Automatic Verification of Cryptographic Protocols

Hubert Comon-Lundh

LSV, CNRS

INRIA project SECSI

École Normale Supérieure de Cachan

comon@lsv.ens-cachan.fr

SUMMARY OF THE LECTURES

Part 0: introduction

Part 1:local theories

Part 2: protocols

Part 3: algebraic properties

PART 0
INTRODUCTION

CRYPTOGRAPHIC PROTOCOLS

A model checking problem:

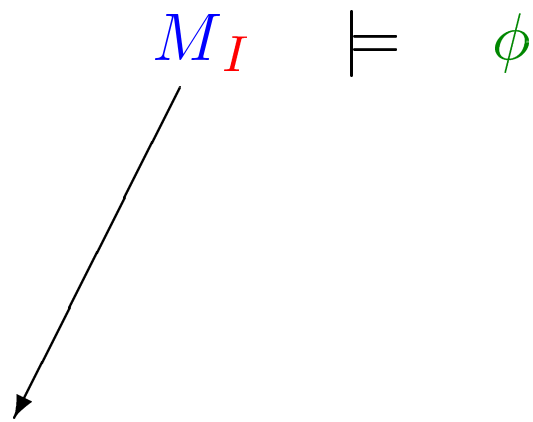
Cryptographic Protocols

A model checking problem:

$$M_I \quad \models \quad \phi$$

CRYPTOGRAPHIC PROTOCOLS

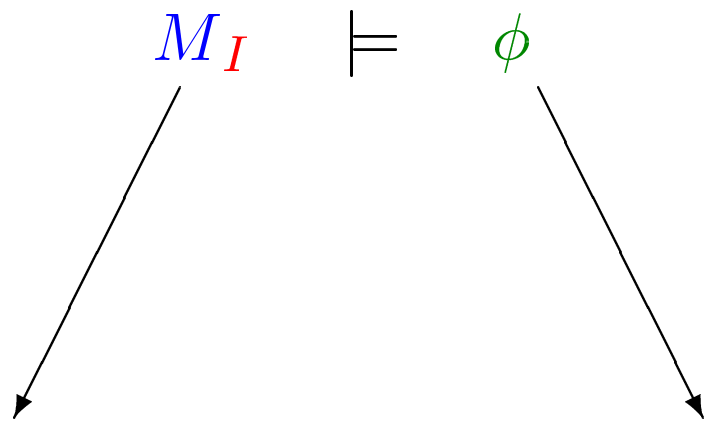
A model checking problem:



Infinite state, infinitely branching, with some specifi cities

CRYPTOGRAPHIC PROTOCOLS

A model checking problem:

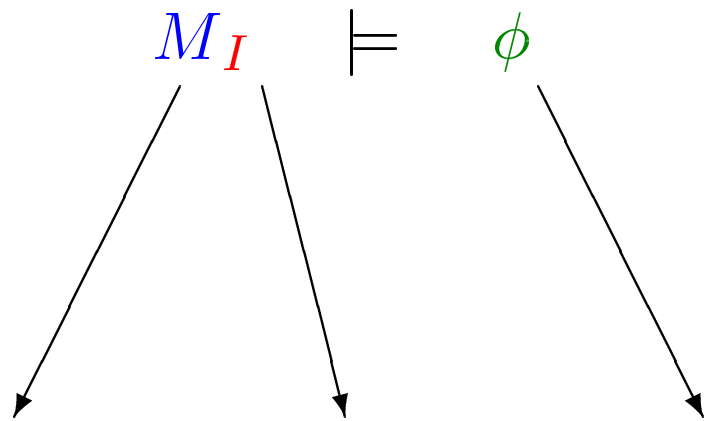


Infinite state, infinitely branching, with some specificities

A security property ($\neq$ temporal formula)

CRYPTOGRAPHIC PROTOCOLS

A model checking problem:



Infinite state, infinitely branching, with some specifi cities

Intruder theory

A security property ($\neq$ temporal formula)

AUTOMATIC VERIFICATION

Why automatic ?

- Verifi cation of many small variants of a protocol. (Nonce implementation, memory constraints, bandwidth constraints,...)
- Refi ne 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.

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 fi nd an attack

Optimistic : use upper approximations, trying to fi nd a proof.

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. Diffi cult to take algebraic properties into account.
- There is a huge variety of security properties, whose proofs can hardly be automatized

BOUNDED NUMBER OF SESSIONS

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

BOUNDED NUMBER OF SESSIONS

We fi x 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 fi x 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.

BOUNDED NUMBER OF SESSIONS

We fi x 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, ...

GOALS OF THE LECTURES

Design proof strategies which are

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

EXAMPLES OF PROTOCOLS

TMN:

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

NS:

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

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.

Part 1:
Local Theories

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 decidabable in linear time and is* PTIME-*complete*

Many equivalent problems (under constant space reductions):

- AND/OR graph reachability
- Tree automata emptiness

PROOF OF THE THEOREM (I)

Reduce fi rst the problem to a fi xed point computation, separating the purely negative clauses from the others.

PROOF OF THE THEOREM (I)

Reduce fi rst the problem to a fi xed 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 fi rst the problem to a fi xed 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`

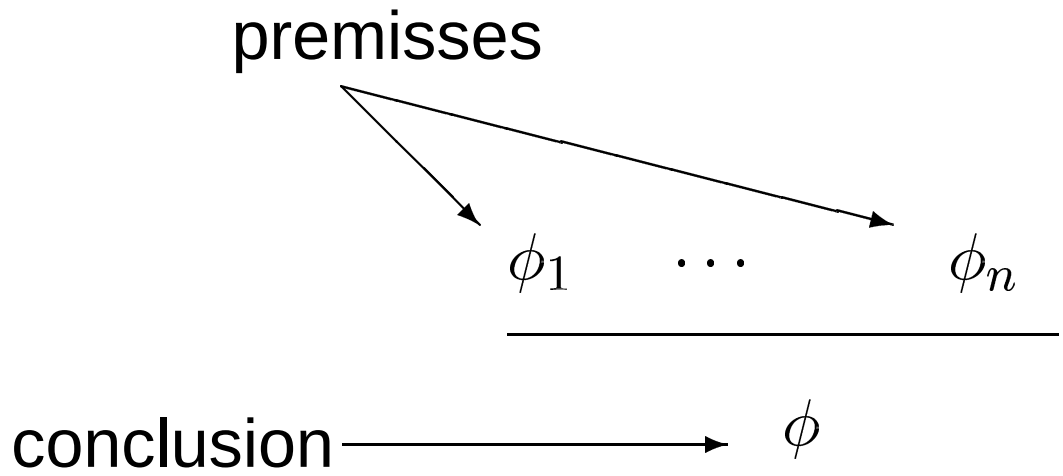
push P on σ

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).

