Verification of Security Protocols

Véronique Cortier[1]

January 18th, 2009

VMCAI'09

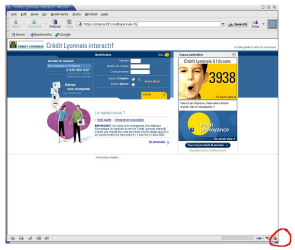
[1]LORIA, CNRS - INRIA Cassis project, Nancy Universités

Context : cryptographic protocols

Cryptographic protocols are widely used in everyday life.

Credit card payment

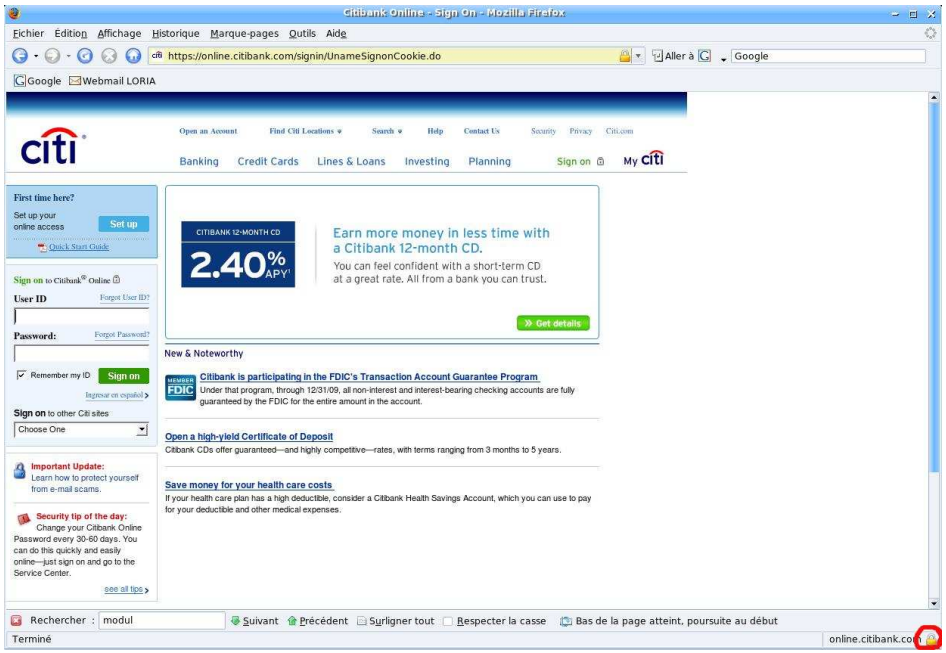
On the web (SSH, SSL, ...)



Pay-per-view

Electronic purse

On the web



- HTTPS, i.e. the SSL protocol for ensuring confidentiality

- password-based authentication

Credit Card payment

- It is a real card ?
- Is the pin code protected ?

Example : Credit Card Payment Protocol

- The waiter introduces the credit card.
- The waiter enters the amount m of the transaction on the terminal.
- The terminal authenticates the card.
- The customer enters his secret code.
  If the amount m is greater than 100 euros
  (and in only 20% of the cases)
  - The terminal asks the bank for authentication of the card.
  - The bank provides authentication.

More details

4 actors : Bank, Customer, Card and Terminal.

Bank owns

- a signing key K_B^{-1}, secret,
- a verification key K_B, public,
- a secret symmetric key for each credit card K_{CB}, secret.

Card owns

- Data : last name, first name, card's number, expiration date,
- Signature's Value $VS = \{hash(\text{Data})\}_{K_B^{-1}}$,
- secret key K_{CB}.

Terminal owns the verification key K_B for bank's signatures.

Credit card payment Protocol (in short)

The terminal reads the card :

1. $Ca \rightarrow T : \text{Data}, \{hash(\text{Data})\}_{K_B^{-1}}$

Credit card payment Protocol (in short)

The terminal reads the card :

$$1. \quad Ca \rightarrow T : \mathrm{Data}, \{hash(\mathrm{Data})\}_{K_B^{-1}}$$

The terminal asks for the secret code :

$$2. \quad T \rightarrow Cu : secret\ code?$$
$$3. \quad Cu \rightarrow Ca : 1234$$
$$4. \quad Ca \rightarrow T : ok$$

Credit card payment Protocol (in short)

The terminal reads the card :

$$1. \quad Ca \rightarrow T : \text{Data}, \{hash(\text{Data})\}_{K_B^{-1}}$$

The terminal asks for the secret code :

$$2. \quad T \rightarrow Cu : secret\ code?$$
$$3. \quad Cu \rightarrow Ca : 1234$$
$$4. \quad Ca \rightarrow T : ok$$

The terminal calls the bank :

$$5. \quad T \rightarrow B : auth?$$
$$6. \quad B \rightarrow T : N_b$$
$$7. \quad T \rightarrow Ca : N_b$$
$$8. \quad Ca \rightarrow T : \{N_b\}_{K_{CB}}$$
$$9. \quad T \rightarrow B : \{N_b\}_{K_{CB}}$$
$$10. \quad B \rightarrow T : ok$$

Some flaws

The security was initially ensured by :

- the cards were very difficult to reproduce,
- the protocol and the keys were secret.

But

- cryptographic flaw : 320 bits keys can be broken (1988),
- logical flaw : no link between the secret code and the authentication of the card,
- fake cards can be build.

Some flaws

The security was initially ensured by :

- the cards were very difficult to reproduce,
- the protocol and the keys were secret.

But

- cryptographic flaw : 320 bits keys can be broken (1988),
- logical flaw : no link between the secret code and the authentication of the card,
- fake cards can be build.

→ "YesCard" build by Serge Humpich
(1998 in France).

How does the "YesCard" work ?

Logical flaw

1. $Ca \rightarrow T$: Data, $\{hash(\text{Data})\}_{K_B^{-1}}$
2. $T \rightarrow Ca$: *secret code?*
3. $Cu \rightarrow Ca$: 1234
4. $Ca \rightarrow T$: *ok*

How does the "YesCard" work ?

