

Combining algorithms for deciding knowledge in security protocols

Mathilde Arnaud, Véronique Cortier and Stéphanie Delaune

LORIA, CNRS & INRIA project Cassis, Nancy, France

September 10, 2007



Cryptographic protocols

- small programs designed to **secure** communication (*e.g.* secrecy)
- use **cryptographic primitives** (*e.g.* encryption, hash function, ...)

Presence of an attacker

- may **read** every message sent on the network,
- may **intercept** and **send** new messages according to its deduction capabilities.

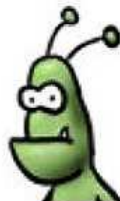


Cryptographic protocols

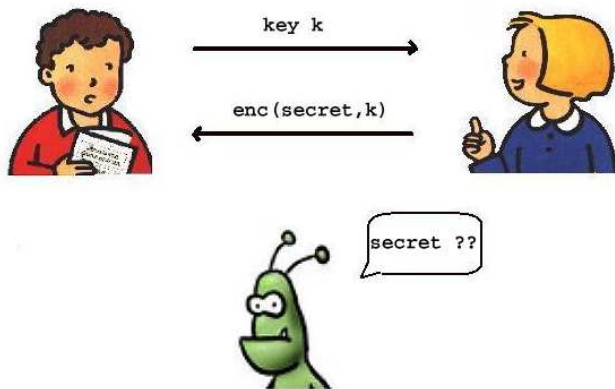
- small programs designed to **secure** communication (*e.g.* secrecy)
- use **cryptographic primitives** (*e.g.* encryption, hash function, ...)

Presence of an attacker

- may **read** every message sent on the network,
- may **intercept** and **send** new messages according to its deduction capabilities.



A simple protocol



→ Does the attacker **know** secret?

Attacker power (in formal models)

→ The attacker can do **symbolic manipulations** on messages.

Messages are abstracted by terms ...

- encryption $\{x\}_y$,
- pairing $\langle x, y \rangle$, ...

... together with an equational theory

- classical theory (E_{enc}):

$$\text{proj}_1(\langle x, y \rangle) = x \quad \text{proj}_2(\langle x, y \rangle) = y \quad \text{dec}(\text{enc}(x, y), y) = x$$

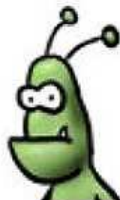
- exclusive or (E_{xor}):

$$\begin{array}{lcl} (x \oplus y) \oplus z & = & x \oplus (y \oplus z) \\ x \oplus 0 & = & x \end{array} \quad \begin{array}{lcl} x \oplus y & = & y \oplus x \\ x \oplus x & = & 0 \end{array}$$

Understanding security protocols often requires reasoning about **knowledge** of the attacker.

Two main kinds of knowledge

- deduction,
- static equivalence – indistinguishability



→ rely on an underlying **equational theory**

→ often used as **subroutines** in many decision procedures

Deduction

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

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

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

$$\frac{\frac{}{T \vdash \text{enc}(\text{secret}, k)} \quad \frac{}{T \vdash k}}{T \vdash \text{dec}(\text{enc}(\text{secret}, k), k)} \quad f \in \Sigma}{T \vdash \text{secret}} \text{dec}(\text{enc}(x, y), y) = x$$

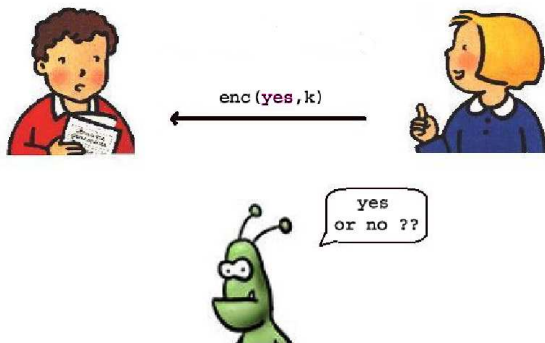
$$\frac{}{T \vdash_E M} M \in T \qquad \frac{T \vdash_E M_1 \quad \dots \quad T \vdash_E M_k}{T \vdash_E f(M_1, \dots, M_k)} f \in \Sigma$$

