AUTOMATIC VERIFICATION OF CRYPTOGRAPHIC PROTOCOLS

Hubert Comon-Lundh
LSV, CNRS
INRIA project SECSI
École Normale Supérieure de Cachan
`comon@lsv.ens-cachan`

ADVERTISEMENT

There is a grant programme for master/PhD students at ENS Cachan

Deadline for applications: Feb 15th, 2005

Look at ENS Cachan web pages `www.ens-cachan.fr`.

PART 0
INTRODUCTION

SUMMARY OF THE LECTURES

Part 0: introduction

Part 1:local theories

Part 2: protocols

Part 3: algebraic properties

A model checking problem:

$$M_I \models \phi$$

CRYPTOGRAPHIC PROTOCOLS

A model checking problem:

Cimpa school, Feb 2005
5-Automatic Verification of Cryptographic Protocols

CRYPTOGRAPHIC PROTOCOLS

A model checking problem:

$$M_I \models \phi$$

Infinite state, infinitely branching, with some specificities

A security property ($\neq$ temporal formula)

Cimpa school, Feb 2005
5-Automatic Verification of Cryptographic Protocols

CRYPTOGRAPHIC PROTOCOLS

A model checking problem:

$$M_I \models \phi$$

Infinite state, infinitely branching, with some specificities

Cimpa school, Feb 2005
5-Automatic Verification of Cryptographic Protocols

AUTOMATIC VERIFICATION

Why automatic ?

- Verification of many small variants of a protocol. (Nonce implementation, memory constraints, bandwidth constraints,...)
- Refine the model: include more properties of the primitives, depending on the encryption algorithms (e.g. malleability, encryption and decryption commute... See F. Morain's lecture).

Alternative: use machine assisted proofs Paulson 97 – 04.

CRYPTOGRAPHIC PROTOCOLS

A model checking problem:

$$M_I \models \phi$$

Infinite state, infinitely branching, with some specificities

Intruder theory

A security property ($\neq$ temporal formula)

THE OPTIMISTIC APPROACH

- ProVerif (See C. Fournet's lecture)
- The EVA project: LSV, VERIMAG, TRUSTED LOGIC.
- Many others CAPSL, ...

Many papers and results, using various techniques: Clauses, Set constraints, Tree automata,... (See Ramanujam lecture)

Weaknesses:

- A failure doesn't mean that there is an attack
- A success means no attack, assming some hypothesis on the cryptographic primitives. Difficult to take algebraic properties into account.
- There is a huge variety of security properties, whose proofs can hardly be automatized

THE TWO APPROACHES

The security problem is Π^1_1-hard: there is no decision and even no semi-decision algorithm.

This result holds even under strong additional hypotheses (see Ramanujam lecture).

The two approaches:

Pessimistic : try to find an attack

Optimistic : use upper approximations, trying to find a proof.

BOUNDED NUMBER OF SESSIONS

We fix the number of protocol instances ; no guarantee that the protocol is secure for more instances.

M. Rusinowitch and M. Turuani, 2001: security is co-NP-complete for a bounded number of sessions, *In the Dolev-Yao model* (perfect cryptography)

BOUNDED NUMBER OF SESSIONS

We fix the number of protocol instances ; no guarantee that the protocol is secure for more instances.

BOUNDED NUMBER OF SESSIONS

We fix the number of protocol instances ; no guarantee that the protocol is secure for more instances.

M. Rusinowitch and M. Turuani, 2001: security is co-NP-complete for a bounded number of sessions, *In the Dolev-Yao model* (perfect cryptography)

The PROUVÉ project: LSV, VERIMAG, LORIA, FRANCE TELECOM, CRIL

Case studies: Electronic money, Vote. Properties are not reduced to secrecy and authentication.

Many tools based on model checking, boundind the number of sessions and often also the instances: CSP/FDR, ATHENA, CASRUL, AVISPA, ...

BOUNDED NUMBER OF SESSIONS

We fix the number of protocol instances ; no guarantee that the protocol is secure for more instances.

