)$ or $\neg P(t_1, \ldots, t_n)$
closed under	$\vee, \wedge, \neg, \exists, \forall$

Clauses: Only universal quantifiers

Horn Clauses: at most one positive literal (where A_i, B are atoms.)

$$\forall \tilde{x}.\ A_1, \ldots, A_n \Rightarrow B$$

Modelling using Horn clauses

Modelling the attacker

Horn clauses $\mathcal{C}_{att}$ reflects the capabilities of the attacker.

$$
\begin{array}{rcll}
\mathsf{att}(x),\ \mathsf{att}(y) & \Rightarrow & \mathsf{att}(\langle x, y\rangle) & \text{pairing} \\
\mathsf{att}(\langle x, y\rangle) & \Rightarrow & \mathsf{att}(x) & \text{projection} \\
\mathsf{att}(\langle x, y\rangle) & \Rightarrow & \mathsf{att}(y) & \text{projection}
\end{array}
$$

$$
\begin{array}{rcll}
\mathsf{att}(x),\ \mathsf{att}(y) & \Rightarrow & \mathsf{att}(\{x\}_y) & \text{encryption} \\
\mathsf{att}(\{x\}_y), \mathsf{att}(y) & \Rightarrow & \mathsf{att}(x) & \text{decryption}
\end{array}
$$

Modelling the protocol (on an example)

$\{pin\}_{Ka}$ ←

$\{\{pin\}_{Ka}\}_{Kb}$ →

$\{pin\}_{Kb}$ ←

$\longrightarrow \{\{pin\}_{Ka}\}_{Kb} = \{\{pin\}_{Kb}\}_{Ka}.$

Modelling the protocol (on an example)

$\{pin\}_{Ka}$ ←

$\{\{pin\}_{Ka}\}_{Kb}$ →

$\{pin\}_{Kb}$ ←

This protocol does not work! (authentication problem)

Modelling the protocol (on an example)

$\{pin\}_{Ka}$ ←

$\{\{pin\}_{Ka}\}_{Kb}$ →

$\{pin\}_{Kb}$ ←

This protocol does not work! (authentication problem)

$\{pin\}_{Ka}$ ←

$\{\{pin\}_{Ka}\}_{Ki}$ →

$\{pin\}_{Ki}$ ←

Modelling the protocol (using Horn clauses)

Protocol:

$$
\begin{array}{lcl}
A \to B & : & \{\mathsf{pin}\}_{K_a} \\
B \to A & : & \{\{\mathsf{pin}\}_{K_a}\}_{K_b} \\
A \to B & : & \{\mathsf{pin}\}_{K_b}
\end{array}
$$

Horn clauses $\mathcal{C}_P$:

$$
\begin{array}{rcl}
 & \Rightarrow & \mathsf{att}(\{\mathsf{pin}\}_{K_a}) \\
\mathsf{att}(x) & \Rightarrow & \mathsf{att}(\{x\}_{K_b}) \\
\mathsf{att}(\{x\}_{K_a}) & \Rightarrow & \mathsf{att}(x)
\end{array}
$$

$\longrightarrow$ These clauses model an arbitrary number of executions of the protocol between the two honest participants A and B.

Modelling the security property

We consider secrecy as a reachability (accessibility) property, and we consider the Horn clause

$$\neg \mathrm{att}(\mathrm{pin})$$

There exists an attack (in this model) iff $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg \mathrm{att}(\mathrm{pin})$ is NOT satisfiable.

Modelling the security property

We consider secrecy as a reachability (accessibility) property, and we consider the Horn clause

$$\neg \mathrm{att}(\mathrm{pin})$$

There exists an attack (in this model) iff $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg \mathrm{att}(\mathrm{pin})$ is NOT satisfiable.

Exercise

Do you think that $\mathcal{C}_{prot} + \mathcal{C}_{att} + \neg \mathrm{att}(\mathrm{pin})$ is satisfiable or not? Justify your answer.
What about $\mathcal{C}_{prot} + \mathcal{C}_{att}$? and $\mathcal{C}_{prot}$?

How to decide satisfiability?