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

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

$$\frac{\frac{}{T \vdash \text{enc}(\text{secret}, k)} \quad \frac{}{T \vdash k}}{T \vdash \text{dec}(\text{enc}(\text{secret}, k), k)} f \in \Sigma}{T \vdash \text{secret}} \text{dec}(\text{enc}(x, y), y) = x$$

Deduction is not always sufficient



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

The real question

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

Static equivalence (indistinguishability relation)

frame = set of **restricted names** + sequence of messages

$$\phi = \nu \tilde{n}. \{M_1/x_1, \dots, M_\ell/x_\ell\}$$

Examples:

- If the key k is **not revealed**, we have that

$$\phi_1 = \nu k. \{\text{enc}(\text{yes}, k)/x\} \quad \text{and} \quad \phi_2 = \nu k. \{\text{enc}(\text{no}, k)/x\}$$

- If the key k is **revealed**, we have that

$$\psi_1 = \nu k. \{k/x_1, \text{enc}(\text{yes}, k)/x_2\} \quad \text{and} \quad \psi_2 = \nu k. \{k/x_1, \text{enc}(\text{no}, k)/x_2\}$$

Static equivalence (indistinguishability relation)

frame = set of **restricted names** + sequence of messages

$$\phi = \nu \tilde{n}. \{M_1/x_1, \dots, M_\ell/x_\ell\}$$

Examples:

- If the key **k is not revealed**, we have that

$$\phi_1 = \nu k. \{\text{enc}(\text{yes}, k)/x\} \quad \text{and} \quad \phi_2 = \nu k. \{\text{enc}(\text{no}, k)/x\}$$

- If the key **k is revealed**, we have that

$$\psi_1 = \nu k. \{k/x_1, \text{enc}(\text{yes}, k)/x_2\} \quad \text{and} \quad \psi_2 = \nu k. \{k/x_1, \text{enc}(\text{no}, k)/x_2\}$$

Static equivalence (indistinguishability relation)

frame = set of **restricted names** + sequence of messages

$$\phi = \nu \tilde{n}. \{M_1/x_1, \dots, M_\ell/x_\ell\}$$

Examples:

- If the key **k is not revealed**, we have that

$$\phi_1 = \nu k. \{\text{enc}(\text{yes}, k)/x\} \quad \text{and} \quad \phi_2 = \nu k. \{\text{enc}(\text{no}, k)/x\}$$

- If the key **k is revealed**, we have that

$$\psi_1 = \nu k. \{k/x_1, \text{enc}(\text{yes}, k)/x_2\} \quad \text{and} \quad \psi_2 = \nu k. \{k/x_1, \text{enc}(\text{no}, k)/x_2\}$$

Static equivalence (indistinguishability relation)

frame = set of **restricted names** + sequence of messages

$$\phi = \nu \tilde{n}. \{M_1/x_1, \dots, M_\ell/x_\ell\}$$

Examples:

- If the key **k is not revealed**, we have that

$$\phi_1 = \nu k. \{\text{enc}(\text{yes}, k)/x\} \quad \text{and} \quad \phi_2 = \nu k. \{\text{enc}(\text{no}, k)/x\}$$

→ **indistinguishable**

- If the key **k is revealed**, we have that

$$\psi_1 = \nu k. \{k/x_1, \text{enc}(\text{yes}, k)/x_2\} \quad \text{and} \quad \psi_2 = \nu k. \{k/x_1, \text{enc}(\text{no}, k)/x_2\}$$

→ **distinguishable**

Goal of this paper

Our contribution

We propose **combination** algorithms (**PTIME**) for **deduction** and **static equivalence** for **disjoint** equational theories.

A modular approach

→ Deciding interesting theories can be done by **reducing** the decision to **simpler theories**.

