RAPPORT TECHNIQUE PROUVÉ

Sufficient conditions on properties for an automated verification: theoretical report on the verification of protocols for an extended model of the intruder

Auteurs	:	Vincent Bernat, Hubert Comon-Lundh, Véronique Cortier, Stéphanie Delaune, Florent Jacquemard, Pascal Lafourcade, Yassine Lakhnech, Laurent Mazaré

Date	:	December 17, 2004
Rapport PROUVÉ numéro	:	4
Version	:	1.0

Loria
CNRS UMR 7503,
Campus Scientifique - BP 239
54506 Vandoeuvre-lès-nancy cedex
www.loria.fr

Laboratoire Spécification Vérification
CNRS UMR 8643, ENS Cachan
61, avenue du président-Wilson
94235 Cachan Cedex, France
www.lsv.ens-cachan.fr

Laboratoire Verimag
CNRS UMR 5104,
Univ. Joseph Fourier, INPG
2 av. de Vignate,
38610 Gières, France
www-verimag.imag.fr

Cril Technology
9/11 rue Jeanne Braconnier
92360 Meudon La Foret Cedex, France
www.cril.fr

France Telecom
Div. Recherche et Développement
38, 40 rue du Général Leclerc
92794 Issy Moulineaux Cedex
www.rd.francetelecom.fr

Résumé : Les primitives cryptographiques sont le plus souvent représentées par des symboles libres de fonctions, suivant ainsi l'*hypothèse du chiffrement parfait*. Cependant certaines attaques ou même le déroulement honnête de certains protocoles utilisent les propriétés algébriques d'opérateurs comme le « ou exclusif », l'exponentiation modulaire ou l'addition par exemple.
Ce rapport se découpe en deux parties. Dans un premier temps, nous présentons un état de l'art général des résultats obtenus pour la vérification automatique des protocoles cryptographiques, pour certaines propriétés algébriques. Dans un deuxième temps, nous détaillons les résultats obtenus dans le cadre projet RNTL PROUVÉ, par les différents partenaires du projet. Ces travaux ont fait l'objet de publications [AC04, DJ04, Maz04a, Maz04b, Maz04c, BEL05, BEL04b] ou de rapports [CDL04, LLT04].

Contents

Sufficient conditions on properties for an automated verification: theoretical report on the verification of protocols for an extended model of the intruder

Vincent Bernat, Hubert Comon-Lundh, Véronique Cortier,
Stéphanie Delaune, Florent Jacquemard,
Pascal Lafourcade, Yassine Lakhnech, Laurent Mazaré

December 17, 2004

Abstract: Cryptographic protocols are successfully analyzed using formal methods. However, formal approaches usually consider the encryption schemes as black boxes and assume that an adversary cannot learn anything from an encrypted message except if he has the key. Such an assumption is too strong in general since some attacks exploit in a clever way the interaction between protocol rules and properties of cryptographic operators. Moreover, the executability of some protocols relies explicitly on some algebraic properties of cryptographic primitives such as commutative encryption. We first give an overview of the existing methods in formal approaches for analyzing cryptographic protocols. Then we describe more precisely the results obtained by the partners of the RNTL project PROUVÉ.

1 Introduction

Cryptographic protocols are short programs designed to ensure secure communications on channels that may be controlled by an attacker. Considering the increasing size of networks and their dependence on cryptographic protocols, a high level of assurance is needed in the correctness of such protocols. These protocols are notoriously difficult to design and test, and serious flaws have been found in many protocols. Consequently, there has been a growing interest in applying formal methods for validating cryptographic protocols. These protocols use cryptographic primitives as public and symmetric encryption. These functions are based on mathematical notions (like modular exponentiation or elliptic curves) and on algorithmically hard problems (as extracting the modular logarithm or factorization into prime numbers). A first approach is to verify a protocol with its actual cryptographic primitives and to show that attacking the protocol can be reduced to solving an algorithmically hard problem. Such proofs are done by hand and are often long and difficult. In particular, they seem very hard to automate.

An other approach is to abstract from cryptographic primitives. This may be justified by the observation that many attacks rely only on the logical structure of the protocols and simply consist in replaying some messages at the right steps. That is why formal methods usually consider encryption schemes as black boxes and assume that an adversary cannot learn anything from an encrypted message except if he has the key. This is called the *perfect encryption assumption*. More generally, formal methods assume *perfect cryptography*: the other cryptographic primitives (like pairing or hashing) are also idealized in order to enable automatic verification. Even if these assumptions are not realistic, many real attacks have been discovered using this approach. The most famous flaw is the man-in-the-middle attack on the Needham-Schroeder protocol with public key encryption, found by G. Lowe [Low96] using an automatic tool.

Many decidability results have been obtained under this perfect cryptography hypothesis: the secrecy preservation is co-NP-complete for a bounded number of sessions [RT01], and decidable for an unbounded number of sessions under some additional restrictions [Low98, DLMS99, CLC03]. Many tools have also been developed to automatically verify cryptographic protocols like [Mea96, Low97, Bla01].

Recent works investigate how to relax the perfect cryptography assumption by refining the abstraction on cryptographic primitives. The aim is to take into account some of the algebraic properties of the cryptographic primitives. Most of the algebraic properties studied so far and presented in this survey are properties that can be modeled using equations. For example a commutative encryption is expressed by the equality $\{\{x\}_y\}_z = \{\{x\}_z\}_y$. Such a representation of the algebraic properties is natural for many cryptographic primitives and very convenient since it enables to reuse classical methods on first order logic for terms modulo equational theories. The interest of studying the algebraic properties of the cryptographic primitives is that some attacks may be missed when abstracting encryption as a perfect black box. For example, Bull's protocol has been proven secure [Pau97] under the perfect encryption assumption. However; the protocol uses the exclusive or operator, which is associative, commutative, nilpotent, and has a neutral element. An attack on this protocol relying on these properties has been given in [RS98]. Even without searching for attacks, one may need algebraic properties to simply be able to specify some protocols. For example, the simple Three Pass protocol proposed by R. Shamir [CJ97] requires a commutative encryption like the RSA encryption. All these examples will be developed in the following sections.

The aim of this report is to present sufficient conditions to verify cryptographic protocols with algebraic properties. This survey contains two main sections. In Section 2, we first present an overview of decidability results or incomplete decision procedures that have been obtained so far for some algebraic properties using formal methods. After a brief description on how cryptographic protocols are modeled (Section 2.1), we give in Section 2.2 a summary of the results obtained assuming perfect cryptography. This work has been partially published in a research report [CDL04]. Then in Section 3, we provide a detailed description of the results obtained by the partners of the RNTL project PROUVÉ. When looking for decidability results for cryptographic protocols under equational theories, the basic step consists in understanding what results we can obtain for ground deducibility. Deciding whether a message is deducible from a finite set of messages has been studied for several equational theories with AC symbol and homomorphism. This is described in Section 3.1, full details may be found in [LLT04]. Decidability of deducibility and static equivalence for general classes of equational theories is studied in Section 3.2. This work has also been published in [AC04]. In Section 3.3, we consider an intruder that may guess certain parts of the messages and introduce a notion of *opacity* for which we show decidability [Maz04a, Maz04b, Maz04c]. Next, we have obtained decidability results for an active intruder and a finite number of sessions. In Section 3.4, we consider general equational theories with explicit destructors. This encompasses encryption, decryption and pairing for example. We prove that secrecy remains co-NP-complete [DJ04]. In Section 3.5, we also consider protocols with timestamps and provide an algorithm for checking security properties [BEL05, BEL04b]. We conclude this section by the presentation of an on-going work (Section 3.6) on the decidability and complexity of deduction of an active intruder for a finite number of sessions and general equational theories.

2 Existing Results

Many decidability results have been obtained under the perfect cryptography assumption. Recently, several works try to extend these results to protocols with some algebraic properties. We give here an overview of these two kinds of results, after briefly describing how cryptographic protocols are modeled in formal methods. Results under the perfect cryptography assumption are summarized in Table 1, results for algebraic properties are summarized in Table 2.

