

Analyzing privacy properties: decision procedures for equivalence

Véronique Cortier

Colloquium in honour of Martin Abadi

June 25-26, 2015, Cachan



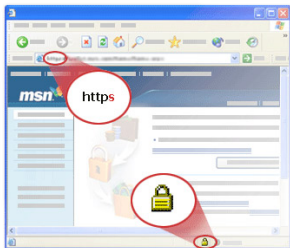
Context : Cryptographic protocols



Cryptographic protocols

- small programs designed to **secure** communication
→ **e.g.** secrecy, authentication, anonymity, . . . ,
- use **cryptographic primitives**
→ **e.g.** encryption, signature, hash, . . .

Example : HTTPS



- TLS : Transport Security Layer, depuis 1994
- various implementations : OpenSSL, SecureTransport, JSSE, ...
- Lots of bugs and attacks, with fixes every month



Memory overflow

Go to fail

Missing checks
(MACs, signatures, ...)

FREAK attack - February 2015

Bhargavan et al.



Electronic voting

Promises :

- Facilitated tally
- Voters may vote from home, at any time
- Already used in many countries : Estonia, Australia, France, Switzerland



Should offer strong security guarantees :

- Vote privacy : is my vote secret ?
- Verifiability : how can I trust the result ?
- Coercion-resistance, no vote buying, voter authentication, ...

Why is it difficult to design a secure protocol ?



Attackers may :

- read
- intercept
- forge messages
- own some compromised identities.



Why is it difficult to design a secure protocol ?



Attackers may :

- read
- intercept
- forge messages
- own some compromised identities.



Formal methods can help !

Security properties : accessibility

Trace properties

e.g. safety properties : “something bad never occurs on any execution trace of P ”

→ secrecy, authentication, ...



Security properties : accessibility

Trace properties

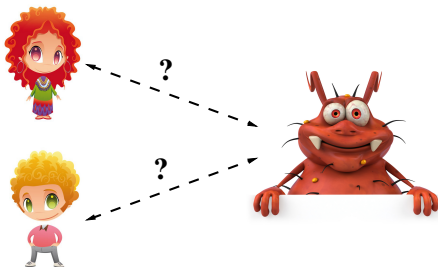
e.g. safety properties : “something bad never occurs on any execution trace of P ”
→ **secrecy, authentication, ...**



A lot of existing results :

- several **procedures** to deal with a variety of cryptographic primitives, e.g. encryption, signature, exclusive or, ...
- several **automatic tools**
e.g., ProVerif, AVISPA, Scyther, Tamarin ...

Security properties : indistinguishability



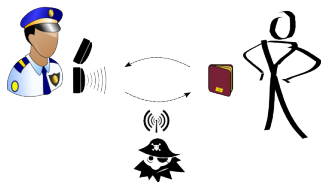
$$P[Alice] \stackrel{?}{\approx} P[Bob]$$

Equivalence based properties

“An observer cannot observe any difference between P and Q ”
→ **anonymity properties, privacy related properties in e-voting, strong secrecy, ...**

Anonymity of passport holders

An attacker should not distinguish between A and B .



$$\text{Pass}(A) \approx \text{Pass}(B)$$

May also model :

- Anonymity in a network (TOR, group signatures, ...)
- Untraceability (RFIDs, mobile phones, ...)

Vote privacy

Is my vote secret ?



$\text{Alice}(1) \approx \text{Bob}(1)$

Vote privacy

Is my vote secret ?



~~$Alice(1) \approx Bob(1)$~~

$Alice(1) \approx Alice(0)$

Vote privacy

Is my vote secret ?



~~$Alice(1) \approx Bob(1)$~~

~~$Alice(1) \approx Alice(0)$~~

$Alice(1) \mid Bob(0) \approx Alice(0) \mid Bob(1)$

How to analyse security protocols?



Methodology

- ① Proposing accurate models
 - symbolic models
 - cryptographic/computational models
- ② Proving security
 - decidability/undecidability results
 - tools

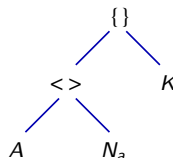
Modeling messages

Idea 1 : keeping only the structure of the messages

→ Messages are abstracted by terms.

Example :

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



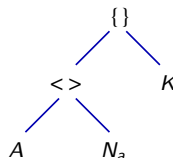
Modeling messages

Idea 1 : keeping only the structure of the messages

→ Messages are abstracted by terms.