New decidability results

Deduction and **static equivalence** are decidable in **PTIME** for subterm theories (e.g. E_{enc}) and exclusive or (E_{xor}) [Abadi&Cortier,06], [Chevalier *et al.*,03].

→ those problems are also **decidable in PTIME** for $E_{\text{enc}} \cup E_{\text{xor}}$.

Goal of this paper

Our contribution

We propose **combination** algorithms (**PTIME**) for **deduction** and **static equivalence** for **disjoint** equational theories.

A modular approach

→ Deciding interesting theories can be done by **reducing** the decision to **simpler theories**.

New decidability results

Deduction and **static equivalence** are decidable in **PTIME** for subterm theories (e.g. E_{enc}) and exclusive or (E_{xor}) [Abadi&Cortier,06], [Chevalier *et al.*,03].

→ those problems are also **decidable in PTIME** for $E_{\text{enc}} \cup E_{\text{xor}}$.

Combination for unification

Our procedures rely on combination algorithms for solving **unification** modulo $E = E_1 \cup E_2$ (E_1 and E_2 are disjoint)

→ [Schmidt-Schauss,89], [Baader&Schulz,96]

Combination for deduction (active case)

We follow the approach developed in [Chevalier&Rusinowitch,05]

→ combination algorithm for **deduction** in the presence of an **active** attacker (they take into account the rules of the protocol)

→ they do **not** consider **static equivalence**

Outline of the talk

- 1 Introduction
- 2 Deduction
- 3 Static equivalence
- 4 Conclusion

Lemma (characterization of deduction)

$\phi \vdash_E M$ if and only if there exists a term ζ such that $\zeta\phi =_E M$.

→ Such a term ζ is a *recipe* of the term M .

Example: $E := \text{dec}(\text{enc}(x, y), y) = x$.

$$\phi = \nu k. \nu s. \{ \text{enc}(s, k) / x_1, k / x_2 \}$$

We have that $\phi \vdash_E s$. Indeed $\zeta = \text{dec}(x_1, x_2)$ is a recipe of s .

Deduction problem for the equational theory E built over Σ .

Entries: A frame ϕ and a term M (both built over Σ)

Question: $\phi \vdash_E M$?

Lemma (characterization of deduction)

$\phi \vdash_E M$ if and only if there exists a term ζ such that $\zeta\phi =_E M$.

→ Such a term ζ is a *recipe* of the term M .

Example: $E := \text{dec}(\text{enc}(x, y), y) = x$.

$$\phi = \nu k. \nu s. \{ \text{enc}(s, k) / x_1, k / x_2 \}$$

We have that $\phi \vdash_E s$. Indeed $\zeta = \text{dec}(x_1, x_2)$ is a recipe of s .

Deduction problem for the equational theory E built over Σ .

Entries: A frame ϕ and a term M (both built over Σ)

Question: $\phi \vdash_E M$?

Main result for deduction

Theorem (Combination for deduction)

Let (Σ_1, E_1) and (Σ_2, E_2) be two *disjoint* equational theories. If *deduction* is decidable for (Σ_1, E_1) and (Σ_2, E_2) then *deduction is decidable* for $(\Sigma_1 \cup \Sigma_2, E_1 \cup E_2)$.

Our algorithm

Let ϕ be a frame and M be a term built over $\Sigma_1 \cup \Sigma_2$.

- 1 compute the *subterms* (alien subterms) of ϕ and M .
- 2 *saturation* of ϕ by subterms which are deducible either in E_1 or in E_2
→ abstraction of alien factors by fresh names
- 3 check if $M \in \text{sat}(\phi)$.

→ *completeness* obtained thanks to a *locality* lemma.

→ our algorithm is *polynomial* (in the DAG-size of the inputs)

Main result for deduction

Theorem (Combination for deduction)

Let (Σ_1, E_1) and (Σ_2, E_2) be two *disjoint* equational theories. If *deduction* is decidable for (Σ_1, E_1) and (Σ_2, E_2) then *deduction is decidable* for $(\Sigma_1 \cup \Sigma_2, E_1 \cup E_2)$.