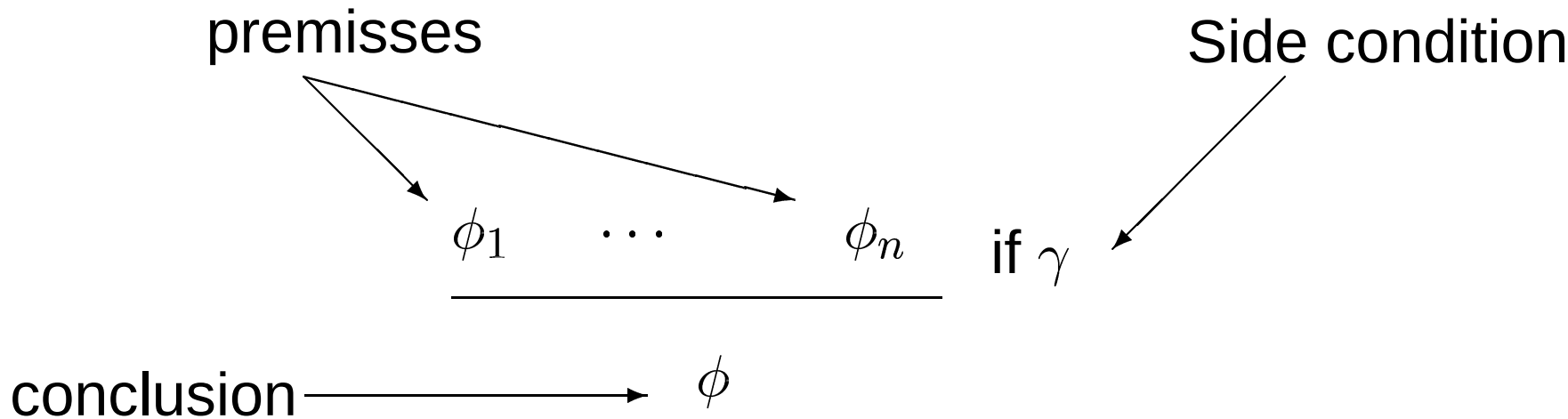
INFERENCE SYSTEMS

INFERENCE SYSTEMS



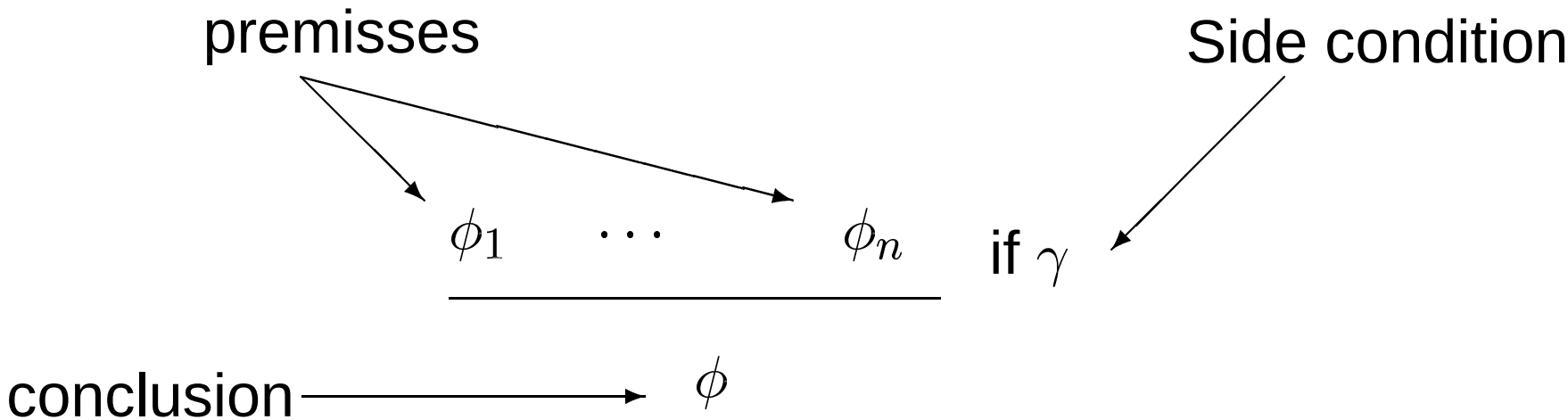
$\phi_1, \ldots, \phi_n, \phi$ are formulas in a term algebra $T(\mathcal{F}, X)$.

INFERENCE SYSTEMS



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

INFERENCE SYSTEMS



$\phi_1, \ldots, \phi_n, \phi$ 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 $\theta \models \gamma$.

LOCALITY

$\mathcal{I}$ is a fi nite 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 $F(\{\phi_1, \ldots, \phi_n, \phi\})$.

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

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)

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 fi nite 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 fi nite 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 fi nite 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 pub(_), 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.

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

$$\frac{\begin{array}{ccc}\Pi_1 & & \Pi_n \\ \hline t_1 & \cdots & t_n\end{array}}{f(t_1, \ldots, t_n)}$$

LOCALITY PROOF (CNTD)

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

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

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

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

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.

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 $\{u\}_{\mathsf{pub}(v)}$:

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

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.

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 fi xed point by decryption and unpairing.
2. Next try to build the term t using encryption and pairing from the set obtained in the fi rst step

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

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 fi xed point by decryption and unpairing.
2. Next try to build the term t using encryption and pairing from the set obtained in the fi rst step

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

Solution: Take $T = \{\{c\}_{<a,b>}, a, b\}$. The procedure is complete only when keys are atomic

More exercises

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

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

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:

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

Again, show that we get a local theory.

More exercises

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

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

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

Solution: Consider $F(T) = Sub(T)$: + for every $\mathsf{priv}(s) \in Sub(T)$, the term $\mathsf{pub}(s)$.

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

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

Again, show that we get a local theory.

More exercises

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

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

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

Solution: Consider $F(T) = Sub(T)$: + for every $\mathsf{priv}(s) \in Sub(T)$, the term $\mathsf{pub}(s)$.

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

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

Again, show that we get a local theory.

Solution: Take for $F(T)$ the set obtained by saturating $F(T)$ with the inference rule. Consider the inference rule as a decomposition. Change the size defi nition of a proof (fi rst the usual size, then the number of applications of the new rules)

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.

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.

Solution: Assume a single final state q_f. Complete the tree automaton, using for instance the following rule:

$$\frac{\{q_1\}_{\mathsf{pub}(q_2)} \xrightarrow[\mathcal{A}]{*} q_f \quad \mathsf{priv}(q_3) \xrightarrow[\mathcal{A}]{*} q_f}{q_1 \xrightarrow[\mathcal{A}]{} q_f} \quad \text{If } L_{\mathcal{A}}(q_2) \cap L_{\mathcal{A}}(q_3) \neq \emptyset$$

EXCLUSIVE OR AXIOMS

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

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

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).

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).

Solution: Gather together the $\oplus$ rules. Use induction, assuming that the $\oplus$ rule is a construction if it yields a term headed with $\oplus$. It is a decomposition otherwise. At the induction step for $\oplus$, premisses are split among those, which are headed with $\oplus$ (hence obtained by decomposition) and those which are not headed with $\oplus$. The latter can either be removed or are subterms of a premisse of the fi rst form.

Part 2
Proof Normalization

PROTOCOL SYNTAX

A fi nite number of roles:

$A(x_1, \ldots, x_k)$ role name and parameters

$\nu N_1, \ldots, N_k$ name generation (nonces)

$$\left.\begin{array}{rcl} u_1 & \rightarrow & v_1 \\ & \cdots & \\ u_n & \rightarrow & v_n \end{array}\right\}$$ protocol rules

$u_1, \ldots, u_n, v_1, \ldots, v_n$ are terms with variables; variables stand for nonces generated by other roles, or encrypted data, which cannot be decrypted.

u_i, v_i can be empty (0). Terms are untyped.

$Var(v_i) \subseteq \{x_1, \ldots, x_k\} \cup Var(u_1, \ldots, u_i)$.

MODELS OF PROTOCOLS

A state is composed of a set of terms I (intruder knowledge) and for each agent a local state, which maps each integer out of a fi nite set to

- a role name
- a binding for the parameters of that role
- a step number
- a binding for variables introduced before that step
- possibly a list of terms (for agreement properties)

MODELS OF PROTOCOLS (CNTD)

The set of agents names is divided into honest agents and dishonest agents. All private data and newly generated data from dishonest agents belong to I.
A transition between two states is given by a local state change of a single agent:

- either starting a new session: add to the local state of some agent a new entry with initial values
- or else: increase the step number of a local entry, add the new bindings

and increase the intruder knowledge with the appropriate instance $v_i\sigma$. The transition can only occur if $u_i\theta$ can be deduced from I for a substitution θ which is compatible with σ.

Example of transitions

$$
\begin{array}{lll}
A(B) : \nu N_A. & 1. & 0 \rightarrow \{< A, N_A >\}_{pub(B)} \\
 & 2. & \{< N_A, x >\}_{pub(A)} \rightarrow \{x\}_{pub(B)} \\
 & & \\
B(A) : \nu N_B. & 1. & \{< A, x >\}_{pub(B)} \rightarrow \{< x, N_B >\}_{pub(A)} \\
 & 2. & \{N_B\}_{pub(B)} \rightarrow 0
\end{array}
$$

$$
\begin{pmatrix} a: \\ b: \\ c: \\ I : a, b, c, \mathsf{priv}(c) \end{pmatrix} \rightarrow \begin{pmatrix} a(1) = & (A, 1, B = c) \\ b: & \\ I: & a, b, c, \mathsf{priv}(c), \{< a, N_A(1) >\}_{\mathsf{pub}(c)} \end{pmatrix}
$$

$$
\rightarrow \begin{pmatrix} a(1) = & (A, 1, B = c) \\ b(2) = & (B, 1, A = a, x = N_A(1)) \\ I: & a, b, c, \mathsf{priv}(c), \{< a, N_A(1) >\}_{\mathsf{pub}(c)}, \\ & \{< N_A(1), N_B(2) >\}_{\mathsf{pub}(a)} \end{pmatrix}
$$

EXAMPLE OF TRANSITIONS (CNTD)

$$
\begin{array}{lrcl}
A(B) : \nu N_A. & 1. & 0 & \rightarrow \quad \{< A, N_A >\}_{pub(B)} \\
& 2. & \{< N_A, x >\}_{pub(A)} & \rightarrow \quad \{x\}_{pub(B)}
\end{array}
$$

$$
\begin{array}{lrcl}
B(A) : \nu N_B. & 1. & \{< A, x >\}_{pub(B)} & \rightarrow \quad \{< x, N_B >\}_{pub(A)} \\
& 2. & \{N_B\}_{pub(B)} & \rightarrow \quad 0
\end{array}
$$

$$
\left(\begin{array}{ll}
a(1) = & (A, 1, B = c) \\
b(2) = & (B, 1, A = a, x = N_A(1) \\
I : & a, b, c, \mathsf{priv}(c), \\
& \{< a, N_A(1) >\}_{\mathsf{pub}(c)}, \\
& \{< N_A(1), N_B(2) >\}_{\mathsf{pub}(a)}
\end{array}\right)
\rightarrow
\left(\begin{array}{ll}
a(1) = & (A, 2, B = c, x = N_B(2)) \\
b(2) = & (B, 1, A = a, x = N_A(1)) \\
I : & a, b, c, \mathsf{priv}(c), \\
& \{< a, N_A(1) >\}_{\mathsf{pub}(c)}, \\
& \{< N_A(1), N_B(2) >\}_{\mathsf{pub}(a)} \\
& \{N_B(2)\}_{\mathsf{pub}(c)}
\end{array}\right)
$$

violated property: "$\forall n b, a, k$ such that a, b are honest and$b(n) = (B, k, y = a, ...)$, $N_B(n)$ remains unknown to the intruder".