Example :

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



Idea 2 : Equations for reflecting the properties of the primitives

Decryption

$$\text{dec}(\{x\}_y, y) = x$$

Homomorphic encryption

$$\{x_1\}_y * \{x_2\}_y = \{x_1 + x_2\}_y$$

Deduction

Deduction

$T \vdash M$ models that the attacker can deduce M out of the set T .

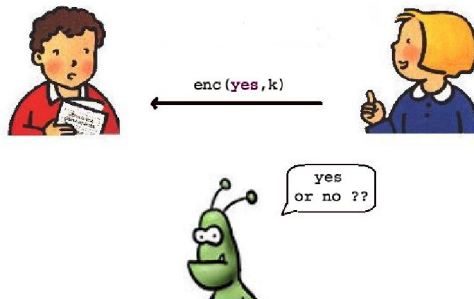
$$\frac{}{T \vdash_E M} M \in T$$

$$\frac{T \vdash_E M_1 \quad \dots \quad T \vdash_E M_k}{T \vdash_E f(M_1, \dots, M_k)} f \in \Sigma$$

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



Deduction is not always sufficient



→ The intruder **knows** the values **yes** and **no** !

The real question

Is the intruder able to tell whether Alice sent **yes** or **no** ?

Static equivalence - Intuition

The adversary sees sequences of messages $\{M_1/x_1, \dots, M_\ell/x_\ell\}$

Static equivalence - Intuition

Two sequences of messages are equivalent if the adversary cannot build a test that differentiates them.

$$\begin{aligned} \{\text{enc}(\text{yes}, k)/x_1\} &\sim \{\text{enc}(\text{no}, k)/x_1\} \\ \{\text{enc}(\text{yes}, k)/x_1, k/x_2\} &\not\sim \{\text{enc}(\text{no}, k)/x_1, k/x_2\} \end{aligned}$$

Since $\text{dec}(x_2, x_1) = \text{yes}$ holds on the left but not on the right.

Static equivalence - Decidability

Theorem [Abadi,C.]

$\phi_1 \sim_E \phi_2$ is decidable (PTIME) for convergent subterm theories **E**.

Convergent subterm theories : encryption (symmetric and asymmetric), signatures, hashes, MACs, ...

$$\text{dec}(\text{enc}(x, y), y) = x$$

$$\text{adec}(\text{aenc}(x, \text{pub}(y)), y) = x$$

$$\pi_1(\langle x, y \rangle) = x$$

$$\pi_2(\langle x, y \rangle) = y$$

Blind signatures

First phase :

$$V \rightarrow A : \text{sign}(\text{blind}(\text{vote}, r), V)$$
$$A \rightarrow V : \text{sign}(\text{blind}(\text{vote}, r), A)$$


Voting phase :

$$V \rightarrow C : \text{sign}(\text{vote}, A)$$

Equational theory for blind signatures

[Kremer Ryan 05]

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

Other examples of theories

EXclusive Or

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

Diffie-Hellmann

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

More decidability results

Problem : $M_1 \sim_E^? M_2$ for the theory E (modeling primitives).

- **PTIME** for **locally stable** convergent theories

$\text{unblind}(\text{sign}(\text{blind}(x, y), z), y) = \text{sign}(x, z)$

In collaboration with Martin Abadi

More decidability results

Problem : $M_1 \stackrel{?}{\sim}_E M_2$ for the theory E (modeling primitives).

- PTIME for **locally stable** convergent theories

$$\text{unblind}(\text{sign}(\text{blind}(x, y), z), y) = \text{sign}(x, z)$$

In collaboration with Martin Abadi

- Decidability of **monoidal theories** with associative and commutative operators, by reduction to classical algebraic problems (e.g. **Gröbner basis**)

$$x+y = y+x, \quad x+(y+z) = (x+y)+z, \quad h(x+y) = h(x)+h(y)$$

In collaboration with Stéphanie Delaune

More decidability results

Problem : $M_1 \stackrel{?}{\sim}_E M_2$ for the theory E (modeling primitives).

