Algorithmic Verification Methods for Cryptographic Protocols

Liana Bozga

Verimag - Joseph Fourier University

Cryptographic Protocols

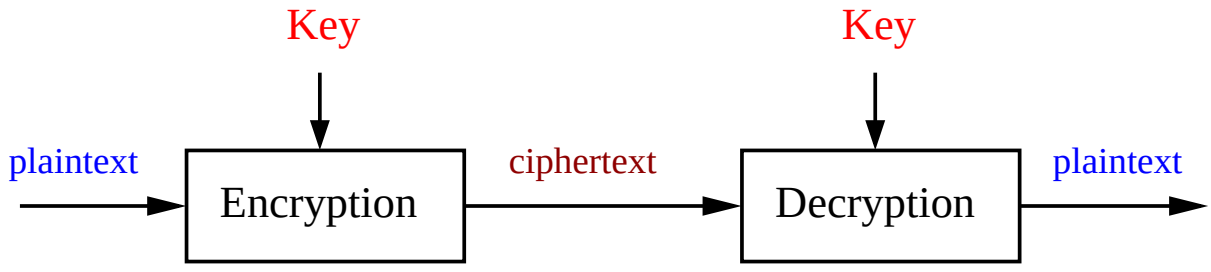
- are rules for exchanging messages
- ensure *secure communication* on an open network in the presence of adversaries
- application: ATM, e-commerce, electronic vote or contract signing, etc.
- properties:

  Secrecy: only authorized parties have access to information

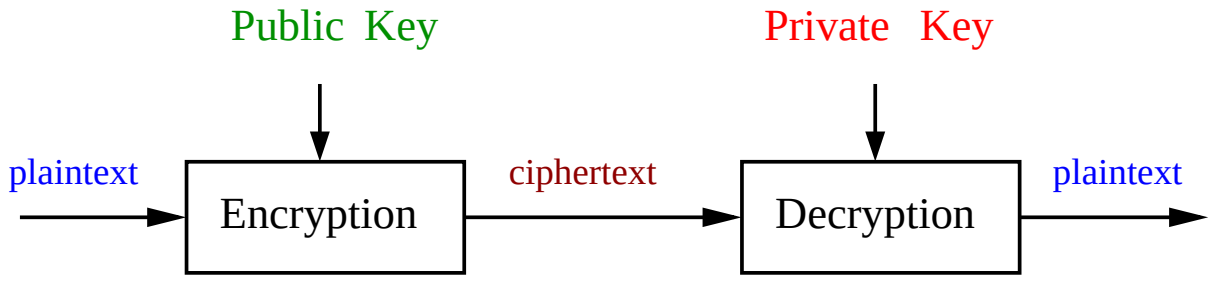
  Authenticity: identity claims (user, message)

Cryptographic Primitives

- Symmetric encryption



- Asymmetric encryption



The Needham-Schroeder Protocol with Asymmetric Keys

Purpose: Participants A and B exchange the secret nonce Nb

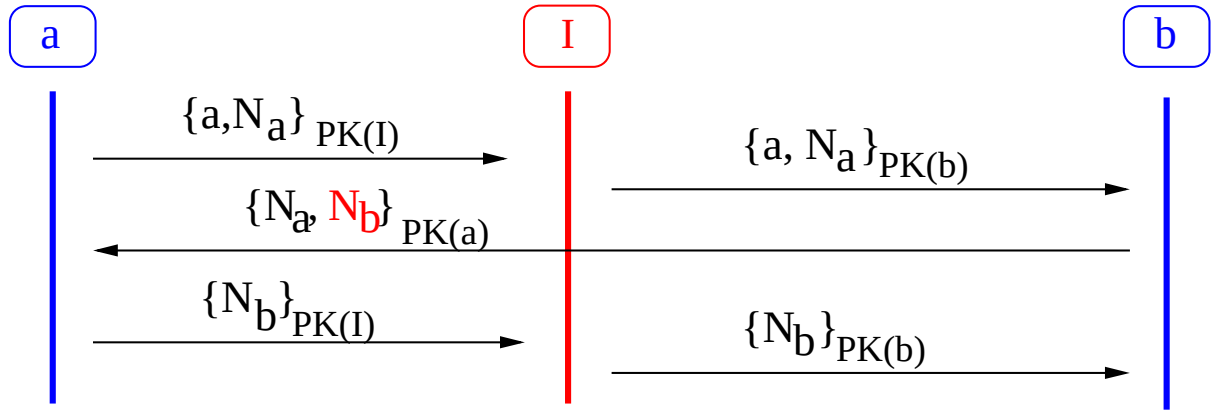
$$
\begin{aligned}
A \rightarrow B : &\ \{A, N_a\}_{PK(B)} \\
B \rightarrow A : &\ \{N_a, N_b\}_{PK(A)} \\
A \rightarrow B : &\ \{N_b\}_{PK(B)}
\end{aligned}
$$

The Needham-Schroeder Protocol with Asymmetric Keys

Purpose: Participants A and B exchange the secret nonce Nb

$$A \rightarrow B: \{A, N_a\}_{PK(B)}$$
$$B \rightarrow A: \{N_a, N_b\}_{PK(A)}$$
$$A \rightarrow B: \{N_b\}_{PK(B)}$$

Lowe'95:

