

YAPA: A generic tool for computing intruder knowledge

Mathieu Baudet¹

Joint work with Véronique Cortier² and Stéphanie Delaune³

¹DCSSI, France

²LORIA, CNRS & INRIA project Cassis, France

³LSV, ENS Cachan & CNRS & INRIA, France

RTA'2009, Brasília, June 29.

Content of the talk

1 Motivations

- Why study static equivalence?
- Why a new tool?

2 Results

- Overview of the procedure
- Examples
- Proving termination and non-failure

3 Conclusion

Content of the talk

1 Motivations

- Why study static equivalence?
- Why a new tool?

2 Results

- Overview of the procedure
- Examples
- Proving termination and non-failure

3 Conclusion

Static equivalence (teaser)

- 1 A *useful* logical tool for security protocols.
- 2 A *nice* and general algebraic notion.

Algebraic framework

- Consider a set \mathcal{F}_{pub} of first-order symbols
 $f : s \times \cdots \times s \rightarrow s$. (Single sort s assumed for simplicity.)
- A \mathcal{F}_{pub} -algebra is a set \mathcal{A} together with functions
 $f_{\mathcal{A}} : \mathcal{A} \times \cdots \times \mathcal{A} \rightarrow \mathcal{A}$.
- Standard definitions : \mathcal{F}_{pub} -morphisms, generated sub-algebras $\mathcal{F}_{\text{pub}}[S] \subseteq \mathcal{A}$, free algebra $\mathcal{F}_{\text{pub}}[X]$, ...

Static equivalence (algebraic definition)

- Consider the tuples $\varphi = (t_1, \dots, t_n)$ in \mathcal{A}^n , also called *frames* and written $\varphi = \{w_1 \triangleright t_1, \dots, w_n \triangleright t_n\}$.
- A formal **equation** on \mathcal{A}^n is a pair $M_1 \bowtie M_2$ where $M_1, M_2 \in \mathcal{F}_{\text{pub}}[w_1, \dots, w_n]$ are terms built upon special constants w_i .

Static equivalence (algebraic definition)

- Consider the tuples $\varphi = (t_1, \dots, t_n)$ in \mathcal{A}^n , also called *frames* and written $\varphi = \{w_1 \triangleright t_1, \dots, w_n \triangleright t_n\}$.
- A formal **equation** on \mathcal{A}^n is a pair $M_1 \bowtie M_2$ where $M_1, M_2 \in \mathcal{F}_{\text{pub}}[w_1, \dots, w_n]$ are terms built upon special constants w_i .

Definition

Two frames φ_1 and φ_2 in \mathcal{A}^n are *statically equivalent* (from [Abadi and Fournet, 2001]), written $\varphi_1 \approx \varphi_2$, iff

$$\text{eq}(\varphi_1) = \text{eq}(\varphi_2)$$

where $\text{eq}(\varphi) = \{M_1 \bowtie M_2 \mid M_1\varphi =_{\mathcal{A}} M_2\varphi\}$.

A mathematical example

Example

Let $n = 1$, $\mathcal{A} = \mathbb{C}$ and the terms $M \in \mathbb{Q}[w_1]$ be rational polynomials with single variable w_1 . We have $\varphi_1 \approx \varphi_2$ iff φ_1 and φ_2 are both *transcendental* or are *conjugated elements* (i.e. have the same minimal polynomial over \mathbb{Q}).

For instance, $\pi \approx e$ and $\sqrt{2} \approx -\sqrt{2}$.

We are currently investigating further links with the fundamentals of algebraic geometry. (Ask me for more details!)

Back to logics and security protocols I

- We are interested in modeling **cryptographic messages** : we let \mathcal{A} be an \mathcal{F} -algebra of ground terms taken modulo an equational theory \mathbf{E} , where $\mathcal{F}_{\text{pub}} \subsetneq \mathcal{F}$.
- Typically, the symbols in $\mathcal{F} - \mathcal{F}_{\text{pub}}$ are free constants modeling **secret keys** or random numbers.
- \mathbf{E} is generated by a finite set of equations modeling the **cryptographic primitives**.