Logical flaw

1. $Ca \longrightarrow T$: Data, $\{hash(\text{Data})\}_{K_B^{-1}}$
2. $T \longrightarrow Ca$: *secret code?*
3. $Cu \longrightarrow Ca'$: 2345
4. $Ca' \longrightarrow T$: *ok*

How does the "YesCard" work ?

Logical flaw

1. $Ca \rightarrow T$: Data, $\{hash(\text{Data})\}_{K_B^{-1}}$
2. $T \rightarrow Ca$: *secret code?*
3. $Cu \rightarrow Ca'$: 2345
4. $Ca' \rightarrow T$: *ok*

Remark : there is always somebody to debit.
$\rightarrow$ creation of a fake card

How does the "YesCard" work ?

Logical flaw

1. $Ca \rightarrow T$: Data, $\{hash(\text{Data})\}_{K_B^{-1}}$
2. $T \rightarrow Ca$: *secret code?*
3. $Cu \rightarrow Ca'$: 2345
4. $Ca' \rightarrow T$: *ok*

Remark : there is always somebody to debit.
$\rightarrow$ creation of a fake card

1. $Ca' \rightarrow T$: XXX, $\{hash(\text{XXX})\}_{K_B^{-1}}$
2. $T \rightarrow Cu$: *secret code?*
3. $Cu \rightarrow Ca'$: 0000
4. $Ca' \rightarrow T$: *ok*

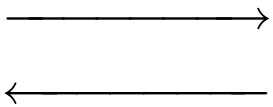
Electronic signature

- authenticates the signer

- should be verifiable by anyone

- ensures non-repudiation
(I never signed that message !)

Pay-per-view devices



- Checks your identity
- You should be granted access to the movie only once
- You should not be able to broadcast the movie to other people

Electronic voting

- The result corresponds to the votes.
- Each vote is confidential.
- No partial result is leaked before the end of the election
- Only voters can vote and at most once
- Coercion resistance

Electronic purse

- It should not possible to add money without paying.
- It should not be possible to create fake electronic purse.

Secure databases

Patients	Social security number	Diseases	Treatments
Smith	123 - 45 - 6789	Flu	...
Smith	123 - 45 - 6789	Arthritis	...
Williams	789 - 56 - 1234	Dehydration	...
Johnson	012 - 34 - 5678	Arthritis	...

- Authorized persons have access only to a partial view of the database (different for doctors, nurses, researchers, ...)
- Data may be exchanged between e.g. doctors, hospitals, medical laboratories.
- Data are regularly updated.

Security goals

Cryptographic protocols aim at

- preserving confidentiality of data
  (e.g. pin code, medical files, ...)
- ensuring authenticity
  (Are you really talking to your bank ? ?)
- ensuring anonymous communications
  (for e-voting protocols, ...)
- protecting against repudiation
  (I never sent this message ! !)
- ...

$\Rightarrow$ Cryptographic protocols vary depending on the application.

Difficulty

Presence of an attacker

- may read every message sent on the net,
- may intercept and send new messages.

$\Rightarrow$ The system is infinitely branching

Outline of the talk

Motivation : Cryptography does not suffice to ensure security !

Example : Commutative encryption (RSA)

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

Motivation : Cryptography does not suffice to ensure security !

Example : Commutative encryption (RSA)

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\leftarrow \left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}}$$

Motivation : Cryptography does not suffice to ensure security !

Example : Commutative encryption (RSA)

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}} \longleftarrow$$

$$\{\text{pin} : 3443\}_{k_{\text{bob}}} \longrightarrow$$

$$\text{Since } \left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}} = \left\{\{\text{pin} : 3443\}_{k_{\text{bob}}}\right\}_{k_{\text{alice}}}$$

Motivation : Cryptography does not suffice to ensure security !

Example : Commutative encryption (RSA)

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}} \longleftarrow$$

$$\{\text{pin} : 3443\}_{k_{\text{bob}}} \longrightarrow$$

$\longrightarrow$ It does not work ! (Authentication problem)

Motivation : Cryptography does not suffice to ensure security !

Example : Commutative encryption (RSA)

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\leftarrow \left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{bob}}}$$

$$\{\text{pin} : 3443\}_{k_{\text{bob}}} \longrightarrow$$

→ It does not work ! (Authentication problem)

$$\{\text{pin} : 3443\}_{k_{\text{alice}}} \longrightarrow$$

$$\leftarrow \left\{\{\text{pin} : 3443\}_{k_{\text{alice}}}\right\}_{k_{\text{intruder}}}$$

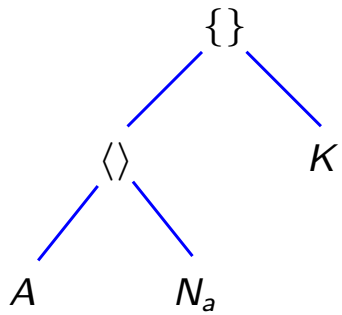
$$\{\text{pin} : 3443\}_{k_{\text{intruder}}} \longrightarrow$$

Messages

Messages are abstracted by terms.

Agents : $a, b, \ldots$ Nonces : $n_1, n_2, \ldots$ Keys : $k_1, k_2, \ldots$
Cyphertext : $\{m\}_k$ Concatenation : $\langle m_1, m_2 \rangle$

Example : The message $\{A, N_a\}_K$ is represented by :



Intruder abilities

Composition rules

$$\frac{T \vdash u \quad T \vdash v}{T \vdash \langle u\,, v\rangle} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathsf{enc}(u, v)} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathsf{enca}(u, v)}$$

Intruder abilities

Composition rules

$$\frac{T \vdash u \quad T \vdash v}{T \vdash \langle u\,, v\rangle} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathrm{enc}(u, v)} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathrm{enca}(u, v)}$$

Decomposition rules

$$\frac{}{T \vdash u}\, u \in T \qquad \frac{T \vdash \langle u\,, v\rangle}{T \vdash u} \qquad \frac{T \vdash \langle u\,, v\rangle}{T \vdash v}$$

$$\frac{T \vdash \mathrm{enc}(u, v) \quad T \vdash v}{T \vdash u} \qquad \frac{T \vdash \mathrm{enca}(u, \mathrm{pub}(v)) \quad T \vdash \mathrm{priv}(v)}{T \vdash u}$$