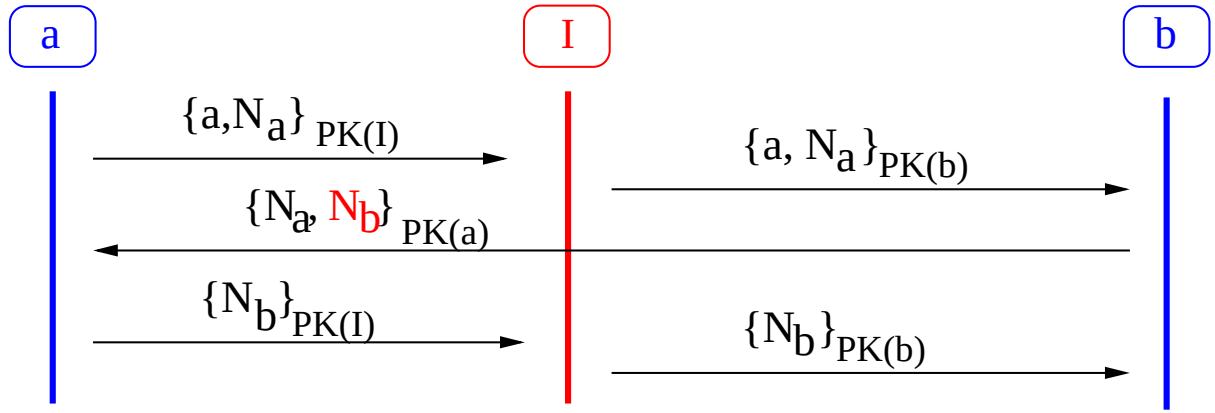


The Needham-Schroeder Protocol with Asymmetric Keys

Purpose: Participants A and B exchange the secret nonce Nb

$$A \rightarrow B : \{A, N_a\}_{PK(B)}$$
$$B \rightarrow A : \{N_a, N_b\}_{PK(A)}$$
$$A \rightarrow B : \{N_b\}_{PK(B)}$$

Lowe'95:



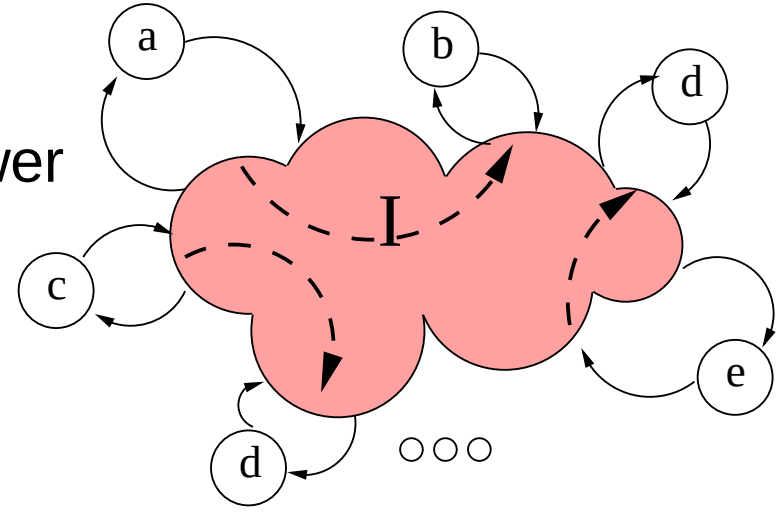
We consider that cryptographic algorithms are perfect

We are interested in logical flaws of protocols

Difficulties of the Verification

Adversary (Intruder)

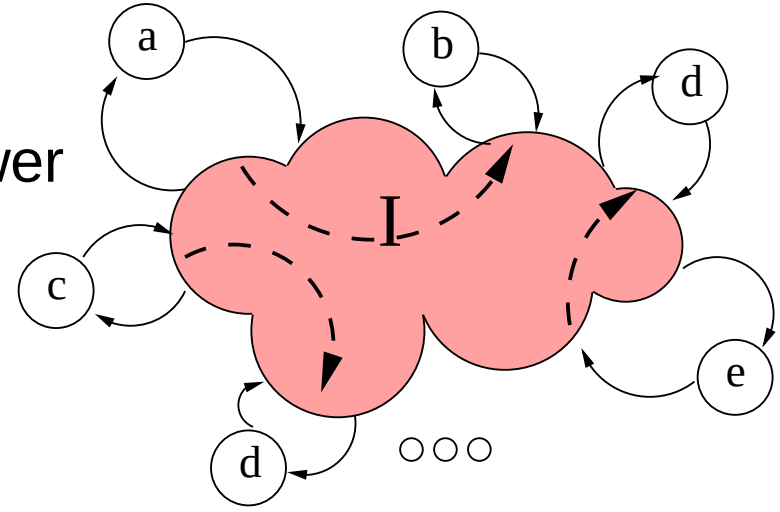
- complete control of network
- no bound on computation power
- no bound on memory



Difficulties of the Verification

Adversary (Intruder)

- complete control of network
- no bound on computation power
- no bound on memory



Protocol

- unbounded size of messages
- unbounded number of sessions $(a, b) \parallel (c, d) \parallel (e, d) \parallel \cdots$
- unbounded number of participants
- unbounded nonce creation

Decidability results for secrecy

	nb. of sessions	nb. of nonce	size of mess.	secrecy
1	bounded	bounded	bounded	decidable
2			unbounded	NP-complete
3	unbounded	bounded	bounded	DEXPTIME
4			unbounded	undecidable
5		unbounded	bounded	undecidable

1 [Schneider'96, Mitchell, Mitchell and Stern'97, Clarke, Jha and Morrero'98]