2.1 Modeling Cryptographic Protocols

Security protocols are typically specified as sets of roles which are abstract patterns of communication specifying which messages are sent, and how to respond to the reception of any message. The messages are represented by terms built over a given alphabet of function symbols containing constants, pairing, and encryption. It may also contain some other symbols such as decryption, exclusive or, and multiplication. In such a case we cannot continue to model messages in terms of free algebras. Instead, we have to consider in addition an equational theory defined by a set of equations to take into account algebraic properties of the operators.

While there are many properties that a security protocol may aim to guarantee, the main results relaxing the *perfect cryptography assumption* regard only trace properties, and in particular *secrecy*: a secret, generated by an honest agent, should not be leaked to the intruder, who is assumed to have a complete control of network communication. For verifying such a property the attacker is typically represented as an active entity who is able for instance to eavesdrop and replay messages, impersonate honest agents, and generate nonces. Deciding whether a protocol preserves secrecy against such an active intruder is called the *security problem*. However; the security decision problem in presence of a passive attacker, who can only eavesdrop messages, is also a significant question and is in general the first step to obtain decision procedures for the security problem and the search of attacks. We can formulate this *intruder deduction problem*, in the following way: given a finite set of messages T and a message s (the secret), can the intruder deduce s? We present existing results for both the security problem and the intruder deduction problem.

Although the results have been obtained in different models (multiset rewriting, strand spaces, process calculus, ...) we give the results without specifying the models since it is quite well accepted that these results are in general also valid in the other models. Some translations between different models have been proposed *e.g.* by [BCLM03].

2.2 Results under the Perfect Cryptography Assumption

The analysis techniques discussed in this section and summarized in Table 1 assume *perfect cryptography*. This means that cryptographic primitives (pairing, encryption, ...) are considered without any algebraic properties. In particular, under the perfect encryption assumption, the encryption is modeled as a black box and the only way to obtain the plain text from the cipher text is by knowing the decryption key. We are going to give a brief overview of decidability results concerning the security problem.

Some Undecidability Results Though cryptographic protocols are often described in a concise way, the verification problem is difficult because of many sources of unboundedness in their modeling, for instance the number of sessions, the length of messages, or generation of nonces.

<table>
<tr><th rowspan="2">Bounded number of sessions</th><th colspan="2">Unbounded number of sessions</th></tr>
<tr><th>Without nonces</th><th>With nonces</th></tr>
<tr><td rowspan="4">co-NP-complete
[RT01]</td><td>Bounded message length:
DEXPTIME-complete
[DLMS99, CKR$^+$03b]</td><td>Bounded message length:
Undecidable
[DLMS99, AC02]</td></tr>
<tr><td>Tagged protocols:
EXPTIME
[BP03]</td><td>Strongly typed protocols:
Decidable [Low98]
Tagged protocols:
Decidable [RS03]</td></tr>
<tr><td>One copy:
3-EXPTIME
[CLC03]</td><td>Ping-pong protocols:
PTIME
[DEK83]</td></tr>
<tr><td colspan="2">General case:
Undecidable [EG83, CC01]</td></tr>
</table>

Table 1: Summary of Results for the Security Problem under the Perfect Cryptography Assumption.

When considering an unbounded number of sessions, the main sources of undecidability are the nonces generation and the possibility to copy arbitrary messages. Several codings of the Post Correspondence Problem [DW83] have been proposed, *e.g.* S. Even and O. Goldreich show in this way the undecidability of the security problem using only a bounded number of nonces [EG83]. Some other codings exist in order to obtain subtler undecidability results, for example H. Comon-Lundh and V. Cortier show in [CC01] that the problem is undecidable even without using composed keys.

Using nonces generation, N. Durgin *et al.* [DLMS99] show that the secrecy problem for protocols is undecidable, even when the length of the messages is bounded. R. Amadio and W. Charatonik [AC02] are even more careful in their analysis since they only consider the encryption primitive to obtain the undecidability result.

Some Decidability Results We have seen that the prominent sources of undecidability are unbounded message length and unbounded number of nonces. Now, we are going to give some decidability results which can be obtained by setting strong conditions on the protocols. One of the first results is a PTIME complexity result which has been obtained by D. Dolev *et al.* for ping-pong protocols between two participants [DEK83]: in each step of the protocol, one of the agents applies a sequence of operators to the last received message, and sends it to the other agent. In [Low98], G. Lowe studies also the security problem with an unbounded number of nonces and shows decidability for subclasses of protocols. In particular, he assumes that each participant can completely analyze any messages he receives. On the contrary, some other works [DLMS99, CKR$^+$03b, CLC03] studied the secrecy problem in the setting of a bounded number of nonces and additional strong restrictions on the protocols. For example the technique described in [CLC03] by H. Comon-Lundh and V. Cortier use stringent criteria by considering protocols in which at each transition an agent may copy at most one unknown component of the received message. Y. Chevalier *et al.* [CKR$^+$03b] and N. Durgin *et al.* [DLMS99] assume that the message size is bounded. Some other works such as the work of B. Blanchet [BP03] use tagging schemes to obtain decidability of secrecy. R. Ramanujam and S. Suresh show that secrecy is decidable in the presence of a bounded number of nonces [RS03]. However; in this last case, some protocols such as the Yahalom protocol do not follow the restricted syntax since agents have to forward message components which cannot be decrypted by them.

Even if it is assumed that there is a bounded number of sessions, it is still not easy to design a decision algorithm since the number of messages that can be created by the attacker is unbounded. M. Rusinowitch and M. Turuani extend in [RT01] the work of R. Amadio *et al.* [ALV02] by giving an NP-complete procedure for deciding protocol security for the Dolev-Yao attacker as long as the number of sessions is bounded. Some similar results [Bor01, MS01] have been obtained in other models. H. Huttel [Hut02] shows a similar result in a context of process algebra for a stronger secrecy property which says that a datum s is secret if the session which contains this datum is indistinguishable of all the sessions containing a datum s' at the place of s. This notion of secrecy is known as an observational equivalence property.

2.3 Results under Relaxation of the Perfect Cryptography Assumption

Certain algebraic properties of encryption, such as the homomorphic properties of RSA or the properties induced by chaining methods for block ciphers, are widely used in protocol constructions. Many real attacks rely on these properties. Recently, several procedures have been proposed to decide insecurity of cryptographic protocols when considering some algebraic properties of the cryptographic primitives, mostly for a finite number of sessions.

The results presented in this section and summarized in Table 2 are first steps towards reducing the gap between formal methods and mathematical proofs typically employed in cryptographic analysis of security protocols. Even though mathematical properties of the underlying cryptographic primitives are taken into account, the analysis is still done on an abstract model, and thus attacks can be missed due to this idealized treatment of cryptography.

Very few automatic tools can handle algebraic properties. The most flexible one is the automatic verifier developed by B. Blanchet [Bla04b]. Any property can be defined provided that it can be expressed in Horn clauses. The treatment of equations is still very naïve and preliminary. In particular, the system may not terminate when more complex equations are entered. The NRL protocol analyzer [Mea96] of C. Meadows has been enriched [Mea00] in order to analyze group protocols with Diffie-Hellman exponentiation, such as the IKA-1 protocol. In the Casper tool [Low97], G. Lowe partially models the Vernam encryption (which uses *exclusive or*) and finds a known flaw on the TMN protocol [TMN89]. The Casper tool also enables to analyze protocols with timestamps under typing assumptions and assumptions on the time window to bound the search space. Finally, the Casrul tool [JRV00] of M. Rusinowitch *et al.* is able to find most of the attacks relying on the associativity property of the pairing function.

Theoretical results are much more numerous; we present them in the following sections, sorted by the algebraic properties.