M. Rusinowitch and M. Turuani, 2001: security is co-NP-complete for a bounded number of sessions, *In the Dolev-Yao model* (perfect cryptography)

The PROUVÉ project: LSV, VERIMAG, LORIA, FRANCE TELECOM, CRIL

Case studies: Electronic money, Vote. Properties are not reduced to secrecy and authentication.

GOALS OF THE LECTURES

Design proof strategies which are

- Refutation complete
- complete for a fixed number of sessions
- work for various intruder theories
- can take into account several algebraic theories for cryptographic primitives

EXAMPLES OF PROTOCOLS

TMN:

$$\begin{array}{lll}
1. & A \to S: & A, B, \{K_A\}_{pub(S)} \\
2. & S \to B: & A \\
3. & B \to S: & A, \{K_B\}_{pub(S)} \\
4. & S \to A: & B, K_B \oplus K_A
\end{array}$$

NS:

$$\begin{array}{lll}
1. & A \to B: & \{< A, N_A >\}_{pub(B)} \\
2. & B \to A: & \{< N_A, N_B >\}_{pub(A)} \\
3. & A \to B: & \{N_B\}_{pub(B)}
\end{array}$$

SPORE – the protocol library

`//www.lsv.ens-cachan.fr/spore/`

SUMMARY OF THE LECTURES

Part 0: introduction

Part 1:local theories

1. Tractable Decision problems HORNSAT
2. Tractable inference systems: LOCAL THEORIES. Mc Allester 93
3. Examples of local theories: the Dolev-Yao intruder deduction systems
4. Exercises

Part 2: proof normalization

1. Protocols: A quick reminder of the trace semantics
2. Proof systems; the particular case of a bounded number of sessions
3. Protocols rules as intruder oracles
4. A normal proof result in the simplest case
5. co-NP completeness in the case of a bounded number of sessions. Rusinowitch and Turuani, 2001
6. Extensions to other intruder theories

SUMMARY OF THE LECTURES (CNTD)

Part 3: algebraic properties

1. Basic on rewriting and narrowing
2. Another local theory
3. Computing variants
4. Locality and variants.

THE HORNSAT DECISION PROBLEM

Data : a finite set of propositional Horn clauses : there is at most one positive litteral in each clause

Question : is the set of clauses satisfiable ?

Theorem 1 HORNSAT *is decidable in linear time and is* PTIME-*complete*

Many equivalent problems (under constant space reductions):

- AND/OR graph reachability
- Tree automata emptiness

PART 1: LOCAL THEORIES

PROOF OF THE THEOREM (I)

Reduce first the problem to a fixed point computation, separating the purely negative clauses from the others.

Assume the data are organized in two arrays:

- A_1 is indexed by propositional variables and $A_1[P] = (s(P), LC(P))$ where $s(P)$ is a status flag and $LC(P)$ is the list of clauses in which P occurs negatively.
- A_2 is indexed by clauses and $A_2[C] = (n(C), H(C))$ where $n(C)$ is an integer, initially set to the number of distinct negative litterals in C. $H(C)$ is the litteral in the head.

PROOF OF THE THEOREM (I)

Reduce first the problem to a fixed point computation, separating the purely negative clauses from the others.

Proof of the theorem (I)

Reduce first the problem to a fixed point computation, separating the purely negative clauses from the others.

Assume the data are organized in two arrays:

- A_1 is indexed by propositional variables and $A_1[P] = (s(P), LC(P))$ where $s(P)$ is a status flag and $LC(P)$ is the list of clauses in which P occurs negatively.
- A_2 is indexed by clauses and $A_2[C] = (n(C), H(C))$ where $n(C)$ is an integer, initially set to the number of distinct negative litterals in C. $H(C)$ is the litteral in the head.

The array computation can be done in linear time. (Note: numbers can be written in base 1).

In addition, we consider a list M, which is initially empty (the least model) and a stack σ.