2 [Rusinowitch and Turuani'01, Boreale'01, Amadio, Lugiez and Vanackère'01]

3 [Chevalier, Kusters, Rusinowitch, Turuani and Vigneron'03]

3-4 [Durgin, Lincoln, Mitchell and Scederov'99]

5 [Amadio, Lugiez and Vanackère'01, Comon, Cortier and Mitchell'01]

Partial Decision Methods

- Resolution-based methods, termination is not guaranteed:
  [Cortier, Mitchell and Ruess'01, Blanchet'03]

- Abstraction-based methods, tree automata for representing the intruder knowledge:
  [Monniaux'99, Goubault-Larecq'00, Genet and Klay'00]

Contributions

Bounded number of sessions: NP-complete decision procedure

- secrecy, authentication (aliveness, weak agreement, agreement) and other prop.
- time sensitive cryptographic protocols
- unbounded initial intruder knowledge
- unbounded size of messages, but atomic keys
- Bozga, Ene and Lakhnech FOSSACS'04, CONCUR'04 and JLAP (to appear)

Unbounded number of sessions: Partial decision method

- combining the approach for bounded with abstract interpretation techniques
- unbounded initial intruder knowledge
- secrecy properties
- unbounded size of messages, but atomic keys
- Bozga, Lakhnech and Perin TACAS'03, CAV'03 and STTT (to appear)

Plan

Introduction

Model

- Terms
- Intruder Model
- Protocol Model

Bounded Protocol Verification

Unbounded Protocol Verification

Conclusions and Perspectives

Terms and Messages

Terms:

$$t ::= x \mid N \mid P \mid K \mid (t_1, t_2) \mid \{t\}_K$$

- x - *message variable*
- N - *nonce*
- P - *participants*
- K - *key*

Messages are ground terms.

The Intruder Model - Dolev Yao

Derivability of a message m from a set E:

$$\frac{m \in E}{E \vdash m}\ axiom \qquad \frac{E \vdash (m_1, m_2)}{E \vdash m_1}\ pr_l$$

$$\frac{E \vdash m_1,\ E \vdash m_2}{E \vdash (m_1, m_2)}\ pair \qquad \frac{E \vdash (m_1, m_2)}{E \vdash m_2}\ pr_r$$

$$\frac{E \vdash m,\ E \vdash k \in \mathcal{K}}{E \vdash \{m\}_k}\ encr \qquad \frac{E \vdash \{m\}_k,\ E \vdash k^{-1}}{E \vdash m}\ decr$$

Protocol Description

Labeled Input / Output actions

$$\left.\begin{array}{ll} a_0: & !\{A, N_a\}_{PK(B)} \\ & \\ & \\ b_0: & ?\{y, z\}_{PK(B)} \end{array}\right\} \quad A \rightarrow B: \ \{A, N_a\}_{PK(B)}$$

$$\left.\begin{array}{ll} & \\ b_1: & !\{z, N_b\}_{PK(y)} \\ & \\ a_1: & ?\{N_a, x\}_{PK(A)} \end{array}\right\} \quad B \rightarrow A: \{N_a, N_b\}_{PK(A)}$$

$$\left.\begin{array}{ll} a_2: & !\{x\}_{PK(B)} \\ & \\ & \\ b_2: & ?\{N_b\}_{PK(B)} \end{array}\right\} \quad A \rightarrow B: \qquad \{N_b\}_{PK(B)}$$

Protocol Description

Labeled Input / Output actions

$$\left.\begin{array}{ll} a_0: & !\{A, N_a\}_{PK(B)} \\ b_0: & ?\{y, z\}_{PK(B)} \end{array}\right\} \quad A \to B: \ \{A, N_a\}_{PK(B)}$$

$$\left.\begin{array}{ll} b_1: & !\{z, N_b\}_{PK(y)} \\ a_1: & ?\{N_a, x\}_{PK(A)} \end{array}\right\} \quad B \to A: \{N_a, N_b\}_{PK(A)}$$

$$\left.\begin{array}{ll} a_2: & !\{x\}_{PK(B)} \\ b_2: & ?\{N_b\}_{PK(B)} \end{array}\right\} \quad A \to B: \quad \{N_b\}_{PK(B)}$$

Bounded protocols: interleaving of actions $\sum_{i=1}^{n} \alpha_1^i \cdots \alpha_{n_i}^i$

Plan

Introduction

Model

Bounded Protocol Verification

- Security Properties Logic TTL
- Weakest precondition calculus
- Decision procedure for the satisfiability problem
- A timed extension Timed TTL

Unbounded Protocol Verification

Conclusions and Perspectives

A Logic for Security Properties

$$P, Q ::= \quad Secret(t) \mid pc = \ell \mid x = t \mid \top \mid P \wedge Q \mid \forall x P \mid \neg P$$