Associativity The associativity property is typically the property of the pairing function. It is expressed by the equation `[[x,y],z] = [x,[y,z]]`. Up to our knowledge, there is no theoretical work studying this single property applied to cryptographic protocols. However, the Casrul tool [JRV00] takes partially into account this property: it manages to find most of the attacks relying on the associativity of the pairing function. It does not capture the full theory of associativity, mostly for reason of efficiency.

	Intruder deduction problem	Security problem	
		Bounded number of sessions	Unbounded number of sessions
Commutativity	*PTIME [CKRT04]*	*co-NP-complete [CKRT04]*	**Ping-pong protocols:** *co-NP-complete [Tur03]*
Exclusive or	*PTIME [CKRT03]*	**General case:** *Decidable [CLS03]* **Restricted protocols:** *co-NP-complete [CKRT03]*	**One copy:** *3-EXPTIME [CLC03]* **Two-way automata:** *Decidable [Ver03]*
Abelian Groups	*NP [CLS03]*	*Decidable [Shm04]*	**Two-way automata:** *Decidable [Ver03]*
Homomorphism	*PTIME [CLT03]*	**Fixed blocks** *co-NP-complete [Che03]*	
Prefix	*PTIME [CKRT03]*	**Fixed blocks** *co-NP-complete [CKRT03]*	
Abelian Groups and Modular Exponentiation	*PTIME [CKR+03a]*	**General case:** *Decidable [Shm04]* **Restricted protocols:** *co-NP-complete [CKR+03a]*	*Decision Procedure [JGLV04]*
Time-stamps	*PTIME* (1)	*Decidable [BEL04b]*	*Decision Procedure [DG04]*

(1) For the intruder deduction problem, there is no notion of time. Thus deciding whether a message with timestamps can be deduced from a set of messages with timestamps correspond to the intruder deduction problem where timestamps are considered as constants known to the intruder.

Empty boxes mean that, to our knowledge, no result has been obtained so far.

Table 2: Summary of Decidability Results or Decision Procedures for some Equational Theories.

Commutativity Chevalier *et al.* present in [CKRT04] a NP decision procedure for the security problem of protocols that employ commutating encryption, *i.e.* $\{\{x\}_y\}_z = \{\{x\}_z\}_y$. One of the most important instances of commutating encryption is RSA encryption with common modulus. For ping-pong protocols (described in Section 2.2), M. Turuani [Tur03] shows that the security problem is co-NP-complete for an unbounded number of sessions.

Note that the first result in formal analysis of security protocols that go beyond the perfect encryption hypothesis is certainly the one of S. Even *et al.* [EGS86], who are interested in RSA encryption. In particular, they deal with the multiplication operator $\times$ and properties such as the homomorphic properties of RSA, *i.e.* $\{x\}_k \times \{y\}_k = \{x \times y\}_k$. The authors also consider commutative encryption but in a restricted way, assuming that different modulus are used to generate different keys. More precisely, they consider commutative encryption between a key and its inverse, which can be encoded by the equations $\{\{x\}_k\}_{k^{-1}} = x$ and $\{\{x\}_{k^{-1}}\}_k = x$. They show that RSA properties are of no concern to the security of ping-pong protocols: if a ping-pong protocol is secure in the abstract model then its implementation using RSA is also secure.

In [RT01], M. Rusinowitch and M. Turuani consider involutive encryption, by adding some capabilities to the intruder: from the messages $\{\{m\}_k\}_k$, he can retrieve the plaintext m. This coding does not allow to capture the whole theory of involution since the rules can only be applied at the top of messages. However, such a theory falls into a particular class of equational theories that can be treated by a generic result of S. Delaune and F. Jacquemard [DJ04]. This result is detailed at the end of this section.

Exclusive or The $\oplus$ symbol denotes the binary operation called *exclusive or*, also denoted by `xor`. The properties of `xor` are:

$$\begin{array}{rcll} x \oplus (y \oplus z) & = & (x \oplus y) \oplus z & \text{(associativity)} \\ x \oplus y & = & y \oplus x & \text{(commutativity)} \\ x \oplus 0 & = & x & \text{(neutral element)} \\ x \oplus x & = & 0 & \text{(nilpotence)} \end{array}$$

This operation is used in many protocols and has aroused a lot of interest during the last years. H. Comon-Lundh and V. Shmatikov present in [CLS03] a decision procedure based on constraint solving techniques for solving the insecurity of cryptographic protocols employing `xor`. In [CKRT03], Y. Chevalier *et al.* improve this result by abstracting from intruder rules using *so-called* oracle rules, *i.e.* deduction rules that satisfy some conditions. As an instance of the general framework, they obtain that the insecurity problem is in NP for a large class of protocols in case of an intruder who can exploit the properties of the `xor` operator.

In [CLC03], H. Comon-Lundh and V. Cortier prove some decidability results of an extension of the Skolem class of first-order logic for the equational theory of `xor`. As an application, they get a decidability result in formal analysis of security protocols in presence of an unbounded number of sessions. They assume a finite number of nonces (which is a correct abstraction) and suppose that at each transition an agent may copy at most one unknown component of the received message. For an other subclass of first-order logic, corresponding to two-way tree automata, K. Verma [Ver03] also gives a decidability result.

The Casper tool [Low97] enables to consider *exclusive or* in the case of Vernam encryption. The Vernam encryption of m by m' is simply $m \oplus m'$. G. Lowe models it by adding new deduction rules to the intruder. For example, the intruder is able to get m from $m \oplus m'$ and m', to get m' from $m \oplus m'$ and m, and to get $m \oplus m'$ from $m' \oplus m$. Using this tool, he retrieves a known flaw on the TMN protocol [TMN89] and proves the security of an improved version.

Abelian Groups The $\times$ symbol denotes the multiplicative binary operation of Abelian groups. The properties of Abelian groups are:

$$\begin{array}{rcll} x\times(y\times z) & = & (x\times y)\times z & \text{(associativity)} \\ x\times y & = & y\times x & \text{(commutativity)} \\ x\times 1 & = & x & \text{(neutral element)} \\ x\times x^{-1} & = & 1 & \text{(inverse)} \end{array}$$

The intruder deduction problem can be decided in non-deterministic polynomial time in the case of Abelian groups. This result has been shown by H. Comon-Lundh and V. Shmatikov in [CLS03].

The security problem has been investigated by J. Millen and V. Shmatikov in [MS03, Shm04]. They present in [MS03] a constraint solving technique that reduces the problem to a system of quadratic Diophantine equations, but decidability of such equations remains an open question. However in [Shm04], V. Shmatikov succeeds in reducing the initial problem to the solvability of a particular system of quadratic Diophantine equations, proving in this way the decidability of the insecurity problem for a bounded number of sessions.

For an unbounded number of sessions and a finite number of nonces, K. Verma [Ver03] gets a decidability result for a restricted class of protocols, corresponding to protocols which can be encoded using two-way tree automata. He extends this work to two other equational theories: the first one defined by the three first equations (associativity, commutativity and neutral element) and the second one defined by the same three first equations plus the equations $-(x+y)=(-x)+(-y)$ and $-(-x)=x$.

Homomorphism We consider operators that verify equalities of the form $f(g(x,y))=g(f(x),f(y))$. In [CLT03], H. Comon-Lundh and R. Treinen investigate for which class of equational axioms the standard intruder model can be extended such that the intruder deduction problem is decidable. As an instance of this general framework, they obtain that the intruder deduction problem is decidable in PTIME, in presence of the following homomorphism property:

$$\{\langle u,v\rangle\}_k=\langle\{u\}_k,\{v\}_k\rangle.$$

This property is in particular verified when using the Electronic Code Book (ECB) mode. The ECB mode is the most obvious way to extend a block cipher to a text of arbitrary length. A block cipher encrypts plain text in fixed-sized-n-bits blocks (often n=64). For messages exceeding n bits, ECB consists in partitioning the message into n-bits blocks and encrypting each of them separately.