Proof of Theorem (II)

First scan A_2 once:
`for` every clause `do`

`if` $n(C) = 0$ `then`

`let` $P = H(C)$ `in`

`if` $s(P) = 0$ `then` push P on σ; set $s(P)$ to 1

Proof of the theorem (III)

`while` σ is not empty do

Pop a proposition P from σ

`For every` $C \in LC(P)$,

decrement $n(C)$

`if`$n(C) = 0$ `then`

`let` $P = H(C)$ `in` `if` $s(P) = 0$ `then`

push P on σ

set $s(P)$ to 1.

Exercise 1 (level 2): show that every variable is pushed at most once on the stack. Conclude that the algorithm works in linear time (assuming decrementation can be done in constant time).

Inference Systems

INFERENCE SYSTEMS

premisses   Side condition

$$\frac{\phi_1 \quad \cdots \quad \phi_n}{\phi} \quad \text{if } \gamma$$

conclusion

$\phi_1, \ldots, \phi_n, \phi$ are formulas in a term algebra $T(\mathcal{F}, X)$.

INFERENCE SYSTEMS

premisses

$$\frac{\phi_1 \quad \cdots \quad \phi_n}{\phi}$$

conclusion

$\phi_1, \ldots, \phi_n, \phi$ are formulas in a term algebra $T(\mathcal{F}, X)$.

LOCALITY

$\mathcal{I}$ is a finite set of inference rules, $\vdash_{\mathcal{I}}$ the (many-steps) deduction relation.

Given a function $F : 2^{T(\mathcal{F})} \rightarrow 2^{T(\mathcal{F})}$ An inference system $\mathcal{I}$ is F-local if, for every formula ϕ such that $\phi_1, \ldots, \phi_n \vdash_{\mathcal{I}} \phi$, there is a proof of ϕ, which only involves formulas of $F(\{\phi_1, \ldots, \phi_n, \phi\})$.

INFERENCE SYSTEMS

premisses   Side condition

$$\frac{\phi_1 \quad \cdots \quad \phi_n}{\phi} \quad \text{if } \gamma$$

conclusion

$\phi_1, \ldots, \phi_n, \phi$ are formulas in a term algebra $T(\mathcal{F}, X)$.

ψ is one step derivable from $\psi_1, \ldots, \psi_n$ if there is a θ such that $\phi_i\theta = \psi_i$, $\phi\theta = \psi$ and $\theta \models \gamma$.

ORDER OF AN INFERENCE RULE

An inference rule r has *order* $k \in \mathbb{N}$ if there are expressions $e_1, \ldots, e_k$ such that each e_i is a subexpression of some formula in r and every (meta)-variable of r occurs in some e_i.

The inference rule

$$\frac{T \vdash k^{-1} \quad T \vdash \{x\}_k}{T \vdash x}$$

has order 1 (and any larger integer)

ORDER OF AN INFERENCE RULE

An inference rule r has *order* $k \in \mathbb{N}$ if there are expressions $e_1, \ldots, e_k$ such that each e_i is a subexpression of some formula in r and every (meta)-variable of r occurs in some e_i.

The inference rule

$$\frac{T \vdash k^{-1} \quad T \vdash \{x\}_k}{T \vdash x}$$

has order

TRACTABILITY OF LOCAL INFERENCE SYSTEMS

The size of a term (resp. a set of terms) is the number of its distinct subterms.

Theorem 2: If

- F is computable in linear time (resp. polynomial time),
- $\mathcal{I}$ is F-local and
- every rule as order k

then, given a finite set of formulas S and a formula ϕ, we can decide whether $S \vdash_{\mathcal{I}} \phi$ in time $O(n^k)$. (resp.), where $n = |S| + |\phi|$.

Proof: Compute $T = F(S \cup \{\phi\})$, each of them is a propositional variable. Compute for each inference rule the $O(n^k)$ Horn clauses obtained by solving the k matching equations for every $t \in T$. Use HORNSAT

TRACTABILITY OF LOCAL INFERENCE SYSTEMS

The size of a term (resp. a set of terms) is the number of its distinct subterms.

