Comparing State Spaces in Automatic Security Protocol Verification

Pascal Lafourcade & Cas Cremers

Cryptographic Protocols

Cryptographic Protocols

Cryptographic Protocols

Information Security Everywhere

- The world is distributed and based on networked information systems.
- Protocols essential to developing networked services and new applications.

Security errors in protocol design are costly

Example: Needham-Schroeder Protocol 1978

Example: Needham-Schroeder Protocol 1978

Example: Needham-Schroeder Protocol 1978

Example: Needham-Schroeder Protocol 1978

Example: Needham-Schroeder Protocol 1978

Lowe Attack on the Needham-Schroeder

so-called “Man in the middle attack”

Lowe Attack on the Needham-Schroeder

so-called “Man in the middle attack”

Lowe Attack on the Needham-Schroeder

so-called "Man in the middle attack"

Lowe Attack on the Needham-Schroeder

so-called “Man in the middle attack”

Lowe Attack on the Needham-Schroeder

so-called “Man in the middle attack”

Lowe Attack on the Needham-Schroeder

so-called “Man in the middle attack”

Lowe Attack on the Needham-Schroeder

so-called “Man in the middle attack”

Necessity of Tools

- Protocols are small recipes.
- Non trivial to design and understand.
- The number and size of new protocols.
- Out-pacing human ability to rigourously analyze them.

GOAL : A tool is finding flaws or establishing their correctness.

- completely automated

How can we compare all these tools “fairly”?

State of the art

- **Time performence comparison of AVISPA Tools**
  L. Vigano “Automated Security Protocol Analysis With the AVISPA Tool” ENTCS 2006.

Outline

1. Motivations
2. State Spaces
   - Notations
   - Results
3. Settings
   - Tools
   - Protocols and PC

Outline

Terminology

- A *run* is a single (possibly partial) instance of a role, performed by an agent.
- A *run description* of a protocol with $|R|$ roles is a set of roles. An element of a run description is of the form $r(a_1, a_2, \ldots, a_{|R|})$, where r denotes the role that the run is performing.

Definitions and Properties (I)

Let n be an integer, and let s be a scenario.

- *Traces* is the set of all traces (possible executions of the protocol) of any length, and any combination of agents.
- *MaxRuns(n)* is the set of traces with at most n runs.

$$\forall n \in N : \text{MaxRuns}(n) \subset \text{Traces} \tag{1}$$

Definitions and Properties (II)

- *RepScen(s)* is the set of traces built only with runs that are present in s. The runs defined by the scenario s can be executed any number of times. In other words, each run in each trace corresponds to an element of s.

Number of Agents

According to [Comon & Cortier 2004]

- Only a single dishonest (compromised) agent e, is enough.
- For the verification of secrecy, only a single honest agent a is sufficient.
- For the verification of authentication, we

Minimal Number of Scenarios

With 2 agents and 1 intruder for $X(a_1, \ldots, a_{|R|})$, we get $|R| * 2 * 3^{(|R|-1)}$ different possible run descriptions. Now we choose a multiset of n run descriptions:

$$\binom{|R| * 2 * 3^{(|R|-1)} + n - 1}{n}$$

Using Burnside Lemma

- $\{a \to a, b \to b\}$ (the trivial renaming)
- $\{a \to b, b \to a\}$

We get

$$k(n, |R|) = \frac{\binom{2*|R|*3^{(|R|-1)}+n-1}{n} + \epsilon_n \binom{|R|*3^{(|R|-1)}+\frac{n}{2}-1}{\frac{n}{2}}}{2}$$

Outline

6 Tools Compared

- **Avispa** :

OFMC: On-the-fly Model-Checker employs several symbolic techniques to explore the state space in a demand-driven way.

CL-AtSe: Constraint-Logic-based Attack Searcher applies constraint solving with simplification heuristics and redundancy elimination techniques.

SATMC: SAT-based Model-Checker builds a propositional formula encoding all the

4 Protocols analyzed

- Needham-Schroeder
- Needham-Schroeder Lowe
- EKE: Encrypted Key Exchange (using symetric and asymetric encryption)
- TLS: Transport Layer Security (larger protocol)

