Algorithmes pour l'équivalence statique

Ştefan Ciobâcă Véronique Cortier

September 1, 2009

Les résultats présentés dans ce rapport ont été obtenus par Mathieu Baudet, Narjes Ben Rajeb, Mouhebeddine Berrima, Ştefan Ciobâcă, Véronique Cortier, Stéphanie Delaune et Steve Kremer.

Abstract

Les propriétés de sécurité pour les protocoles de vote électroniques s'expriment de manière naturelle à l'aide d'*équivalences* entre des processus. Une brique de base pour traiter ces propriétés est de savoir décider l'*équivalence statique*. Avant le début du projet AVOTÉ, plusieurs algorithmes avaient été proposés pour déterminer si des séquences de messages sont statiquement équivalentes mais aucun d'entre eux ne permettaient de considérer les primitives propres aux protocoles de vote comme le rechiffrement ou les fonctions de commitment.

Dans ce rapport, nous rappelons la définition de l'équivalence statique ainsi qu'un bref état de l'art sur les résultats existants avant le début du projet. Puis nous décrivons les différents algorithmes proposés au sein du projet. Plusieurs d'entre eux ont déjà été implémentés.

1 Introduction

Avant même de considérer l'interaction d'un attaquant avec un protocole, il est nécessaire de d'évaluer la connaissance qu'un intrus obtient à partir d'un ensemble de message. La façon la plus classique de modéliser la connaissance est d'utiliser une notion de *déduction* de l'intrus : à partir d'un ensemble de messages S, quels sont les messages déductibles (constructibles) de S? Cette relation, notée $\vdash$, peut être définie pour de nombreuses primitives cryptographiques. Cette notion ne reflète cependant pas toute la connaissance accessible par un intrus. Prenons l'exemple de l'expression d'un vote et supposons qu'un votant envoie le message $\{i\}_{\mathsf{pub}(S)}$ à un serveur où $i \in \{0,1\}$ représente la valeur de son vote. Un attaquant peut alors connaître la valeur du vote, non pas en déchiffrant le message, mais en construisant les messages $\{0\}_{\mathsf{pub}(S)}$ et $\{1\}_{\mathsf{pub}(S)}$ et en comparant les deux chiffrés au message envoyé (à condition que le chiffrement soit déterministe).

Pour refléter la capacité de comparaison d'un intrus, Martin Abadi et Cédric Fournet ont introduit [AF01] la notion d'*équivalence statique* notée $\approx$. Deux ensembles de messages sont dits équivalents statiquement si un attaquant ne peut pas les différencier, c'est-à-dire s'il ne peut pas construire de test qui les différencie. Cette notion est proche de la notion d'*indistinguabilité* très utilisée en cryptographie [GM84]. Certains travaux récents

établissent sous quelles conditions l'équivalence statique de messages symboliques implique l'indistingabilité des distributions associées [BCK09].

Nous rappelons la définition formelle des notions de déduction et équivalence statique à la partie 2 puis nous rappelons quelques résultats existants dans les parties 3 4, 5 et 6. Cependant, ces résultats ne permettent pas de traiter la plupart des théories équationelles pertinentes pour les protocoles de vote, que nous identifions à la partie 7. La suite du rapport est ensuite consacrée à la présentation des résultats obtenus au sein du projet. À la partie 8, nous proposons deux nouveaux résultats de décidabilité pour deux théories équationelles particulières aux protocoles de vote électronique: la théorie pour le rechiffrement et la preuve à vérificatuer désigné ainsi que la théorie du "trapdoor bit-commitment". Dans les parties suivantes, nous avons proposé deux algorithmes génériques pour décider l'équivalence statique. Chacun de ces deux algortithmes a été implémenté. Ainsi, nous avons proposé un premier prototype (partie 9) permettant de décider automatiquement l'équivalence statique pour une classe de théories équationelles couvrant la plupart des primitives habituelles ainsi que la signature en aveugle. L'outil correspondant est YAPA (Yet Another Protocol Analyzer). À la partie 10, nous proposons une nouvelle procédure générique pour l'équivalence statique dans le cas des théories équationelles convergentes. La procédure termine pour certaines théories comme les théories sous-terme convergentes, la théorie du "trapdoor bit-commitment" ainsi que la signature en aveugle. La procedure est implementée dans l'outil KiSs (Knowledge in Security protocols). Les performances des deux outils sont comparées à la partie 11. Enfin, nous discutons du bilan et des perspectives de nos résultats à la partie 12.

2 Définitions

Nous considérons une signature $\mathcal{F}$, un ensemble de variables $\mathcal{X}$ et un ensemble de noms $\mathcal{N}$, munis d'une théorie équationnelle E. Les séquences de messages $M_1, \ldots, M_l$ sont organisées en *structure* (« frame » en anglais) de la forme $\phi = \nu\tilde{n}.\sigma$ où $\tilde{n}$ est un ensemble fini de noms dits *restreints* (initialement inconnus de l'intrus) et σ est une substitution de la forme

$$\sigma = \{^{M_1}/_{x_1}, \ldots, ^{M_l}/_{x_l}\}.$$

Les variables x_i peuvent être vues comme des pointeurs sur les messages M_i. L'opérateur ν est l'opérateur de restriction du pi calcul [Mil99]. La taille d'une structure $\phi = \nu\tilde{n}.\{^{M_1}/_{x_1}, \ldots, ^{M_l}/_{x_l}\}$ est $|\phi| = \sum_{i=1}^{l} |M_i|$. Les noms $\tilde{n}$ sont liés dans ϕ et peuvent être renommés.

2.1 Déduction

Étant donnée une théorie équationnelle E, la notion de déduction est définie de manière canonique par les règles suivantes:

$$\frac{}{\nu\tilde{n}.\sigma \vdash_E M} \text{ si } \exists x \in \mathsf{dom}(\sigma) \text{ tq. } x\sigma = M \qquad \frac{}{\nu\tilde{n}.\sigma \vdash_E s} \; s \notin \tilde{n}$$

$$\frac{\phi \vdash_E M_1 \quad \cdots \quad \phi \vdash_E M_k}{\phi \vdash_E f(M_1, \ldots, M_k)} \; f \in \Sigma \qquad \frac{\phi \vdash_E M \quad M =_E M'}{\phi \vdash_E M'}$$

Intuitivement, les messages déductibles de ϕ sont les messages de ϕ ainsi que les noms non restreints de ϕ, clos par égalité dans E et par application de symboles fonctionnels.

La proposition suivante permet d'introduire la notion de *recette* associée à un terme déductible.

Proposition 1 ([AC04]) *Soit M un terme clos et $\nu\tilde{n}.\sigma$ une structure. Alors $\nu\tilde{n}.\sigma \vdash_E M$ si et seulement si il existe un terme ζ tel que $\mathsf{noms}(\zeta) \cap \tilde{n} = \emptyset$ and $\zeta\sigma =_E M$. Le terme ζ est appelé* recette *associée à M.*

Un terme déductible peut admettre plusieurs recettes.