$\longrightarrow$ using resolution techniques

Binary resolution

$$\frac{\neg A \vee C \quad B \vee D}{C\theta \vee D\theta}\ \theta = \mathrm{mgu}(A, B) \quad \text{Binary resolution}$$

Theorem (Soundness and Completeness)

Binary resolution is sound and refutationally complete for Horn clauses, i.e. a set of Horn clauses $\mathcal{C}$ is not satisfiable if and only if $\square$ (the empty clause) can be obtained from $\mathcal{C}$ by binary resolution.

Example

$$\mathcal{C} = \{\neg\mathsf{att}(s), \quad \mathsf{att}(k_1), \quad \mathsf{att}(\{s\}_{\langle k_1,k_1\rangle}),$$
$$\mathsf{att}(\{x\}_y), \mathsf{att}(y) \Rightarrow \mathsf{att}(x), \qquad \mathsf{att}(x), \mathsf{att}(y) \Rightarrow \mathsf{att}(\langle x, y\rangle)\}$$

Example

$$\mathcal{C} = \{\neg\mathsf{att}(s),\quad \mathsf{att}(k_1),\quad \mathsf{att}(\{s\}_{\langle k_1,k_1\rangle}),$$
$$\mathsf{att}(\{x\}_y), \mathsf{att}(y) \Rightarrow \mathsf{att}(x),\qquad \mathsf{att}(x), \mathsf{att}(y) \Rightarrow \mathsf{att}(\langle x, y\rangle)\}$$

$$\dfrac{\dfrac{\mathsf{att}(\{s\}_{\langle k_1,k_1\rangle}) \quad \mathsf{att}(\{x\}_y), \mathsf{att}(y) \Rightarrow \mathsf{att}(x)}{\mathsf{att}(\langle k_1, k_1\rangle) \Rightarrow \mathsf{att}(s)} \quad \dfrac{\mathsf{att}(k_1) \qquad \dfrac{\mathsf{att}(k_1) \quad \mathsf{att}(x), \mathsf{att}(y) \Rightarrow \mathsf{att}(\langle x, y\rangle)}{\mathsf{att}(y) \Rightarrow \mathsf{att}(\langle k_1, y\rangle)}}{\mathsf{att}(\langle k_1, k_1\rangle)} \qquad \neg\mathsf{att}(s)}{\Box} \qquad \mathsf{att}(s)$$

But it is not terminating!

$$\dfrac{\mathsf{att}(y) \Rightarrow \mathsf{att}(\langle s, y\rangle) \quad \dfrac{\dfrac{\mathsf{att}(y) \Rightarrow \mathsf{att}(\langle s, y\rangle) \quad \dfrac{\mathsf{att}(s) \quad \dfrac{\mathsf{att}(s) \quad \mathsf{att}(x), \mathsf{att}(y) \Rightarrow \mathsf{att}(\langle x, y\rangle)}{\mathsf{att}(y) \Rightarrow \mathsf{att}(\langle s, y\rangle)}}{\mathsf{att}(\langle s, s\rangle)}}{\mathsf{att}(\langle s, \langle s, s\rangle\rangle)}}{\dfrac{\mathsf{att}(\langle s, \langle s, \langle s, s\rangle\rangle\rangle)}{\ldots}}$$

$\rightarrow$ This does not yield any decidability result.

How does ProVerif work?

ProVerif in a nutshell

Two main ideas (extending [Weidenbach, CADE'99]):

1. a simple abstract representation of these protocols, by a set of Horn clauses;

   $\longrightarrow$ relying on parametrized terms (called patterns)

2. an efficient solving algorithm based on resolution to find which facts can be derived from these clauses.

   $\longrightarrow$ ordered resolution with selection

Using this, ProVerif can prove secrecy properties of protocols, or exhibit attacks showing why a message is not secret.

Modelling the attacker using Horn clauses

Public key encryption

$$\begin{aligned} \mathsf{att}(x) &\Rightarrow \mathsf{att}(\mathsf{pk}(x)) \\ \mathsf{att}(x),\ \mathsf{att}(\mathsf{pk}(y)) &\Rightarrow \mathsf{att}(\mathsf{aenc}(x, \mathsf{pk}(y))) \\ \mathsf{att}((\mathsf{aenc}(x, \mathsf{pk}(y))),\ \mathsf{att}(y) &\Rightarrow \mathsf{att}(x) \end{aligned}$$