- t - a term
- x - a variable
- pc, ℓ - control points

It allows us to express security properties as

- secrecy
- authentication

The $Secret$ Predicate and WP calculus

- $Secret(s)$ means $E \not\vdash s$ which is not suitable for induction as

$$E \not\vdash s \wedge m \not\vdash s \text{ does not imply } E, m \not\vdash s.$$

$$E = \{N_a\} \;\; m = N_b \;\; s = (N_a, N_b)$$

The $Secret$ Predicate and WP calculus

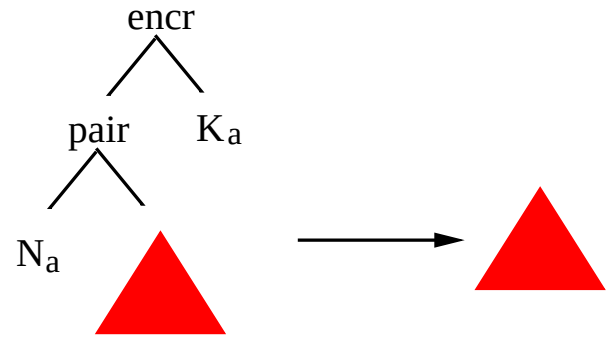
- $Secret(s)$ means $E \not\vdash s$ which is not suitable for induction as

$$E \not\vdash s \wedge m \not\vdash s \text{ does not imply } E, m \not\vdash s.$$

$$E = \{N_a\} \quad m = N_b \quad s = (N_a, N_b)$$

- protocol execution involves some oracle rules

$$a_1 : \quad ?\{N_a, x\}_{K_a}$$
$$a_2 : \quad !x$$



Term Transducer

$$w \quad ::= \quad \lambda x \cdot \text{if } x = \{t\}_k \text{ then } x_{|p} \mid pr_l \mid pr_r \mid \sum_{k \notin K} decr(\cdot, k^{-1})$$
$$\mid w_1 \cdot w_2 \mid w_1 + w_2 \mid w^*$$

New predicate: $t \lhd w \rhd_K s$

- t, s are terms
- w is a term transducer
- and K is a set of keys

no instance of s is obtained from an instance of t by applying w

The Term Transducer Logic TTL

$$P, Q ::= \quad E\lhd w \rhd_K S \mid x\lhd w \rhd_K S \mid pc = \ell \mid x = t \mid \top \mid P \wedge Q \mid \forall x P \mid \neg P$$

$TTL_\forall$ - universal fragment of TTL

- security properties are expressible (secrecy and authentication)
- weakest precondition calculus is closed

$TTL_\exists$ - existential fragment of TTL

- initial conditions are expressible (i.e. the intruder initial knowledge)
- decidability of the satisfiability problem

$\{P_\exists\}\Pi\{Q_\forall\}$ is true iff $\underbrace{\neg(P_\exists \Rightarrow wp(\Pi, Q_\forall))}_{\in\ TTL_\exists}$ is not satisfiable

Weakest Precondition calculus

- Bounded cryptographic protocol: $\Pi = \sum_{i=1}^{n} \alpha_1^i \cdots \alpha_{n_i}^i$

$$wp(\Pi, Q) = \bigvee_{i=1}^{n} wp(\alpha_1^i, wp(\alpha_2^i, \cdots wp(\alpha_{n_i}^i, Q) \cdots))$$

- Distributivity of the weakest precondition operator:

$$wp(\Pi, P \wedge Q) = wp(\Pi, P) \wedge wp(\Pi, Q)$$

$$wp(\Pi, \forall X \cdot P) = \forall X \cdot wp(\Pi, P)$$

$wp(\Pi, \neg P) = \neg wp(\Pi, P)$ - for deterministic programs

Defined inductively on the structure of the postcondition

- Defined by axioms for atomic formulae and I/O actions

Decidability

The existential fragment of TTL is decidable

- define solved form
- define rewriting rules to transform any formula into a set of solved form formulae
- prove soundness and completeness of these rules
- prove termination in solved form

NP-complete - polynomial reduction of 3-SAT problem and
- the solution is polynomially bounded by the formula size

Full TTL logic is undecidable, inspired from Venkataraman 87

Time Sensitive Protocols

Cryptographic protocols use time values to

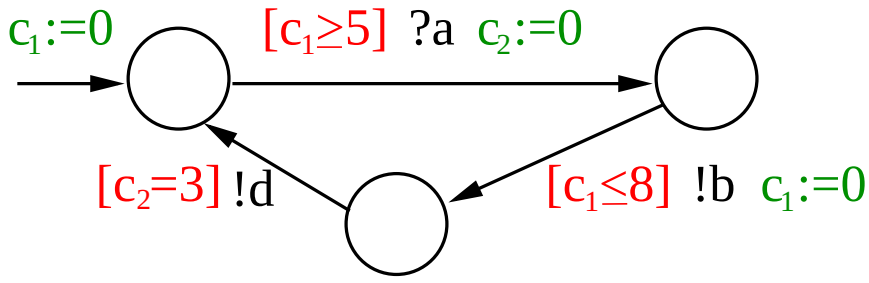
- avoid the reuse of old messages
- stamp the short term keys or the public key certifications

Nonces are not a good approximation for time values

+ time values are ordered, nonces are not
- time values may be guessed, nonces may not

Model of Timed Bounded Protocols

Inspired by timed automata [Alur and Dill'94]



Timed automata are automata extended with

- clocks - their values increases as time elapses
- timed actions - input/output actions with guards and resets

Semantics:

- Discrete transitions
- Time passing transitions

Model of Timed Bounded Protocols

But timed protocols are more general:

$A \rightarrow B : \{A, Na\}_{Kb}$

$B \rightarrow A : \{Na, Tb, Kab\}_{Ka}$

$a_0 : \ !\{A, Na\}_{Kb}\{c := 0\}$

$b_0 : \ ?\{A, x\}_{Kb}$

$b_1 : \ !\{x, c_{now}, Kab\}_{Ka}$

$a_1 : [c < \delta_1 \wedge c_{now} - t < \delta_2] \ ?\{Na, t, y\}_{Ka}$

δ_1 - timeout

δ_2 - key validity

In our model:

- messages carry time information
- we consider timestamps and time variables
- guards are linear constraints over clocks and time variables

Verification of Timed Bounded Protocols

We extend the untimed method:

- define Timed TTL as TTL with time constraints
- extend the wp-calculus to timed actions and Timed TTL
- extend the decidability results to existential fragment of Timed TTL

$\Rightarrow$ we have an effective decision procedure for time sensitive bounded protocols and the Timed TTL

Plan

Introduction

Model

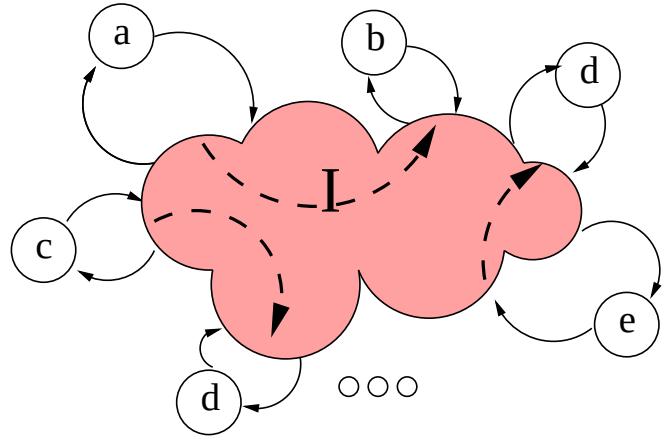
Bounded Protocol Verification

Unbounded Protocol Verification

- Abstraction
- Partial decision method
- Enforcing termination
- Hermes

Conclusions and Perspectives

Unbounded Number of Sessions

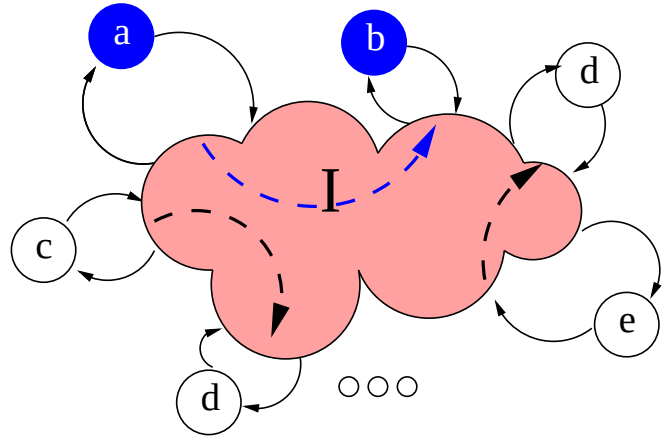


- the verification is quantified universally

  For all session instances with honest participants the property holds

- infinitely many participants
- infinitely many keys
- infinitely many nonces
- unbounded size of messages

Data Abstraction

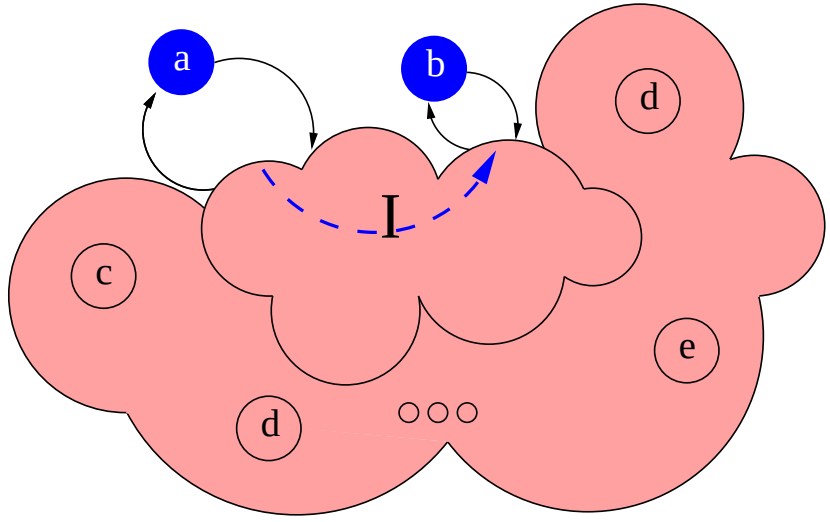


- the verification is quantified universally → fix an arbitrary session

  For all session instances with honest participants the property holds

- infinitely many participants
- infinitely many keys
- infinitely many nonces
- unbounded size of messages