Back to logics and security protocols II

- Static equivalence models **indistinguishability between messages** from an attacker's point of view.
- Another classical problem is **deducibility** :

Given $\varphi \in \mathcal{A}^n$ and $t \in \mathcal{A}$, does there exist $M \in \mathcal{F}_{\text{pub}}[w_1, \dots, w_n]$ such that $M\varphi =_{\mathcal{A}} t$?

N.B. Such an M is often called a *recipe* of t .

Example : deterministic symmetric encryption

- $M \in \mathcal{F}_{\text{pub}}[w_1, \dots, w_n]$ *(recipes)*
 $::= w_i \mid \text{enc}(M_1, M_2) \mid \text{dec}(M_1, M_2)$
- $t \in \mathcal{F}[\emptyset] ::= k_j \mid \text{enc}(t_1, t_2) \mid \text{dec}(t_1, t_2)$ *(plain terms)*
- Let E be generated by $\text{dec}(\text{enc}(x, y), y) = x$.
- Consider $\varphi_1 = \{w_1 \triangleright \text{enc}(k_1, k_2), w_2 \triangleright k_2\}$ *(frames)*
 and $\varphi_2 = \{w_1 \triangleright \text{enc}(k_1, k_2), w_2 \triangleright k_3\}$.
- We have $\varphi_1 \not\approx_E \varphi_2$ *(φ_1, φ_2 not E -equivalent)*
 because $\text{enc}(\text{dec}(w_1, w_2), w_2)\varphi_1 =_E w_1\varphi_1$
 but $\text{enc}(\text{dec}(w_1, w_2), w_2)\varphi_2 \neq_E w_1\varphi_2$

Equational approach to security protocols I

- Similar equational settings used in popular specification languages such as the **applied pi calculus** [Abadi and Fournet, 2001], or **Proverif's** language [Blanchet, 2001, Blanchet et al., 2008].
- Studying full protocols requires a more general notion of **observational equivalence**.

Equational approach to security protocols II

- Proof techniques for observational equivalence include
 - labelled bisimulations built on the top of static equivalence [Abadi and Fournet, 2001],
 - and symbolic semantics based on a generalization of static equivalence [Baudet, 2005, Delaune et al., 2007].
- Static equivalence also applied to characterize guessing attacks [Corin et al., 2004, Baudet, 2005]
- Correspondance between static equivalence and cryptographic (a.k.a. computational) indistinguishability investigated in several papers, e.g. [Abadi et al., 2006].

More equational theories I

- More involved examples of cryptographic equational theories include (see e.g. [Cortier et al., 2006])
 - public-key encryption : $\text{pdec}(\text{penc}(x, \text{pub}(y), z), y) = x$
 - signatures : $\text{checksign}(\text{sign}(x, y), \text{pub}(y)) = \text{ok}$

More equational theories I

- More involved examples of cryptographic equational theories include (see e.g. [Cortier et al., 2006])
 - public-key encryption : $\text{pdec}(\text{penc}(x, \text{pub}(y), z), y) = x$
 - signatures : $\text{checksign}(\text{sign}(x, y), \text{pub}(y)) = \text{ok}$
 - XOR symbol : $AC[\oplus] \quad x \oplus x = 0$
 - XOR-homomorphic symbols : $h(x \oplus y) = h(x) \oplus h(y)$
 - Diffie-Hellman exponents : $(g^x)^y = (g^y)^x$