Considérons la théorie E_{enc}, induite par les équations ci-dessous :

$$E_{\mathsf{enc}} = \{\mathsf{dec}(\mathsf{enc}(x,y),y) = x, \quad \mathsf{fst}(\mathsf{pair}(x,y)) = x, \quad \mathsf{snd}(\mathsf{pair}(x,y)) = y\}.$$

Soit ϕ la structure $\nu\{k,s\}.\{\mathsf{enc}(s,k)/x, k/y\}$. Alors $\phi \vdash_{E_{\mathsf{enc}}} k$ et $\phi \vdash_{E_{\mathsf{enc}}} s$. De plus, une recette pour k est y puisque $k =_{E_{\mathsf{enc}}} y\phi$ et une recette pour s est $\mathsf{dec}(x,y)$ puisque $s =_{E_{\mathsf{enc}}} \mathsf{dec}(x,y)\phi$.

2.2 Équivalence statique

Deux termes M et N sont égaux dans la structure φ pour la théorie équationnelle E, ce qui sera noté $(M =_E N)\varphi$, si et seulement si $\varphi = \nu\tilde{n}.\sigma$, $M\sigma =_E N\sigma$ et $\{\tilde{n}\} \cap (\mathsf{noms}(M) \cup \mathsf{noms}(N)) = \emptyset$ pour un choix de $\tilde{n}$ et d'une substitution σ.

Deux structures φ et ψ sont *statiquement équivalentes*, noté $\varphi \approx_E \psi$, si $\mathsf{dom}(\varphi) = \mathsf{dom}(\psi)$ et si, pour tous termes M et N, la propriété suivante est vérifiée:

$$(M =_E N)\varphi \;\Leftrightarrow (M =_E N)\psi.$$

Intuitivement, φ et ψ sont statiquement équivalentes si elles vérifient les mêmes égalités.

Considérons par exemple la structure $\phi_1 \stackrel{\mathsf{def}}{=} \nu k.\{\mathsf{enc}(0,k)/x, k/y\}$ correspondant au cas où un votant vote 0 à l'aide de la clef k ainsi que la structure $\phi_2 \stackrel{\mathsf{def}}{=} \nu k.\{\mathsf{enc}(1,k)/x, k/y\}$, correspondant au cas où un votant vote 1 à l'aide de la clef k. Les valeurs 0 et 1 sont des symboles constants donc connus de l'intrus. ϕ_1 satisfait l'égalité $(\mathsf{dec}(x,y) =_{E_{\mathsf{enc}}} 0)\phi_1$ ce qui n'est pas le cas de ϕ_2: $(\mathsf{dec}(x,y) \neq_{E_{\mathsf{enc}}} 0)\phi_2$. On en déduit $\phi_1 \not\approx_{E_{\mathsf{enc}}} \phi_2$ alors que $\nu k.\{\mathsf{enc}(0,k)/x\} \approx_{E_{\mathsf{enc}}} \nu k.\{\mathsf{enc}(1,k)/x\}$.

2.3 Comparaison des deux notions

Intuitivement, l'équivalence statique est plus forte que la déduction. Ceci est vrai dès que la théorie équationnelle contient un symbole fonctionnel libre.

Proposition 2 ([AC04, AC06]) *Soit E une théorie équationnelle associée à une signature $\mathcal{F}$. Soit $\mathcal{F}' \stackrel{\mathsf{def}}{=} \mathcal{F} \uplus \{h\}$, où h est un symbole unaire. Soit E' la plus petite théorie équationnelle étendant E aux termes de $\mathcal{F}'$. Soit $\phi = \nu\tilde{n}.\{{}^{x_1}/_{M_1}, \ldots, {}^{x_l}/_{M_l}\}$ une structure sur $\mathcal{F}$, M un terme clos sur $\mathcal{F}$ et k un nom frais. Alors $\phi \vdash_E M$ si et seulement si*

$$\nu\tilde{n}.\{{}^{M_1}/_{x_1}, \ldots, {}^{M_l}/_{x_l}, {}^{h(M)}/_{x_{l+1}}\} \not\approx_{E'} \nu(\tilde{n}. \cup \{k\})\{{}^{M_1}/_{x_1}, \ldots, {}^{M_l}/_{x_l}, {}^{k}/_{x_{l+1}}\}$$

Nous pouvons en déduire que si $\approx_{E'}$ est décidable, alors $\vdash_E$ est aussi décidable (avec au plus la même complexité). Cependant, l'existence d'un symbole libre (ou d'une construction jouant le même rôle, comme le chiffrement par exemple) est important pour la réduction de la déduction à l'équivalence statique. Ainsi, si on considère uniquement la théorie AC pure E_{AC} (définie

page 5) alors $\approx_{E_{\mathsf{AC}}}$ est décidable en temps polynomial alors que $\vdash_{E_{\mathsf{AC}}}$ est NP-complet, ce qui montre que la réduction proposée à la proposition 2 n'est pas toujours possible.

À l'inverse, il est possible de construire une théorie équationnelle E telle que $\vdash_E$ est décidable alors que $\approx_E$ ne l'est pas. Une première construction avait été proposée dans [AC04] avec une esquisse de preuve, une preuve complète a été proposée par Borgström [Bor05].

La suite de ce rapport est consacrée à la caractérisation de théories équationnelles E pour lesquelles les relations $\vdash_E$ et $\approx_E$ sont décidables, en particulier en ce qui concerne les théories équationelles pertinentes pour les protocoles de vote électronique.

3 Cas des théories sous-termes

Une première classe de théories équationnelles pour lesquelles déduction et équivalence statique sont décidables est la classe des théories sous-termes convergentes.

Définition 1 (Théories sous-termes) *Une théorie E est dite* sous-terme *si elle peut être définie par un ensemble fini d'équations de la forme $M = N$ où N est un sous-terme de M ou une constante.*

Une théorie E est sous-terme convergente si E est sous-terme et convergente.

Ainsi, la théorie E_{enc} est sous-terme convergente. C'est également le cas des trois théories définies ci-dessous:

$$\begin{array}{ll} E_{\mathsf{inv}} : & \{I(I(x)) = x, I(x) \times x = 1, x \times I(x) = 1\} \\ E_{\mathsf{idem}} : & \{h(h(x)) = h(x)\} \\ E_{\mathsf{sym}} : & \{\mathsf{enc}(\mathsf{enc}(x, y), y) = x\} \end{array}$$

La théorie E_{inv} modélise la fonction inverse dans les groupes par exemple. La théorie E_{idem} représente une fonction de hachage idempotente sur des entrées de petite taille (puisque le hachage d'un message haché $h(m)$ produit à nouveau $h(m)$). La théorie E_{sym} représente une fonction de chiffrement qui permet également de déchiffrer.