- **PTIME** for **locally stable** convergent theories

$$\text{unblind}(\text{sign}(\text{blind}(x, y), z), y) = \text{sign}(x, z)$$

In collaboration with Martin Abadi

- Decidability of **monoidal theories** with associative and commutative operators, by reduction to classical algebraic problems (e.g. **Gröbner basis**)

$$x+y = y+x, \quad x+(y+z) = (x+y)+z, \quad h(x+y) = h(x)+h(y)$$

In collaboration with Stéphanie Delaune

- **Combination result :** (for E_1, E_2 disjoint theories)

If $\approx$ is decidable for E_1 and for E_2 then it is decidable for $E_1 \cup E_2$.

In collaboration with Stéphanie Delaune

Some tools : YAPA, KISS, ...

More decidability results

Problem : $M_1 \stackrel{?}{\sim}_E M_2$ for the theory E (modeling primitives).

- **PTIME** for **locally stable** convergent theories

$\text{unblind}(\text{sign}(\text{blind}(x, y), z), y) = \text{sign}(x, z)$

In collaboration with Martin Abadi

- Decidability of **monoidal theories** with associative and commutative operators, by reduction to classical algebraic problems (e.g. **Gröbner basis**)

$x+y = y+x, \quad x+(y+z) = (x+y)+z, \quad h(x+y) = h(x)+h(y)$

In collaboration with Stéphanie Delaune

- **Combination result** : (for E_1, E_2 disjoint theories)

If $\approx$ is decidable for E_1 and for E_2 then it is decidable for $E_1 \cup E_2$.

In collaboration with Stéphanie Delaune

Some tools : YAPA, KISS, ...

→ This is just w.r.t. a **passive adversary** (an **eavesdropper**).

Active case : applied-pi calculus

Applied pi-calculus : Abadi & Fournet, 2001 based on the π -calculus

$P, Q, R :=$ plain processes

0

null process

$\text{in}(u, x).P$

message input

$\text{out}(u, M).P$

message output

if $M = N$ then P else Q

conditional

$\nu n.P$

name restriction

$P \mid Q$

parallel

$!P$

replication

Active case : applied-pi calculus

Applied pi-calculus : Abadi & Fournet, 2001 based on the π -calculus

$P, Q, R :=$ plain processes

0

null process

$\text{in}(u, x).P$

message input

$\text{out}(u, M).P$

message output

if $M = N$ then P else Q

conditional

$\nu n.P$

name restriction

$P \mid Q$

parallel

$!P$

replication

Trace equivalence - Intuition

Two processes P and Q are in trace equivalence $P \approx Q$ if for any sequence of identical observable actions, the resulting sequences of emitted messages are in **static equivalence**.

Some results for trace equivalence

$$P \stackrel{?}{\approx} Q$$

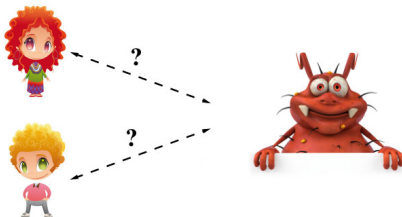
Decision procedures / Tools :

- **ProVerif**, not guaranteed to terminate, P and Q have to be similar See Bruno's talk tomorrow !
- **Akiss**, not guaranteed to terminate, bounded number of sessions, arbitrary primitives
- **SPEC**, bounded number of sessions, standard primitives
- **APTE**, bounded number of sessions, standard primitives, else branches

→ None of these approaches take into account that **length is a distinguishing factor**.

→ No decidability result for an **unbounded number of sessions**.

Time and length

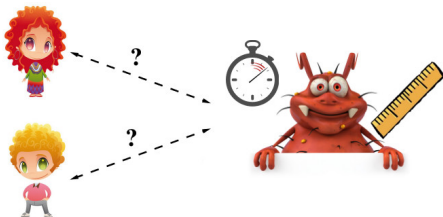


Decidability of trace equivalence

A decision procedure for trace equivalence :

- all the standard primitives (encryption, signature, hash, MACs, etc.)
- bounded number of sessions

Time and length



Decidability of trace equivalence

A decision procedure for trace equivalence :

- all the standard primitives (encryption, signature, hash, MACs, etc.)
- bounded number of sessions
- **active adversary that can measure length and time**
- integrated in the APTE tool

[http ://projects.lsv.ens-cachan.fr/APTE/](http://projects.lsv.ens-cachan.fr/APTE/)

Static equivalence does not account for length differences

$$\{\text{enc}(\text{yes}, k) /_{x_1}\} \sim \{\text{enc}(\text{no}, k) /_{x_1}\}$$

Static equivalence does not account for length differences

$$\{\text{enc}(\text{yes}, k) /_{x_1}\} \sim \{\text{enc}(\text{no}, k) /_{x_1}\}$$

