Associative-Commutative deducibility constraints

Sergiu Bursuc
LSV, CNRS
INRIA project SECSI
École Normale Supérieure de Cachan
bursuc@lsv.ens-cachan.fr

joint work with Hubert Comon-Lundh and Stéphanie Delaune

STACS Aachen, February 2007

Plan

1. Motivation: Security protocols
2. The problem: Diophantine systems
3. Decision result

Security protocols

- Concurrent programs aiming at some secure transaction
- Even when cryprographic primitives are idealized, there are many attacks
- SPORE – the protocol library
  `//www.lsv.ens-cachan.fr/spore/`

Search for attacks

1. Choose a fixed number of copies of each process, with parameter values
2. Guess an interleaving of the actions of each process
3. Solve a deducibility constraint system (solution = attack)

Deducibility constraints

Parts of messages that cannot be analyzed by the agents are abstracted with variables.

Syntax:

$$\begin{array}{rcl} T_0 & \Vdash & u_1 \\ T_0, v_1 & \Vdash & u_2 \\ & \vdots & \\ T_0, v_1, \ldots, v_n & \Vdash & s \end{array}$$

Semantics: An equational theory E and a deducibility relation $\vdash$ on terms modulo E.
σ is an attack if, for every i, $T_i\sigma \vdash u_{i+1}\sigma$

Decision procedures

Exclusive Or [CKRT, LICS'03], [CLS, LICS'03]

Modular exponentiation [CKRT, FST/TCS'03] [Shmat, FOSSACS'04]

Homomorphisms and exclusive or [DLLT, ICALP'06]

...

Decision procedures

Exclusive Or [CKRT, LICS'03], [CLS, LICS'03]

Modular exponentiation [CKRT, FST/TCS'03] [Shmat, FOSSACS'04]

Homomorphisms and exclusive or [DLLT, ICALP'06]

...

But does not solve our case studies !

Decision procedures

Exclusive Or [CKRT, LICS'03], [CLS, LICS'03]

Modular exponentiation [CKRT, FST/TCS'03] [Shmat, FOSSACS'04]

Homomorphisms and exclusive or [DLLT, ICALP'06]

...

But does not solve our case studies !

[CLD, RTA'05]: reduction of many relevant theories to the pure associative-commutative case.

Associative-Commutative constraints

Terms : linear combinations of variables and constants

Deduction : the intruder can only add terms.

Example:

$$2a \Vdash X \qquad\qquad a + b \Vdash Y$$

Associative-Commutative constraints

Terms : linear combinations of variables and constants

Deduction : the intruder can only add terms.

Example:

$$2a \Vdash X \qquad a + b \Vdash Y$$

$$a, X + b \Vdash Y$$

Associative-Commutative constraints

Terms : linear combinations of variables and constants

Deduction : the intruder can only add terms.

Example:

$$2a \Vdash X \qquad a + b \Vdash Y$$
$$a, X + b \Vdash Y$$

Solutions of the two left contraints are of the form:

$$X \mapsto 2\lambda a; \quad Y \mapsto (2\lambda\lambda'' + \lambda')a + \lambda'' b$$

AC-constraints and Diophantine equations.

- AC-constraints $\Longrightarrow$ Diophantine equations.

AC-constraints and Diophantine equations.

- AC-constraints $\Longrightarrow$ Diophantine equations.

$$\left\{\begin{array}{rcl} 2a & \Vdash & X \\ a, X+b & \Vdash & Y \end{array}\right. \Longrightarrow \left\{\begin{array}{l} X_a = 2\lambda \\ Y_a = \lambda' + \lambda'' X_a \\ Y_b = \lambda'' \end{array}\right.$$

AC-constraints and Diophantine equations.

- AC-constraints $\Longrightarrow$ Diophantine equations.

$$\left\{\begin{array}{rcl} 2a & \Vdash & X \\ a, X+b & \Vdash & Y \end{array}\right. \Longrightarrow \left\{\begin{array}{l} X_a = 2\lambda \\ Y_a = \lambda' + \lambda'' X_a \\ Y_b = \lambda'' \end{array}\right.$$

- Diophantine equations $\Longrightarrow$ AC-constraints

AC-constraints and Diophantine equations.

- AC-constraints $\Longrightarrow$ Diophantine equations.

$$\left\{\begin{array}{rcl} 2a & \Vdash & X \\ a, X+b & \Vdash & Y \end{array}\right. \Longrightarrow \left\{\begin{array}{l} X_a = 2\lambda \\ Y_a = \lambda' + \lambda'' X_a \\ Y_b = \lambda'' \end{array}\right.$$

- Diophantine equations $\Longrightarrow$ AC-constraints
  Code the sum and the product.

AC-constraints and Diophantine equations.

- AC-constraints $\Longrightarrow$ Diophantine equations.

$$\left\{\begin{array}{rcl} 2a & \Vdash & X \\ a, X+b & \Vdash & Y \end{array}\right. \Longrightarrow \left\{\begin{array}{l} X_a = 2\lambda \\ Y_a = \lambda' + \lambda'' X_a \\ Y_b = \lambda'' \end{array}\right.$$

- Diophantine equations $\Longrightarrow$ AC-constraints
  Code the sum and the product.
- Undecidability in general.

Some restrictions

We assume:

Monotonicity: left members of constraints are linearly ordered by inclusion.

Determinacy: each constraint introduces at most one variable, on its right.

In addition: each right member contains at most one variable.

Counter examples:

- $a \Vdash X \wedge b \Vdash Y$
- $a + b \Vdash Y \wedge a + b, a + X \vdash 2Y$
- $a + b \Vdash X + Y$
- $a \Vdash X \wedge a, a + b \Vdash X + Y$

Examples

- Every semi-linear set can be expressed as the solutions of a well-formed system
- $$\begin{cases} a & \Vdash & X \\ a, X + b & \Vdash & Y \end{cases}$$

  has solutions $\{X \mapsto xa;\ Y \mapsto (xy + z)a + yb\}$. i.e. $\{(x, 0, xy + z, y) \mid x, y, z \in \mathbb{N}\}$, which is not semi-linear, and not trivially simple semi-polynomial (cf. Karianto, Krieg, Thomas, ICALP'06).

Decision result

Theorem: the existence of a solution to a simple AC-deducibility constraint system is in NEXPTIME.

Proof:

1. partition the variables w.r.t. an occurrence preorder
2. solutions of a constraint only using variables in the minimal class (minimal strongly connected component) form a semi-linear set
3. minimal solutions of an arbitrary simple system, when restrited to the variables of a minimal class are “not too far” from minimal solutions of the minimal class.

Other intruder capabilities

Intruder now has two rules:

$$\frac{x \qquad y}{x + y} \qquad\qquad \frac{x + y \qquad y}{x}$$

Theorem: Deducibility contraints with such capabilities are solvable in NEXPTIME
Moreover, we do not need the restriction that right members contain at most one variable

Conclusions

- There is still a challenge on AC-deducibility constraints (without the additional restriction)
- Applicability has to be demonstrated
- Another research direction: combination techniques