Théorème 1 ([AC04, AC06]) *Soit E une théorie sous-terme convergente. Les deux problèmes $\phi \vdash M$ et $\phi \approx \phi'$ sont décidables en temps polynomial en la taille de ϕ, ϕ' et $|M|$.*

Ce résultat a été étendu par Mathieu Baudet [Bau05, Bau07] au cas des théories convergentes, définies par un ensemble fini d'équations de la forme $M = N$ où N est un sous-terme de M ou un *terme constant* (et non une constante).

4 Conditions suffisantes pour des théories équationnelles AC-convergentes

Soit E une théorie équationnelle. Les symboles AC de E sont les symboles binaires $\oplus_1, \ldots, \oplus_k$ tels que les équations $x \oplus_i (y \oplus_i z) = (x \oplus_i y) \oplus_i z$ (associativité) et $x \oplus_i y = y \oplus_i x$ (commutativité) sont dans E. Nous écrivons qu'une théorie E est *AC-convergente* si elle est induite par un système de réécriture $\rightarrow$ terminant et confluent modulo AC, tel que $U =_E V$ si et seulement si l'intersection de leurs formes normales modulo AC est non vide. Si E ne contient pas de symbole AC alors on retrouve la notion habituelle de convergence.

L'idée générale pour montrer la décidabilité de l'équivalence statique est de se ramener à un ensemble de *petites équations*

$$\mathsf{Eq}(\phi) = \{(M = N) \mid (M =_E N)\phi \text{ et } |M|, |N| \leqslant c\}$$

où c est une constante bien choisie et où la notion de taille ne compte pas les symboles AC: $|t_1 \oplus t_2| = \max(|t_1|, |t_2|)$. L'ensemble des petites équations $\mathsf{Eq}(\phi)$ est fini si E ne contient pas de symbole AC mais peut être infini si E contient des symboles AC.

Il a été montré [AC05, AC06] que la déduction et l'équivalence statique sont décidables pour une théorie E AC-convergente dès lors que :

1. E est localement stable: pour toute structure ϕ, on peut construire un ensemble $\mathsf{sat}(\phi)$, stable par application d'un petit contexte

   $$\forall t_1, \ldots, t_k \in \mathsf{sat}(\phi)\ \forall C \text{ tq. } |C| \leq c',\ C[t_1, \ldots, t_k] \to_{AC} C'[t'_1, \ldots, t'_n]$$

   pour des termes $t'_1, \ldots, t'_n \in \mathsf{sat}(\phi)$ et C' un (petit) contexte. La constante c' dépend de la taille de la théorie E.

2. E est localement décidable: savoir si ϕ' satisfait les équations de $\mathsf{Eq}(\phi)$ est décidable.

La condition 1 est en particulier vérifiée pour toutes les théories sous-terme convergentes. La condition 2 est toujours vérifiée pour les théories E sans symbole AC puisque $\mathsf{Eq}(\phi)$ est alors fini. Dans le cas de la théorie $E_\oplus$ du XOR avec un symbole AC $\oplus$, l'ensemble $\mathsf{Eq}(\phi)$ est également fini (modulo XOR). Dans le cas de la théorie E_{AC} AC pure, induite par les équations

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z \quad \text{et} \quad x \oplus y = y \oplus x,$$

on peut montrer qu'on peut réduire $\mathsf{Eq}(\phi)$ à un ensemble fini en calculant le noyau d'un $\mathbb{Z}$-module [Sch86]. Cette approche sera détaillée et généralisée à la partie 5.

À l'aide de ce résultat, nous retrouvons la décidabilité de la déduction et de l'équivalence statique pour les théories sous-terme convergentes. Il est également possible de montrer la décidabilité de la déduction et de l'équivalence statique pour d'autres théories non sous-terme convergentes comme les théories $E_\oplus$, E_{AC} ou E_{blind} et E_{homo} définies ci-dessous.

$$E_{\mathsf{blind}} = E_{\mathsf{enc}} \cup \left\{ \begin{array}{rcl} \mathsf{open}(\mathsf{commit}(x,y),y) & = & x \\ \mathsf{getpk}(\mathsf{host}(x)) & = & x \\ \mathsf{checksign}(\mathsf{sign}(x,y),\mathsf{pk}(y)) & = & x \\ \mathsf{unblind}(\mathsf{blind}(x,y),y) & = & x \\ \mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(x,y),z),y) & = & \mathsf{sign}(x,z) \end{array} \right\}$$

$$E_{\mathsf{homo}} = E_{\mathsf{enc}} \cup \left\{ \begin{array}{rcl} \mathsf{enc}(\mathsf{pair}(x,y),z) & = & \mathsf{pair}(\mathsf{enc}(x,z),\mathsf{enc}(y,z)) \\ \mathsf{dec}(\mathsf{pair}(x,y),z) & = & \mathsf{pair}(\mathsf{dec}(x,z),\mathsf{dec}(y,z)) \end{array} \right\}$$

La théorie E_{blind} a été introduite par S. Kremer and M. Ryan pour modéliser la signature en aveugle, primitive utilisée en particulier dans certains protocoles de vote électronique [KR05]. La théorie E_{homo} représente un schéma de chiffrement homomorphique comme le mode de chiffrement ECB (Encryption Chaining Block) par exemple, où chaque bloc est chiffré indépendemment des précédents.

D'autres exemples de théories localement stables et localement décidables (donc pour lesquelles la déduction et de l'équivalence statique sont décidables) ont été proposés dans [AC06].

5 Théories monoïdales

Les théories monoïdales, définies par W. Nutt [Nut90], regroupent de nombreuses théories avec symbole AC. Une méthode générale a été proposée [CD07] pour réduire la décidabilité de la déduction l'équivalence statique à des problèmes algébriques mieux connus.

5.1 Définitions

Définition 2 (Théorie monoïdale) *Une théorie E sur la signature $\mathcal{F}$ est dite* monoïdale *si elle satisfait les trois propriétés suivantes:*

1. *La signature $\mathcal{F}$ contient un symbole binaire $+$ ainsi qu'un symbole constant 0 et tous les autres symboles fonctionnels de $\mathcal{F}$ sont unaires.*
2. *Le symbole $+$ est associatif et commutatif, avec pour unité 0,* i.e. *les équations $x + (y + z) = (x + y) + z$ (A), $x + y = y + x$ (C) et $x + 0 = x$ (U) sont dans E.*
3. *Chaque symbole unaire $h \in \mathcal{F}$ est un endomorphisme pour $+$ et 0, i.e. $h(x + y) = h(x) + h(y)$ et $h(0) = 0$.*

Les théories ci-dessous sont monoïdales.