And also :

$$\{\text{enc}(\text{yes}, k) /_x\} \sim \{\text{enc}(\langle \text{yes}, \text{yes} \rangle, k) /_x\},$$

Static equivalence does not account for length differences

$$\{\text{enc}(\text{yes}, k) /_{x_1}\} \sim \{\text{enc}(\text{no}, k) /_{x_1}\}$$

And also :

$$\begin{aligned} \{\text{enc}(\text{yes}, k) /_x\} &\sim \{\text{enc}(\langle \text{yes}, \text{yes} \rangle, k) /_x\}, \\ \{\text{enc}(\text{yes}, k) /_x\} &\sim \{\text{enc}(\langle \text{yes}, \text{yes}, \text{yes}, \text{yes} \rangle, k) /_x\}, \\ \{\text{enc}(\text{yes}, k) /_x\} &\sim \{\text{enc}(\langle \text{yes}, \text{yes}, \text{yes}, \text{yes}, \text{yes}, \text{yes}, \text{yes} \rangle, k) /_x\}, \\ \{\text{enc}(\text{yes}, k) /_x\} &\sim \{\text{enc}(\langle \text{yes}, \text{yes}, \text{yes}, \text{yes}, \text{yes}, \text{yes}, \text{yes}, \dots \rangle, k) /_x\}, \dots \end{aligned}$$

Length functions

A length function is a function $\ell :$
 $T(F, N, X) \mapsto \mathbb{R}^+.$



Length functions

A length function is a function ℓ :
 $T(F, N, X) \mapsto \mathbb{R}^+$.



A **linear length function** satisfies :

$$\ell(f(t_1, \dots, t_n)) = \beta^f + \alpha_1^f \ell(t_1) + \dots + \alpha_n^f \ell(t_n)$$

Length functions

A length function is a function ℓ :
 $T(F, N, X) \mapsto \mathbb{R}^+$.



A **linear length function** satisfies :

$$\ell(f(t_1, \dots, t_n)) = \beta^f + \alpha_1^f \ell(t_1) + \dots + \alpha_n^f \ell(t_n)$$

Example :

$$\begin{aligned}\ell(k) &= 1 \\ \ell(\langle u, v \rangle) &= 1 + \ell(u) + \ell(v) \\ \ell(\text{enc}(u, v)) &= 1 + \ell(u) + \ell(v) \\ \ell(\text{aenc}(u, v)) &= 2 + \ell(u) + \ell(v)\end{aligned}$$

Decidability for time and length

joint work with V. Cheval

Theorem (decidability [CAV'13])

The following problem is decidable

Input *Two processes P and Q and a linear length function ℓ*

Output *Whether $P \approx^\ell Q$*

Decidability for time and length

joint work with V. Cheval

Theorem (decidability [CAV'13])

The following problem is decidable

Input Two processes P and Q and a linear length function ℓ

Output Whether $P \approx^\ell Q$

Theorem (extension to time [POST'15])

Time and length equivalence can be reduced to time equivalence

$$P \approx^t Q \Leftrightarrow P \approx^\ell Q$$

Decidability for time and length

joint work with V. Cheval

Theorem (decidability [CAV'13])

The following problem is decidable

Input *Two processes P and Q and a linear length function ℓ*

Output *Whether $P \approx^\ell Q$*

Theorem (extension to time [POST'15])

Time and length equivalence can be reduced to time equivalence

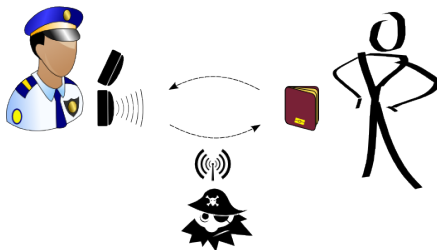
$$P \approx^t Q \Leftrightarrow P \approx^\ell Q$$

→ Implemented in the **APTE** tool.

<http://projects.lsv.ens-cachan.fr/APTE/>

Case study : biometric passport

Passive Authentication protocol of the biometric passport



New attack for tracing users : even if data are encrypted

- Tested (on a small number of passport : ~ 20)
- Relies on an usual weakness : **messages of variable lengths** (e.g. jpeg picture)
- May be fixed using padding

How to decide trace equivalence for unbounded sessions ?

Quite undecidable...