Intruder abilities

Composition rules

$$\frac{T \vdash u \quad T \vdash v}{T \vdash \langle u\,, v\rangle} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathrm{enc}(u, v)} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathrm{enca}(u, v)}$$

Decomposition rules

$$\frac{}{T \vdash u}\, u \in T \qquad \frac{T \vdash \langle u\,, v\rangle}{T \vdash u} \qquad \frac{T \vdash \langle u\,, v\rangle}{T \vdash v}$$

$$\frac{T \vdash \mathrm{enc}(u, v) \quad T \vdash v}{T \vdash u} \qquad \frac{T \vdash \mathrm{enca}(u, \mathrm{pub}(v)) \quad T \vdash \mathrm{priv}(v)}{T \vdash u}$$

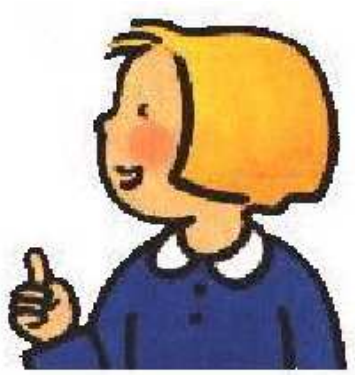
Deducibility relation

A term u is deducible from a set of terms T, denoted by $T \vdash u$, if there exists a prooftree witnessing this fact.

A simple protocol

$\langle Bob, k \rangle$ →

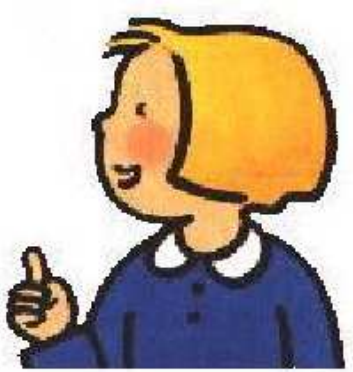
← $\langle Alice, enc(s, k) \rangle$



A simple protocol

$\langle Bob, k \rangle$ →

← $\langle Alice, enc(s, k) \rangle$



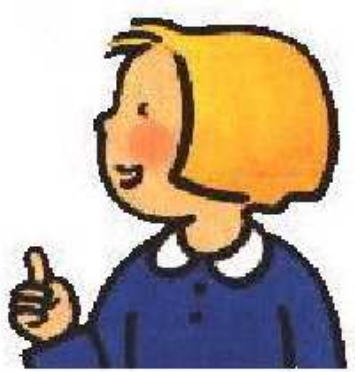
Question ?

Can the attacker learn the secret s ?

A simple protocol

$\langle Bob, k \rangle$ →

← $\langle Alice, enc(s, k) \rangle$



Answer : Of course, Yes !

$$\frac{\dfrac{\langle Alice, enc(s, k) \rangle}{enc(s, k)} \qquad \dfrac{\langle Bob, k \rangle}{k}}{s}$$

Decision of the intruder problem

Given A set of messages S and a message m

Question Can the intruder learn m from S that is $S \vdash m$?

This problem is decidable in polynomial time

Decision of the intruder problem

Given A set of messages S and a message m

Question Can the intruder learn m from S that is $S \vdash m$?

This problem is decidable in polynomial time

Lemma (Locality)

If there is a proof of $S \vdash m$ then there is a proof that only uses the subterms of S and m.

Protocol description

Protocol :

$$\begin{array}{lcl} A \longrightarrow B & : & \{\mathsf{pin}\}_{k_a} \\ B \longrightarrow A & : & \{\{\mathsf{pin}\}_{k_a}\}_{k_b} \\ A \longrightarrow B & : & \{\mathsf{pin}\}_{k_b} \end{array}$$

A protocol is a finite set of roles :

- role $\Pi(1)$ corresponding to the 1^{st} participant played by a talking to b :

$$\begin{array}{rcl} \mathsf{init} & \xrightarrow{k_a} & \mathsf{enc}(\mathsf{pin}, k_a) \\ \mathsf{enc}(x, k_a) & \longrightarrow & x. \end{array}$$

Protocol description

Protocol :

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

A protocol is a finite set of roles :

- role $\Pi(1)$ corresponding to the 1^{st} participant played by a talking to b :

$$\begin{array}{rcl} \mathsf{init} & \xrightarrow{k_a} & \mathsf{enc}(\mathsf{pin}, k_a) \\ \mathsf{enc}(x, k_a) & \to & x. \end{array}$$

- role $\Pi(2)$ corresponding to the 2^{nd} participant played by b with a :

$$\begin{array}{rcl} x & \xrightarrow{k_b} & \mathsf{enc}(x, k_b) \\ \mathsf{enc}(y, k_b) & \to & \mathsf{stop}. \end{array}$$

Secrecy via constraint solving [Millen et al]

Constraint systems are used to specify secrecy preservation under a particular, finite scenario.

Scenario

$$\begin{array}{l} \mathrm{rcv}(u_1) \xrightarrow{N_1} \mathrm{snd}(v_1) \\ \mathrm{rcv}(u_2) \xrightarrow{N_2} \mathrm{snd}(v_2) \\ \quad \ldots \\ \mathrm{rcv}(u_n) \xrightarrow{N_n} \mathrm{snd}(v_n) \end{array}$$

Constraint System

$$\mathcal{C} = \begin{cases} T_0 \Vdash u_1 \\ T_0, v_1 \Vdash u_2 \\ \quad \ldots \\ T_0, v_1, .., v_n \Vdash s \end{cases}$$

where T_0 is the initial knowledge of the attacker.

Remark : Constraint Systems may be used more generally for trace-based properties, e.g. authentication.

Secrecy via constraint solving [Millen et al]

Constraint systems are used to specify secrecy preservation under a particular, finite scenario.