Data Abstraction

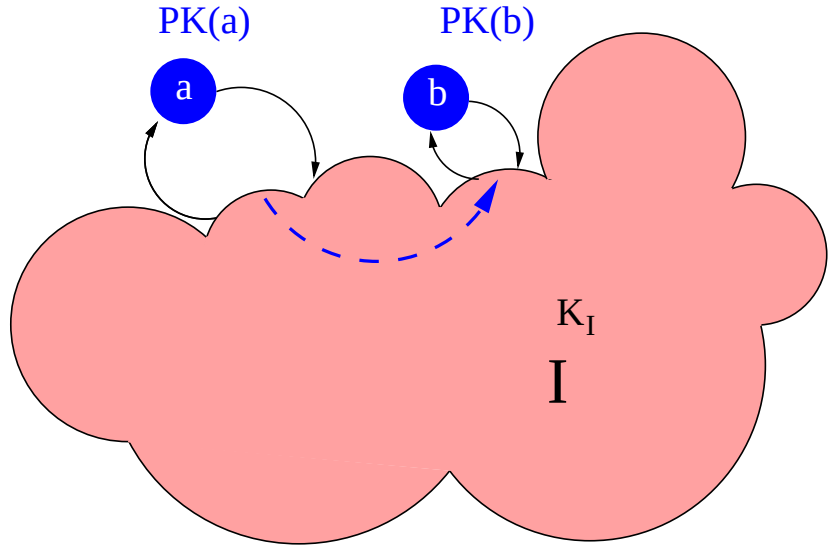


- the verification is quantified universally → fix an arbitrary session

  For all session instances with honest participants the property holds

- infinitely many participants → fixed session participants & the intruder
- infinitely many keys
- infinitely many nonces
- unbounded size of messages

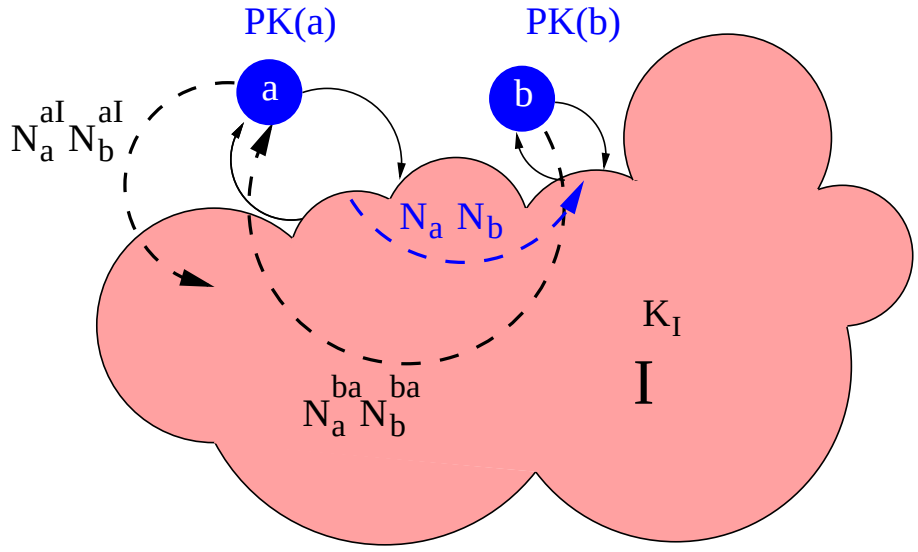
Data Abstraction



- the verification is quantified universally → fix an arbitrary session

  For all session instances with honest participants the property holds
- infinitely many participants → fixed session participants & the intruder
- infinitely many keys → only keys for abstract participants
- infinitely many nonces
- unbounded size of messages

Data Abstraction



- the verification is quantified universally → fix an arbitrary session

  For all session instances with honest participants the property holds
- infinitely many participants → fixed session participants & the intruder
- infinitely many keys → only keys for abstract participants
- infinitely many nonces → one element per type of session
- unbounded size of messages

Control Abstraction

- Unbounded number of sessions

  $\longrightarrow$ unbounded number sessions between the abstract participants

  1. remove the control points from the actions
  2. input actions becomes guards for output actions

$$
\begin{array}{lll}
a_0: & !\{A, N_a\}_{PK(B)} & - \longrightarrow \{A, N_a\}_{PK(B)} \\
a_1: & ?\{N_a, x\}_{PK(A)} & \{N_a, x\}_{PK(A)} \longrightarrow \{x\}_{PK(B)} \\
a_2: & !\{x\}_{PK(B)} &
\end{array}
$$

$$
\begin{array}{lll}
b_0: & ?\{A, y\}_{PK(B)} & \\
b_1: & !\{y, N_b\}_{PK(A)} & \{A, y\}_{PK(B)} \longrightarrow \{y, N_b\}_{PK(A)} \\
b_2: & ?\{N_b\}_{PK(B)} &
\end{array}
$$

The Verification Problem

Consider abstract protocols $\Pi^\sharp$ defined by a set T of abstract transitions of the form $t_p \longrightarrow t_c$.

Given E_0 and S_0 two sets of messages we want to verify:

$$\text{If } E_0 \rightarrow_T^* E \text{ then } E \not\vdash S_0$$

Solution: Compute transducer W and secrets S such that

$$E{\lhd}W{\rhd}_K S = wp^*(\Pi^\sharp, E{\lhd}\varepsilon{\rhd}_K S_0)$$

and check if

$$E_0{\lhd}W{\rhd}_K S \text{ is valid}$$

The Verification Problem

Consider abstract protocols $\Pi^\sharp$ defined by a set T of abstract transitions of the form $t_p \rightarrow t_c$.