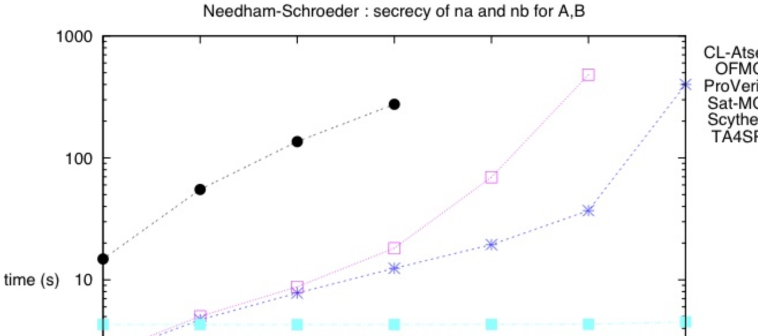
EKE

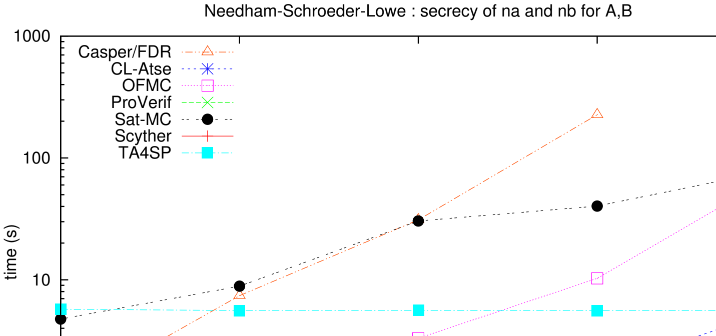
```
0. A->B: {Ea}_Kab         |    Key exchange part
1. B->A: {{K}_Ea}_Kab     |
2. A->B: {Ca}_K           |
3. B->A: {Ca,Cb}_K        |    Challenge/Response
4. A->B: {Cb}_K           |    Authentication part
```

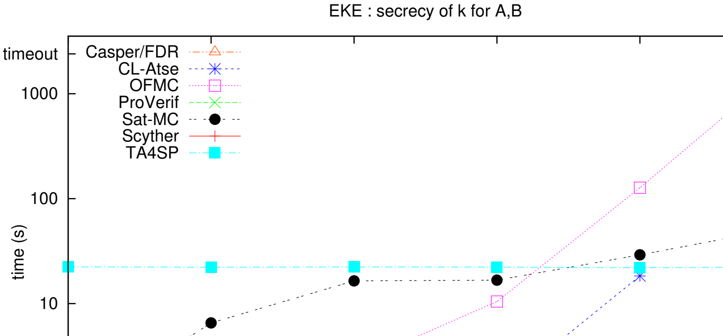
TLS

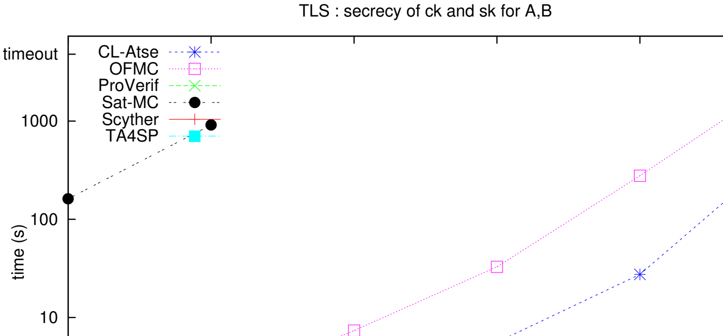
```
0. A->B: A, Na, Sid, Pa       | Pa is a cryptosuite offe
```

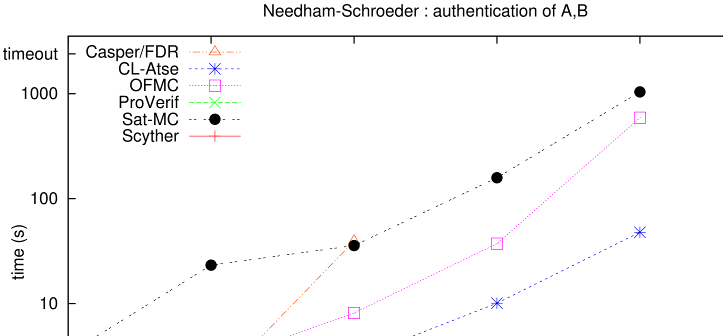
Outline

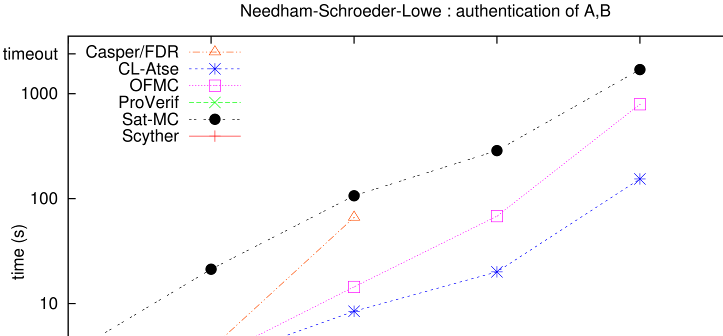

Needham-Schroeder : secrecy of na and nb for A,B
1000
100
time (s)
10
CL-Atse
OFMC
ProVeri
Sat-MC
Scythe
TA4SP