- La théorie ACU sur $\mathcal{F} = \{+, 0\}$ induite par les axiomes (A),(C) et (U).
- La théorie ACUI sur $\mathcal{F} = \{+, 0\}$ induite par les axiomes (A),(C), (U) et (I): $x + x = x$ (Idempotence).
- La théorie ACUN (théorie du *ou exclusif*, notée $E_{\oplus}$ jusqu'ici) induite par les axiomes (A),(C), (U) et (N): $x + x = 0$.
- La théorie AG (*groupes Abéliens*) sur $\mathcal{F} = \{+, -, 0\}$ induite par les axiomes (A),(C), (U) et $x + -(x) = 0$.
- Les théories ACUh, ACUIh, ACUNh sur $\mathcal{F} = \{+, h, 0\}$ et AGh sur $\mathcal{F} = \{+, -, h, 0\}$: ces théories correspondent aux théories précédentes, étendues par les lois d'homomorphisme: $h(x + y) = h(x) + h(y)$ et $h(0) = 0$.
- La théorie $\mathsf{AGh_1} \ldots \mathsf{h_n}$ sur $\mathcal{F} = \{+, -, h_1, \ldots, h_n, 0\}$, induite par les axiomes de AG, les lois d'homomorphisme pour chaque h_i et la commutativité des symboles unaires: $h_i(h_j(x)) = h_j(h_i(x))$ pour tout $1 \leqslant i, j \leqslant n$.

D'autres exemples de théories monoïdales sont présentés dans [Nut90].

Stéphanie Delaune a montré [Del06] que le problème de la déduction pour la théorie ACU se réduit à la résolution linéaire d'équations dans $\mathbb{N}$ tandis que le problème de la déduction pour la théorie AGh se réduit à la résolution linéaire d'équations dans $\mathbb{Z}[h]$, l'anneau des polynômes à une indéterminée, à coefficients dans $\mathbb{Z}$.

Ces résultats ont été généralisés en associant un semi-anneau à chaque théorie monoïdale. Cela permet de réduire la décision de la déduction et de l'équivalence statique à des problèmes plus classiques en algèbre. Les théories monoïdales présentent une structure algébrique proche de celles des anneaux, à ceci près que certains éléments peuvent ne pas avoir d'inverse (pour la loi de « groupe »). Une telle structure est appelée *semi-anneau*.

Définition 3 (Semi-anneau) *Un* semi-anneau *est un ensemble $\mathcal{S}$ (appelé* univers *du semi-anneau) contenant les éléments 0 et 1 et muni de deux opérations binaires $+$ et $\cdot$ telles que $(\mathcal{S}, +, 0)$ est un monoïde commutatif, $(\mathcal{S}, \cdot, 1)$ est un monoïde, et telles que les propriétés suivantes sont vérifiées pour tous $\alpha, \beta, \gamma \in \mathcal{S}$:*

- $(\alpha + \beta) \cdot \gamma = \alpha \cdot \gamma + \beta \cdot \gamma$ *(distributivité à droite)*
- $\alpha \cdot (\beta + \gamma) = \alpha \cdot \beta + \alpha \cdot \gamma$ *(distributivité à gauche)*
- $0 \cdot \alpha = \alpha \cdot 0 = 0$ *(loi du zéro).*

Les opérations $+$ et $\cdot$ sont appelées respectivement *addition* et *multiplication* du semi-anneau $\mathcal{S}$. Les éléments 0 et 1 sont respectivement le *zéro* et l'*unité*.

W. Nutt a montré [Nut90] qu'on peut associer un semi-anneau $\mathcal{S}_E$ à toute théorie monoïdale E. Le semi-anneau $\mathcal{S}_E$ est construit de la manière suivante. Son univers est $T(\mathcal{F}, \{\mathbf{1}\})/E$, où $\mathbf{1}$ est une nouvelle constante libre ($\mathbf{1} \notin \mathcal{F}$). La constante 0 est le zéro de $\mathcal{S}_E$, le symbole $+$ en est l'addition. La multiplication est définie par $s \cdot t := s[\mathbf{1} \mapsto t]$. La constante $\mathbf{1}$ agit donc comme élément neutre de la multiplication.

Nous illustrons cette définition par quelques exemples.

1. Le semi-anneau $\mathcal{S}_{\mathsf{ACU}}$ est isomorphe à $\mathbb{N}$, le semi-anneau des entiers naturels.
2. Le semi-anneau $\mathcal{S}_{\mathsf{ACUN}}$ est isomorphe au corps $\mathbb{Z}/2\mathbb{Z}$.
3. Le semi-anneau $\mathcal{S}_{\mathsf{AGh}}$ à l'anneau commutatif $\mathbb{Z}[\mathsf{h}]$.

5.2 Réduction de la déduction et de l'équivalence statique

Considérons la théorie ACU ainsi que la structure

$$\phi = \nu n_1, n_2, n_3.\{{}^{3n_1+2n_2+3n_3}/_{x_1}, {}^{n_2+3n_3}/_{x_2}, {}^{3n_2+n_3}/_{x_3}, {}^{3n_1+n_2+4n_3}/_{x_4}\},$$

où la notation kn avec $k \in \mathbb{N}$ représente le terme $n + \cdots + n$ (k fois). On peut associer à ϕ la matrice M_ϕ où $i$ème ligne représente la décomposition de x_i sur les constantes n_1, n_2, n_3.

$$M_\phi = \begin{pmatrix} 3 & 2 & 3 \\ 0 & 1 & 3 \\ 0 & 3 & 1 \\ 3 & 1 & 4 \end{pmatrix}$$

Soit $t = 7n_1 + 3n_2 + 8n_3$. On associe à t le vecteur $U_t = \begin{pmatrix} 7 & 3 & 8 \end{pmatrix}$. On remarque facilement que t est déductible à partir de ϕ si et seulement si il existe $X \in \mathbb{N}^4$ tel que $XM_\phi = U_t$.

Cette réduction peut être généralisée à toutes les théories monoïdales.

Théorème 2 ([CD07]) *Soit E une théorie monoïdale et $\mathcal{S}_E$ son semi-anneau associé. La déduction dans E se réduit en temps polynomial au problème suivant:*

Données*: Une matrice A de taille $\ell \times m$ et un vecteur de taille ℓ, à coefficients dans $\mathcal{S}_E$.*
Question*: Est-ce qu'il existe un vecteur X (à coefficients dans $\mathcal{S}_E$) tel que $X \cdot A = b$?*

Considérons maintenant la structure

$$\phi' = \nu n_1, n_2, n_3.\{^{n_1+2n_2+7n_3}/_{x_1}, ^{n_1+5n_2}/_{x_2}, ^{5n_2+8n_3}/_{x_3}, ^{3n_1+2n_2+4n_3}/_{x_4}\}$$

On lui associe la matrice $M_{\phi'}$ définie ci-dessous.

$$M_{\phi'} = \begin{pmatrix} 1 & 2 & 7 \\ 1 & 5 & 0 \\ 0 & 5 & 8 \\ 3 & 2 & 4 \end{pmatrix}$$

Les structures ϕ et ϕ' sont statiquement équivalentes si et seulement si, pour tous termes M, N de $T(\{0, +, x_1, x_2, x_3, x_4\})$,

$$(M =_{\mathsf{ACU}} N)\phi \quad \Leftrightarrow \quad (M =_{\mathsf{ACU}} N)\phi'$$

Ceci est équivalent aux relations matricielles suivantes: pour tout $X \in \mathbb{Z}^4$, $XM_\phi = 0$ si et seulement si $XM_{\phi'} = 0$. C'est-à-dire, ϕ et ϕ' sont statiquement équivalentes si et seulement si les matrices M_ϕ et $M_{\phi'}$ ont même «'noyau ».