Scenario

$$\begin{array}{l} \mathrm{rcv}(u_1) \xrightarrow{N_1} \mathrm{snd}(v_1) \\ \mathrm{rcv}(u_2) \xrightarrow{N_2} \mathrm{snd}(v_2) \\ \quad \ldots \\ \mathrm{rcv}(u_n) \xrightarrow{N_n} \mathrm{snd}(v_n) \end{array}$$

Constraint System

$$\mathcal{C} = \begin{cases} T_0 \Vdash u_1 \\ T_0, v_1 \Vdash u_2 \\ \quad \ldots \\ T_0, v_1, .., v_n \Vdash s \end{cases}$$

where T_0 is the initial knowledge of the attacker.

Solution of a constraint system

A substitution σ such that

for every $T \Vdash u \in \mathcal{C}$, $u\sigma$ is deducible from $T\sigma$, that is $u\sigma \vdash T\sigma$.

Example of a system constraint

$$\begin{array}{lcl} A \to B & : & \{\mathsf{pin}\}_{k_a} \\ B \to A & : & \{\{\mathsf{pin}\}_{k_a}\}_{k_b} \\ A \to B & : & \{\mathsf{pin}\}_{k_b} \end{array}$$ and the attacker initially knows $T_0 = \{\mathsf{init}\}$.

One possible associated constraint system is :

$$\mathcal{C} = \left\{ \begin{array}{l} \{\mathsf{init}\} \Vdash \mathsf{init} \\ \{\mathsf{init}, \{\mathsf{pin}\}_{k_a}\} \Vdash \{\mathsf{x}\}_{k_a} \\ \{\mathsf{init}, \{\mathsf{pin}\}_{k_a}, x\} \Vdash \mathsf{pin} \end{array} \right.$$

Is there a solution ?

Example of a system constraint

$$\begin{array}{lcl} A \to B & : & \{\mathsf{pin}\}_{k_a} \\ B \to A & : & \{\{\mathsf{pin}\}_{k_a}\}_{k_b} \\ A \to B & : & \{\mathsf{pin}\}_{k_b} \end{array}$$ and the attacker initially knows $T_0 = \{\mathsf{init}\}$.

One possible associated constraint system is :

$$\mathcal{C} = \left\{ \begin{array}{l} \{\mathsf{init}\} \Vdash \mathsf{init} \\ \{\mathsf{init}, \{\mathsf{pin}\}_{k_a}\} \Vdash \{\mathsf{x}\}_{k_a} \\ \{\mathsf{init}, \{\mathsf{pin}\}_{k_a}, x\} \Vdash \mathsf{pin} \end{array} \right.$$

Is there a solution ?

Of course yes, simply consider $x = \mathsf{pin}$!

How to solve constraint system ?

$$\text{Given } \mathcal{C} = \begin{cases} T_0 \Vdash u_1 \\ T_0, v_1 \Vdash u_2 \\ \quad \dots \\ T_0, v_1, .., v_n \Vdash u_{n+1} \end{cases}$$

Question Is there a solution σ of $\mathcal{C}$?

An easy case : "solved constraint systems"

$$\text{Given } \mathcal{C} = \begin{cases} T_0 \Vdash x_1 \\ T_0, v_1 \Vdash x_2 \\ \quad \dots \\ T_0, v_1, .., v_n \Vdash x_{n+1} \end{cases}$$

Question Is there a solution σ of $\mathcal{C}$?

An easy case : "solved constraint systems"

$$\text{Given } \mathcal{C} = \begin{cases} T_0 \Vdash x_1 \\ T_0, v_1 \Vdash x_2 \\ \quad \dots \\ T_0, v_1, .., v_n \Vdash x_{n+1} \end{cases}$$

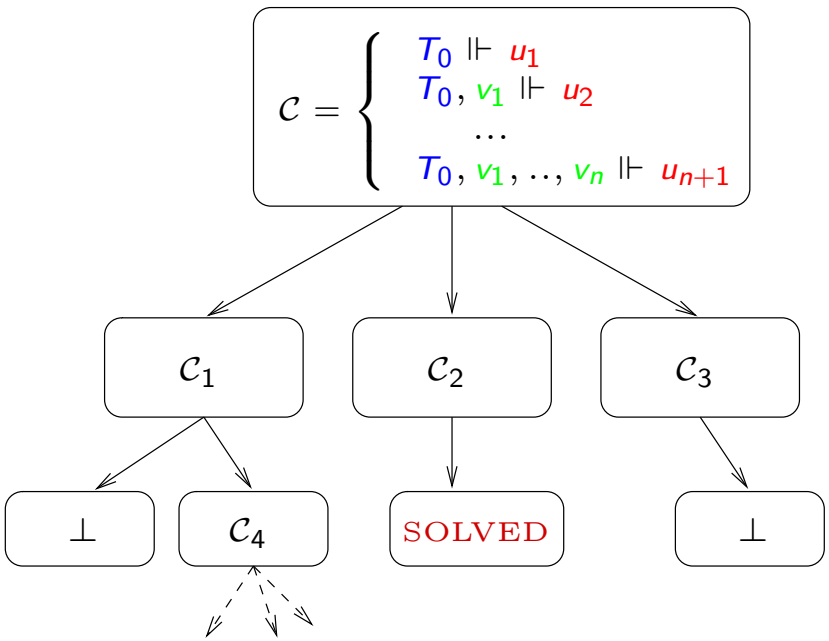
Question Is there a solution σ of $\mathcal{C}$?

Of course yes !

Consider e.g. $\sigma(x_1) = \cdots = \sigma(x_{n+1}) = t \in T_0$.

Decision procedure [Millen / Comon-Lundh]

Goal : Transformation of the constraints in order to obtain a solved constraint system.



$\mathcal{C}$ has a solution iff $\mathcal{C} \rightsquigarrow \mathcal{C}'$ with $\mathcal{C}'$ in solved form.