More equational theories I

- More involved examples of cryptographic equational theories include (see e.g. [Cortier et al., 2006])
 - public-key encryption : $\text{pdec}(\text{penc}(x, \text{pub}(y), z), y) = x$
 - signatures : $\text{checksign}(\text{sign}(x, y), \text{pub}(y)) = \text{ok}$
 - XOR symbol : $AC[\oplus] \quad x \oplus x = 0$
 - XOR-homomorphic symbols : $h(x \oplus y) = h(x) \oplus h(y)$
 - Diffie-Hellman exponents : $(g^x)^y = (g^y)^x$
 - pair-homomorphic encryption :
 - ... $\text{enc}(\langle x, y \rangle, z) = \langle \text{enc}(x, z), \text{enc}(y, z) \rangle$
 - prefix-homomorphic encryption :
 - ... $\text{pref}(\text{enc}(\langle x, y \rangle, z)) = \text{enc}(x, z)$
 - blind signatures : $\text{checksign}(\text{sign}(x, y), \text{pub}(y)) = \text{ok}$
 $\text{unblind}(\text{blind}(x, y), y) = x$
 $\text{unblind}(\text{sign}(\text{blind}(x, y), z), y) = \text{sign}(x, z)$

More equational theories II

- Each of these theories yields new deduction and static-equivalence problems to decide.
- So far the only applicable tool to static equivalence has been **Proverif** [Blanchet et al., 2008], but it does not make use of the **specialized, existing decision procedures** for static equivalence [Abadi and Cortier, 2006, Cortier and Delaune, 2007].

Our contributions

Focusing on theories E generated by **convergent** rewrite systems \mathcal{R} :

- We present a **uniform procedure** for deducibility and static equivalence, that is
 - sound and complete, up to explicit failure cases,
 - provably non failing on a syntactic class of theories called *layered*,
 - “as much terminating as possible” in non-failing cases (termination implied by finite representation of deducible terms).
- We provide an efficient Ocaml implementation :
<http://www.lsv.ens-cachan.fr/~baudet/yapa/>

Content of the talk

1 Motivations

- Why study static equivalence?
- Why a new tool?

2 Results

- Overview of the procedure
- Examples
- Proving termination and non-failure

3 Conclusion

Overview of the procedure I

- We saturate a set of *deduction facts* $\Phi = \{M_i \triangleright t_i\}$ and a set of *visible equations* $\Psi = \{\forall \bar{x}. M_j \bowtie N_j\}$ by means of transformation rules $st \Longrightarrow st'$.
- The initial state $\text{Init}(\varphi)$ is (roughly) $(\Phi_0, \Psi_0) \simeq (\varphi \downarrow_{\mathcal{R}}, \emptyset)$.
- The final state is either \perp (failure) or a *saturated* state (Φ_1, Ψ_1) (success).

Overview of the procedure II

- Saturated states are **finite syntactic representations** of the sets of deducible terms and equations of the initial frame φ .

Theorem (soundness and completeness)

If $\text{Init}(\varphi) \Longrightarrow^* (\Phi, \Psi)$ is saturated, then

- For all recipes M and ground terms t ,
 $M\varphi =_{\text{E}} t \Leftrightarrow \exists N \text{ s.t. } \Psi \vdash M \bowtie N \text{ and } N \triangleright_{\Phi} t \downarrow_{\mathcal{R}}$
- For all recipes M and N ,
 $M\varphi =_{\text{E}} N\varphi \Leftrightarrow \Psi \vdash M \bowtie N$.

where $M \triangleright_{\Phi} t \Leftrightarrow \exists C, \{M_i \triangleright t_i\} \subseteq \Phi, \begin{cases} M = C[M_1, \dots, M_n] \\ t = C[t_1, \dots, t_n] \end{cases}$.