Cette réduction se généralise à nouveau à toutes les théories monoïdales.

Théorème 3 ([CD07]) *Soit E une théorie monoïdale et $\mathcal{S}_E$ son semi-anneau associé. L'équivalence statique dans E se réduit en temps polynomial au problème suivant:*

Données*: Deux matrices A_1 et A_2 de taille $\ell \times m$ et à coefficients dans $\mathcal{S}_E$.*
Question*: Est-ce que l'égalité suivante est vérifiée?*

$$\{(X, Y) \in \mathcal{S}_E^\ell \times \mathcal{S}_E^\ell \mid X \cdot A_1 = Y \cdot A_1\} = \{(X, Y) \in \mathcal{S}_E^\ell \times \mathcal{S}_E^\ell \mid X \cdot A_2 = Y \cdot A_2\}$$

Considérons la signature $\mathcal{F}_1 = \{+, 0, -, \mathsf{h}_1, \mathsf{h}_2\}$ et la théorie E_1 induite par les axiomes de AG, par l'équation $\mathsf{h}_1(\mathsf{h}_2(x)) = \mathsf{h}_2(\mathsf{h}_1(x))$, les lois d'homomorphisme

$$\begin{array}{rclcrcl} \mathsf{h}_1(x+y) & = & \mathsf{h}_1(x) + \mathsf{h}_1(y) & \qquad & \mathsf{h}_1(0) & = & 0 \\ \mathsf{h}_2(x+y) & = & \mathsf{h}_2(x) + \mathsf{h}_2(y) & & \mathsf{h}_2(0) & = & 0 \end{array}$$

ainsi que les équations

$$\begin{array}{rcl} \mathsf{h}_1(\mathsf{h}_1(\mathsf{h}_2(x))) + \mathsf{h}_2(\mathsf{h}_2(x)) & = & 0 \\ \mathsf{h}_1(x) + \mathsf{h}_1(\mathsf{h}_2(\mathsf{h}_2(x))) & = & 0 \end{array}$$

La théorie E_1 est une théorie monoïdale et son semi-anneau associé $\mathcal{S}_{E_1}$ est isomorphe à $\mathbb{Z}[\mathsf{h}_1, \mathsf{h}_2]/_{(\mathsf{h}_1^2\mathsf{h}_2+\mathsf{h}_2^2, \mathsf{h}_1+\mathsf{h}_1\mathsf{h}_2^2)}$, *i.e.* l'anneau $\mathbb{Z}[\mathsf{h}]$ quotienté par l'idéal engendré par les polynômes $\mathsf{h}_1^2\mathsf{h}_2 + \mathsf{h}_2^2$ et $\mathsf{h}_1 + \mathsf{h}_1\mathsf{h}_2^2$. Les théorèmes 2 et 3 montrent que décider la déduction et l'équivalence statique revient à résoudre des systèmes linéaires dans $\mathbb{Z}[\mathsf{h}_1, \mathsf{h}_2]/_{(\mathsf{h}_1^2\mathsf{h}_2+\mathsf{h}_2^2, \mathsf{h}_1+\mathsf{h}_1\mathsf{h}_2^2)}$. Ce type de systèmes peut en général être résolu en utilisant les bases de Gröbner [Eis99] et des implémentations de ce type de résolution sont disponibles (comme dans le logiciel libre Sage[1]). La décision de la déduction et de l'équivalence statique peut donc être implémentée facilement à l'aide de ces outils.

6 Composition

Chacun des résultats de décidabilité présentés dans les parties précédentes n'est valide que pour des théories équationnelles particulières ou pour des classes de théories particulières. Il

[1] `http://www.sagemath.org/`

a été montré [ACD07] qu'il est facile de composer les résultats de décidabilité dès lors que les théories sont disjointes.

Étant donnée une signature $\mathcal{F}$ et un ensemble E d'équations sur $\mathcal{F}$, on appelle théorie équationnelle $(\mathcal{F}, E)$ la relation d'équivalence induite sur $T(\mathcal{F}, \mathcal{X}, \mathcal{N})$ par les équations de E.

Théorème 4 ([ACD07]) *Soient $(\mathcal{F}_1, E_1)$ et $(\mathcal{F}_2, E_2)$ deux théories équationnelles telles que $\mathcal{F}_1 \cap \mathcal{F}_2 = \emptyset$. Si la déduction et l'équivalence statique sont décidables pour $(\mathcal{F}_1, E_1)$ et pour $(\mathcal{F}_2, E_2)$, alors la déduction et l'équivalence statique sont décidables $(\mathcal{F}_1 \cup \mathcal{F}_2, E_1 \cup E_2)$.*

Ce résultat nous permet donc de combiner tous les résultats précédentes, dès lors que les théories sont disjointes (mais peuvent partager les noms libres de $\mathcal{N}$).

7 Théories pour les protocoles de vote

Les protocoles de vote électronique utilisent souvent des primitives cryptographiques non standard qui ne peuvent pas toutes être traitées par les résultats précédents. Nous décrivons ci-dessous trois théories équationelles importantes pour le vote électronique, tirées de [DKR09].

7.1 Signature en aveugle

Les signatures en aveugle sont notament utilisées dans le protocole de Fujioka, Okamoto and Ohta [FOO92]. Elles permettent à un votant de transmettre son vote « caché » à un serveur qui contrôle que le votant a la permission de voter. Le vote (caché) est alors signé par le serveur et renvoyé au votant qui peut alors extraire son vote signé par le serveur sans que ce dernier n'ait pu avoir accès au vote. Cette primitive est formalisée par la théorie induite par les équations ci-dessous.

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

7.2 Théorie pour le « Trapdoor bit-commitment »

Le protocole de vote proposé par Fujioka, Okamoto et Ohta [FOO92] utilise également une fonction d'engagement (commitment) $\mathsf{tdcommit}(x,y,z)$. Le premier argument de $\mathsf{tdcommit}$ est la valeur sur laquelle l'agent s'engage, le deuxième argument est la « trappe » qui permet d'ouvrir l'engagement:

$$\mathsf{open}(\mathsf{tdcommit}(x,y,z),y) = x \tag{1}$$

Cette fonction a la particularité de ne pas constituer un véritable engagement: le votant peut choisir la valeur de la trappe de manière à ouvrir l'engagement de la manière qu'il souhaite:

$$\mathsf{tdcommit}(x, f_1(y,z,w,x), w) = \mathsf{tdcommit}(y,z,w) \tag{2}$$

La théorie E_{Trap}, définie dans [DKR09], est formée des équations 1, 2 et 8 ainsi que des trois équations ci-dessous pour la signature en aveugle.