Given E_0 and S_0 two sets of messages we want to verify:

$$\text{If } E_0 \rightarrow_T^* E \text{ then } E \not\vdash S_0$$

Solution: Compute transducer W and secrets S such that

$$E{\prec}W{\succ}_K S = wp^*(\Pi^\sharp, E{\prec}\varepsilon{\succ}_K S_0)$$

and check if

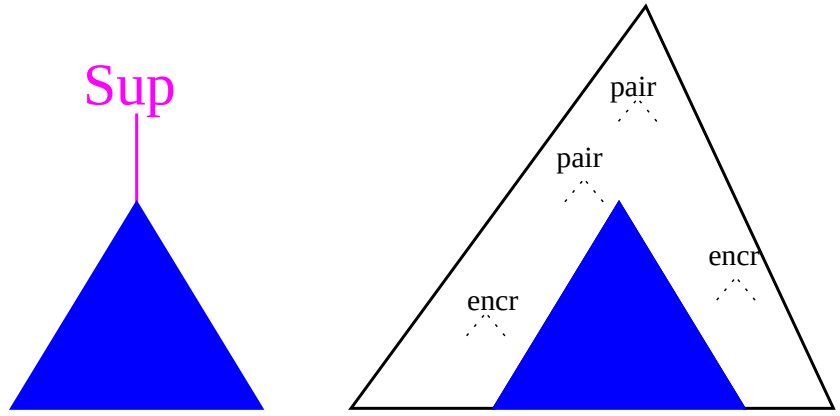
$$E_0{\prec}W{\succ}_K S \text{ is valid}$$

Termination?

Enforcing Termination

1. Use **patterns** and **pattern transducers** as symbolic representation

   Patterns are terms with the operator Sup.
   $Sup(t)$ - represents all terms containing t as subterm

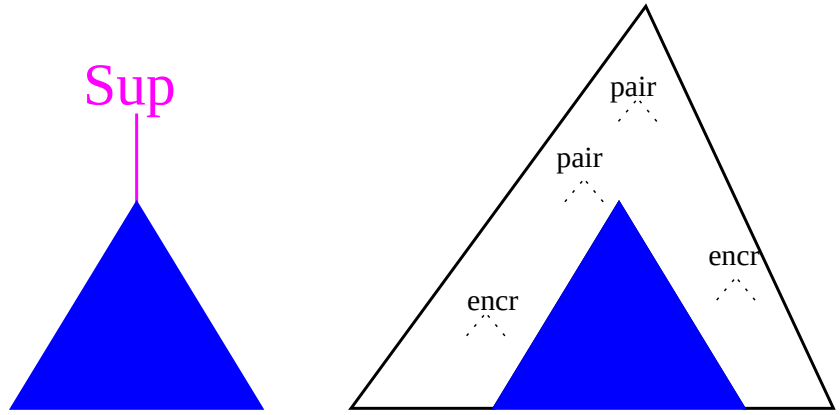


Pattern transducers are transducers defined over patterns

Enforcing Termination

1. Use **patterns** and **pattern transducers** as symbolic representation

   Patterns are terms with the operator Sup.
   $Sup(t)$ - represents all terms containing t as subterm

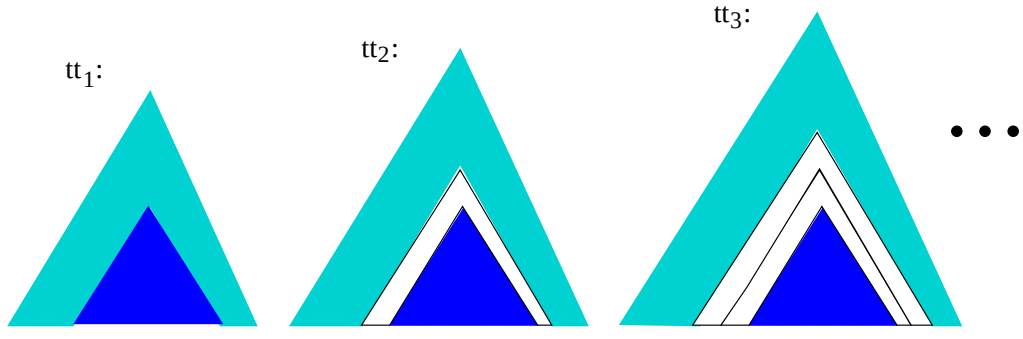


   Pattern transducers are transducers defined over patterns

2. Define a **widening technique** using pattern transducers

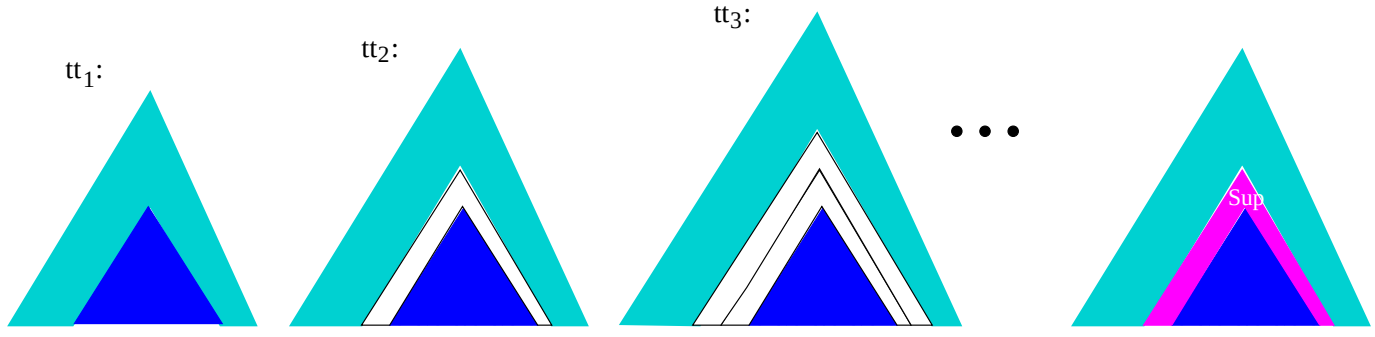
Widening

- detect increasing sequence (tt_i) of pattern transducers where



Widening

- detect increasing sequence (tt_i) of pattern transducers where



- approximate the whole sequence by a pattern transducer

Example - Widening

Consider: $K = \{K_a\}$

$$\Pi^\sharp = \left\{ \begin{array}{l} \{(I,x)\}_{K_a} \rightarrow x \\ \{(A,(N_a,y))\}_{K_a} \rightarrow \{y\}_{K_a} \end{array} \right\} \text{ and } S_0 = \{K_a\}$$