Overview of the procedure III

From saturated states $\text{Init}(\varphi_i) \Longrightarrow^* (\Phi_i, \Psi_i)$, it is easy to deduce procedures to check whether

(i) t is deducible from φ_1 , that is :

$$t \downarrow_{\mathcal{R}} \in \mathcal{F}_{\text{pub}}[\text{im}(\Phi_1)]$$

(ii) $\text{eq}_E(\varphi_1) \subseteq \text{eq}_E(\varphi_2)$, that is :

$$\text{for all } (\forall \bar{x}. M \bowtie N) \in \Psi_1, \quad (M_{\varphi_2}) \downarrow_{\mathcal{R}} = (N_{\varphi_2}) \downarrow_{\mathcal{R}}.$$

Simple example

Let $\mathcal{R} = \{\text{dec}(\text{enc}(x, y), y) \rightarrow x\}$
 and $\varphi_1 = \{w_1 \triangleright \text{enc}(k_1, k_2), w_2 \triangleright k_2\}$.

Deductions steps for saturating φ_1 :

$$\frac{w_1 \triangleright \text{enc}(k_1, k_2) \quad w_2 \triangleright k_2}{\text{dec}(w_1, w_2) \triangleright k_1} \quad \frac{}{\forall x, y. \text{dec}(\text{enc}(x, y), y) \bowtie x}$$

$$\frac{\text{dec}(w_1, w_2) \triangleright k_1 \quad w_2 \triangleright k_2 \quad w_1 \triangleright \text{enc}(k_1, k_2)}{\text{enc}(\text{dec}(w_1, w_2), w_2) \bowtie w_1}$$

Less simple example (with apologies for the wrong definition of $\mathcal{R}_{\text{blind}}$ in the proc.)

$$\text{Let } \mathcal{R} = \left\{ \begin{array}{l} \text{checksign}(\text{sign}(x, y), \text{pub}(y)) \rightarrow \text{ok} \\ \text{unblind}(\text{blind}(x, y), y) \rightarrow x \\ \text{unblind}(\text{sign}(\text{blind}(x, y), z), y) \rightarrow \text{sign}(x, z) \end{array} \right\}$$

and $\varphi_1 = \{w_1 \triangleright \text{blind}(k, r), w_2 \triangleright r\}$.

(1) Trivial equations :

$$\overline{\forall x, y. \text{checksign}(\text{sign}(x, y), \text{pub}(y)) \bowtie \text{ok}}$$

$$\overline{\forall x, y. \text{unblind}(\text{blind}(x, y), y) \bowtie x}$$

$$\overline{\forall x, y. \text{unblind}(\text{sign}(\text{blind}(x, y), z), y) \bowtie \text{sign}(x, z)}$$

Less simple example (with apologies for the wrong definition of $\mathcal{R}_{\text{blind}}$ in the proc.)

$$\text{Let } \mathcal{R} = \left\{ \begin{array}{l} \text{checksign}(\text{sign}(x, y), \text{pub}(y)) \rightarrow \text{ok} \\ \text{unblind}(\text{blind}(x, y), y) \rightarrow x \\ \text{unblind}(\text{sign}(\text{blind}(x, y), z), y) \rightarrow \text{sign}(x, z) \end{array} \right\}$$

and $\varphi_1 = \{w_1 \triangleright \text{blind}(k, r), w_2 \triangleright r\}$.

(2) Failure case that must be postponed :

$$\frac{w_1 \triangleright \text{blind}(k, r) \quad w_2 \triangleright r}{\text{unblind}(\text{sign}(w_1, z), w_2) \triangleright \text{sign}(k, z)}$$

Rationals for failure cases

In YAPA, accumulated deduction facts $M \triangleright t$ must be ground. Only equations $\forall x, y. M \bowtie N$ may use quantifiers.

Less simple example (with apologies for the wrong definition of $\mathcal{R}_{\text{blind}}$ in the proc.)

$$\text{Let } \mathcal{R} = \left\{ \begin{array}{l} \text{checksign}(\text{sign}(x, y), \text{pub}(y)) \rightarrow \text{ok} \\ \text{unblind}(\text{blind}(x, y), y) \rightarrow x \\ \text{unblind}(\text{sign}(\text{blind}(x, y), z), y) \rightarrow \text{sign}(x, z) \end{array} \right\}$$

and $\varphi_1 = \{w_1 \triangleright \text{blind}(k, r), w_2 \triangleright r\}$.