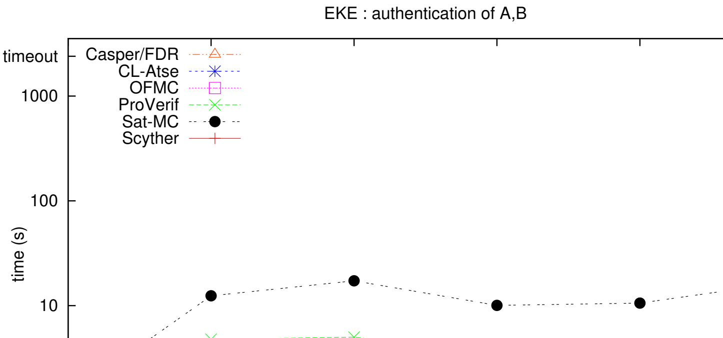

Needham-Schroeder-Lowe : secrecy of na and nb for A,B
1000
100
10
time (s)
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther
TA4SP

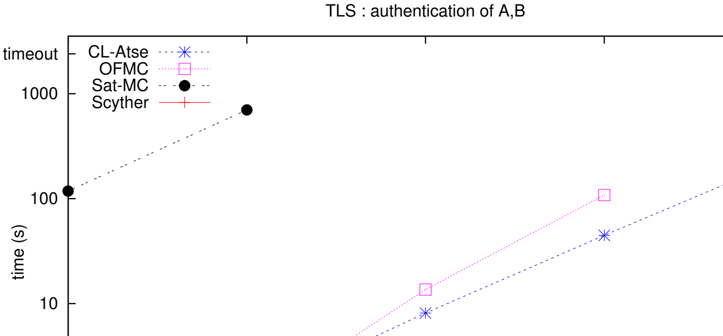

EKE : secrecy of k for A,B
timeout
1000
100
10
time (s)
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther
TA4SP


TLS : secrecy of ck and sk for A,B
timeout
1000
100
10
time (s)
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther
TA4SP


Needham-Schroeder : authentication of A,B
timeout
1000
100
10
time (s)
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther


Needham-Schroeder-Lowe : authentication of A,B
timeout
1000
100
10
time (s)
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther

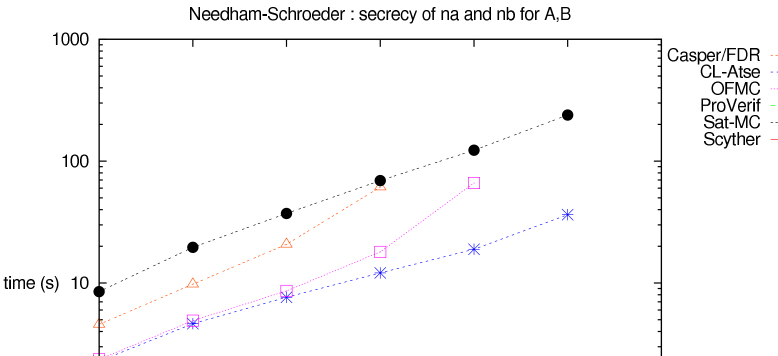

EKE : authentication of A,B
timeout
1000
100
10
time (s)
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther

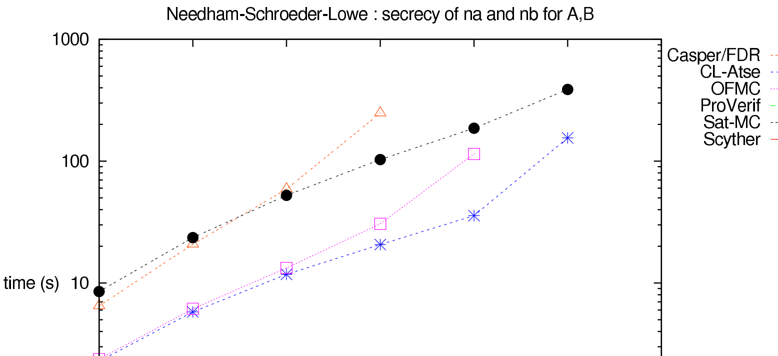

TLS : authentication of A,B
timeout
1000
100
10
time (s)
CL-Atse
OFMC
Sat-MC
Scyther

Outline

Conclusion

- Automatic verification is necessary.
- Tool are very helpful for design and verification.
- Use your favorite tool.
- Modeling of a protocol is quite tricky.
- Know the limitations of the tool and what you are checking.


Needham-Schroeder : secrecy of na and nb for A,B
1000
100
10
time (s)
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther


Needham-Schroeder-Lowe : secrecy of na and nb for A,B
1000
100
10
time (s)
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther

Thank you for your attention.

Questions ?