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

7.3 Rechiffrement et preuve à vérificateur désigné

Le protocole de Lee *et al* [LBD+03] fait appel à une primitive de rechiffrement $\mathsf{rencrypt}$ qui permet de modifier l'aléa du message chiffré

$$\mathsf{rencrypt}(\mathsf{enca}(x, \mathsf{pub}(y), z), w) = \mathsf{enca}(x, \mathsf{pub}(y), f_0(z, w)) \tag{3}$$

Le terme $\mathsf{enca}(m, \mathsf{pub}(a), r)$ représente le chiffrement à clefs publiques du message m par la clef $\mathsf{pub}(a)$. Le terme r représente l'aléa utilisé dans le chiffrement. Il permet de distinguer deux chiffrements d'un même message. Le symbole f_0 représente le fait que l'aléa est modifié par le rechiffrement.

Le protocole de Lee *et al* fait également appel à une primitive de « preuve à vérificateur désigné ». Un agent peut produire la preuve $\mathsf{dvp}(x, y, z, \mathsf{pub}(w))$ que deux messages chiffrés x et y contiennent le même plaintext. Cette preuve n'est vérifiable que par l'agent w:

$$\mathsf{checkdvp}(\mathsf{dvp}(x, \mathsf{rencrypt}(x, y), y, \mathsf{pub}(z)), x, \mathsf{rencrypt}(x, y), \mathsf{pub}(z)) = \mathsf{ok} \tag{4}$$

$$\mathsf{checkdvp}(\mathsf{dvp}(x, y, z, w), x, y, \mathsf{pub}(w)) = \mathsf{ok} \tag{5}$$

La théorie E_{DVP}, définie dans [DKR09], est formée des équations 3, 5 et 4 ainsi que des trois équations plus classiques ci-dessous:

$$\mathsf{checksign}(\mathsf{sign}(x, y), \mathsf{pub}(y)) = x \tag{6}$$

$$\mathsf{dec}(\mathsf{enca}(x, \mathsf{pub}(y), z), y) = x \tag{7}$$

$$\mathsf{getpk}(\mathsf{host}(x)) = x \tag{8}$$

La dernière équation permet à un agent d'obtenir la clef publique d'un autre agent.

8 Deux nouveux résultats de décidabilité

Les théories E_{DVP} et E_{Trap} peuvent être complétées de manière à être convergentes. Cependant, aucun des résultats présentés dans les parties précédentes ne s'applique. En adaptant la technique développée à la partie 4, nous avons montré la décidabilité de la déduction et de l'équivalence statique pour ces deux théories.

Théorème 5 ([BBRC09]) *La déduction et l'équivalence statique sont décidables en temps polynomial pour les théories E_{DVP} et E_{Trap}.*

Ce résultat fait l'objet d'un rapport de recherche. Nous en rédigeons actuellement une version journal en vue d'une soumission.

9 Une première implémentation pour l'équivalence statique : l'outil YAPA

Le théorème 1 de la partie 3 a été généralisé par Mathieu Baudet [Bau05, Bau07] dans le cas actif, pour les théories sous-terme convergentes. En collaboration avec Mathieu Baudet, nous avons ensuite revisité sa procédure pour proposer un algorithme efficace (et implémenté) qui permet de décider la déduction et de l'équivalence statique pour une large classe de théories équationnelles, incluant en particulier les théories sous-termes convergentes.

Mathieu Baudet a étendu [Bau05, Bau07] le théorème 1 au cas actif, pour décider l'existence d'attaques par dictionnaire pour un nombre borné de sessions. On appelle « attaques par dictionnaire » les attaques consistant pour l'intrus à essayer par force brute les différentes valeurs d'un secret faible comme un mot de passe par exemple.

Nous avons revisité [BCD09] l'algorithme proposé par Mathieu Baudet, dans le cas passif uniquement, de manière à pouvoir traiter une classe plus large de théories que les théories sous-termes (mais sans symbole AC). Le principe de la procédure consiste à saturer une structure ϕ en ajoutant les termes déductibles par application d'un petit contexte, en n'ajoutant que les termes qui n'étaient pas déjà constructibles à partir des termes courants. Les recettes associées aux termes déductibles sont calculées au vol. L'algorithme est correct et complet au sens où, pour toute théorie convergente, si l'outil parvient à saturer la structure ϕ, alors

- un terme est déductible si et seulement si il est syntaxiquement déductible à partir de la structure saturée,
- une structure ϕ' est statiquement équivalente à ϕ si et seulement si elle vérifie les égalités entre recettes calculées lors de la saturation.

Par contre, il se peut que la saturation échoue ou ne termine pas. Nous avons montré que la procédure de saturation n'échoue jamais pour la classe des théories *convergentes en couche* comme les théories sous-terme convergents ainsi que les théories E_{blind} ou E_{homo} ou la théorie E_{pref}, définie ci-dessous.

$$E_{\mathsf{pref}} = E_{\mathsf{enc}} \cup \left\{ \ \mathsf{pref}(\mathsf{enc}(\mathsf{pair}(x,y),z)) \quad = \quad \mathsf{enc}(x,z) \ \right\}$$

Nous avons également proposé un critère pour assurer la terminaison de la procédure pour les théories convergentes en couche. Ce critère est en particulier satisfait par toutes les théories localement stables (sans symbole AC). Cet algorithme nous a permis de déduire la décidabilité de la déduction et de l'équivalence statique pour la théorie E_{pref} (décidabilité qui aurait aussi pu être établie à l'aide du résultat de [AC05, AC06]).

Cet algorithme a été implémenté dans l'outil YAPA[2] et ses performances sont discutées dans la partie 11.

10 L'outil KISS

Si nous avons pu montrer la décidabilité de l'équivalence statique pour les théories E_{DVP} et E_{Trap}, il se trouve que l'outil YAPA ne termine pour aucune de ces deux théories, pourtant importantes pour l'analyse des protocoles de vote électronique. L'outil KiSs a été développé pour tenter de remédier à cette faiblesse. Il implémente une représentation symbolique des structures et un processus de saturation symbolique qui étend intuitivement la procédure de YAPA. En particulier, l'outil KiSs dans le cas de la théorie E_{Trap}. La procedure implémentée par l'outil KiSs est décrite dans [CDK09]. Comme YAPA, notre procédure est une procédure *générique* pour des théories équationnelle dites convergentes, au sens où toute théorie convergente peut être donnée en entrée de l'outil. Comme le probleme sous-jacent est indécidable même dans le cas de théories équationnelle convergentes, notre procédure ne termine pas pour certaines théories.