Modelling the attacker using Horn clauses

Public key encryption

$$\begin{aligned} \mathrm{att}(x) &\Rightarrow \mathrm{att}(\mathrm{pk}(x)) \\ \mathrm{att}(x),\ \mathrm{att}(\mathrm{pk}(y)) &\Rightarrow \mathrm{att}(\mathrm{aenc}(x, \mathrm{pk}(y))) \\ \mathrm{att}((\mathrm{aenc}(x, \mathrm{pk}(y))),\ \mathrm{att}(y) &\Rightarrow \mathrm{att}(x) \end{aligned}$$

Signature

$$\begin{aligned} \mathrm{att}(x),\ \mathrm{att}(y) &\Rightarrow \mathrm{att}(\mathrm{sign}(x, y)) \\ \mathrm{att}(\mathrm{sign}(x, y)) &\Rightarrow \mathrm{att}(x) \end{aligned}$$

Symmetric encryption

$$\begin{aligned} \mathrm{att}(x),\ \mathrm{att}(y) &\Rightarrow \mathrm{att}(\mathrm{senc}(x, y)) \\ \mathrm{att}((\mathrm{senc}(x, y)),\ \mathrm{att}(y) &\Rightarrow \mathrm{att}(x) \end{aligned}$$

Initial knowledge

$$\Rightarrow \mathrm{att}(\mathrm{pk}(sk_A)) \qquad \Rightarrow \mathrm{att}(sk_I) \qquad \Rightarrow \mathrm{att}(\mathrm{pk}(sk_B))$$

Modelling the protococol using Horn clauses

Denning-Sacco protocol . . .

$$\begin{aligned} A \to B &: \mathsf{aenc}(\mathsf{sign}(k, \mathsf{priv}(A)), \mathsf{pub}(B)) \\ B \to A &: \mathsf{senc}(s, k) \end{aligned}$$

. . . using Horn clauses

Modelling the protococol using Horn clauses

Denning-Sacco protocol . . .

$$\begin{array}{lcl} A \to B & : & \mathsf{aenc}(\mathsf{sign}(k, \mathsf{priv}(A)), \mathsf{pub}(B)) \\ B \to A & : & \mathsf{senc}(s, k) \end{array}$$

. . . using Horn clauses

- A talks with any principal represented by its public key $\mathsf{pk}(x)$.

$$\mathsf{att}(\mathsf{pk}(x)) \Rightarrow \mathsf{att}(\mathsf{aenc}(\mathsf{sign}(k, sk_A), \mathsf{pk}(x)))$$

Modelling the protococol using Horn clauses

Denning-Sacco protocol . . .

$$\begin{array}{lcl} A \to B & : & \mathrm{aenc}(\mathrm{sign}(k, \mathrm{priv}(A)), \mathrm{pub}(B)) \\ B \to A & : & \mathrm{senc}(s, k) \end{array}$$

. . . using Horn clauses

- A talks with any principal represented by its public key $\mathrm{pk}(x)$.

$$\mathrm{att}(\mathrm{pk}(x)) \Rightarrow \mathrm{att}(\mathrm{aenc}(\mathrm{sign}(k, sk_A), \mathrm{pk}(x)))$$

- When B receives a message of the expected form, he replies accordingly

$$\mathrm{att}(\mathrm{aenc}(\mathrm{sign}(y, sk_A), \mathrm{pk}(sk_B))) \Rightarrow \mathrm{att}(\mathrm{senc}(s, y))$$

Modelling the protococol using Horn clauses

Denning-Sacco protocol . . .

$$\begin{array}{lcl} A \to B & : & \mathsf{aenc}(\mathsf{sign}(k, \mathsf{priv}(A)), \mathsf{pub}(B)) \\ B \to A & : & \mathsf{senc}(s, k) \end{array}$$

. . . using Horn clauses

- A talks with any principal represented by its public key $\mathsf{pk}(x)$.

$$\mathsf{att}(\mathsf{pk}(x)) \Rightarrow \mathsf{att}(\mathsf{aenc}(\mathsf{sign}(k[x], sk_A), \mathsf{pk}(x)))$$

- When B receives a message of the expected form, he replies accordingly

$$\mathsf{att}(\mathsf{aenc}(\mathsf{sign}(y, sk_A), \mathsf{pk}(sk_B))) \Rightarrow \mathsf{att}(\mathsf{senc}(s, y))$$

$\longrightarrow$ names are **parametrized** to partially modelled their freshness

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

Is $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg \mathrm{att}(s)$ satisfiable or not?