(3) Other deduction steps

$$\frac{w_1 \triangleright \text{blind}(k, r) \quad w_2 \triangleright r}{\text{unblind}(w_1, w_2) \triangleright k}$$

$$\frac{\text{unblind}(w_1, w_2) \triangleright k \quad w_2 \triangleright r \quad w_1 \triangleright \text{blind}(k, r)}{\text{blind}(\text{unblind}(w_1, w_2), w_2) \bowtie w_1}$$

Less simple example (with apologies for the wrong definition of $\mathcal{R}_{\text{blind}}$ in the proc.)

$$\text{Let } \mathcal{R} = \left\{ \begin{array}{l} \text{checksign}(\text{sign}(x, y), \text{pub}(y)) \rightarrow \text{ok} \\ \text{unblind}(\text{blind}(x, y), y) \rightarrow x \\ \text{unblind}(\text{sign}(\text{blind}(x, y), z), y) \rightarrow \text{sign}(x, z) \end{array} \right\}$$

and $\varphi_1 = \{w_1 \triangleright \text{blind}(k, r), w_2 \triangleright r\}$.

(4) Failure case solved!

$$\frac{w_1 \triangleright \text{blind}(k, r) \quad w_2 \triangleright r \quad \text{unblind}(w_1, w_2) \triangleright k}{\forall z. \text{unblind}(\text{sign}(w_1, z), w_2) \bowtie \text{sign}(\text{unblind}(w_1, w_2), z)}$$

Proving non-failure

We have formalized these observations and defined a general class of non-failing theories, called *layered*.

A syntactic criterion for non-failure

Definition (Layered rewrite system – simplified)

There exist subsystems $\emptyset = \mathcal{R}_0 \subseteq \mathcal{R}_1 \subseteq \dots \subseteq \mathcal{R}_N = \mathcal{R}$ such that for every $0 \leq i < N$, for every rule $l \rightarrow r$ in $\mathcal{R}_{i+1} - \mathcal{R}_i$, for every $l = D[l_1, \dots, l_n, x_1, \dots, x_m]$, either

(i) $\text{var}(r) \subseteq \text{var}(l_1, \dots, l_n)$, or

(ii) there exists C such that

$$C[l_1, \dots, l_n, x_1, \dots, x_m] \xrightarrow{\leq 1} \mathcal{R}_i r.$$

Proposition

The procedure never fails on layered convergent theories.

Note that the union of two layered systems is layered.

Termination

- Proving **termination by hand** for one theory is generally easy by standard techniques.
- We provide a semantic criterion ((ii) below) as well.

Proposition

Assume $\text{Init}(\varphi) \not\Rightarrow^* \perp$. The following are equivalent :

- (i) There exists a **saturated state** $\text{Init}(\varphi) \Rightarrow^* (\Phi, \Psi)$.
- (ii) There exists a **finite set of deducible terms** S such that $\mathcal{F}_{\text{pub}}[S]$ covers every deducible \mathcal{R} -normal term.
- (iii) There exists no **fair infinite derivation** from $\text{Init}(\varphi)$.

Content of the talk

1 Motivations

- Why study static equivalence?
- Why a new tool?