Homomorphism where the g operator has associativity, commutativity and/or nilpotence, and neutral elements has also been considered by P. Lafourcade, D. Lugiez, and R. Treinen [LLT04]. More precisely, they have shown that the intruder deduction problem is: NP-complete when g is an AC operator; EXPTIME when g is the *exclusive or* operator; EXPTIME when g is the operator of an Abelian group. More details may be found in Section 3.1.

In his thesis [Che03], Y. Chevalier shows that the insecurity problem remains NP-complete, provided that the blocks to which the homomorphism rule is applied are fixed in advance.

Prefix The prefix property is the ability of an intruder to get from an encrypted message the encryption of any of its prefixes: from a message $\{x,y\}_z$, he can deduce the message $\{x\}_z$. This property strongly depends on the encryption algorithm. For example, the ECB algorithm, presented in the previous paragraph, has this property. However; the ECB algorithm is not commonly used. A relatively good method of encrypting several blocks of data is Cipher Block Chaining (CBC). In such a system, the encryption of message block sequence $P_1P_2\cdots P_n$ (where some bits may be added to

P_n such that every block has the same length) with the key K is $C_0C_1C_2\cdots C_n$ where $C_0 = I$ (initialization block) and $C_i = \{C_{i-1} \oplus P_i\}_K$. The CBC encryption system has the following property: if $C_0C_1C_2\cdots C_iC_{i+1}\cdots C_n = \{P_1P_2\cdots P_iP_{i+1}\cdots P_n\}_K$ then $C_0C_1C_2\cdots C_i = \{P_1P_2\cdots P_i\}_K$, that is to say an intruder can get $\{x\}_z$ from $\{x,y\}_z$ if the length of x is a multiple of the block length used by the cryptographic algorithm.

The framework developed in [CKRT03] by Y. Chevalier *et al.* to study the `xor` theory can also be applied to model an intruder that may exploit the prefix property. They show that in this case the insecurity problem remains NP-complete, provided that the blocks to which the prefix rule is applied are fixed in advance.

Abelian Groups and Modular Exponentiation The $\times$ symbol denotes the multiplicative binary operation of Abelian groups. The properties of the Abelian groups and modular exponentiation theory are those of Abelian groups extended with:

$$\begin{aligned} exp(exp(x,y),z) &= exp(x, y \times z) \\ exp(x,1) &= x \end{aligned}$$

This theory allows to take into account simple properties of product and exponentiation operators, such as those of RSA and Diffie-Hellman exponentiation, which are widely used in protocol constructions. All the results described below assume that the Abelian groups operator (multiplication) appears only in the exponents. In particular, exponentials are not multiplied with each other. This restriction is necessary to obtain decidability results. Indeed, D. Kapur *et al.* [KNW02, KNW03] has shown the undecidability of the unification modulo the theory of exponentiation when the distributivity property of exponentiation over multiplication is assumed. Now, if unification is undecidable, the security problem is undecidable too since the unification problem of two terms $(u[x_1,\dots,x_n], v[x_1,\dots,x_n])$ with variables $x_1,\dots,x_n$ can be reduced to the security problem of the following protocol when requiring the secrecy of s:

$$\begin{array}{rll} A & \to B: & M_1,\dots,M_n \qquad (A \text{ sends a sequence of } n \text{ messages.}) \\ B: x_1,\dots,x_n & \to A: & \{u[x_1,\dots,x_n], v[x_1,\dots,x_n]\}_k \qquad (k \text{ is a fresh key}) \\ & & (B \text{ answers by instantiating the variables } x_1,\dots,x_n \text{ by the messages sent by } A) \\ B: \{x,x\}_k & \to A: & s \end{array}$$

J. Millen and V. Shmatikov mentioned in [MS03] that the security problem is still an open problem even in case of a fixed generator of the group.

Partial results for protocol analysis have been obtained by M. Boreale and M.G. Buscemi [BB03] and Y. Chevalier *et al.* in [CKR+03a]. The decision procedure of [BB03] requires an *a priori* upper bound on the number of factors in each product, and they do not provide a complexity result. Y. Chevalier *et al.* [CKR+03a] prove that the security problem is co-NP-complete in the presence of Abelian groups and modular exponentiation from arbitrary bases, but under some restrictions on how agents and intruder learn information in products of exponents.

In [Shm04] the security problem is reduced to the (in)solvability of a special decidable system of quadratic equations in the domain of the integers, providing the expected decidability result without the restrictions described previously.

J. Goubault-Larrecq *et al.* [JGLV04] have developed a practical implementation to verify cryptographic protocols using modular exponentiation on a fixed generator g. This operator is modeled by adding a free symbol *exp* and an associative and commutative operator $\times$. The term $exp(M_1 \times M_2)$

represents the exponentiation $(g^{M_1})^{M_2}$. To represent the ability for an attacker to raise $exp(M_1)$ to the M_2-th power, a deduction rule is added: knowing $exp(M_1)$ and M_2, any agent or attacker can deduce $exp(M_1 \times M_2)$. Using these correct abstractions, they verify the IKA.1 group-key agreement protocol [AST00] for up to 4 principals.

C. Meadows has also enriched her NRL protocol analyzer [Mea96] in order to verify group key protocols using Diffie-Hellman exponentiation. She has in particular analyzed the AGDH-2 protocol [Mea00], very similar to the IKA.1 protocol.

Timestamps Timestamps are often used in cryptographic protocols to prevent replay of messages communicated in the past. However, in most of the existing verification methods and decidability results for cryptographic protocols, timestamps are replaced by nonces because of the complexity of the verification of time-dependent protocols. As a consequence, temporal properties of timestamps are not taken into consideration. Most of the works on timed cryptographic protocols uses theorem-provers [ES00a] or finite-state model-checkers [Low97]. While the first ones need human help, the second ones rely on typing assumptions and assumption on the time window to bound the search space.

More recently, some automatic procedures for proving secrecy have been proposed [DG04, BEL04b] to deal with time-dependent cryptographic protocols. G. Delzanno and P. Ganty's approach [DG04] is based on data structures for symbolically representing sets of configurations of an arbitrary number of parallel protocol sessions. Since verification of secrecy for unbounded protocols is undecidable, termination of state exploration cannot be guaranteed in general. In [BEL04b], L. Bozga *et al.* present a model inspired by timed automata and a symbolic decision procedure to deal with bounded time-dependent cryptographic protocols. Their approach provides an algorithm (and hence a decidability result) for checking security properties of timed cryptographic protocols for a bounded number of sessions.

Classes of Equational Theories We have seen that there exist a lot of decision results when considering a fixed equational theory corresponding to a fixed intruder power. However, some papers [DJ04, AC04] propose generic decision procedures to solve useful problems for verification of security protocols. These procedures can be applied to any model provided that they can be axiomatized by a convergent rewriting system verifying some syntactic conditions. In [AC04], M. Abadi and V. Cortier show that the problems of deducibility and indistinguishability (static equivalence) are both decidable in PTIME for convergent subterm theories, *i.e.* theories described by sets of equations $M = N$ where N is a subterm of M. Simultaneously, S. Delaune and F. Jacquemard [DJ04] prove that the insecurity problem (in presence of an active attacker) is decidable in non-deterministic polynomial time for a class of equational theories which is slightly more restrictive than convergent subterm theories. The class of rewriting systems which are in the scope of these results contains the standard Dolev-Yao theory of [DEK83] and other relevant theories like the theory of involution which is mentioned in [RT01]. Moreover the use of explicit destructors such as decryption and projection operators allows to specify protocols (see [DJ04] for examples) without assuming some kind of integrity checks. An ongoing work, presented in [CL04] by H. Comon-Lundh, consists in proving a generic decidability result for the security problem in presence of a finite number of sessions, parametrized by the equational theory. This generic result requires some conditions on the equational theory.