A SMALL SIMPLIFICATION RESULT

Theorem (CL,Cortier 03):

If there is an attack, then there is an attack two identities are sufficient.

($k + 1$ identities for properties requiring k variables).

We can always assume a fi xed number of agents: a honest one and a dishonest one (2 in what follows).

THE BOUNDED NUMBER OF SESSIONS CASE

We have two role instances in the attack: R_1, R_2:

$$
\begin{array}{lll}
R_1: \quad A = a, B = c \text{ in} & & \\
 & & \Rightarrow \quad \{< A, N_1 >\}_{\mathsf{pub}(B)} \\
 & \{< N_1, y >\}_{\mathsf{pub}(A)} & \Rightarrow \quad \{y\}_{\mathsf{pub}(B)} \\
R_2: \quad B = a, A = a \text{ in} & \{< A, x >\}_{\mathsf{pub}(B)} & \Rightarrow \quad \{< x, N_2 >\}_{\mathsf{pub}(A)} \\
 & \{N_2\}_{\mathsf{pub}(B)} & \Rightarrow \quad .
\end{array}
$$

$$
\left(\begin{array}{l} x_1^a = 0, x_2^a = 0 \\ I : a, c, \mathsf{priv}(c) \end{array} \right) \rightarrow \left(\begin{array}{l} x_1^a = 1, x_2^a = 0 \\ I : a, c, \mathsf{priv}(c), \{< a, N_1 >\}_{\mathsf{pub}(c)} \end{array} \right)
$$

$$
\rightarrow \left(\begin{array}{l} x_1^a = 1, x_2^a = 1, x = N_1 \\ I : a, c, \mathsf{priv}(c), \{< a, N_1 >\}_{\mathsf{pub}(c)}, \{< N_1, N_2 >\}_{\mathsf{pub}(a)} \end{array} \right)
$$

$$
\rightarrow \left(\begin{array}{l} x_1^a = 2, x_2^a = 1, x = N_1, y = N_2 \\ I : a, c, \mathsf{priv}(c), \{< a, N_1 >\}_{\mathsf{pub}(c)}, \\ \quad \{< N_1, N_2 >\}_{\mathsf{pub}(a)}, \{N_2\}_{\mathsf{pub}(c)} \end{array} \right)
$$

From which N_2 can be retrieved.

HORN CLAUSE APPROACH

Guess the role instances and the interleaving of the rules.

Protocol rules:

$$I_k(u_k) \Rightarrow I_{k+1}(v_k)$$

The intruder's knowledge is increasing:

$$I_k(x) \Rightarrow I_{k+1}(x)$$

Initial knowledge:

$$I_0(t)$$

n copies of each inference rule.

$$\neg I_n(s)$$

Is this set of clauses satisfi able ?

HORN CLAUSE APPROACH (2)

$$
\begin{array}{rcl}
 & & I_0(a) \\
 & & I_0(\mathsf{priv}(c)) \\
 & & I(1)(\{< a, N_1 >\}_{\mathsf{pub}(c)}) \\
I_1(\{< a, x >\}_{\mathsf{pub}(a)}) & \Rightarrow & I_2(\{< x, N_2 >\}_{\mathsf{pub}(a)}) \\
I_2(\{< N_1, y >\}_{\mathsf{pub}(a)}) & \Rightarrow & I_3(\{y\}_{\mathsf{pub}(c)})
\end{array}
$$

$$
\begin{array}{rcl}
 & & I_0(c) \\
I_0(x) & \Rightarrow & I_1(x) \\
I_1(x) & \Rightarrow & I_2(x) \\
I_2(x) & \Rightarrow & I_3(x) \\
I_3(N_2) & \Rightarrow & \bot
\end{array}
$$

$$
\begin{array}{rcl}
I_k(x), I_k(y) & \Rightarrow & I_k(< x, y >) \\
I_k(x), I_k(y) & \Rightarrow & I_k(\{x\}_y) \\
I_k(\{x\}_{\mathsf{pub}(y)}), I_k(\mathsf{priv}(y)) & \Rightarrow & I_k(x)
\end{array}
$$

$$
\begin{array}{rcl}
I_k(< x, y >) & \Rightarrow & I_k(x) \\
I_k(< x, y >) & \Rightarrow & I_k(y)
\end{array}
$$

for k = 0, ..., 3.

This set of clauses is unsatisfi able.

HORN CLAUSE APPROACH (3)

Exercise 8 (level 3) Give a protocol example (together with role instances and interleaving of rules) for which the set of clauses is unsatisfi able, while ther is no attack.

HORN CLAUSE APPROACH (3)

Exercise 8 (level 3) Give a protocol example (together with role instances and interleaving of rules) for which the set of clauses is unsatisfi able, while ther is no attack.

To solve this problem: *Rigidify* the protocol clauses

THE REWRITING APPROACH

The deducibility problem of s, given $t_1, \ldots, t_n$ is stated as

Find C such that $C[t_1, \ldots, t_n] \xrightarrow[\mathcal{R}]{*} s$

$\mathcal{R}$ encodes decomposition rules (see part 3):

$$\mathsf{dec}(\{x\}_{\mathsf{pub}(y)}, \mathsf{priv}(y)) \xrightarrow[\mathcal{R}]{} y$$

For a bounded number of sessions: $C_0, C_1, \ldots, C_{n+1}$ and assignments for variables of $u_1, \ldots, u_n$ such that

$$\begin{cases} C_0[t_1, \ldots, t_k] & \xrightarrow[\mathcal{R}]{*} & u_1 \\ C_1[v_1, t_1, \ldots, t_k] & \xrightarrow[\mathcal{R}]{*} & u_2 \\ \vdots & & \\ C_n[v_1, \ldots, v_n, t_1, \ldots, t_k] & \xrightarrow[\mathcal{R}]{*} & s \end{cases}$$

THE CONSTRAINT SOLVING APPROACH (1)

Find a substitution σ, which satisfi es the system:

$$\begin{cases} t_1, \ldots, t_k & \Vdash & u_1 \\ v_1, t_1 \ldots, t_k & \Vdash & u_2 \\ \vdots & & \\ v_1, \ldots, v_n, t_1, \ldots, t_k & \Vdash & s \end{cases}$$

σ is a solution of $T \Vdash u$ if $T\sigma \vdash_I u\sigma$.

THE CONSTRAINT SOLVING APPROACH (2)

R_1	$T \Vdash u$	$\rightsquigarrow$	$\top$	If $T \cup \{x \mid T' \Vdash x \in C, T' \subseteq T, T' \neq T\} \vdash u\}$
R_2	$C \wedge T \Vdash u$	$\rightsquigarrow_\sigma$	$C\sigma \wedge T\sigma \Vdash u\sigma$	If $\sigma = mgu(t, u)$, $t \in F(T), t \neq u$, t, u non variable
R_3	$C \wedge T \Vdash u$	$\rightsquigarrow_\sigma$	$C\sigma \wedge T\sigma \Vdash u\sigma$	If $\sigma = mgu(t_1, t_2)$, $t_1, t_2 \in F(T), t_1 \neq t_2$, t_1, t_2 no
R_4	$T \Vdash \{u\}_v$	$\rightsquigarrow$	$T \Vdash u \wedge T \Vdash v$	
R_5	$T \Vdash < u, v >$	$\rightsquigarrow$	$T \Vdash u \wedge T \Vdash v$	
R_6	$T \Vdash u$	$\rightsquigarrow$	$\bot$	If $T = \emptyset$ or else $Var(T, u) = \emptyset$ and $T \not\vdash u$

Assumes that F is the set of subterms and deduction uses the DY theory. (for simplicity, we only consider symmetric encryption).

EXERCISE 9 (LEVEL 5)

Assume we have a conjunction of constraints $T_i \Vdash u_i$ such that $T_i \subseteq T_{i+1}$ and if $x \in Var(T_i)$, then there is an index $j < i$ such that $x \in Var(u_j)$.

1. Show that the above conditions are invariant by the constraint solving rules
2. Show that, if $C \rightsquigarrow_\sigma C'$, then, for every solution θ of C', $\sigma\theta$ is a solution of C.
3. Show that the constraint solving rules terminate.
4. A *solved form* is a constraint such that every right member is a variable. Show that every solved form has at least one solution.
5. Show that every constraint which is not in solved form can be reduced using one of the constraint solving rules.
6. Explain why the constraint solving rules solve the insecurity decision problem.
7. What is the complexity of the resulting decision procedure ?

PROTOCOLS AS ORACLES

Intruder's point of view: there is an additional deduction rule:

$$\frac{u_i\sigma}{v_i\sigma}$$

if some agent reached step $i-1$ in the role to which belongs the rule $u_i \rightarrow v_i$ and σ is compatible with the partial bindings at this stage.
Confi dentiality is then a deducibility problem in some formal proof system.