Intruder step

The intruder can built messages

$$R_5 : \quad \mathcal{C} \wedge T \Vdash f(u,v) \quad \rightsquigarrow \quad \mathcal{C} \wedge T \Vdash u \wedge T \Vdash v \qquad \text{for } f \in \{\langle\rangle, \mathrm{enc}, \mathrm{enca}\}$$

Intruder step

The intruder can built messages

$$R_5 : \quad \mathcal{C} \wedge T \Vdash f(u,v) \quad \rightsquigarrow \quad \mathcal{C} \wedge T \Vdash u \wedge T \Vdash v \qquad \text{for } f \in \{\langle\rangle, \text{enc}, \text{enca}\}$$

Example :

$$a, k \Vdash \text{enc}(\langle x, y\rangle, k) \quad \rightsquigarrow \quad \begin{array}{l} a, k \Vdash k \\ a, k \Vdash x \\ a, k \Vdash y \end{array}$$

Eliminating redundancies

$$k \Vdash x$$
$$k, \mathrm{enc}(s, x) \Vdash s$$

The constraint $\mathrm{enc}(s, x) \Vdash s$ will be satisfied as soon as $k \Vdash x$ is satisfied.

Eliminating redundancies

$$k \Vdash x$$
$$k, \mathrm{enc}(s, x) \Vdash s$$

The constraint $\mathrm{enc}(s, x) \Vdash s$ will be satisfied as soon as $k \Vdash x$ is satisfied.

$$R_1 : \mathcal{C} \wedge T \Vdash u \rightsquigarrow \mathcal{C} \quad \text{if } T \cup \{x \mid T' \Vdash x \in \mathcal{C}, T' \subsetneq T\} \vdash u$$

Unsolvable constraints

$$R_4 : \mathcal{C} \wedge T \Vdash u \rightsquigarrow \bot \qquad \text{if } \mathrm{var}(T, u) = \emptyset \text{ and } T \not\vdash u$$

Example :

$$\begin{array}{l} \ldots \\ a, \mathrm{enc}(s, k) \Vdash s \quad \rightsquigarrow \quad \bot \\ \ldots \end{array}$$

Guessing equalities

1. Example : $k, \mathrm{enc}(\mathrm{enc}(x, k'), k) \Vdash \mathrm{enc}(a, k')$

$$R_2 : \mathcal{C} \wedge T \Vdash u \rightsquigarrow_\sigma \mathcal{C}\sigma \wedge T\sigma \Vdash u\sigma \qquad u' \in st(T)$$
$$\text{if } \sigma = \mathrm{mgu}(u, u'),\ u, u' \notin \mathcal{X},\ u \neq u'$$

Guessing equalities

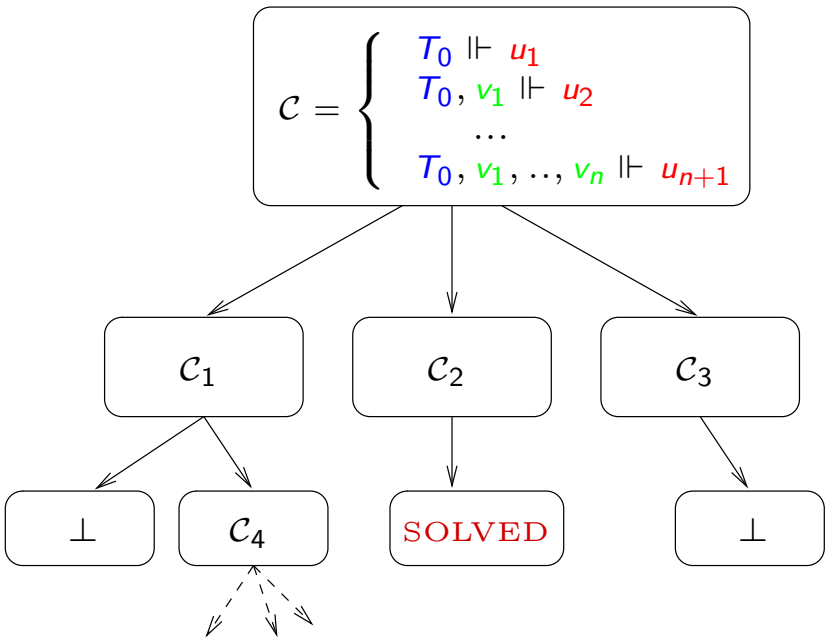
1. Example : $k, \mathrm{enc}(\mathrm{enc}(x, k'), k) \Vdash \mathrm{enc}(a, k')$

$$R_2 : \mathcal{C} \wedge T \Vdash u \rightsquigarrow_\sigma \mathcal{C}\sigma \wedge T\sigma \Vdash u\sigma \qquad u' \in st(T)$$
$$\text{if } \sigma = \mathrm{mgu}(u, u'),\ u, u' \notin \mathcal{X},\ u \neq u'$$

2. Example : $\mathrm{enc}(s, \langle a, x \rangle), \mathrm{enc}(\langle y, b \rangle, k), k \Vdash s$

$$R_3 : \mathcal{C} \wedge T \Vdash v \rightsquigarrow_\sigma \mathcal{C}\sigma \wedge T\sigma \Vdash v\sigma \qquad u, u' \in st(T)$$
$$\text{if } \sigma = \mathrm{mgu}(u, u'),\ u, u' \notin \mathcal{X},\ u \neq u'$$

NP-procedure for solving constraint systems



Theorem

- *$\mathcal{C}$ has a solution iff $\mathcal{C} \rightsquigarrow \mathcal{C}'$ with $\mathcal{C}'$ in solved form.*
- *$\rightsquigarrow$ is terminating in polynomial time.*

What formal methods allow to do ?

- In general, secrecy preservation is undecidable.

What formal methods allow to do ?