Theorem 2: If

- F is computable in linear time (resp. polynomial time),
- $\mathcal{I}$ is F-local and
- every rule as order k

then, given a finite set of formulas S and a formula ϕ, we can decide whether $S \vdash_{\mathcal{I}} \phi$ in time $O(n^k)$. (resp.), where $n = |S| + |\phi|$.

TRACTABILITY OF LOCAL INFERENCE SYSTEMS

The size of a term (resp. a set of terms) is the number of its distinct subterms.

Theorem 2: If

- F is computable in linear time (resp. polynomial time),
- $\mathcal{I}$ is F-local and
- every rule as order k

then, given a finite set of formulas S and a formula ϕ, we can decide whether $S \vdash_{\mathcal{I}} \phi$ in time $O(n^k)$. (resp. $O(n^{m \times k})$), where $n = |S| + |\phi|$.

Proof: Compute $T = F(S \cup \{\phi\})$, each of them is a propositional variable. Compute for each inference rule the $O(n^k)$ Horn clauses obtained by solving the k matching equations for every $t \in T$. Use HORNSAT

EXERCISE 2 (LEVEL 1)

Theorem 2 essentially assumes that there are no side conditions in the inference rules. What must be changed if we allow side conditions ?

DOLEV-YAO LIKE THEORIES

$\mathcal{F}$ be $\mathsf{pub}(\_), \mathsf{priv}(\_), \{\_\}\_, <\_,\_>, [\_]\_$ and constants.

$$\frac{x \quad y}{<x,y>} \qquad \frac{x \quad y}{\{x\}_y} \qquad \frac{x \quad y}{[x]_y}$$

$$\frac{<x,y>}{x} \qquad \frac{<x,y>}{y} \qquad \frac{[x]_y \quad y}{x}$$

$$\frac{\{x\}_{\mathsf{pub}(y)} \quad \mathsf{priv}(y)}{x} \qquad \frac{x}{\mathsf{pub}(x)}$$

DOLEV-YAO RULES ARE F-LOCAL

Theorem Let $F(T)$ be the set of subterms of T. Then the set of Dolev-Yao rules is F-local.

Locality Proof (cntd)

- If the last inference rule is a construction rule, use induction hypothesis.

$$\dfrac{\dfrac{\color{blue}{\Pi_1}}{t_1} \quad \cdots \quad \dfrac{\color{blue}{\Pi_n}}{t_n}}{f(t_1, \ldots, t_n)}$$

Dolev-Yao rules are F-local

Theorem Let $F(T)$ be the set of subterms of T. Then the set of Dolev-Yao rules is F-local.

We divide the rules into two sets: the *constructor rules*, which build new terms and the *decomposition rules*, which consist of the other 5 rules. We prove, by induction on the length of a minimal size proof that, if $T \vdash_{\mathcal{I}} t$ then

1. if the last rule is a construction rule, then all terms in the proof are in $F(T) \cup F(\{t\})$
2. otherwise, all terms in the proof are in $F(T)$.

In case the proof contains no inference step, $t \in T$ and all terms in the proof are in $F(T)$.

Locality Proof (cntd)

- If it is a symmetric decryption:

$$\dfrac{\dfrac{\color{blue}{\Pi_1}}{[u]_v} \quad \dfrac{\color{blue}{\Pi_2}}{v}}{u}$$

The last rule of Π_1 is not a construction. We use induction hypothesis twice and closure of $F(T)$ by subterm.