LIFTING INTRUDER RULES

We consider constrained formulas $u \; [\![E]\!]$. For arbitrary rules in the offline theory:

$$\frac{u_1 \qquad \cdots \quad u_n}{u}$$

LIFTING INTRUDER RULES

We consider constrained formulas $u \; [\![E]\!]$. For arbitrary rules in the offline theory:

$$\frac{u'_1 \;\; [\![E_1]\!] \;\; \cdots \;\; u'_n \;\; [\![E_n]\!]}{u \;\; [\![E_1 \wedge \ldots \wedge E_n \wedge R]\!]}$$

u'_i is the linearized version of the term and R are the co-reference constraints.

LIFTING INTRUDER RULES

We consider constrained formulas $u \; [\![E]\!]$. For arbitrary rules in the offline theory:

$$\frac{u'_1 \; [\![E_1]\!] \quad \cdots \quad u'_n \; [\![E_n]\!]}{u \; [\![E_1 \wedge \ldots \wedge E_n \wedge R]\!]}$$

u'_i is the linearized version of the term and R are the co-reference constraints.

$$\frac{\{x\}_{\mathsf{pub}(y)} \; [\![E_1]\!] \quad \mathsf{priv}(z) \; [\![E_2]\!]}{x \; [\![E_1 \wedge E_2 \wedge y = z]\!]}$$

THE PROTOCOL RULES AS ORACLES

$$\frac{u \; [\![E \wedge x_{s,A,k-1} = 1]\!]}{w_k \; [\![E \wedge x_{s,A,k} = 1 \wedge u = v_k]\!]}$$

$v_k \rightarrow w_k$ is the kth rule of the s instance of role A.

A FORMAL PROOF SYSTEM (CNTD)

Instanciation:

$$\frac{C[x_1, \ldots, x_n]\ [\![E \wedge x_1 = u_1 \wedge \ldots \wedge x_n = u_n]\!]}{C[u_1, \ldots, u_n]\ [\![E \wedge x_1 = u_1 \wedge \ldots \wedge x_n = u_n]\!]}$$

For every (strict) abstraction C of the intruder inference premisses.

For DY-like system C is empty of the context $\{x\}\_$; we have the two rules

$$\frac{x\ [\![x = u \wedge E]\!]}{u\ [\![x = u \wedge E]\!]}$$

$$\frac{\{x\}_y\ [\![y = u \wedge E]\!]}{\{x\}_u\ [\![y = u \wedge E]\!]}$$

A FORMAL PROOF SYSTEM (CNTD)

Constraint solving rule: $E \mapsto E'$ if E' is a solved form of E. (DAG representation).

Weakening:

$$\frac{x \; [\![E_1]\!] \quad y \; [\![E_2]\!]}{x \; [\![E_1 \wedge E_2]\!]}$$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\frac{0 \ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)} \ [\![x_A = 1]\!]}$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\dfrac{\mathsf{priv}(c)\ [\![\emptyset]\!] \qquad \dfrac{0\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)}\ [\![x_A = 1]\!]}}{< a, N_1 >\ [\![x_A = 1]\!]}$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\dfrac{\mathsf{priv}(c)\ [\![\emptyset]\!] \qquad \dfrac{0\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)}\ [\![x_A = 1]\!]}}{< a, N_1 >\ [\![x_A = 1]\!]} \qquad \dfrac{< a, N_1 >\ [\![x_A = 1]\!] \qquad \mathsf{pub}(a)\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1]\!]}$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\dfrac{\dfrac{\mathsf{priv}(c)\ [\![\emptyset]\!] \qquad \dfrac{0\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)}\ [\![x_A = 1]\!]}}{< a, N_1 >\ [\![x_A = 1]\!]} \qquad \dfrac{\mathsf{pub}(a)\ [\![\emptyset]\!] \qquad \{< a, N_1 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1]\!]}{\{< y, N_2 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1 \wedge x_B = 1 \wedge \{< a, y >\}_{\mathsf{pub}(a)} = \{< a, N_1 >\}_{\mathsf{pub}(a)}]\!]}$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\dfrac{\dfrac{\mathsf{priv}(c)\ [\![\emptyset]\!] \qquad \dfrac{0\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)}\ [\![x_A = 1]\!]}}{< a, N_1 >\ [\![x_A = 1]\!]} \qquad \dfrac{\mathsf{pub}(a)\ [\![\emptyset]\!] \qquad \{< a, N_1 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1]\!]}{\{< y, N_2 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1 \wedge x_B = 1 \wedge y = N_1]\!]}$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\dfrac{\mathsf{priv}(c)\ [\![\emptyset]\!] \qquad \dfrac{0\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)}\ [\![x_A = 1]\!]}}{< a, N_1 >\ [\![x_A = 1]\!]} \qquad \dfrac{\mathsf{pub}(a)\ [\![\emptyset]\!] \qquad \dfrac{\{< a, N_1 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1]\!]}{\{< y, N_2 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1 \wedge x_B = 1 \wedge y = N_1]\!]}}{\{x\}_{\mathsf{pub}(c)}\ [\![x_A = 2 \wedge x_B = 1 \wedge y = N_1 \wedge \{< N_1, x >\}_{\mathsf{pub}(a)} = \{< y, N_2 >\}_{\mathsf{pub}(a)}]\!]}$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\dfrac{\dfrac{\mathsf{priv}(c)\ [\![\emptyset]\!] \qquad \dfrac{0\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)}\ [\![x_A = 1]\!]}}{< a, N_1 >\ [\![x_A = 1]\!]} \qquad \dfrac{\mathsf{pub}(a)\ [\![\emptyset]\!] \qquad \{< a, N_1 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1]\!]}{\{< y, N_2 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1 \wedge x_B = 1 \wedge y = N_1]\!]}}{\{x\}_{\mathsf{pub}(c)}\ [\![x_A = 2 \wedge x_B = 1 \wedge y = N_1 \wedge x = N_2]\!]}$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

PROOF EXAMPLE: NS ATTACK

A:(with a, c)

$0 \Rightarrow \{< a, N_1 >\}_{\mathsf{pub}(c)}$

$\{< N_1, x >\}_{\mathsf{pub}(a)} \Rightarrow \{x\}_{\mathsf{pub}(c)}$

B (with a, a)

$\{< a, y >\}_{\mathsf{pub}(a)} \Rightarrow \{< y, N_2 >\}_{\mathsf{pub}(a)}$

$\{N_2\}_{\mathsf{pub}(a)} \Rightarrow 0$

$$\dfrac{\dfrac{\mathsf{priv}(c)\ [\![\emptyset]\!] \qquad \dfrac{0\ [\![\emptyset]\!]}{\{< a, N_1 >\}_{\mathsf{pub}(c)}\ [\![x_A = 1]\!]}}{< a, N_1 >\ [\![x_A = 1]\!]} \qquad \mathsf{pub}(a)\ [\![\emptyset]\!]}{\dfrac{\dfrac{\{< a, N_1 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1]\!]}{\{< y, N_2 >\}_{\mathsf{pub}(a)}\ [\![x_A = 1 \wedge x_B = 1 \wedge y = N_1]\!]}}{\{x\}_{\mathsf{pub}(c)}\ [\![x_A = 2 \wedge x_B = 1 \wedge y = N_1 \wedge x = N_2]\!]} \qquad \mathsf{priv}(c)\ [\![\emptyset]\!]}$$

$$x\ [\![x_A = 2 \wedge x_B = 1 \wedge y = N_1 \wedge x = N_2]\!]$$

$T \supseteq \{0, \mathsf{pub}(a), a, \mathsf{priv}(c)\}$

EXAMPLE OF DEDUCTION (2)

$$A(B) : \nu s. \qquad \Rightarrow \quad \{< A, \{s\}_{\mathsf{pub}(B)} >\}_{\mathsf{pub}(B)}$$
$$B(A) : \quad \{< A, \{x\}_{\mathsf{pub}(B)} >\}_{\mathsf{pub}(B)} \quad \Rightarrow \quad \{< B, \{x\}_{\mathsf{pub}(A)} >\}_{\mathsf{pub}(A)}$$

s has to remain secret (when generated by a honest agent for an honest agent).

Exercise 10 (level 4) Give an attack and its proof in the deduction system (Ind: use 1 instance of the fi rst role and 3 instances of the second role).

ADEQUACY OF THE DEDUCTION SYSTEM

Theorem 3 There is an attack on confi dentiality iff the secret can be deduced from the intruder intial knowledge, using the extended inference system.

This is true also for an unbounded number of sessions.

Back to F-locality

R: (renamed) protocol rules, L: abstractions of left sides of R. D: right sides of R. F: $2^{T(F,X)} \rightarrow 2^{T(F,X)}$, $S \subseteq T(F,X)$

$$T_R{}^F(S) = \{g\theta \mid g \in L, \forall x.x\theta \in F(S) \cup F(D)\}$$

If E is a solvable constraint, σ_E is the mgu of E.

The extended deduction system is F-local if, whenever there is a proof of s $[\![E]\!]$ with hypotheses S, then there is a proof of s' $[\![E']\!]$ with the same hypotheses and such that

- All intermediate constrained formulas u $[\![E]\!]$ are such that $u \in T_R{}^F(S) \cup F(\{s'\})$ and equations in E are in $(F(L) \cup F(D) \cup F(T))^2$
- $s\sigma_E = s'\sigma_{E'}$

LOCALITY OF EXTENDED SYSTEMS

Let F be the set of subterms (possibly with a $\mathsf{pub}(\_)$ in front).