[2] Accessible sur la page `http://www.lsv.ens-cachan.fr/~baudet/yapa/`

Cependant, nous avons démontré qu'elle termine de nombreuses classes intéressantes des théories: théories sous-termes, la théorie de la signature en aveugle, ainsi qu'une extension convergente de la théorie du "trapdoor bit-commitment". De plus, la terminaison est en temps polynomial sur ces théories.

À la base de l'algorithme sous-jacent se trouve un processus de *saturation*. A chaque structure on associe un ensemble initial fini des objets dits "faits de déduction" et "faits équationels" qui décrivent la capacité de l'intrus d'appliquer des symboles de fonction sur les termes de la structure. Par exemple, pour la structure

$$\phi \stackrel{\mathsf{def}}{=} \nu k.\{\mathsf{enc}(0,k)/x, k/y\}$$

l'ensemble initial contiendra en particulier les "faits de déduction" suivants:

$$\begin{array}{rr} x \rhd enc(0,k) \mid & \emptyset \\ enc(U,V) \rhd enc(u,v) \mid & U \rhd u, V \rhd v \end{array}$$

Le premier fait indique que x est une recette pour le terme $enc(0,k)$, alors que le deuxième fait indique que $enc(U,V)$ est une recette pour $enc(u,v)$ si U (resp. V) est une recette pour u (resp. v).

L'ensemble initial est *saturé* (c'est-à-dire qu'on ajoute des nouveaux "faits" en respectant certaines regles syntaxiques) de façon à permettre à l'intrus de travailler modulo la théorie équationnelle. Pour la structure ϕ définie précédement, pendant le processus de saturation, on va obtenir le "fait équationnel"

$$x \sim enc(0,y) \mid \emptyset$$

qui indique que $(x =_E enc(0,y))\phi$.

Si la saturation termine, l'ensemble de faits obtenus est une description complète de la structure. Pour tester l'équivalence statique $(\phi_1 \stackrel{?}{\approx}_E \phi_2)$ il suffit ensuite de déterminer si chaque structure satisfait les faits équationnels de l'autre.

En utilisant une stratégie de saturation *équitable*, on peut retrouver un autre résultat de décision, celui pour le chiffrement homomorphique. Il faut aussi noter que notre procédure est "best effort", au sens où elle termine pour certaines structures même si ce n'est pas un algorithme de décision pour la théorie considérée.

La procédure a été implementée dans l'outil KiSs[3]. KiSs implémente une stratégie de saturation *faiblement équitable* (qui permet en particulier de décider la théorie du chiffrement homomorphique) et stocke les termes sous forme DAG (ce qui permet d'obtenir des performances polynomiales pour certaines théories). Une faiblesse notable de KiSs est qu'en général, l'outil ne termine pas pour la théorie de rechiffrement E_{renc}, formée simplement de l'équation 3 (KiSs ne termine pas pour la théorie E_{DVP} non plus, dont l'équation 3 fait partie).

11 Comparaison des outils YAPA et KISS

On ne connaît pas pour l'instant si les classes des théories équationnelles sur lesquelles YAPA et respectivement KiSs terminent sont incomparable ou si KiSs englobe YAPA. Inversement, nous savons que YAPA ne termine pas sur la théorie du "trapdoor bit-commitment" alors que

[3]disponible en ligne à l'addresse `http://www.lsv.ens-cachan.fr/~ciobaca/kiss`

KiSs termine en temps polynomial). Il sera intéressant de comparer les deux outils de ce point de vue. D'autre part, il faudra étendre les outils pour traiter la théorie du rechiffrement.

YAPA et KiSs fonctionnent de manière efficace, comme l'illustre le tableau ci-dessous.

Théorie équationnelle	E_{enc} $n = 10$	E_{enc} $n = 14$	E_{enc} $n = 18$	E_{enc} $n = 20$	E_{blind}	E_{pref}	E_{homo}
Temps d'exécution YAPA	0.13s	2.22s	41.91s	3m0.33s	0.03s	1.17s	3.01s
Temps d'exécution KiSs	0.65s	1.83s	4.73s	6.85s	0.1s	1.63s	16.03s

Dans le cas de la théorie équationnelle E_{enc}, nous avons testé YAPA et KiSs sur les structures $\varphi_n = \{{}^{t_n^0}/_{x_1}, {}^{c_0}/_{x_2}, {}^{c_1}/_{x_3}\}$ et $\varphi'_n = \{{}^{t_n^1}/_{x_1}, {}^{c_0}/_{x_2}, {}^{c_1}/_{x_3}\}$, où $t_0^i = c_i$ et $t_{n+1}^i = \mathsf{pair}(\mathsf{enc}(t_n^i, k_n^i), k_n^i)$, $i \in \{0, 1\}$. Ces exemples permettent d'accroître exponentiellement en n la taille (non DAG) des tests permettant de distinguer φ_n et φ'_n tandis que la taille de φ_n et φ'_n croît linéairement. Les fichiers d'exemples utilisés sont accessibles depuis la page de l'outil YAPA.

YAPA et KiSs sont les deux premiers outils dédiés à la décision de la déduction et l'équivalence statique. Ils proposent des algorithmes de décision unifiés, fonctionnant pour de nombreuses théories convergentes sans symbole AC. Le seul autre outil capable de décider l'équivalence statique est l'outil ProVerif [Bla01, Bla05], développé par Bruno Blanchet. Cet outil est conçu pour analyser des propriétés d'accessibilité et d'équivalence pour les protocoles cryptographiques, pour un nombre non borné de sessions. Il s'applique donc à un contexte plus large (et plus difficile) que YAPA et KiSs. En retour, ProVerif est nettement moins efficace pour décider l'équivalence statique et échoue sur certaines théories comme E_{homo}.

12 Bilan et perspectives

Les différents résultats de décidabilité décrits au cours de ce rapport sont résumés à la figure 1. Les citations en gras font références aux résultats obtenus dans le cadre du projet AVOTÉ.

Au sein du projet AVOTÉ, nous avons ainsi proposé de nouveaux résultats de décision pour l'équivalence statique et pour des théories équationelles pertinentes pour les protocoles de vote électronique. Nous avons également proposé les deux premiers outils permettant de décider automatiquement l'équivalence statique.

Il est possible que de nouvelles théories seront à étudier, en fonction des primitives utilisées par les protocoles de vote électronique.

La décidabilité de la déduction et de l'équivalence statique ne sont qu'une brique de base pour analyser les protocoles contre un intrus actif. Une suite logique du travail présenté dans ce rapport consiste à étendre les résultats de décidabilité au cas actif, au moins pour un nombre borné de sessions. Des résultats préliminaires ont été obtenus pour la classe des théories sous-terme convergentes [CD09]. Par contre, aucun résultat n'existe à l'heure actuelle pour analyser les protocoles de vote dans le cas d'un intrus actif.