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

$$\text{Is } \mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s) \text{ satisfiable or not?}$$

Denning Sacco protocol

1. $\text{att}(sk_I)$ initial knowledge

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

$$\text{Is } \mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s) \text{ satisfiable or not?}$$

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

$$\text{Is } \mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s) \text{ satisfiable or not?}$$

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1
3. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_I)))$	using protocol (rule 1) on 2

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

$$\text{Is } \mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s) \text{ satisfiable or not?}$$

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1
3. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_I)))$	using protocol (rule 1) on 2
4. $\text{att}(\text{pk}(sk_B))$	initial knowledge

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

Is $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s)$ satisfiable or not?

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1
3. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_I)))$	using protocol (rule 1) on 2
4. $\text{att}(\text{pk}(sk_B))$	initial knowledge
5. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_B))$	using attacker rules on 3 with 1/4

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

Is $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s)$ satisfiable or not?

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1
3. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_I)))$	using protocol (rule 1) on 2
4. $\text{att}(\text{pk}(sk_B))$	initial knowledge
5. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_B))$	using attacker rules on 3 with 1/4
6. $\text{att}(\text{senc}(s, k[sk_I]))$	using protocol (rule 2) on 5

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

Is $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s)$ satisfiable or not?

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1
3. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_I)))$	using protocol (rule 1) on 2
4. $\text{att}(\text{pk}(sk_B))$	initial knowledge
5. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_B))$	using attacker rules on 3 with 1/4
6. $\text{att}(\text{senc}(s, k[sk_I]))$	using protocol (rule 2) on 5
7. $\text{att}(k[sk_I])$	using attacker rules on 3 with 1

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

Is $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s)$ satisfiable or not?

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1
3. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_I)))$	using protocol (rule 1) on 2
4. $\text{att}(\text{pk}(sk_B))$	initial knowledge
5. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_B))$	using attacker rules on 3 with 1/4
6. $\text{att}(\text{senc}(s, k[sk_I]))$	using protocol (rule 2) on 5
7. $\text{att}(k[sk_I])$	using attacker rules on 3 with 1
8. $\text{att}(s)$	attacker rule on 6 with 7.

Modelling the security property using Horn clauses

We consider secrecy as a reachability (accessibility) property.

Is $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s)$ satisfiable or not?

Denning Sacco protocol

1. $\text{att}(sk_I)$	initial knowledge
2. $\text{att}(\text{pk}(sk_I))$	using attacker rules on 1
3. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_I)))$	using protocol (rule 1) on 2
4. $\text{att}(\text{pk}(sk_B))$	initial knowledge
5. $\text{att}(\text{aenc}(\text{sign}(k[sk_I], sk_A), \text{pk}(sk_B))$	using attacker rules on 3 with 1/4
6. $\text{att}(\text{senc}(s, k[sk_I]))$	using protocol (rule 2) on 5
7. $\text{att}(k[sk_I])$	using attacker rules on 3 with 1
8. $\text{att}(s)$	attacker rule on 6 with 7.

Contradiction ! $\mathcal{C}_{att} + \mathcal{C}_{prot} + \neg\text{att}(s)$ is **not** satisfiable.

$\longrightarrow$ This derivation corresponds to an attack.

Exercise

Consider the Horn clauses given on the previous slides to model the Denning Sacco protocol.

Exercise

Apply binary resolution to derive the empty clause.

ProVerif

ProVerif implements a resolution strategy well-adapted to cryptographic protocols.

ordered resolution with selection

Approximation of the translation in Horn clauses:

- the freshness of nonces is partially modeled;
- the number of times a message appears is ignored, only the fact that is has appeared is taken into account;
- the state of the principals is not fully modeled.

$\longrightarrow$ These approximations are keys for an efficient verification.

Experimental results

Pentium III, 1 GHz.

Protocol	Result	ms
Needham-Schroeder public key	Attack [Lowe]	10
Needham-Schroeder public key corrected	Secure	7
Needham-Schroeder shared key	Attack [Denning]	52
Needham-Schroeder shared key corrected	Secure	109
Denning-Sacco	Attack [AN]	6
Denning-Sacco corrected	Secure	7
Otway-Rees	Secure	10
Otway-Rees, variant of Paulson98	Attack [Paulson]	12
Yahalom	Secure	10
Simpler Yahalom	Secure	11
Main mode of Skeme	Secure	23