Our algorithm

Let ϕ be a frame and M be a term built over $\Sigma_1 \cup \Sigma_2$.

- 1 compute the **subterms** (alien subterms) of ϕ and M .
- 2 **saturation** of ϕ by subterms which are deducible either in E_1 or in E_2
→ abstraction of alien factors by fresh names
- 3 check if $M \in \text{sat}(\phi)$.

→ **completeness** obtained thanks to a **locality** lemma.

→ our algorithm is **polynomial** (in the DAG-size of the inputs)

Our algorithm on an example

Equational theory: $E = E_{\text{enc}} \cup E_{\text{xor}}$

$$\phi = \nu n_2, n_3. \{ \text{enc}(\langle n_1 \oplus n_2, n_3 \rangle, n_4) /_{x_1} \} \quad M = n_2 \oplus n_3$$

① **subterms** (alien) of ϕ and M :

$$\text{enc}(\langle n_1 \oplus n_2, n_3 \rangle, n_4), M, n_1 \oplus n_2, n_1, n_2, n_3, n_4$$

- ② saturation of ϕ by deducible subterms either in E_{enc} or in E_{xor}
- $n_1 \oplus n_2$ deducible in E_{enc} with $\zeta_3 = \text{proj}_1(\text{dec}(x_1, n_4)) - x_2$
 - n_3 deducible in E_{enc} with $\zeta_3 = \text{proj}_2(\text{dec}(x_1, n_4)) - x_3$
 - $n_2 \oplus n_3$ deducible in E_{xor} with $\zeta_4 = n_1 \oplus x_2 \oplus x_3$
 - ...
- ③ it is now easy to check that $M \in \text{sat}(\phi)$

Our algorithm on an example

Equational theory: $E = E_{\text{enc}} \cup E_{\text{xor}}$

$$\phi = \nu n_2, n_3. \{ \text{enc}(\langle n_1 \oplus n_2, n_3 \rangle, n_4) /_{x_1} \} \quad M = n_2 \oplus n_3$$

① **subterms** (alien) of ϕ and M :

$$\text{enc}(\langle n_1 \oplus n_2, n_3 \rangle, n_4), M, n_1 \oplus n_2, n_1, n_2, n_3, n_4$$

② saturation of ϕ by deducible subterms either in E_{enc} or in E_{xor}

→ $n_1 \oplus n_2$ deducible in E_{enc} with $\zeta_3 = \text{proj}_1(\text{dec}(x_1, n_4)) - x_2$

→ n_3 deducible in E_{enc} with $\zeta_3 = \text{proj}_2(\text{dec}(x_1, n_4)) - x_3$

→ $n_2 \oplus n_3$ deducible in E_{xor} with $\zeta_4 = n_1 \oplus x_2 \oplus x_3$

→ ...

③ it is now easy to check that $M \in \text{sat}(\phi)$

Our algorithm on an example

Equational theory: $E = E_{\text{enc}} \cup E_{\text{xor}}$

$$\phi = \nu n_2, n_3. \{ \text{enc}(\langle n_1 \oplus n_2, n_3 \rangle, n_4) /_{x_1} \} \quad M = n_2 \oplus n_3$$

① **subterms** (alien) of ϕ and M :

$$\text{enc}(\langle n_1 \oplus n_2, n_3 \rangle, n_4), M, n_1 \oplus n_2, n_1, n_2, n_3, n_4$$

② saturation of ϕ by deducible subterms either in E_{enc} or in E_{xor}

$$\longrightarrow n_1 \oplus n_2 \text{ deducible in } E_{\text{enc}} \text{ with } \zeta_3 = \text{proj}_1(\text{dec}(x_1, n_4)) - x_2$$

$$\longrightarrow n_3 \text{ deducible in } E_{\text{enc}} \text{ with } \zeta_3 = \text{proj}_2(\text{dec}(x_1, n_4)) - x_3$$

$$\longrightarrow n_2 \oplus n_3 \text{ deducible in } E_{\text{xor}} \text{ with } \zeta_4 = n_1 \oplus x_2 \oplus x_3$$