Locality Proof (cntd)

- If the last inference rule is a construction rule, use induction hypothesis.

$$\dfrac{\dfrac{\color{blue}{\Pi_1}}{t_1} \quad \cdots \quad \dfrac{\color{blue}{\Pi_n}}{t_n}}{f(t_1, \ldots, t_n)}$$

- If it is unpairing, then the last rule of Π cannot be a pairing rule:

$$\dfrac{\dfrac{\dfrac{\color{blue}{\Pi_1}}{u} \quad \dfrac{\color{blue}{\Pi_2}}{v}}{< u, v >}}{u}$$

is not minimal in size: Π_1 is a shorter proof of the same term. Then we use induction hypothesis.
The other unpairing rule yields a similar proof.

Locality Proof (cntd)

- If it is a symmetric decryption:

$$\dfrac{\dfrac{\Pi_1}{[u]_v} \quad \dfrac{\Pi_2}{v}}{u}$$

The last rule of Π_1 is not a construction. We use induction hypothesis twice and closure of $F(T)$ by subterm.

- If it is an asymetric decryption of $\{u\}_{\mathsf{pub}(v)}$:

$$\dfrac{\dfrac{\Pi_1}{\{u\}_{\mathsf{pub}(v)}} \quad \dfrac{\Pi_2}{\mathsf{priv}(v)}}{u}$$

The last rule of Π_1 is not a construction rule. By induction hypothesis, all terms in Π_1 belong to $F(T)$. In particular, $u, \mathsf{pub}(v) \in F(T)$. Next, there is no construction rule yielding $\mathsf{priv}(v)$, hence apply the induction hypothesis.

Locality Proof (cntd)

- If it is a symmetric decryption:

$$\dfrac{\dfrac{\Pi_1}{[u]_v} \quad \dfrac{\Pi_2}{v}}{u}$$

The last rule of Π_1 is not a construction. We use induction hypothesis twice and closure of $F(T)$ by subterm.

- If it is an asymetric decryption of $\{u\}_{\mathsf{pub}(v)}$:

$$\dfrac{\dfrac{\Pi_1}{\{u\}_{\mathsf{pub}(v)}} \quad \dfrac{\Pi_2}{\mathsf{priv}(v)}}{u}$$

More exercises

Exercise 4 (level 2) Assume we add the following rule

$$\dfrac{\{x\}_{\mathsf{priv}(y)} \quad \mathsf{pub}(y)}{x}$$

Show that this yields also a local theory (possibly using another function F)

Exercise 5 (level 3)
Assume we add the following rule, which is assumed to model some kind of cipher-block chaining property:

$$\dfrac{\{< x, y >\}_z}{\{x\}_z}$$

Again, show that we get a local theory.

Passive attacks are easy to find

Corollary Deducibility can be decided in linear time for the Dolev-Yao rules.

Exercise 3 (level 2) In early papers, the following procedure was proposed for the intruder deduction problem: given $t_1, \ldots, t_n, t$

1. First decompose as much as possible $t_1, \ldots, t_n$: compute the fixed point by decryption and unpairing.
2. Next try to build the term t using encryption and pairing from the set obtained in the first step

Why is this procedure incomplete (Give an example) ? Under which additional hypotheses is it complete ?

Exclusive Or axioms

$$\begin{array}{rcl} x \oplus x \oplus y & \rightarrow & y \\ x \oplus x & \rightarrow & 0 \\ x \oplus 0 & \rightarrow & x \end{array} \qquad \begin{array}{rcl} x \oplus (y \oplus z) & = & (x \oplus y) \oplus z \\ x \oplus y & = & y \oplus x \end{array}$$

The rewrite system is AC-convergent: there are unique normal forms $t\downarrow$, up to AC.

More exercises (cntd)

Exercise 6 (level 3)
Show that, if S is a recognizable tree language, then the set of terms deducible from S in the DY inference system is also a recognizable tree language.

Extending DY with exclusive or

Add to DY the following rule(s):

$$\frac{x_1 \quad \cdots \quad x_n}{(x_1 \oplus \ldots \oplus x_n)\downarrow}$$

Exercise 7 (level 4). Show that the new inference system, with exclusive or, is F-local. (Ind: consider for F the set of subterms, when $\oplus$ is viewed as a varyadic symbol).