Some recent works of J. Millen and C. Lynch [Mil03, LM04] compare the approach using explicit destructors to the standard one for the case of the decryption operator, and they give conditions under which security for the free algebra implies security for the rewrite rule model. In order to render the analysis done by formal methods closer to more concrete cryptographic models, C. Meadows suggests in [Mea03] a hierarchy of models at varying degrees of abstraction. In such an approach, the choice of the model in which the analysis is performed depends on the conditions verified by the protocol under consideration.

After this overview on existing results regarding analysis of cryptographic protocols, we focus on results recently obtained in the PROUVÉ project.

3 Results obtained within the RNTL Project PROUVÉ

3.1 Intruder Deduction for *AC*-like Equational Theories with Homomorphisms

We investigate the intruder deduction problem in presence of several variants of the equational theory of associativity and commutativity (*AC* in short) of a binary operator $\oplus$, plus the homomorphism property of a binary function symbol over the *AC* operator. The variants of *AC* that we consider are:

- Associativity and commutativity of the binary operator $\oplus$, plus homomorphism of an unary function symbol f with respect to $\oplus$, we denote the following equational theory by *ACh*:
  - $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativity)
  - $x \oplus y = y \oplus x$ (Commutativity)
  - $f(x \oplus y) = f(x) \oplus f(y)$ (Homomorphism)
- Associativity, commutativity, and nilpotence of the binary operator $\oplus$ and existence of a zero element for $\oplus$ that is the theory of *exclusive or*, plus homomorphism of a unary function symbol f with respect to $\oplus$, we denote the following equational theory by *ACUNh*:
  - $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativity)
  - $x \oplus y = y \oplus x$ (Commutativity)
  - $0 \oplus x = x$ (Unit)
  - $x \oplus x = 0$ (Nilpotence)
  - $f(x \oplus y) = f(x) \oplus f(y)$ (Homomorphism)
- Abelian groups, plus homomorphism of a unary function symbol with respect to the group operator, we denote the following equational theory by *AGh*:
  - $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativity)
  - $x \oplus y = y \oplus x$ (Commutativity)
  - $0 \oplus x = x$ (Unit)
  - $x \oplus I(x) = 0$ (Inversion)
  - $f(x \oplus y) = f(x) \oplus f(y)$ (Homomorphism)

We are furthermore interested in the combination of these *AC*-like theories with a generalization of one homomorphic function to some form of distributivity of the encryption operator over the binary operator $\oplus$. We are interested in decidability and complexity of the intruder deduction problem in any of these equational theories, that are the most frequent axioms arising in cryptographic protocols. We follow the approach of [CLS03] and [CLT03] which consists in adapting Mc Allester's *locality* method. The full version of this work can be found in [LLT04].

3.1.1 Extended Dolev-Yao Model

We consider the classic model of deduction rules introduced by Dolev and Yao [DY83] in order to model the deductive capacities of a passive intruder. In this model, an intruder may use any term he has previously observed on the network, and construct new terms by pairing, unpairing, using a free constructor, encryption and decryption, where in the last two cases the intruder also has to know the encryption key.

We are in particular interested in the case where the equational theory contains the axioms of *associativity and commutativity* (*AC*) of a distinguished binary function symbol. We separate the construction rule for this binary *AC* operator from the construction rule for the free function symbols, and furthermore generalize this rule into a vary-aric rule.

We introduce a variant of the extended Dolev-Yao model for the case where the equational theory can be presented by a convergent term rewriting system modulo a background equational theory (which usually is *AC*).

The three theories defined previously can be presented by a convergent term rewriting system modulo *AC*, and hence we can work with normal forms of terms modulo the background theory.

3.1.2 General Locality

Our starting point is the locality technique introduced by David McAllester [McA93]. He considers deduction systems which are represented by finite sets of Horn clauses. He shows that if a deduction system has the so-called *locality property* then there is a polynomial-time algorithm to decide the deducibility of a term w from a finite set of terms T_0. A deduction system has the *locality property* if any proof can be transformed into a local proof. A *local proof* is a proof where all the nodes are syntactic subterms of T_0 and w. There are two main restrictions in this approach :

- The deduction system must be finite.
- The notion of locality is restricted to syntactic subterms.

These restrictions pose a serious problem when we are working modulo *AC*, as it is already observed in [CLS03]: there is in general an exponential number of subterms modulo *AC* of a given term. The solution proposed in [CLS03], and which we also adopt here, is to use a construction of $\oplus$ with an arbitrary number of hypotheses. In this way, we can avoid the exponential number of subterms. However, we are now struck with an infinite number of different deduction rules. Fortunately, we can still obtain a polynomial algorithm adapting the McAllester algorithm by deciding one-step deducibility in our adapted algorithm, as explained in Section 3.1.4. The notion of locality restricted to syntactic subterms is not sufficent in our three cases. We have to define in each cases specific notions of subterms, which give us the complexity of the intruder problem in each studied theories.

3.1.3 Locality Results

In our three cases we define an operator S to generate the set of the "subterms" and prove the S-locality. We make the distinction between the *binary case* (each sum contains only two arguments) and general case. We prove in each case the theorem of locality and indicate here the complexity of the construction of S for a proof P of $T \vdash w$. In the *ACh* case the construction of the set of the subterms is in PTime in size of $T \cup \{w\}$. In the binary case for *ACUNh* and *AGh* we obtain a construction in PTime in size of $T \cup \{w\}$, using a pumping lemma for one-counter automata. However in the general case for *ACUNh* and *AGh* generate the set of the subterms is EXP-time in size of $T \cup \{w\}$, more details are given in [LLT04]

3.1.4 One-step Deducibility

We show how to decide one-step deducibility for our three cases. If the equational theory is just *AC* and if we consider the case of a binary proof, then one-step deducibility is obviously in PTime. We transform the problem of testing one-step deducibility into solvability of a system of linear Diophantine equations. The domain over which this system of equations is to be solved depends on the equational theory considered:

- *ACh*: Solvability of a system of linear equations over $\mathbb{N}$ is a NP-complete problem [Pap94].
- *ACUNh*: Solvability of a system of linear Diophantine equations over $\mathbb{Z}/2\mathbb{Z}$ is in PTIME [KKS87].
- *AGh*: Solvability of a system of linear Diophantine equations over $\mathbb{Z}$ is in PTIME [Sch86].

3.1.5 Extension

The homomorphism law is now replaced by a law stating that the encryption of the $\oplus$ of two messages is equal to the $\oplus$ of the encryptions of the two messages using the same encryption key. More precisely, we consider the equation $\{x \oplus y\}_k = \{x\}_k \oplus \{y\}_k$. This can be seen as the extension to an infinite family of homomorphisms, one for each possible encoding key. All results of the previous sections carry over to this extension.

3.1.6 Results

A summary of the results obtained on the complexity of the intruder deduction system modulo *AC*-like equational theories with homomorphism is given in the following table. The same results are obtained for our extension. The results for homomorphism only (without *AC* axioms) have been shown in a different paper [CLT03] and are here cited only for completeness.

Intruder deduction problem

	Binary case	**General case**
Homomorphism	*PTIME [CLT03]*	
ACh	*PTIME*	*NP-Complete*
ACUNh	*PTIME*	*EXPTIME*
AGh	*PTIME*	*EXPTIME*

3.2 Deciding Knowledge and Static Equivalence under Equational Theories

We study the decidability of message deduction and static equivalence. We define a relation $\phi \vdash M$ denoting that M can be deduced from ϕ, and a relation $\varphi \approx_s \psi$ denoting that φ and ψ are statically equivalent; here ϕ, φ, and ψ are all essentially lists of messages, each with a name, represented by formal expressions. Our first main positive results assume only that the equational theory is defined by a convergent rewriting system with a finite number of rules of the form $M \to N$ where N is a proper subterm of M or a constant symbol. Such theories, which we call convergent subterm theories, appear frequently in applications. For them, we obtain that both $\phi \vdash M$ and $\varphi \approx_s \psi$ are decidable in polynomial time. Details may be found in [AC04]. Then we have extended these results to more general theories with AC-symbols like the theory of the *exclusive or* and of modular exponentiation.