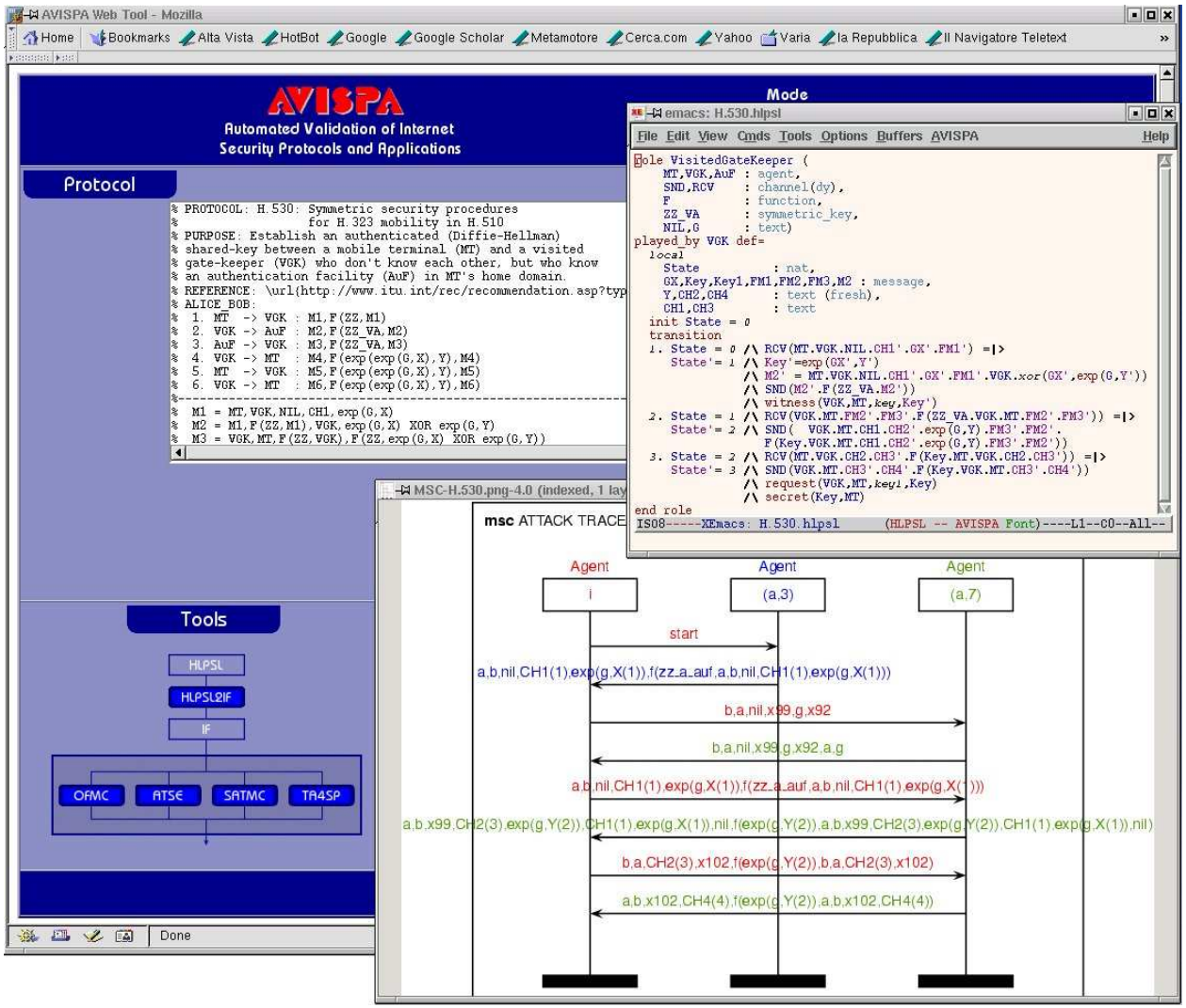
- In general, secrecy preservation is undecidable.
- For a bounded number of sessions, secrecy is co-NP-complete [RusinowitchTuruani CSFW01]
  $\longrightarrow$ several tools for detecting attacks (Casper, Avispa platform...)

What formal methods allow to do ?

- In general, secrecy preservation is undecidable.
- For a bounded number of sessions, secrecy is co-NP-complete [RusinowitchTuruani CSFW01]
  $\longrightarrow$ several tools for detecting attacks (Casper, Avispa platform...)
- For an unbounded number of sessions
  - for one-copy protocols, secrecy is DEXPTIME-complete [CortierComon RTA03] [SeildVerma LPAR04]
  - for message-length bounded protocols, secrecy is DEXPTIME-complete [Durgin et al FMSP99] [Chevalier et al CSL03]

  $\longrightarrow$ some tools for proving security (ProVerif, EVA Platform)

Example of tool : Avispa Platform



Collaborators

- LORIA, France
- DIST, Italy
- ETHZ, Switzerland
- Siemens, Germany

www.avispa-project.org

Outline of the talk

Motivation

Back to our running example :

$$\begin{array}{lcl} A \rightarrow B & : & \{\mathsf{pin}\}_{k_a} \\ B \rightarrow A & : & \{\{\mathsf{pin}\}_{k_a}\}_{k_b} \\ A \rightarrow B & : & \{\mathsf{pin}\}_{k_b} \end{array}$$

We need the equation for the commutativity of encryption

$$\{\{z\}_x\}_y = \{\{z\}_y\}_x$$

Some other examples

Encryption-Decryption theory

$$\mathsf{dec}(\mathsf{enc}(x,y),y) = x \quad \pi_1(\langle x,y\rangle) = x \quad \pi_2(\langle x,y\rangle) = y$$

EXclusive Or

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

Diffie-Hellmann

$$\exp(\exp(z,x),y) = \exp(\exp(z,y),x)$$

E-voting protocols

First phase :

$$V \to A : \quad \mathrm{sign}(\mathit{blind}(\mathrm{vote}, r), V)$$
$$A \to V : \quad \mathrm{sign}(\mathit{blind}(\mathrm{vote}, r), A)$$

Voting phase :

$$V \to C : \quad \mathrm{sign}(\mathrm{vote}, A)$$

...