Compute without widening:

$$\begin{array}{ll} W_1 = & (\{(I,x)\}_{K_a})^* \\ W_2 = (W_1 + & \{(A,(N_a,(I,x)))\}_{K_a})^* \\ W_3 = (W_2 + & \{(A,(N_a,(A,(N_a,(I,x)))))\}_{K_a})^* \\ \vdots & \end{array}$$

Example - Widening

Consider: $K = \{K_a\}$

$$\Pi^\sharp = \left\{ \begin{array}{l} \{(I,x)\}_{K_a} \rightarrow x \\ \{(A,(N_a,y))\}_{K_a} \rightarrow \{y\}_{K_a} \end{array} \right\} \text{ and } S_0 = \{K_a\}$$

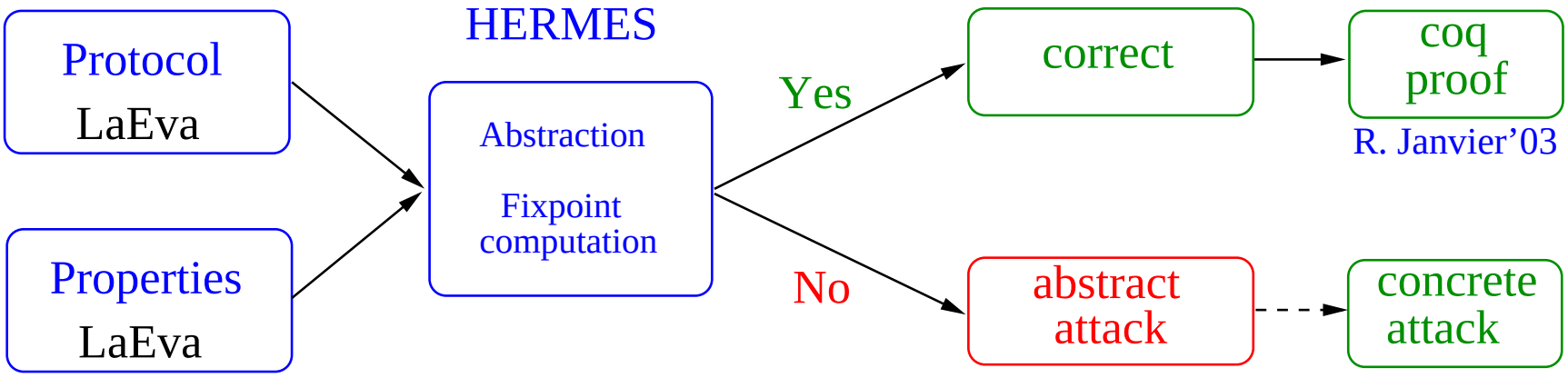
Compute without widening:

$$\begin{array}{ll} W_1 = & (\{(I,x)\}_{K_a})^* \\ W_2 = (W_1 + & \{(A,(N_a,(I,x)))\}_{K_a})^* \\ W_3 = (W_2 + & \{(A,(N_a,(A,(N_a,(I,x)))))\}_{K_a})^* \\ \vdots & \end{array}$$

Compute with widening:

$$\begin{array}{ll} W_3 = (W_2 + & \{(A,(N_a,Sup((I,x))))\}_{K_a})^* \\ W_4 = W_3 & \end{array}$$

Presentation of HERMES



Available online at: http://www-verimag.imag.fr/∼Liana.Bozga/eva/hermes.php

HERMES Results

Protocol Name	Time (sec)	Result
Needham Schroeder Public Key	0.04	Attack
Needham Schroeder Lowe	0.02	OK
Yahalom	29.42	OK
Otway Rees	0.04	OK
Denning Sacco Key Distribution with Public Key	0.03	Attack
Wide Mouthed Frog (modified)	0.03	OK
Kao Chow	1.08	OK
Neumann Stubblebine	0.10	OK
Needham Schroeder Symmetric Key	0.08	Attack
TMN	0.01	Attack
Andrew Secure RPC	0.01	OK
Woo and Lam Mutual Authentication (modified)	0.10	OK
Skeme (modified)	0.03	OK

Conclusions

Verification approach which:

- is complete and effective for
  - bounded time sensitive cryptographic protocols
  - a powerful logic to express properties
  - unbounded initial intruder knowledge
  - unbounded size of messages, but atomic keys
- allows proof correctness for
  - unbounded cryptographic protocols
  - secrecy properties
  - unbounded number of participants, nonces and keys
  - unbounded initial intruder knowledge
  - unbounded size of messages, but atomic keys

HERMES tool which implement this approach.

Perspectives

Weaker intruder and protocol model:

- secure channels
- iterative sessions for same participants

Other security properties:

- Anonymity: nobody may obtain who talks
- Non-repudiation: message exchange can be proved by sender and receiver
- Fairness: no one of participants may obtain advantage

To study:

- composed key
- authentication for unbounded protocols