3.2.1 Definitions

After a protocol execution, an attacker may know a sequence of messages $M_1, \ldots, M_l$. This means that he knows each message but he also knows in which order he received the messages. So it is not enough for us to say that the attacker knows the set of terms $\{M_1, \ldots, M_l\}$. Furthermore, we should distinguish those names that the attacker had before the execution from those that were freshly generated and which may remain secret from the attacker; both kinds of names may appear in the terms.

In the applied pi calculus [AF01], such a sequence of messages is organized into a *frame* $\nu\widetilde{n}\sigma$, where $\widetilde{n}$ is a finite set of names (intuitively, the fresh names), and σ is a substitution of the form:

$$\{^{M_1}/_{x_1}, \ldots, ^{M_l}/_{x_l}\} \quad \text{with} \quad dom(\sigma) \stackrel{\text{def}}{=} \{x_1, \ldots, x_l\}.$$

The variables enable us to refer to each M_i, for example for keeping track of their order of transmission. We always assume that the terms M_i are closed.

Given a frame ϕ that represents the information available to an attacker, we may ask whether a given closed term M may be deduced from ϕ. This relation is written $\phi \vdash M$ (following Schneider [Sch96]). It is axiomatized by the rules:

$$\frac{}{\nu\widetilde{n}\sigma \vdash M} \text{ if } \exists x \in dom(\sigma) \text{ s.t. } x\sigma = M \qquad \frac{}{\nu\widetilde{n}\sigma \vdash s} \; s \notin \widetilde{n}$$

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

Deduction does not always suffice for expressing the knowledge of an attacker, as discussed in the introduction. For example, consider $\phi_1 \stackrel{\text{def}}{=} \nu k\{\mathsf{enc}(0,k)/x, k/y\}$ and $\phi_2 \stackrel{\text{def}}{=} \nu k\{\mathsf{enc}(1,k)/x, k/y\}$, where $0, 1 \in \Sigma$ are constant symbols. The attacker can deduce the same set of terms from these two frames since he knows 0 and 1. But he could tell the difference between these two frames by checking whether the decryption of x with y produces 0 or 1.

We say that two terms M and N are equal in the frame φ for the equational theory E, and write $(M =_E N)\varphi$, if and only if $\varphi = \nu\widetilde{n}.\sigma$, $M\sigma =_E N\sigma$, and $\{\widetilde{n}\} \cap (fn(M) \cup fn(N)) = \emptyset$ for some names $\widetilde{n}$ and substitution σ. Then we say that two frames φ and ψ are *statically equivalent*, and write $\varphi \approx_s \psi$, when $dom(\varphi) = dom(\psi)$ and when, for all terms M and N, we have $(M =_E N)\varphi$ if and only if $(M =_E N)\psi$. We write $\approx_{sE}$ when E is not clear from the context.

In our example, we have $(\mathsf{dec}(x,y) =_{E_1} 0)\phi_1$ but not $(\mathsf{dec}(x,y) =_{E_1} 0)\phi_2$. Therefore, $\phi_1 \not\approx_s \phi_2$ although $\nu k\{\mathsf{enc}(0,k)/x\} \approx_s \nu k\{\mathsf{enc}(1,k)/x\}$.

3.2.2 Results

We have shown that $\vdash$ and $\approx_s$ may be undecidable in general; that the decidability of $\vdash$ reduces to the one of $\approx_s$ but that the converse does not hold: there exists an equational theory such that $\vdash$ is decidable and $\approx_s$ is not.

To obtain decidability results, we have first considered *subterm theories*, defined by a finite set of equations of the form $M = N$ where N is a proper subterm of M or a constant symbol. The definition of subterm theories is almost vacuous on its own. Even equality may be undecidable for subterm theories. Any equational theory defined by a finite set of equations $M = M'$ with variables can be encoded as a subterm theory, with the two equations:

$$\mathsf{Whichever}(M, M') = M \qquad \mathsf{Whichever}(M, M') = M'$$

for each original equation $M = M'$. In light of this encoding, we should add the assumption that, by orienting the equations that define a subterm theory from left to right, we obtain a convergent rewriting system:

Definition 1 *A equational theory E, defined by a finite set of equations $\bigcup_{i=1}^{n}\{M_i = N_i\}$ where $fn(M_i) = fn(N_i) = \emptyset$, is a* convergent subterm theory *if the set of rewriting rules $r(E) \stackrel{def}{=} \bigcup_{i=1}^{n}\{M_i \to N_i\}$ is convergent and if each N_i is a proper subterm of M_i or a constant. We write $U \to V$ if U and V are closed terms and U may be rewritten to V (in one step) using a rule of $r(E)$.*

Examples Important destructor-constructor rules like those for pairing, encryption, and signature may be expressed in subterm theories (typically convergent ones):

$$\begin{array}{rclrcl} \mathsf{fst}(<x,y>) & = & x & \mathsf{dec}(\mathsf{enc}(x,y),y) & = & x \\ \mathsf{snd}(<x,y>) & = & y & \mathsf{check}(x,\mathsf{sign}(x,\mathsf{sk}(y)),\mathsf{pk}(y)) & = & \mathsf{ok} \end{array}$$

Additional examples can be found in previous work (e.g., [AF01, Bla04a]). Convergent subterm theories also enable us to capture sophisticated but sensible properties, as in:

$$\begin{array}{ll} E_4: & \{I(I(x)) = x, I(x) \times x = 1, x \times I(x) = 1\}, \\ E_5: & \{h(h(x)) = x\}, \\ E_6: & \{\mathsf{enc}(\mathsf{enc}(x,y),y) = x\}. \end{array}$$

The theory E_4 models an inverse function. The theory E_5 models a hash function that is idempotent on small inputs (since the hash of a hash gives the same hash). The theory E_6 represents an encryption function that also decrypts: the encryption of a plaintext, twice with the same key, returns the plaintext.

Decidability of $\vdash$ and $\approx_s$ For convergent subterm theories, both $\vdash$ and $\approx_s$ become decidable.

Theorem 1 *Let E be a convergent subterm theory. For any frames ϕ and ϕ', for any closed term M, we can decide $\phi \vdash M$ and $\phi \approx_s \phi'$ in polynomial time in $|\phi|$, $|\phi'|$, and $|M|$.*

3.2.3 Extensions

We relax our two hypotheses on equational theories: convergence and subterm property. Instead of considering convergent theories, we consider equational theories E with some associative and commutative symbols, that come with a rewriting system $\mathcal{R}$ such that a $\mathcal{R}$ is convergent modulo AC rewriting. Moreover, instead of considering a syntactic condition (subterm property) on the equational theory, we define a property of saturation for each frame ϕ. Several equational theories fit this generalization, like the homomorphism equational theory:

$$\left\{ \begin{array}{rclrcl} \mathsf{fst}(\langle x,y\rangle) & = & x, & \mathsf{dec}(\mathsf{enc}(x,y),y) & = & x, \\ \mathsf{snd}(\langle x,y\rangle) & = & y, & \mathsf{enc}(\langle x,y\rangle,z) & = & \langle \mathsf{enc}(x,z),\mathsf{enc}(y,z)\rangle \end{array} \right\},$$

the pure AC theory:

$$\bigcup_{i=1}^{k}\{(x \oplus_i y) \oplus_i z = x \oplus_i (y \oplus_i z), x \oplus_i y = y \oplus_i x\},$$

and the *exclusive or*:

$$E_9 = \left\{ \begin{array}{rcl} x \oplus (y \oplus z) & = & (x \oplus y) \oplus z \\ x \oplus y & = & y \oplus x \\ x \oplus x & = & 0 \\ x \oplus 0 & = & x \end{array} \right\}.$$

3.3 Opacity and Guessing Attacks

We study an information flow property called opacity [Maz04c]. This notion, close to non-interference, means that an intruder is not able to deduce whether a certain property ψ is verified. The property ψ is defined on initial variables of the protocol. For example, ψ can have the form $v = 1$ where v is a vote that is casted during the protocol execution. The intruder observes a finite number of sessions in a passive way. For that purpose, we introduce a similarity relation over messages: $E \vdash m_1 \sim m_2$ if an intruder is not able to differentiate message m_1 from message m_2 by using knowledge E. Opacity is equivalent to satisfiability of constraints involving equalities and similarities. We have proven decidability of satisfiability for such constraints and so decidability of opacity when considering a passive intruder. Details can be found in [Maz04a].