Equational theory for blind signatures

[Kremer Ryan 05]

$$\begin{aligned} \mathsf{checksign}(\mathsf{sign}(x,y),\mathsf{pk}(y)) &= x \\ \mathsf{unblind}(\mathsf{blind}(x,y),y) &= x \\ \mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(x,y),z),y) &= \mathsf{sign}(x,z) \end{aligned}$$

Deduction

$$\frac{}{T \vdash_E M} M \in T \qquad \frac{T \vdash_E M_1 \quad \cdots \quad T \vdash_E M_k}{T \vdash_E f(M_1, \ldots, M_k)} f \in \Sigma$$

$$\frac{T \vdash M}{T \vdash M'} M =_E M'$$

Deduction

$$\frac{}{T \vdash_E M} M \in T \qquad \frac{T \vdash_E M_1 \quad \cdots \quad T \vdash_E M_k}{T \vdash_E f(M_1, \ldots, M_k)} f \in \Sigma$$

$$\frac{T \vdash M}{T \vdash M'} M =_E M'$$

Example : $E := \mathrm{dec}(\mathrm{enc}(x, y), y) = x$ and $T = \{\mathrm{enc}(\mathit{secret}, k),\ k\}$.

$$\dfrac{\dfrac{\dfrac{}{T \vdash \mathrm{enc}(\mathit{secret}, k)} \qquad \dfrac{}{T \vdash k}}{T \vdash \mathrm{dec}(\mathrm{enc}(\mathit{secret}, k), k)}}{T \vdash \mathit{secret}} \quad f \in \Sigma \quad \mathrm{dec}(\mathrm{enc}(x, y), y) = x$$

Rewriting systems

For analyzing equational theories, we (try to) associate to E a finite convergent rewriting system $\mathcal{R}$ such that :

$$u =_E v \quad \text{iff} \quad u\downarrow = v\downarrow$$

Definition (Characterization of the deduction relation)

Let $t_1, \ldots t_n$ and u be terms in normal form.

$$\{t_1, \ldots t_n\} \vdash u \quad \text{iff} \quad \exists C \text{ s.t. } C[t_1, \ldots, t_n] \rightarrow^* u$$

(Also called Cap Intruder problem [Narendran et al])

Some results with equational theories

	Security problem	
	Bounded number of sessions	**Unbounded number of sessions**
Commutative encryption	*co-NP-complete* *[CKRT04]*	**Ping-pong protocols :** *co-NP-complete* *[Turuani04]*
Exclusive Or	*Decidable* *[CS03,CKRT03]*	**One copy - No nonces :** *Decidable* *[CLC03]* **Two-way automata - No nonces :** *Decidable* *[Verma03]*
Abelian Groups	*Decidable* *[Shmatikov04]*	**Two-way automata - No nonces :** *Decidable* *[Verma03]*
Prefix encryption	*co-NP-complete* *[CKRT03]*	
Abelian Groups and Modular Exponentiation	**General case :** *Decidable* *[Shmatikov04]* **Restricted protocols :** *co-NP-complete* *[CKRT03]*	**AC properties of the Modular Exponentiation No nonces :** *Semi-Decision Procedure* *[GLRV04]*

Outline of the talk

Specificity of cryptographic models

- Messages are bitstrings
- Real encryption algorithm
- Real signature algorithm
- General and powerful adversary

$\longrightarrow$ very little abstract model

Encryption nowadays

$\longrightarrow$ Based on algorithmically hard problems.

RSA Function $n = pq$, p et q primes.
e : public exponent

- $x \mapsto x^e \mod n$ easy (cubic)
- $y = x^e \mapsto x \mod n$ difficult
  $x = y^d$ où $d = e^{-1} \mod \phi(n)$

Encryption nowadays

$\longrightarrow$ Based on algorithmically hard problems.

RSA Function $n = pq$, p et q primes.
e : public exponent

- $x \mapsto x^e \mod n$ easy (cubic)
- $y = x^e \mapsto x \mod n$ difficult
  $x = y^d$ où $d = e^{-1} \mod \phi(n)$

Diffie-Hellman Problem

- Given $A = g^a$ and $B = g^b$,
- Compute $\mathrm{DH}(A, B) = g^{ab}$

Encryption nowadays

$\longrightarrow$ Based on algorithmically hard problems.

RSA Function $n = pq$, p et q primes.
e : public exponent

- $x \mapsto x^e \mod n$ easy (cubic)
- $y = x^e \mapsto x \mod n$ difficult
  $x = y^d$ où $d = e^{-1} \mod \phi(n)$

Diffie-Hellman Problem

- Given $A = g^a$ and $B = g^b$,
- Compute $\mathrm{DH}(A, B) = g^{ab}$

$\longrightarrow$ Based on hardness of integer factorization.

Estimations for integer factorization

Module (bits)	Operations (in $\log_2$)
512	58
1024	80
2048	111
4096	149
8192	156

$\approx 2^{60}$ years

$\rightarrow$ Lower bound for RSA and Diffie-Hellman.

Setting for cryptographic protocols

Protocol :

- Message exchange program
- using cryptographic primitives

Adversary $\mathcal{A}$: any probabilistic polynomial Turing machine, *i.e.* any probabilistic polynomial program.

- polynomial : captures what is feasible
- probabilistic : the adversary may try to guess some information

Definition of secrecy preservation

→ Several notions of secrecy :

One-Wayness : The probability for an adversary $\mathcal{A}$ to compute the secret s against a protocol $\mathcal{P}$ is negligible (smaller than any inverse of polynomial).

$$\forall p \text{ polynomial } \exists \eta_0 \, \forall \eta \geq \eta_0 \quad \Pr^{\eta}_{m,r}[\mathcal{A}(\mathcal{P}_K) = s] \leq \frac{1}{p(\eta)}$$

η : security parameter = key length

Not strong enough !

- The adversary may be able to compute half of the secret message.
- There is no guarantee in case that some partial information on the secret is known.

Not strong enough !

- The adversary may be able to compute half of the secret message.
- There is no guarantee in case that some partial information on the secret is known.

→ Introduction of a notion of indistinguishability.

Indistinguishability

The secrecy of s is defined through the following game :

- Two values n_0 and n_1 are randomly generated instead of s ;
- The adversary interacts with the protocol where s is replaced by n_b, $b \in \{0, 1\}$;
- We give the pair (n_0, n_1) to the adversary ;
- The adversary gives b',

The data s is secret if $\Pr[b = b'] - \frac{1}{2}$ is a negligible function.