Théorie	Déduction	Équivalence statique
sous-terme convergente Ex: $E_{\mathsf{enc}}, E_{\mathsf{inv}}, E_{\mathsf{idem}}, E_{\mathsf{sym}}$	PTIME [AC06, CDK09]	
E_{blind}	décidable [AC06], PTIME **[BCD09, CDK09]**	
E_{homo}	décidable [AC06, CDK09], PTIME **[BCD09]**	
E_{pref}	décidable [AC06], PTIME **[BCD09]**	
E_{DVP}	PTIME **[BBRC09]**	
E_{Trap}	PTIME **[BBRC09, CDK09]**	
ACU, E_{AC}	NP-complet	PTIME [AC06, CD07]
ACUI	décidable [DLLT06, DLLT08]	décidable [CD07]
ACUN ($E_{\oplus}$)	PTIME [CKRT03]	PTIME [AC06, CD07]
AG	PTIME [CKR$^+$03]	PTIME [CD07]
monoidale	décidable [CD07]	décidable [CD07]

D'après le résultat de composition [ACD07], la déduction et l'équivalence statique sont également décidables pour l'union de toutes théories disjointes mentionnées dans ce tableau.

Figure 1: Résultats de décidabilité pour la déduction et l'équivalence statique.

References

[AC04] M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. In *Proc. 31st Int. Coll. Automata, Languages, and Programming (ICALP'2004)*, volume 3142 of *Lecture Notes in Computer Science*, pages 46–58, Turku, Finland, July 2004. Springer.

[AC05] M. Abadi and V. Cortier. Deciding knowledge in security protocols under (many more) equational theories. In *Proc. 18th IEEE Computer Security Foundations Workshop (CSFW'05)*, pages 62–76, Aix-en-Provence, France, June 2005. IEEE Comp. Soc. Press.

[AC06] M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*, 387(1-2):2–32, November 2006.

[ACD07] Mathilde Arnaud, Véronique Cortier, and Stéphanie Delaune. Combining algorithms for deciding knowledge in security protocols. In *Proceedings of the 6th International Symposium on Frontiers of Combining Systems (FroCoS'07)*, volume 4720 of *Lecture Notes in Artificial Intelligence*, pages 103–117, Liverpool, UK, September 2007. Springer.

[AF01] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proc. of the 28th ACM Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115, January 2001.

[Bau05] Mathieu Baudet. Deciding security of protocols against off-line guessing attacks. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'05)*, pages 16–25, Alexandria, Virginia, USA, November 2005. ACM Press.

[Bau07] Mathieu Baudet. *Sécurité des protocoles cryptographiques : aspects logiques et calculatoires*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2007.

[BBRC09] Mouhebeddine Berrima, Narjes Ben Rajeb, and Véronique Cortier. Deciding knowledge in security protocols under some e-voting theories. Research Report RR-6903, INRIA, April 2009.

[BCD09] Mathieu Baudet, Véronique Cortier, and Stéphanie Delaune. YAPA: A generic tool for computing intruder knowledge. In *20th International Conference on Rewriting Techniques and Applications (RTA'09)*, volume 5595 of *Lecture Notes in Computer Science*, pages 148–163, Bras??lia, Brazil, June 2009. Springer.

[BCK09] Mathieu Baudet, Véronique Cortier, and Steve Kremer. Computationally sound implementations of equational theories against passive adversaries. *Information and Computation*, 207(4):496–520, April 2009.

[Bla01] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proc. of the 14th Computer Security Foundations Workshop (CSFW'01)*. IEEE Computer Society Press, June 2001.

[Bla05] Bruno Blanchet. An automatic security protocol verifier based on resolution theorem proving (invited tutorial). In *20th International Conference on Automated Deduction (CADE-20)*, Tallinn, Estonia, July 2005.

[Bor05] J. Borgström. Static equivalence *is* harder than knowledge. In Jos Baeten and Iain Phillips, editors, *Proceedings of the 12th International Workshop on Expressiveness in Concurrency (EXPRESS'05)*, Electronic Notes in Theoretical Computer Science, San Francisco, CA, USA, August 2005. Elsevier Science Publishers. To appear.

[CD07] Véronique Cortier and Stéphanie Delaune. Deciding knowledge in security protocols for monoidal equational theories. In *Proc. of the 14th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07)*, volume 4790 of *Lecture Notes in Artificial Intelligence*, pages 196–210, Yerevan, Armenia, October 2007. Springer.

[CD09] Véronique Cortier and Stéphanie Delaune. A method for proving observational equivalence. In *Proceedings of the 22nd IEEE Computer Security Foundations Symposium (CSF'09)*, pages 266–276, Port Jefferson, NY, USA, July 2009. IEEE Computer Society Press.

[CDK09] Ştefan Ciobâcă, Stéphanie Delaune, and Steve Kremer. Computing knowledge in security protocols under convergent equational theories. In Renate Schmidt, editor, *Proceedings of the 22nd International Conference on Automated Deduction (CADE'09)*, Lecture Notes in Artificial Intelligence, Montreal, Canada, August 2009. Springer. To appear.

[CKR+03] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, Mathieu Turuani, and Laurent Vigneron. Deciding the security of protocols with diffie-hellman exponentiation and product in exponents. In *Proc. of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'03)*, 2003.

[CKRT03] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, and Mathieu Turuani. An NP decision procedure for protocol insecurity with xor. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS '03)*, 2003.

[Del06] Stéphanie Delaune. Easy intruder deduction problems with homomorphisms. *Information Processing Letters*, 97(6):213–218, March 2006.

[DKR09] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 2009. To appear.

[DLLT06] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. In Michele Buglesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) — Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 132–143, Venice, Italy, July 2006. Springer.

[DLLT08] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis for monoidal equational theories. *Information and Computation*, 206(2-4):312–351, February-April 2008.

[Eis99] David Eisenbud. *Commutative Algebra with a view toward Algebraic Geometry.* Springer, 1999.

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *Advances in Cryptology - AUSACRYPT'92*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer-Verlag, 1992.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, April 1984.

[KR05] Steve Kremer and Mark. Ryan. Analysis of an electronic voting protocol in the applied pi-calculus. In Mooly Sagiv, editor, *Programming Languages and Systems – Proceedings of the 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 186–200, Edinburgh, U.K., April 2005. Springer.

[LBD+03] Lee, Boyd, Dawson, Kim, Yang, and Yoo. Providing receipt-freeness in mixnet-based voting protocols. In *ICISC: International Conference on Information Security and Cryptology.* LNCS, 2003.

[Mil99] Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999.

[Nut90] W. Nutt. Unification in monoidal theories. In *Proc. 10th Int. Conference on Automated Deduction, (CADE'90)*, volume 449 of *LNCS*, pages 618–632, Kaiserslautern (Germany), 1990. Springer.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1986.