3.3.1 Definitions

The intuitive definition of opacity is that an intruder is not able to distinguish a run where the property is satisfied from a run where it is not. To distinguish two messages, the intruder can decompose them, according to his knowledge but if he does not know the key k for example, he will not be able to make the difference between two different messages encoded by this key k. Two such messages will be called similar messages. This definition will be formalized using inference rules.

An *environment* is a finite set of closed messages. It usually denotes the set of messages known by the intruder. In what follows, we use symmetric key cryptography. However, all our results can easily be generalized to public key cryptography.

Definition 2 (Similar Messages) *Two closed messages m_1 and m_2 are said to be* similar *for the environment env iff $env \vdash m_1 \sim m_2$ where $\sim$ is the smallest (w.r.t set inclusion) binary relation satisfying:*

$$\frac{a \in Atoms}{a \sim a} \qquad \frac{u_1 \sim u_2 \quad v_1 \sim v_2}{\langle u_1, v_1 \rangle \sim \langle u_2, v_2 \rangle}$$

$$\frac{env \vdash k \quad u \sim v}{\{u\}_k \sim \{v\}_k} \qquad \frac{env \nvdash k \quad env \nvdash k'}{\{u\}_k \sim \{v\}_{k'}}$$

The environment name will be omitted when it is not relevant for comprehension.

Let us consider a protocol PR and one of its session σ (σ is a substitution which instantiates the initial variables of PR). We are interested in predicates over σ, namely properties ψ that act over variables instantiated by σ. Such properties may express the identity of an agent, or the value of a vote, for instance. The opacity problem considered here relies on three hypothesis: (1) The intruder C has a passive view of protocol session σ involving two agents A and B. Passive means that the intruder can intercept and view any messages exchanged by A and B but is not able to block, modify nor send any message. (2) The intruder knows the protocol PR. (3) The intruder has an initial knowledge c_0, which is a predicate (for example, $c_0 = (k_1 = k_2)$ means that C knows that the keys that will instantiate k_1 and k_2 are the same).

If we consider the witness run $run(P\sigma) = m_1.m_2...m_n$ then a property ψ is opaque if there exist two possible sessions σ_1 and σ_2 of the protocol giving messages similar (w.r.t. $\sim$) to the witness messages (m_1 to m_n) where for example, $\psi\sigma_1$ is true (i.e. property ψ is verified for initial variables defined by σ_1) and $\psi\sigma_2$ is false. In this case, the intruder will not be able to deduce any knowledge on $\psi\sigma$. Of course, there is no need to find both σ_1 and σ_2: if $\psi\sigma$ is true, then we could use σ instead of σ_1, as exchanged messages are the same and hence they are similar. But we will keep this notation with three substitutions to show the symmetry of this problem.

Definition 3 (Opacity) *A property ψ is said to be* opaque *for a protocol session σ of P iff there exist two sessions of the protocol σ_1 and σ_2 such that*

$$c_0\sigma_1 \wedge p_1 \sim m_1 \wedge \ldots \wedge p_n \sim m_n \wedge \psi\sigma_1 \qquad c_0\sigma_2 \wedge q_1 \sim m_1 \wedge \ldots \wedge q_n \sim m_n \wedge \neg\psi\sigma_2,$$

where $m_1.m_2...m_n$ is the run of protocol P related to σ, $p_1.p_2...p_n$ is related to σ_1 and $q_1.q_2...q_n$ is related to σ_2. Note that the three runs must have the same length n.

The environment *env* used in the previous conjunctions is $\{m_1,...,m_n,p_1,...,p_n,q_1,...,q_n\}$ and can be augmented with an initial knowledge of the intruder env_0.

3.3.2 Results

The former definition of opacity naturally leads to a class of predicates called *initial predicates*. These predicates involve both equalities and similarities but the logical negation is not allowed. The set *IP* of *initial predicates* is given by the following grammar where *m* and *n* are two messages:

$$P ::= P_A | P \wedge P \qquad P_A ::= m \sim n | m = n | \bot | \top$$

Then, it is possible to encode an opacity problem as an initial predicate. Checking opacity is reduced to checking satisfiability of the predicate.

Theorem 2 *Satisfiability of initial predicates is decidable.*

Hence opacity of a property ψ is decidable as soon as both ψ and $\neg\psi$ can be expressed as initial predicates. Note, that this decidability result does not allow any efficient checking as the proof uses a finite model argument and thus gives an algorithm with exponential complexity. However, when considering atomic keys, another algorithm can be used (this algorithm is described in [Maz04c]) which is efficient enough to be used in practical situations.

3.3.3 Adding Guessing Attacks

A description of guessing attacks can be found in [Low02] or in [GLNS93]. These attacks rely on a weakening of the perfect cryptography hypothesis: if a message is encoded two times with the same key, the results are exactly the same string of bits (i.e. the encoding does not add some random informations). Thus, if an intruder intercepts two times the same string of bits and this string corresponds according to the protocol first to $\{m_1\}_{k_1}$ then to $\{m_2\}_{k_2}$, the intruder is able to deduce $m_1 = m_2$ and $k_1 = k_2$.

To model this, it is possible to add syntactic equalities to our constraints. The set of "unifiable" messages corresponding to all possible guessing attacks can be computed in a finite time (we still assume that there is a bounded number of sessions). However, this last part is rather technical, hence the interested reader is referred to [Maz04c]. Using satisfiability of initial predicates, it is possible to decide opacity with guessing attacks.

3.3.4 Application to Electronic Voting

Let us consider the most simple electronic voting protocol. *A* is the voter and *S* the authority that counts the different votes. The objective is that the expressed vote remains opaque. The protocol is written $A \to S : \{v\}_{pub(S)}$ where *v* is chosen among the values *yes* and *no*. Let us suppose that the expressed vote is *yes*, so the substitution σ is defined by $\sigma = [v \backslash yes]$. Then the environment is the set $\{\{yes\}_{pub(S)}, yes, no, pub(S)\}$. We easily obtain the only couple of "unifiable" messages $(\{v\}_{pub(S)}, \{yes\}_{pub(S)})$. The constraints of opacity of property $\psi = (v = yes)$ are:

$$\{v\}_{pub(S)} = \{yes\}_{pub(S)} \wedge \{v\}_{pub(S)} \sim \{yes\}_{pub(S)} \wedge v = yes$$

$$\{v\}_{pub(S)} = \{yes\}_{pub(S)} \wedge \{v\}_{pub(S)} \sim \{yes\}_{pub(S)} \wedge v = no$$

Our decision procedure proves that the second constraint is not satisfiable (it leads to unify *yes* and *no*). Thus property ψ is not opaque: the intruder can guess the vote. Intuitively, the intruder generates messages $\{yes\}_{pub(S)}$ and $\{no\}_{pub(S)}$ and compares them to the intercepted message.

3.3.5 Extensions

The previous definition of opacity makes some restrictive hypothesis: it is only possible to consider a passive intruder. Hence a first extension for this work would be to add the general case of active intruders (as defined in Dolev-Yao model). In that case, we have proven a simplified version of opacity to be decidable [Maz04b] but we still do not have a proof for opacity in general. The main difficulty is that we have to model that the intruder forges its messages in the same way across the two sessions. This can be done using simultaneous deductibility: $E_1, E_2 \vdash m_1, m_2$ which means that m_1 can be deduced from E_1 and m_2 from E_2 by performing the same operations.