2 Results

- Overview of the procedure
- Examples
- Proving termination and non-failure

3 Conclusion

Supported theories

- Altogether, using [Abadi and Cortier, 2006], we deduce **termination and non-failure** for

- subterm convergent theories \mathcal{R} :

$$\forall l \rightarrow r \text{ in } \mathcal{R}, r \in \text{st}(l) \cup \mathcal{F}[\emptyset] \downarrow_{\mathcal{R}}$$

- pair-homomorphic encryption :

$$\dots \quad \text{enc}(\langle x, y \rangle, z) = \langle \text{enc}(x, z), \text{enc}(y, z) \rangle$$

- prefix-homomorphic encryption (*new*) :

$$\dots \quad \text{pref}(\text{enc}(\langle x, y \rangle, z)) = \text{enc}(x, z)$$

- blind signatures : $\text{checksign}(\text{sign}(x, y), \text{pub}(y)) = \text{ok}$

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

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

- a simple theory of addition : $\text{plus}(x, \text{s}(y)) = \text{plus}(\text{s}(x), y)$

$$\text{plus}(x, \text{zero}) = x$$

$$\text{pred}(\text{s}(x)) = x$$

Benchmarks

- We have tested the tool on a few examples and obtained good results, usually faster than Proverif.
- This is not surprising as static equivalence is an easier problem than the (in)security of protocols as studied by Proverif.

Summary

- We have proposed a **unifying approach** to study “intruder knowledge” for convergent theories.
- Many equational theories are **provably and efficiently** supported by the tool YAPA.

Perspectives

- Better comparison with Proverif and with the recent work of [Ciobâcă et al., 2009], which both allow non-ground deduction facts.
- More complex theories e.g. including a XOR symbol (see combination theorem of [Arnaud et al., 2007]).
- More complex algebraic properties, for instance checking whether $\text{eq}(\varphi_1) \cap \text{eq}(\varphi_2) \subseteq \text{eq}(\varphi_3)$.
- Active case, ideally to generalize [Baudet, 2005].

Thank you !

Erratum : Blind signatures

To see that the (corrected) theory of blind signatures is layered, let

$$\mathcal{R}_1 = \left\{ \begin{array}{l} \text{checksign}(\text{sign}(x, y), \text{pub}(y)) \rightarrow \text{ok} \\ \text{unblind}(\text{blind}(x, y), y) \rightarrow x \end{array} \right\}$$

$$\mathcal{R}_2 = \mathcal{R}_1 \cup \{ \text{unblind}(\text{sign}(\text{blind}(x, y), z), y) \rightarrow \text{sign}(x, z) \}$$

Link with algebraic geometry (ongoing work)

- Generalize equations by allowing disjunctions :

$$F ::= \bigvee_{i=1}^n (M_1 \bowtie M_2) \in \mathcal{P}_{\text{fin}}(\mathcal{F}_{\text{pub}}[w_1, \dots, w_n]^2)$$

- $\text{formulas}(\Phi) = \{F \mid \forall \varphi \in \Phi, \varphi \models F\}$ (*≈ radical ideal*)
- $\text{points}(I) = \{\varphi \mid \forall F \in I, \varphi \models F\}$ (*≈ algebraic set*)
- $\Phi \subseteq \overline{\Phi} = \text{points}(\text{formulas}(\Phi))$ (*≈ algebraic closure,*
→ Zariski topology)
- $\varphi_1 \approx \varphi_2$ iff $\text{formulas}(\{\varphi_1\}) = \text{formulas}(\{\varphi_2\})$.

References I



Abadi, M., Baudet, M., and Warinschi, B. (2006).
 Guessing attacks and the computational soundness of static equivalence.
 In [Foundations of Software Science and Computation Structures \(FOSSACS'06\)](#), pages 398–412.



Abadi, M. and Cortier, V. (2006).
 Deciding knowledge in security protocols under equational theories.
[Theoretical Computer Science](#), 387(1-2) :2–32.



Abadi, M. and Fournet, C. (2001).
 Mobile values, new names, and secure communication.
 In [28th ACM Symposium on Principles of Programming Languages \(POPL'01\)](#), pages 104–115. ACM.



Anantharaman, S., Narendran, P., and Rusinowitch, M. (2007).
 Intruders with caps.
 In [18th International Conference on Term Rewriting and Applications \(RTA'07\)](#), volume 4533 of [LNCS](#). Springer.