Theorem 4[Rusinowitch, Turuani, 2001]. Dolev-Yao extended systems are F-local.

PROOF OF THEOREM 4 (I)

fi rst step: delay intanciations and weakenings as much as possible.

$$\dfrac{\dfrac{\dfrac{\Pi_1}{x\ [\![x = s]\!]}}{s\ [\![x = s]\!]} \qquad \dfrac{\Pi_2}{t\ [\![E]\!]}}{< s, t >\ [\![x = s \wedge E]\!]} \quad \Rightarrow \quad \dfrac{\dfrac{\Pi_1}{x\ [\![x = s]\!]} \qquad \dfrac{\Pi_2}{t\ [\![E]\!]}}{< x, t >\ [\![x = s]\!]}$$

Instanciations are followed by decompositions and weakenings are followed by weakening of protocol rules. We group instanciations with the following decomposition. Weakenings are also grouped with the following protocol rule.

Proof of theorem 4 (II)

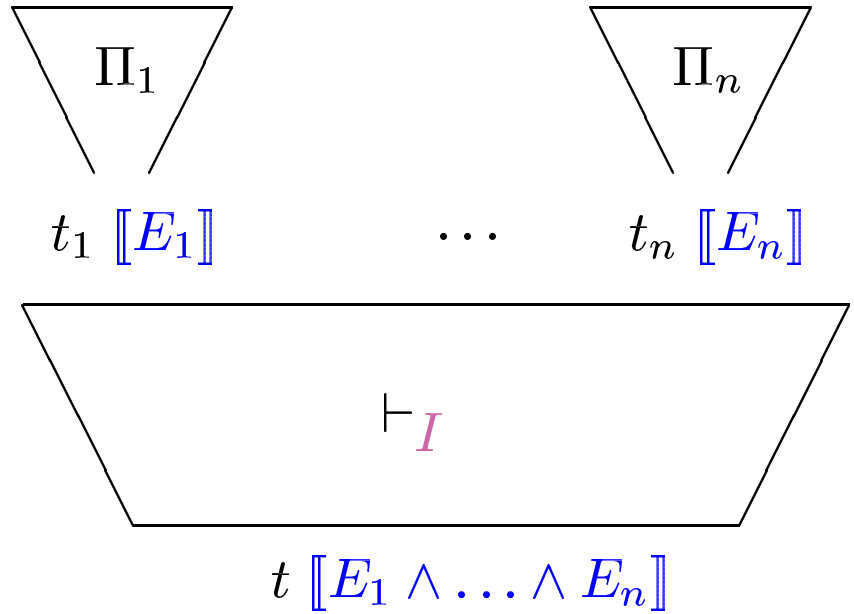
By induction the size of the proof Π : If $S \vdash t \; [\![E]\!]$ then there is a proof $S \vdash t \; [\![E_0]\!]$ such that:

- All terms in the proofs belong to ${T_R}^F \cup F(\{t\})$
- If the last rule is not a construction or a weakening, then the conclusion is in ${T_R}^F$
- Every (solved) constraint in the proof is a conjunction of $x = k$ and $x = v$ with $v \in F(L) \cup F(D) \cup F(S)$
- E_0 is obtained from E by replacing right hand sides s of E which are not in ${T_R}^F$ with 0. For such s there are subproofs of Π yielding $s \; [\![E']\!]$, ending with a construction rule and such that $E \models E'$.

The base case is straightforward

PROOF OF THEOREM 4 (III)

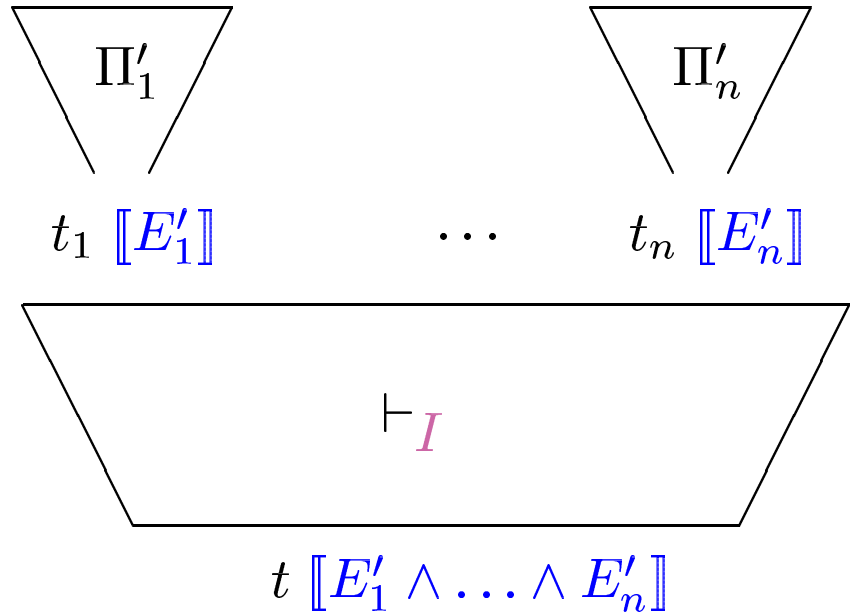
Case 1: intruder rules



The $\vdash_I$ part is maximal.

PROOF OF THEOREM 4 (III)

Case 1: intruder rules

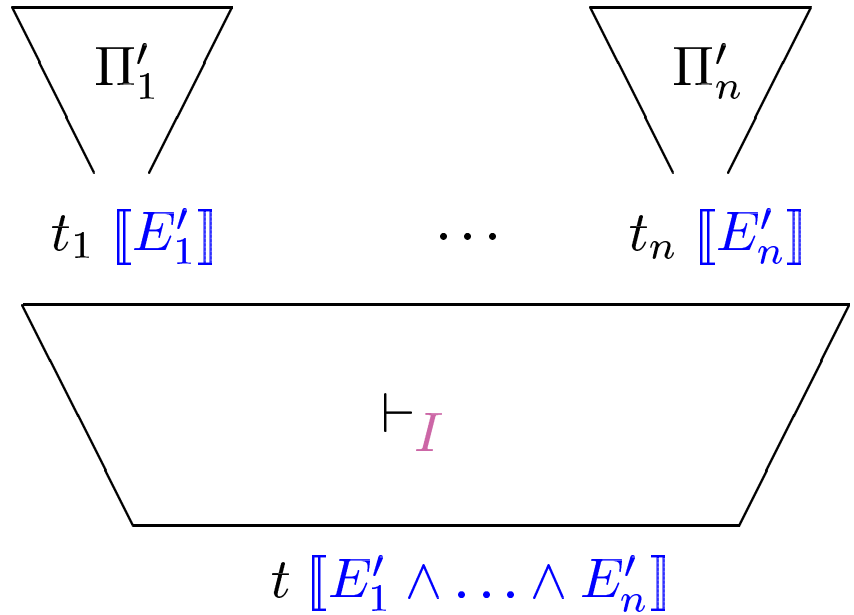


The $\vdash_I$ part is maximal.

Apply the induction hypothesis

PROOF OF THEOREM 4 (III)

Case 1: intruder rules



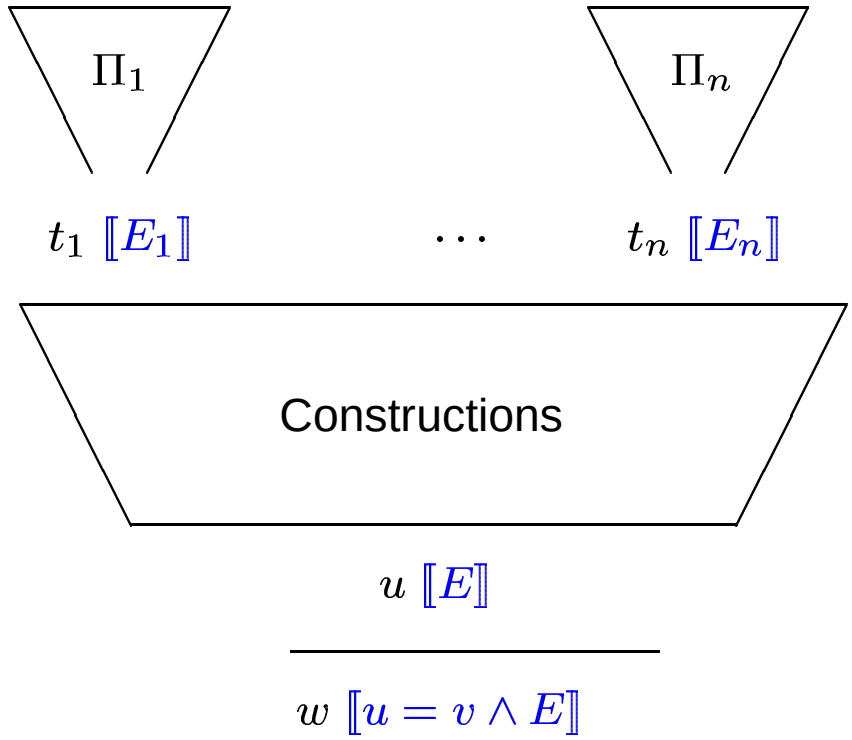
The $\vdash_I$ part is maximal.