$\longrightarrow \dots$

③ it is now easy to check that $M \in \text{sat}(\phi)$

Outline of the talk

- 1 Introduction
- 2 Deduction
- 3 Static equivalence**
- 4 Conclusion

Static equivalence

Definition (static equivalence)

$\phi_1 \approx \phi_2$ iff $\text{dom}(\phi_1) = \text{dom}(\phi_2)$ and for every couple of terms (M, N)

$$(M =_{\mathbf{E}} N)\phi_1 \Leftrightarrow (M =_{\mathbf{E}} N)\phi_2$$

Example: $\mathbf{E} = \text{dec}(\text{enc}(x, y), y) = x$

$$\psi_1 = \nu k. \{k/x_1, \text{enc}(\text{yes}, k)/x_2\} \text{ and } \psi_2 = \nu k. \{k/x_1, \text{enc}(\text{no}, k)/x_2\}$$

→ not statically equivalent, choose $M = \text{dec}(x_2, x_1)$ and $N = \text{yes}$

Static equivalence problem for the theory \mathbf{E} built over Σ .

Entries: Two frames ϕ_1 and ϕ_2 (both built over Σ)

Question: $\phi_1 \approx_{\mathbf{E}} \phi_2?$

Static equivalence

Definition (static equivalence)

$\phi_1 \approx \phi_2$ iff $\text{dom}(\phi_1) = \text{dom}(\phi_2)$ and for every couple of terms (M, N)

$$(M =_{\mathbf{E}} N)\phi_1 \Leftrightarrow (M =_{\mathbf{E}} N)\phi_2$$

Example: $\mathbf{E} = \text{dec}(\text{enc}(x, y), y) = x$

$$\psi_1 = \nu k. \{k/x_1, \text{enc}(\text{yes}, k)/x_2\} \text{ and } \psi_2 = \nu k. \{k/x_1, \text{enc}(\text{no}, k)/x_2\}$$

→ **not statically equivalent**, choose $M = \text{dec}(x_2, x_1)$ and $N = \text{yes}$

Static equivalence problem for the theory \mathbf{E} built over Σ .

Entries: Two frames ϕ_1 and ϕ_2 (both built over Σ)

Question: $\phi_1 \approx_{\mathbf{E}} \phi_2$?

Definition (static equivalence)

$\phi_1 \approx \phi_2$ iff $\text{dom}(\phi_1) = \text{dom}(\phi_2)$ and for every couple of terms (M, N)

$$(M =_{\mathbf{E}} N)\phi_1 \Leftrightarrow (M =_{\mathbf{E}} N)\phi_2$$

Example: $\mathbf{E} = \text{dec}(\text{enc}(x, y), y) = x$

$$\psi_1 = \nu k. \{k/x_1, \text{enc}(\text{yes}, k)/x_2\} \text{ and } \psi_2 = \nu k. \{k/x_1, \text{enc}(\text{no}, k)/x_2\}$$

→ **not statically equivalent**, choose $M = \text{dec}(x_2, x_1)$ and $N = \text{yes}$

Static equivalence problem for the theory \mathbf{E} built over Σ .

Entries: Two frames ϕ_1 and ϕ_2 (both built over Σ)

Question: $\phi_1 \approx_{\mathbf{E}} \phi_2$?

Main result for static equivalence

Theorem (Combination for static equivalence)

Let (Σ_1, E_1) and (Σ_2, E_2) be two *disjoint* equational theories. If deduction *and* static equivalence are decidable for (Σ_1, E_1) and (Σ_2, E_2) then *static equivalence is decidable* for $(\Sigma_1 \cup \Sigma_2, E_1 \cup E_2)$.

Our algorithm

Let ϕ_1 and ϕ_2 two frames built over $\Sigma_1 \cup \Sigma_2$.