Reachability property (e.g. secrecy) : some results, often with many restrictions :

- no nonces, at most one variable
- no nonces, bounded size
- with nonces, strong tagging, no ciphertext forwarding

Equivalence properties

- **no result !**

Small attack property

with R. Chréten and S. Delaune [Concur'14]

Small attack property with typing

For tagged determinate protocols, there is an attack if, and only if, there exists a **well-typed** attack, that is, an attack where all messages comply with honest messages format.

Determinate protocols : the protocol's output is determined by the attacker's actions.

Tagged protocols

Tagged version of Otway-Rees

1. $A \rightarrow B : M, A, B, \{\mathbf{1}, N_a, M, A, B\}_{K_{as}}$
2. $B \rightarrow S : M, A, B, \{\mathbf{1}, N_a, M, A, B\}_{K_{as}}, \{\mathbf{2}, N_b, M, A, B\}_{K_{bs}}$
3. $S \rightarrow B : M, \{\mathbf{3}, N_a, K_{ab}\}_{K_{as}}, \{\mathbf{4}, N_b, K_{ab}\}_{K_{bs}}$
4. $B \rightarrow A : M, \{\mathbf{3}, N_a, K_{ab}\}_{K_{as}}$

Type-compliant protocols

- δ : any typing policy (that supports unification)
- $ESt(P)$: set of encrypted subterm of P

Type-compliant protocols

P is **type-compliant** w.r.t. a typing policy δ if $\forall t, t' \in ESt(P)$
 $t\sigma_1 = t\sigma_2 \Rightarrow \delta(t) = \delta(t')$

Note : can be relaxed for variables not under replication

Type-compliant protocols

- δ : any typing policy (that supports unification)
- $ESt(P)$: set of encrypted subterm of P

Type-compliant protocols

P is **type-compliant** w.r.t. a typing policy δ if $\forall t, t' \in ESt(P)$
 $t\sigma_1 = t\sigma_2 \Rightarrow \delta(t) = \delta(t')$

Note : can be relaxed for variables not under replication

Tagged protocols

P is tagged if :

- any term in $ESt(P)$ is of the form $\text{enc}(\langle c, m \rangle, k)$,
- there exists σ_P such that for any $s, t \in ESt(P)$ with the same tag, $s\sigma_P = t\sigma_P$.

Type-compliant protocols

- δ : any typing policy (that supports unification)
- $ESt(P)$: set of encrypted subterm of P

Type-compliant protocols

P is **type-compliant** w.r.t. a typing policy δ if $\forall t, t' \in ESt(P)$
 $t\sigma_1 = t\sigma_2 \Rightarrow \delta(t) = \delta(t')$

Note : can be relaxed for variables not under replication

Tagged protocols

P is tagged if :

- any term in $ESt(P)$ is of the form $\text{enc}(\langle c, m \rangle, k)$,
- there exists σ_P such that for any $s, t \in ESt(P)$ with the same tag, $s\sigma_P = t\sigma_P$.

→ Tagged protocols are **type-compliant**.

Decidability result, without nonces

Decidability

For tagged protocols with a **bounded number of nonces**, trace equivalence is decidable, for protocols with **symmetric encryption and concatenation**.

Also true for type-compliant protocols w.r.t. a finite typing system (finitely many terms of each type).

Side result : decision procedure for trace equivalence of determinate protocols for a bounded number of sessions **which preserves types**.

Decidability result, with nonces

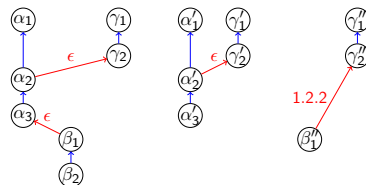
But well, protocols do **use nonces**.

Decidability result, with nonces

But well, protocols do **use nonces**.

Dependency graph : computes dependencies on a well-typed trace

- sequential dependencies
- data dependencies



Theorem

For tagged protocols with *symmetric encryption and concatenation* and an *acyclic graph*, trace equivalence is decidable for an *unbounded number of nonces*.

with R. Cr  tien and S. Delaune [CSF'15]

Conclusion

Can we check for equivalence/privacy properties?

- growing number of decision procedures
- integrated in some tools : ProVerif, APTE, Akiss
- may capture time and length, else branches, ...

What remains to be done?

- more primitives (needed e.g. for voting)
- efficiency
- allow for interactive proofs
- automation of cryptographic proofs?