Apply the induction hypothesis

Apply the locality property of $\vdash_I$

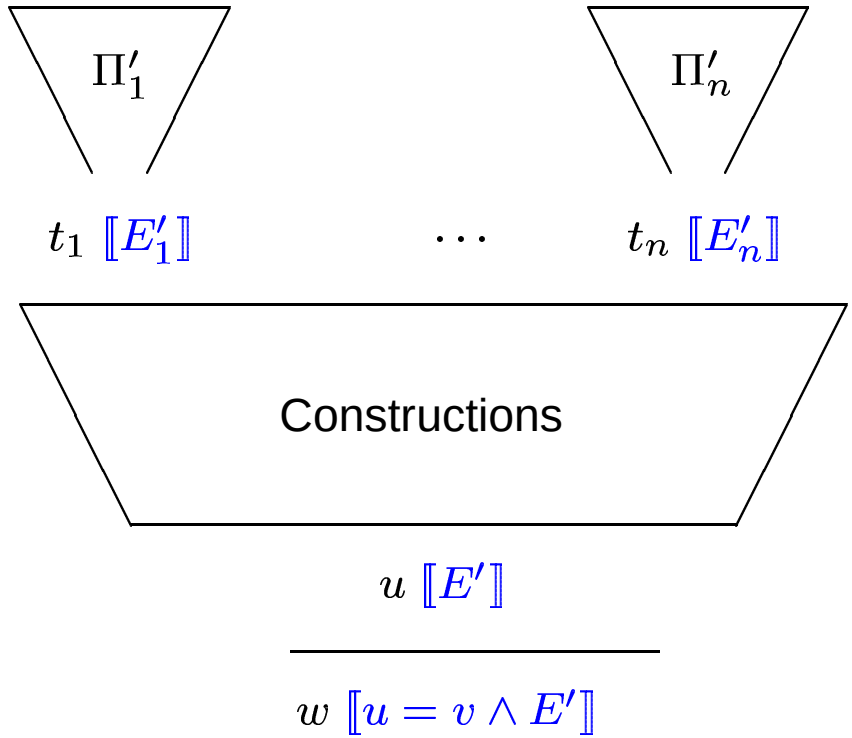
PROOF OF THEOREM 4 (IV)

case 2: protocol rule



PROOF OF THEOREM 4 (IV)

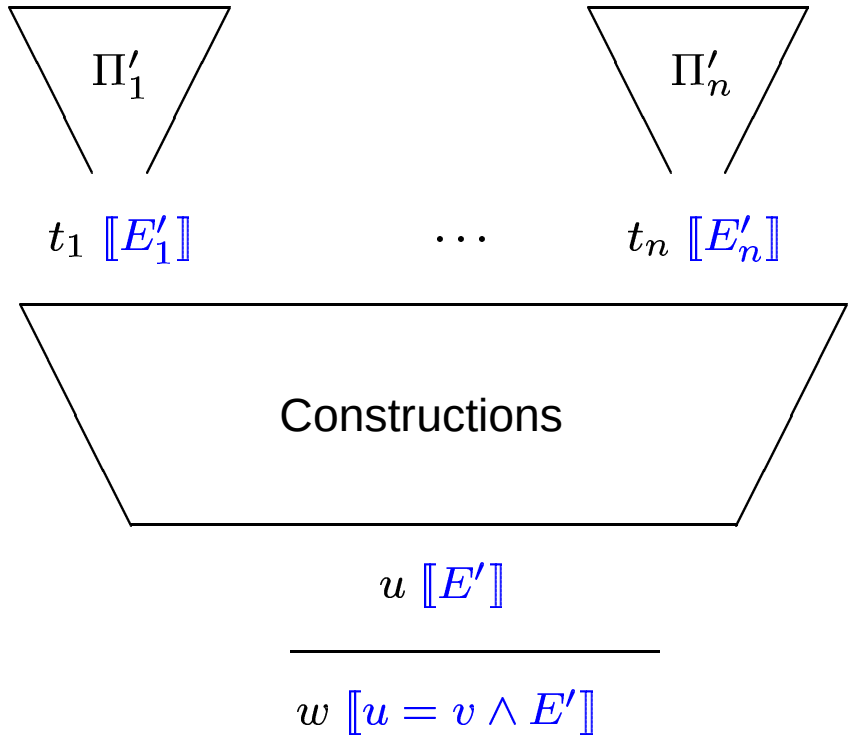
case 2: protocol rule



Apply the induction hypothesis

PROOF OF THEOREM 4 (IV)

case 2: protocol rule

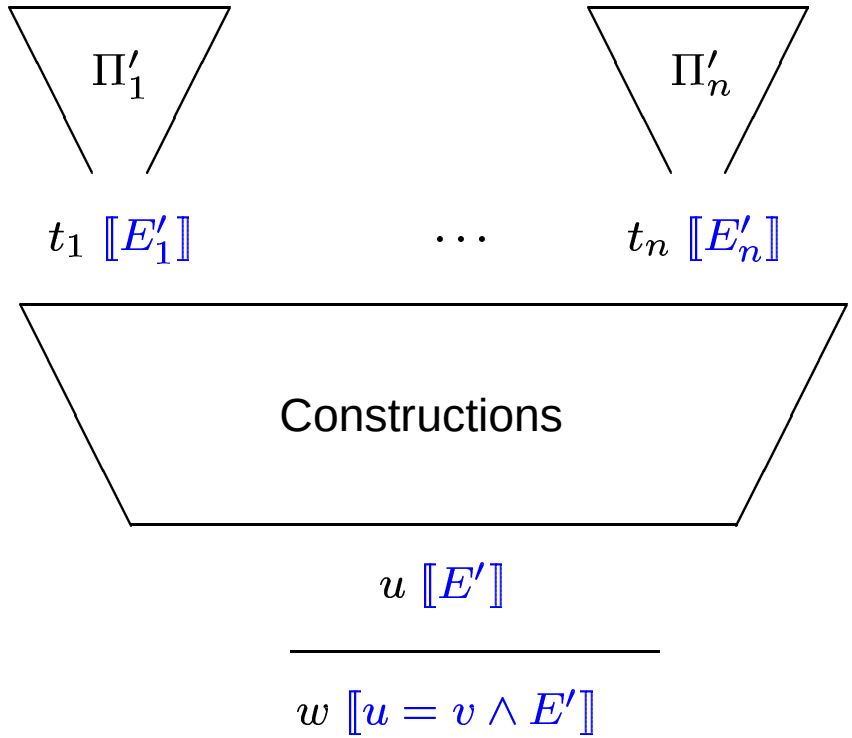


Apply the induction hypothesis

Subcase 1: equation $x = s$ in the solved form of $u = v$ is such that s is a subterm of v or $s \in F(D) \cup F(S)$

PROOF OF THEOREM 4 (IV)

case 2: protocol rule



Apply the induction hypothesis

Subcase 1: equation $x = s$ in the solved form of $u = v$ is such that s is a subterm of v or $s \in F(D) \cup F(S)$

Subcase 2: $x = s$ is such that $s \notin F(L) \cup F(D) \cup F(S)$: we replace s with 0. Possible, by induction on the size of the construction part.

PROOF OF THEOREM 4 (V)

case 3: instanciation

$$\dfrac{\dfrac{\dfrac{\Pi}{x\ [\![x = s \wedge E]\!]}}{<u, v>\ \ [\![x =< u, v > \wedge E]\!]}}{u\ [\![x =< u, v > \wedge E]\!]}$$

When $< u, v >$ is too large, there is a simpler proof of $< u, v >$ (with more general constraints), by induction hypothesis.

CONSEQUENCES

Corollary: The security problem for DY theories for a bounded number of sessions is in co-NP.

Simply guess the terms in ${T_R}^F$.

EXTENSIONS

- More general defi nition of constructions
- Conditions on decomposition/instanciation
- Considering Associative and commutative symbols
  Relies on the fi niteness of equivalence classes.
  Generalizes both **[CL, Shmatikov, 03] [Chevalier et al., 03]**.

PART 3

INTRODUCING ALGEBRAIC PROPERTIES

THE GOALS

- Go beyond the perfect cryptography assumption
- Get rid of algebraic properties (modifying the protocol and deduction rules): be back to perfect cryptography.

EXAMPLE 1: INVERSE KEY

$$\begin{cases} \{\{x\}_k\}_{k^{-1}} & = & x \\ (k^{-1})^{-1} & = & k \end{cases}$$

EXAMPLE 1: INVERSE KEY

$$\left\{ \begin{array}{rcl} \{\{x\}_k\}_{k^{-1}} & = & x \\ (k^{-1})^{-1} & = & k \end{array} \right.$$

$$\begin{array}{rcl} A(B) : \nu K_{AB}. & \Rightarrow & \{< A, \{K_{AB}\}_{\mathsf{pub}(A)^{-1}} >\}_{\mathsf{pub}(B)} \\ B(A) : \nu s. \quad \{< A, x >\}_{\mathsf{pub}(B)} & \Rightarrow & [s]_{\{x\}_{\mathsf{pub}(A)}} \end{array}$$

EXAMPLE 1: INVERSE KEY

$$\begin{cases} \{\{x\}_k\}_{k^{-1}} &= x \\ (k^{-1})^{-1} &= k \end{cases}$$

$$\begin{array}{rlll} A(B) : \nu K_{AB}. & & \Rightarrow & \{< A, \{K_{AB}\}_{\mathsf{pub}(A)^{-1}} >\}_{\mathsf{pub}(B)} \\ B(A) : \nu s. & \{< A, x >\}_{\mathsf{pub}(B)} & \Rightarrow & [s]_{\{x\}_{\mathsf{pub}(A)}} \end{array}$$

One instance of role B with $B = b$, $A = a$ and $x = 0$. Yields $[s]_{\{0\}_{\mathsf{pub}(a)}}$.