① First step:

② Second step: If ϕ'_1 and ϕ'_2 contains all their deducible subterms then

$$\phi_1 \approx_E \phi_2 \Leftrightarrow \begin{cases} \phi'_1 \approx_{E_1} \phi'_2 \\ \text{and} \\ \phi'_1 \approx_{E_2} \phi'_2 \end{cases} \quad \text{where alien factors are abstracted by fresh names}$$

Main result for static equivalence

Theorem (Combination for static equivalence)

Let (Σ_1, E_1) and (Σ_2, E_2) be two *disjoint* equational theories. If deduction *and* static equivalence are decidable for (Σ_1, E_1) and (Σ_2, E_2) then *static equivalence is decidable* for $(\Sigma_1 \cup \Sigma_2, E_1 \cup E_2)$.

Our algorithm

Let ϕ_1 and ϕ_2 two frames built over $\Sigma_1 \cup \Sigma_2$.

① First step:

② Second step: If ϕ'_1 and ϕ'_2 contains all their deducible subterms then

$$\phi'_1 \approx_E \phi'_2 \Leftrightarrow \begin{cases} \phi'_1 \approx_{E_1} \phi'_2 \\ \text{and} \\ \phi'_1 \approx_{E_2} \phi'_2 \end{cases} \quad \text{where alien factors are abstracted by fresh names}$$

Main result for static equivalence

Theorem (Combination for static equivalence)

Let (Σ_1, E_1) and (Σ_2, E_2) be two *disjoint* equational theories. If deduction *and* static equivalence are decidable for (Σ_1, E_1) and (Σ_2, E_2) then *static equivalence is decidable* for $(\Sigma_1 \cup \Sigma_2, E_1 \cup E_2)$.

Our algorithm

Let ϕ_1 and ϕ_2 two frames built over $\Sigma_1 \cup \Sigma_2$.

- First step:** compute ϕ'_1 and ϕ'_2 such that
 - $\longrightarrow \phi'_1$ and ϕ'_2 contain their **deducible subterms**, and
 - $\longrightarrow \phi_1 \approx_E \phi_2 \Leftrightarrow \phi'_1 \approx_E \phi'_2$.
- Second step:** If ϕ'_1 and ϕ'_2 contains all their deducible subterms then

$$\phi'_1 \approx_E \phi'_2 \Leftrightarrow \begin{cases} \phi'_1 \approx_{E_1} \phi'_2 \\ \text{and} \\ \phi'_1 \approx_{E_2} \phi'_2 \end{cases} \quad \begin{array}{l} \text{where alien factors are} \\ \text{abstracted by fresh names} \end{array}$$

Outline of the talk

- 1 Introduction
- 2 Deduction
- 3 Static equivalence
- 4 Conclusion**

Our contribution

We propose **combination** algorithms (**PTIME**) for **deduction** and **static equivalence** for **disjoint** equational theories.

→ A **methodology** to prove deduction and static equivalence for complex equational theories

→ **New** decidability and complexity **results** for interesting theories,

Example

Deduction and **static equivalence** are **decidable** in **PTIME** for

- $E_{\text{enc}} \cup E_{\text{xor}}$,
- $E_{\text{enc}} \cup E_{\text{AG}}$, and more generally
- $E \cup E_{\text{xor}}$ (or $E \cup E_{\text{AG}}$) where E is any subterm convergent theory.

Future work

→ Extension to **non-disjoint** equational theories

Example: fragment of the modular exponentiation theory

- $\exp(x, 1) = x$,
- $\exp(\exp(x, y), z) = \exp(x, y \times z)$,
- $\exp(x, y) \cdot \exp(x, z) = \exp(x, y + z), \dots$

where \times is an Abelian group operator.

→ **Implementation** of the algorithms

→ Extension to **active attacker**

- for deduction – already done [Chevalier *et al.*'05]
- for static equivalence
it will be useful to decide guessing attacks in new equational theories