A typical cryptographic proof

1. Assume that some algorithmic problem P is difficult (E.g. RSA or integer factorization or Discrete Log or CDH, DDH, ...)
2. Suppose that a (polynomial probabilistic) adversary $\mathcal{A}$ breaks the protocol security with non negligible probability

A typical cryptographic proof

1. Assume that some algorithmic problem P is difficult (E.g. RSA or integer factorization or Discrete Log or CDH, DDH, ...)
2. Suppose that a (polynomial probabilistic) adversary $\mathcal{A}$ breaks the protocol security with non negligible probability
3. Build out of $\mathcal{A}$ an adversary $\mathcal{B}$ that solves P.

A typical cryptographic proof

1. Assume that some algorithmic problem P is difficult (E.g. RSA or integer factorization or Discrete Log or CDH, DDH, ...)
2. Suppose that a (polynomial probabilistic) adversary $\mathcal{A}$ breaks the protocol security with non negligible probability
3. Build out of $\mathcal{A}$ an adversary $\mathcal{B}$ that solves P.
4. Conclude that the protocol is secure provided P is difficult.

Formal and Cryptographic approaches

	Formal approach	Cryptographic approach
Messages	terms	bitstrings
Encryption	idealized	algorithm
Adversary	idealized	any polynomial algorithm
Secrecy property	reachability-based property	indistinguishability
Guarantees	unclear	strong
Protocol	may be complex	usually simpler

Formal and Cryptographic approaches

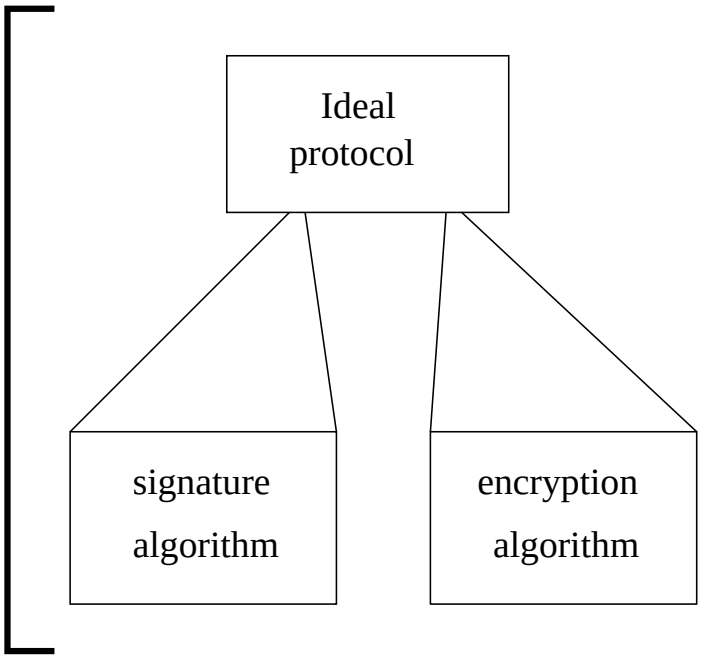
	Formal approach	Cryptographic approach
Messages	terms	bitstrings
Encryption	idealized	algorithm
Adversary	idealized	any polynomial algorithm
Secrecy property	reachability-based property	indistinguishability
Guarantees	unclear	strong
Protocol	may be complex	usually simpler
Proof	automatic	by hand, tedious and error-prone

Link between the two approaches ?

Composition of the two approaches

Automatic cryptographically sound proofs

Implemented protocol



Formal approach: verification of idealized protocols

Cryptographers: verification of the cryptographic primitives

Example : correspondence of secrecy properties

Theorem

For protocols with only public key encryption, signatures and nonces,
Whenever a protocol is proved to ensure secrecy in formal models then it ensures cryptographic indistinguishability in the computational models.

Hypotheses on the Implementation

- encryption : IND-CCA2 (e.g. the OAEP-RSA scheme)
  $\longrightarrow$ the adversary cannot distinguish between $\{n_0\}_k$ and $\{n_1\}_k$ even if he has access to n_0 and n_1 and to encryption and decryption oracles.
- signature : existentially unforgeable under chosen-message attack *i.e.* one can not produce a valid pair (m, σ)
- parsing :
  - each bit-string has a label which indicates his type (identity, nonce, key, signature, ...)
  - one can retrieve the (public) encryption key from an encrypted message.
  - one can retrieve the signed message from the signature

Proof technique

Lemma (Mapping lemma)

Each execution trace of a concrete adversary is captured by a symbolic execution trace of an ideal adversary, except with negligible probability

Proof technique

Lemma (Mapping lemma)

Each execution trace of a concrete adversary is captured by a symbolic execution trace of an ideal adversary, except with negligible probability

Proof technique : Reduce the lemma to the robustness of the primitives (which itself reduces to hardness of algorithmic problem like integer factorization).

Proof technique

Lemma (Mapping lemma)

Each execution trace of a concrete adversary is captured by a symbolic execution trace of an ideal adversary, except with negligible probability

Proof technique : Reduce the lemma to the robustness of the primitives (which itself reduces to hardness of algorithmic problem like integer factorization).

Example : If a computational (concrete) adversary $\mathcal{A}$ is able to compute $\{n_a\}_{K_a}$ out of $\{< A, n_a >\}_{K_a}$,
Then we can build an adversary $\mathcal{A}'$ that breaks the encryption $\{\ \}_{K_a}$.

Conclusion

Formal methods form a powerful approach for analyzing security protocols

- Makes use of classical techniques in formal methods : term algebra, equational theories, clauses and resolution techniques, tree automata, etc.
  $\Rightarrow$ Many decision procedures
- Several automatic tools
  - For successfully detecting attacks on protocols (e.g. Casper, Avispa)
  - For proving security for an arbitrary number of sessions (e.g. ProVerif)
- Provides cryptographic guarantees under classical assumptions on the implementation of the primitives

Some current directions of research

- Enriching the symbolic model
  - Considering more equational theories (e.g. theories for e-voting protocols)
  - Adding more complex structures for data (list, XML, ...)
  - Considering recursive protocols (e.g. group protocol) where the number of message exchanges in a session is not fixed
  - Proving more complex security properties like equivalence-based properties (e.g. for anonymity or e-voting protocols)
- With cryptographic guarantees
  - Combining formal and cryptographic models for more complex primitives and security properties.
  - How far can we go ?
  - Is it possible to consider weaker cryptographic primitives ?