EXAMPLE 2: EXPLICIT DECRYPTION

Don’t consider the decomposition inference rules. Instead:

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & = & x \\ \pi_1(< x, y >) & = & x \\ \pi_2(< x, y >) & = & y \end{cases}$$

EXAMPLE 2: EXPLICIT DECRYPTION

Don’t consider the decomposition inference rules. Instead:

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & = & x \\ \pi_1(< x, y >) & = & x \\ \pi_2(< x, y >) & = & y \\ \{\mathsf{dec}(x, k)\}_{k^{-1}} & = & x \end{cases}$$

EXAMPLE 2: EXPLICIT DECRYPTION

Don't consider the decomposition inference rules. Instead:

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & = & x \\ \pi_1(< x, y >) & = & x \\ \pi_2(< x, y >) & = & y \\ \{\mathsf{dec}(x, k)\}_{k^{-1}} & = & x \end{cases}$$

$$\begin{array}{rrcl} A(B,S) : \nu N. & & \Rightarrow & < A, B, \{N\}_{\mathsf{shr}(A,S)} > \\ S() : & < x, y, z > & \Rightarrow & < x, y, \{\mathsf{dec}(z, \mathsf{shr}(x, S)^{-1})\}_{\mathsf{shr}(y,S)} \\ B(A,S) : \nu s & < A, B, x > & \Rightarrow & \{s\}_{\mathsf{dec}(x,\mathsf{shr}(B,S)^{-1})} > \end{array}$$

EXAMPLE 3: EXCLUSIVE OR

$$\begin{aligned} x \oplus x \oplus y &\rightarrow y \\ x \oplus x &\rightarrow 0 \\ x \oplus 0 &\rightarrow x \end{aligned}$$

$$\begin{aligned} x \oplus (y \oplus z) &= (x \oplus y) \oplus z \\ x \oplus y &= y \oplus x \end{aligned}$$

EXAMPLE 4: MODULAR EXPONENTIATION

$$
\begin{array}{lrcl}
\mathcal{AG} & x * y & = & y * x \\
& x * (y * z) & = & (x * y) * z \\
& (x^{-1})^{-1} & = & x \\
& 1 * x & = & x \\
& 1^{-1} & = & 1 \\
& x * x^{-1} & = & 1 \\
& (x * y)^{-1} & = & x^{-1} * y^{-1} \\
& x * y * y^{-1} & = & x \\
\mathcal{DH} & \mathsf{exp}(x, 1) & = & x \\
& \mathsf{exp}(\mathsf{exp}(x, y), z) & = & \mathsf{exp}(x, y * z)
\end{array}
$$

(group DH : see Madhura)

HOMOMORPHISMS

$$h(< x, y >) =< h(x), h(y) >$$

$$\begin{aligned} h(x \oplus y) &= h(x) \oplus h(y) \\ h(0) &= 0 \end{aligned}$$

$$\{x \oplus y\}_k = \{x\}_k \oplus \{y\}_k$$

CLASSICAL TERM REWRITING

$s \xrightarrow[l \to r]{} t$ if there is a position p in s and a substitution σ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$.

If $\mathcal{R}$ is a set of rewrite rules, this defines a reduction relation $\xrightarrow[\mathcal{R}]{}$.

Termination: no infinite sequence of reductions.

Confluence: $\xleftarrow[\mathcal{R}]{*} \circ \xleftarrow[\mathcal{R}]{*} \subseteq \xrightarrow[\mathcal{R}]{*} \circ \xleftarrow[\mathcal{R}]{*}$

Confluence + Termination $\Rightarrow$ the normal forms exist and are unique.

then $=_{\mathcal{R}}= \xrightarrow[\mathcal{R}]{*} \circ \xleftarrow[\mathcal{R}]{*}$

AC-REWRITING

AC-rewriting: $s \xrightarrow[AC\setminus l\to r]{} t$ if there is a position p in s and a substitution σ such that $s|_p =_{AC} l\sigma$ and $t = s[r\sigma]_p$.

$\mathcal{R}$ is AC-convergent if

- $\xrightarrow[AC\setminus\mathcal{R}]{} \circ =_{AC}$ is terminating,
- $=_{AC} \circ \xrightarrow[AC\setminus\mathcal{R}]{} \subseteq \xrightarrow[AC\setminus\mathcal{R}]{*} \circ =_{AC}$ (local coherence)
- $\xleftarrow[AC\setminus\mathcal{R}]{} \circ \xrightarrow[AC\setminus\mathcal{R}]{} \subseteq \xrightarrow[AC\setminus\mathcal{R}]{*} \circ =_{AC} \circ \xleftarrow[AC\setminus\mathcal{R}]{*}$ (local confluence)

For AC-convergent systems, normal forms are unique, up to AC equality.

All above-mentioned systems are AC-convergent when orienting the (non AC) axioms from left to right.

VARIANTS

$E' \subseteq E$. E converted into an E'-convergent rewrite system $\mathcal{R}$.

S is a fi nite set of variantsof t if

> For every normalized substitution σ, there is a $u \in S$ and a substitution θ s.t. $t\sigma \downarrow_{\mathcal{R}} =_{E'} u\theta$

E has the fi nite variant property (modulo E') if, for every term t, we can compute a fi nite set of variants.

Equivalent property:

$$\forall t.\ \exists n.\ \forall \sigma \in \Sigma_N.\ t\sigma \xrightarrow[AC\setminus\mathcal{R}]{\leq n} t\sigma \downarrow$$

Σ_N is the set of normalized substitutions.

WHY IS IT IMPORTANT ?

Theorem 5 Assume that E has the (AC)-fi nite variant property. Let R be a set of protocol rules And $\mathcal{I}$ be a set of intruder deduction rules.

let $R_1, \ldots, R_n$ be the variants of R and $\mathcal{I}'$ be the union of variants of inference rules in $\mathcal{I}$.

$$\left.\begin{array}{r}\text{Attack with} \\ R, \mathcal{I}, E\end{array}\right\} \Leftrightarrow \exists i. \left\{\begin{array}{l}\text{Attack with} \\ R_i, \mathcal{I}', (AC)\end{array}\right.$$

ALGEBRAIC THEORIES SATISFYING THE PROPERTY

Theorem 6: The theories of Inverse Keys, explicit decryption, exclusive or, Abelian Groups, Abelian Group+DH, have the (AC)-fi nite variant property.

Theorem 7: Exclusive or + Homomorphism does not have the AC fi nite variant property.

STANDARD NARROWING: EXAMPLES

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & \rightarrow & x \\ \pi_1(< x, y >) & \rightarrow & x \\ \pi_2(< x, y >) & \rightarrow & y \\ \{\mathsf{dec}(x, k)\}_{k^{-1}} & \rightarrow & x \end{cases}$$

$$\mathsf{dec}(x, y) \rightsquigarrow_{\{x \mapsto \{z\}_k ; y \mapsto k^{-1}\}} z$$

STANDARD NARROWING: EXAMPLES

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & \rightarrow & x \\ \pi_1(< x, y >) & \rightarrow & x \\ \pi_2(< x, y >) & \rightarrow & y \\ \{\mathsf{dec}(x,k)\}_{k^{-1}} & \rightarrow & x \end{cases}$$

$$\mathsf{dec}(x, y) \rightsquigarrow_{\{x \mapsto \{z\}_k; y \mapsto k^{-1}\}} z$$

$$h(< x, y >) \rightarrow < h(x), h(y) >$$

$$\begin{array}{lll} h(x_0) & \rightsquigarrow_{\{x_0 \mapsto < x_1, x_2 >\}} & < h(x_1), h(x_2) > \\ & \rightsquigarrow_{\{x_1 \mapsto < x_2, y_2 >\}} & << h(x_2), h(y_2) >, h(y_1) > \\ & \cdots & \end{array}$$

COMPUTING A NEW INFERENCE SYSTEM

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & \rightarrow & x \\ \pi_1(< x, y >) & \rightarrow & x \\ \pi_2(< x, y >) & \rightarrow & y \\ \{\mathsf{dec}(x, k)\}_{k^{-1}} & \rightarrow & x \end{cases}$$

$$\frac{x}{\pi_1(x)} \qquad \frac{x}{\pi_2(x)} \qquad \frac{x \quad y}{\mathsf{dec}(x, y)}$$

$$\frac{x \quad y}{\{x\}_y} \qquad \frac{x}{\mathsf{pub}(x)} \qquad \frac{x \quad y}{< x, y >}$$