Arnaud, M., Cortier, V., and Delaune, S. (2007).
 Combining algorithms for deciding knowledge in security protocols.
 In [Proc. 6th International Symposium on Frontiers of Combining Systems \(FroCoS'07\)](#), volume 4720 of [Lecture Notes in Artificial Intelligence](#), pages 103–117. Springer.



Baudet, M. (2005).
 Deciding security of protocols against off-line guessing attacks.
 In [12th ACM Conference on Computer and Communications Security \(CCS'05\)](#), pages 16–25. ACM Press.



Baudet, M. (2007).
[Sécurité des protocoles cryptographiques : aspects logiques et calculatoires.](#)
 Thèse de doctorat, LSV, ENS Cachan, France.

References II



Baudet, M., Cortier, V., and Kremer, S. (2005).

Computationally sound implementations of equational theories against passive adversaries.

In [32nd International Colloquium on Automata, Languages and Programming \(ICALP'05\)](#), volume 3580 of [LNCS](#), pages 652–663. Springer.



Blanchet, B. (2001).

An Efficient Cryptographic Protocol Verifier Based on Prolog Rules.

In [14th Computer Security Foundations Workshop \(CSFW'01\)](#), pages 82–96. IEEE Comp. Soc. Press.



Blanchet, B., Abadi, M., and Fournet, C. (2008).

Automated verification of selected equivalences for security protocols.

[Journal of Logic and Algebraic Programming](#), 75(1) :3–51.



Chevalier, Y., Küsters, R., Rusinowitch, M., and Turuani, M. (2003).

An NP decision procedure for protocol insecurity with XOR.

In [18th IEEE Symposium on Logic in Computer Science \(LICS'03\)](#). IEEE Comp. Soc. Press.



Ciobâcă, Ș., Delaune, S., and Kremer, S. (2009).

Computing knowledge in security protocols under convergent equational theories.

In [Proc. 22nd International Conference on Automated Deduction \(CADE'09\)](#), Lecture Notes in Artificial Intelligence. Springer.

To appear.



Comon-Lundh, H. and Shmatikov, V. (2003).

Intruder deductions, constraint solving and insecurity decision in presence of exclusive or.

In [18th IEEE Symposium on Logic in Computer Science \(LICS'03\)](#). IEEE Comp. Soc. Press.

References III



Corin, R., Doumen, J., and Etalle, S. (2004).

Analysing password protocol security against off-line dictionary attacks.

In [2nd International Workshop on Security Issues with Petri Nets and other Computational Models \(WISP'04\)](#), ENTCS.



Cortier, V. and Delaune, S. (2007).

Deciding knowledge in security protocols for monoidal equational theories.

In [14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning \(LPAR'07\)](#), LNAI. Springer.



Cortier, V., Delaune, S., and Lafourcade, P. (2006).

A survey of algebraic properties used in cryptographic protocols.

[Journal of Computer Security](#), 14(1) :1–43.



Delaune, S. and Jacquemard, F. (2004).

A decision procedure for the verification of security protocols with explicit destructors.

In [11th ACM Conference on Computer and Communications Security \(CCS'04\)](#), pages 278–287.



Delaune, S., Kremer, S., and Ryan, M. D. (2007).

Symbolic bisimulation for the applied pi-calculus.

In [Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science \(FSTTCS'07\)](#), volume 4855 of LNCS, pages 133–145. Springer.



Delaune, S., Kremer, S., and Ryan, M. D. (2008).

Verifying privacy-type properties of electronic voting protocols.

[Journal of Computer Security](#).

To appear.

References IV



Lowe, G. (1996).

Breaking and fixing the Needham-Schroeder public-key protocol using FDR.

In Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96), volume 1055 of LNCS, pages 147–166. Springer-Verlag.



Millen, J. and Shmatikov, V. (2001).

Constraint solving for bounded-process cryptographic protocol analysis.

In 8th ACM Conference on Computer and Communications Security (CCS'01).