$\Rightarrow$

COMPUTING A NEW INFERENCE SYSTEM

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & \rightarrow & x \\ \pi_1(< x, y >) & \rightarrow & x \\ \pi_2(< x, y >) & \rightarrow & y \\ \{\mathsf{dec}(x, k)\}_{k^{-1}} & \rightarrow & x \end{cases}$$

$$\frac{x}{\pi_1(x)} \qquad \frac{x}{\pi_2(x)} \qquad \frac{x \quad y}{\mathsf{dec}(x, y)}$$

$$\frac{x \quad y}{\{x\}_y} \qquad \frac{x}{\mathsf{pub}(x)} \qquad \frac{x \quad y}{< x, y >}$$

$$\Rightarrow \quad \frac{< x_1, x_2 >}{x_1}$$

COMPUTING A NEW INFERENCE SYSTEM

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & \rightarrow & x \\ \pi_1(< x, y >) & \rightarrow & x \\ \pi_2(< x, y >) & \rightarrow & y \\ \{\mathsf{dec}(x, k)\}_{k^{-1}} & \rightarrow & x \end{cases}$$

$$\frac{x}{\pi_1(x)} \qquad \frac{x}{\pi_2(x)} \qquad \frac{x \quad y}{\mathsf{dec}(x, y)}$$

$$\frac{x \quad y}{\{x\}_y} \qquad \frac{x}{\mathsf{pub}(x)} \qquad \frac{x \quad y}{< x, y >}$$

$$\Rightarrow \quad \frac{< x_1, x_2 >}{x_1} \qquad \frac{< x_1, x_2 >}{x_2}$$

COMPUTING A NEW INFERENCE SYSTEM

$$\begin{cases} \mathsf{dec}(\{x\}_k, k^{-1}) & \rightarrow & x \\ \pi_1(< x, y >) & \rightarrow & x \\ \pi_2(< x, y >) & \rightarrow & y \\ \{\mathsf{dec}(x, k)\}_{k^{-1}} & \rightarrow & x \end{cases}$$

$$\frac{x}{\pi_1(x)} \quad \frac{x}{\pi_2(x)} \quad \frac{x \quad y}{\mathsf{dec}(x, y)}$$

$$\frac{x \quad y}{\{x\}_y} \quad \frac{x}{\mathsf{pub}(x)} \quad \frac{x \quad y}{< x, y >}$$

$$\Rightarrow \quad \frac{< x_1, x_2 >}{x_1} \quad \frac{< x_1, x_2 >}{x_2} \quad \frac{\{x_1\}_k \quad k^{-1}}{x_1}$$

EXCLUSIVE OR EXAMPLE

In $\forall t.\ \exists n\ \ \forall \sigma \in \Sigma_N.\ t\sigma \xrightarrow[AC\setminus\mathcal{R}]{\leq n} t\sigma\downarrow$

we can choose $n = 2 \times |t|_{\oplus}$.

EXCLUSIVE OR EXAMPLE

In $\forall t.\ \exists n\ \ \forall \sigma \in \Sigma_N.\ t\sigma \xrightarrow[AC\setminus\mathcal{R}]{\leq n} t\sigma\downarrow$

we can choose $n = 2 \times |t|_{\oplus}$.

$$\frac{x_1 \quad x_2}{(x1 \oplus x_2)\downarrow}$$

becomes

$$\frac{x_1 \quad x_2}{x_1 \oplus x_2} \qquad \frac{x_3 \oplus x_2 \quad x_2}{x_3} \qquad \frac{x_3 \oplus x_4 \quad x_4 \oplus x_5}{x_3 \oplus x_5}$$

EXCLUSIVE OR EXAMPLE

In $\forall t.\ \exists n\ \ \forall \sigma \in \Sigma_N.\ t\sigma \xrightarrow[AC\setminus\mathcal{R}]{\leq n} t\sigma\downarrow$

we can choose $n = 2 \times |t|_{\oplus}$.

$$\frac{x_1 \quad x_2}{(x1 \oplus x_2)\downarrow}$$

becomes

$$\frac{x_1 \quad x_2}{x_1 \oplus x_2} \qquad \frac{x_3 \oplus x_2 \quad x_2}{x_3} \qquad \frac{x_3 \oplus x_4 \quad x_4 \oplus x_5}{x_3 \oplus x_5}$$

The new system is also F-local: this is always the case replacing with variants: F-local systems yield F-local systems.

THE ABELIAN GROUP CASE

$$
\begin{array}{rcl}
x * y & = & y * x \\
x * (y * z) & = & (x * y) * z \\
(x^{-1})^{-1} & \to & x \\
1 * x & \to & x \\
1^{-1} & \to & 1 \\
x * x^{-1} & \to & 1 \\
(x * y)^{-1} & \to & x^{-1} * y^{-1} \\
x * y * y^{-1} & \to & x
\end{array}
$$

The system is AC-convergent. However $t = x^{-1}$, $\sigma = \{x \mapsto a_1 * \cdots * a_n\}$:

$$t\sigma \xrightarrow[AC \setminus \mathcal{R}]{\not\leq \log n} a_1^{-1} * \cdots * a_n^{-1}$$

The Abelian Group case (cntd)

$$
\begin{array}{rcl}
x * y & = & y * x \\
(x^{-1})^{-1} & \rightarrow & x \\
1^{-1} & \rightarrow & 1 \\
(x * y)^{-1} & \leftarrow & x^{-1} * y^{-1}
\end{array}
$$

$$
\begin{array}{rcl}
x * (y * z) & = & (x * y) * z \\
1 * x & \rightarrow & x \\
x * x^{-1} & \rightarrow & 1 \\
x * y * y^{-1} & \rightarrow & x
\end{array}
$$

THE ABELIAN GROUP CASE (CNTD)

$$
\begin{array}{rcl}
x * y & = & y * x \\
(x^{-1})^{-1} & \rightarrow & x \\
1^{-1} & \rightarrow & 1 \\
(x * y)^{-1} & \leftarrow & x^{-1} * y^{-1} \\
(x^{-1} * y)^{-1} & \rightarrow & x * y^{-1} \\
x * (x * y)^{-1} & \rightarrow & y^{-1}
\end{array}
\qquad
\begin{array}{rcl}
x * (y * z) & = & (x * y) * z \\
1 * x & \rightarrow & x \\
x * x^{-1} & \rightarrow & 1 \\
x * y * y^{-1} & \rightarrow & x \\
z * x^{-1} * y^{-1} & \rightarrow & z * (x * y)^{-1} \\
z * x * (x * y)^{-1} & \rightarrow & z * y^{-1}
\end{array}
$$

is AC-convergent.

THE ABELIAN GROUP CASE (CNTD)

$$
\begin{array}{rcl}
x * y & = & y * x \\
(x^{-1})^{-1} & \to & x \\
1^{-1} & \to & 1 \\
(x * y)^{-1} & \leftarrow & x^{-1} * y^{-1} \\
(x^{-1} * y)^{-1} & \to & x * y^{-1} \\
x * (x * y)^{-1} & \to & y^{-1}
\end{array}
\qquad
\begin{array}{rcl}
x * (y * z) & = & (x * y) * z \\
1 * x & \to & x \\
x * x^{-1} & \to & 1 \\
x * y * y^{-1} & \to & x \\
z * x^{-1} * y^{-1} & \to & z * (x * y)^{-1} \\
z * x * (x * y)^{-1} & \to & z * y^{-1}
\end{array}
$$

is AC-convergent. Narrowing still does not terminate:

$$x^{-1} \rightsquigarrow_{\{x \mapsto x_1^{-1} * y_1\}} x_1 * y_1^{-1} \rightsquigarrow \cdots$$

THE ABELIAN GROUP CASE (CNTD)

$$
\begin{array}{rcl}
x * y & = & y * x \\
(x^{-1})^{-1} & \to & x \\
1^{-1} & \to & 1 \\
(x * y)^{-1} & \leftarrow & x^{-1} * y^{-1} \\
(x^{-1} * y)^{-1} & \to & x * y^{-1} \\
x * (x * y)^{-1} & \to & y^{-1}
\end{array}
\qquad
\begin{array}{rcl}
x * (y * z) & = & (x * y) * z \\
1 * x & \to & x \\
x * x^{-1} & \to & 1 \\
x * y * y^{-1} & \to & x \\
z * x^{-1} * y^{-1} & \to & z * (x * y)^{-1} \\
z * x * (x * y)^{-1} & \to & z * y^{-1}
\end{array}
$$

is AC-convergent. Narrowing still does not terminate:

$$x^{-1} \rightsquigarrow_{\{x \mapsto x_1^{-1} * y_1\}} x_1 * y_1^{-1} \rightsquigarrow \cdots$$

However we can choose $n = 2 \times |t|$ in

$$\forall t.\ \exists n\ \ \forall \sigma \in \Sigma_N.\ \ t\sigma \xrightarrow[AC \setminus \mathcal{R}]{\leq n} t\sigma\downarrow$$

CONCLUSION: WHAT DOESN'T WORK ?

Conclusion: what doesn't work ?

- On slide 73

CONCLUSION: WHAT DOESN'T WORK ?

- On slide 73 Current conditions are not met for the AG-variant inference system

CONCLUSION: WHAT DOESN'T WORK ?

- On slide 73 Current conditions are not met for the AG-variant inference system
- Homomorphisms ?