Another interesting extension would be to add an equational theory in the definition of similarity.

3.4 A Decision Procedure for Security Protocols with Explicit Destructors

We propose a decision procedure solving the problem of insecurity for cryptographic protocols containing explicit destructor symbols (like decryption and projection) and equality tests, in the presence of a bounded number of sessions and of a passive or an active intruder. The destructor operators are axiomatized by an arbitrary convergent rewrite system satisfying some syntactic restrictions. This approach, with parameterized semantics, allows us to weaken the security hypotheses for verification, *i.e.* to address a larger class of attacks than for models based on free algebra.

Our decision procedure is defined by an inference system for symbolic constraint solving based on basic narrowing techniques. It is polynomial in time in the case of a passive intruder and non-deterministic polynomial for an active intruder. Details may be found in [DJ04].

3.4.1 Protocol Model

We shall consider as a running example the following protocol for the exchange of a symmetric key K_{ab} between two participants A and B in an asymmetric cryptosystem. It is a simplified version of the Denning-Sacco key distribution protocol [DS81], omitting certificates and timestamps.

$$\begin{array}{llll} 0. & A & \rightarrow & B: \ \langle A, \{\{K_{ab}\}^a_{pub(A)^{-1}}\}^a_{pub(B)}\rangle \\ 1. & B & \rightarrow & A: \ \{secret\}^s_{K_{ab}} \end{array}$$

The symbols $\{\ \}^a$, $\{\ \}^s$, $^{-1}$, and $\langle\ \rangle$ represents respectively asymmetric and symmetric encryption, inverse and pairing. The function symbol *pub* associate to an agent name its public key.

In our model, a *protocol* is defined as a finite set of programs, each program being a finite sequence of *instructions* of the form $\mathsf{recv}(x); \mathcal{E}; \mathsf{send}(s)$ where x is a variable of a fixed set $\mathcal{X}$, the message s is represented as a first order term of $\mathcal{T}(\mathcal{F}, \mathcal{X})$ (built with symbols of a given signature $\mathcal{F}$ and variables of $\mathcal{X}$) and $\mathcal{E}$ is a set (possibly empty) of equations on terms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Note that we are able to deal with first-order equations, in order to specify explicitly some tests performed by the participants at some stage of the protocol. Moreover, some destruction operators may be used in the protocol specification in order to specify unambiguously the actions taken by the agents in protocol execution.

Example 1 *The Denning-Sacco protocol described above is made of two programs:*

$$\begin{array}{ll} \textit{A's role}: & \mathsf{recv}(x_0^0); x_0^0 = 0; \mathsf{send}\big(\langle x_A^0, \{\{x_{Kab}^0\}^a_{pub(x_A^0)^{-1}}\}^a_{pub(x_B^0)}\rangle\big) \\ & \mathsf{recv}\big(x_1^0\big); \mathsf{send}(0) \\ & \\ \textit{B's role}: & \mathsf{recv}\big(x_0^1\big); \mathsf{send}\big(\{x_S^1\}^s_{ad(ad(\pi_2(x_0^1), pub(x_B^1)^{-1}), pub(\pi_1(x_0^1)))}\big) \end{array}$$

The symbols $x_0^0, x_1^0, x_A^0 \ldots$ are all distinct variables of $\mathcal{X}$. Note the explicit use of the asymmetric decryption, denoted by ad, and left and right projection on pairs (π_1 and π_2 respectively) in order to specify the way that B build the answer to the received message x_0^1. The second instruction of program A implements simply the reception of the last message by A.

3.4.2 Equational theories

We assume given a set $\mathcal{R}$ of rewrite rules defining the semantics of the symbols of the signature $\mathcal{F}$ ($\mathcal{R}$ must fulfill some syntactic restrictions, see Section 3.4.5). Following the approach of applied pi-calculus [AF01] (see also Section 3.2 in this report) we are based on semantics for the protocol and the intruder where the messages (terms) are considered modulo the rewrite relation $\xleftrightarrow{*}_{\mathcal{R}}$ induced by $\mathcal{R}$. Moreover, the signature $\mathcal{F}$ is assumed partitioned into a subset $\mathcal{VF}$ of *visible* function symbols (the functions known to everybody, typically $\{\ \}^a$, $\{\ \}^s$, $\langle\ \rangle$, sd, ad, π_1, π_2) and a subset $\mathcal{PF}$ of *private* function symbols (typically $^{-1}$).

Our results are valid for a large class of equational theories which can be represented by a convergent *public-collapsing* rewriting system. The rules of such systems must have the form $\ell \to x$ with $x \in \mathcal{X}$ or $\ell \to a$ where a is a constant symbol of $\mathcal{F}$, with additional technical restrictions on ℓ. We present below several theories, relevant to cryptographic protocols verification, which fall in this class.

Dolev-Yao theory. The following TRS corresponds to the theory of [DY83] for public key encryption. This theory has been studied in many works but the use of explicit decryption and projections symbols and equations in protocol specifications allows to generalize other approaches.

$$\begin{array}{ll} sd(\{x\}^s_y, y) \to x, & ad(\{x\}^a_y, y^{-1}) \to x, \\ x^{-1^{-1}} \to x, & ad(\{x\}^a_{y^{-1}}, y) \to x, \\ \multicolumn{2}{c}{\pi_i(\langle x_1, x_2\rangle) \to x_i \ (i = 1, 2)} \end{array}$$

Inverse-key theory. The three following rules extend the Dolev-Yao theory:

$$\{sd(x,y)\}^a_y \ \to \ x, \{ad(x,y)\}^a_{y^{-1}} \to x, \{ad(x,y^{-1})\}^a_y \to x$$

.

They are useful when we assume that decryption is just an encryption with the inverse key like for the cryptosystem RSA.

3.4.3 Protocol Model

We are interested in the verification of systems in which *agents* follow the programs of the given protocol and communicate through an insecure network. We assume a finite number of sessions. In our model, the set of messages deducible by the network is modeled by an (infinite) set N of ground terms of $\mathcal{T}(\mathcal{F})$ (terms without variables).

The agent's step corresponding to the execution of the instruction $\mathsf{recv}(x);\mathcal{E};\mathsf{send}(s)$ consists in reading a message x in N, verifying the equations of $\mathcal{E}$, and, in case of success, adding the answer s to N. After every such step, we assume that the intruder is able to read any message in N (*passive* behavior), to deduce new messages from the read messages, and, sometimes, to send some message deduced under a fake identity. In the latter case, we call him an *active* intruder. The deduction abilities of the intruder are formally defined as follows: given a set of ground terms T, the set of terms deducible from T, also called *intruder set*, $I_{\mathcal{R}}(T)$ is the smallest set of ground terms containing T, closed under $\xleftrightarrow{*}_{\mathcal{R}}$, and such that for all $t_1,\ldots,t_n \in I_{\mathcal{R}}(T)$ and all $f \in \mathcal{VF}$ of arity n, $f(t_1,\ldots,t_n) \in I_{\mathcal{R}}(T)$.

The problem of *insecurity* is, given a protocol, a bounded number of agents, a finite sequence I of agent's instructions (called *interleaving*) and a secret data $s \in \mathcal{T}(\mathcal{F})$ to decide whether the intruder can learn s when the agents run in the order defined by I.

3.4.4 Verification of Cryptographic Protocols via Constraint Solving

In this section, we show how the problem of insecurity can be reduced to solving systems of intruder constraints and equations.

We shall consider below constraint systems which are made up of intruder constraints and equations:

1. An equation is denoted by $t_1 = t_2$ and a $\mathcal{R}$-solution of an equation is a grounding substitution σ such that $t_1\sigma \xleftrightarrow{*}_{\mathcal{R}} t_2\sigma$.

2. An *intruder constraint* is a tuple written $t_1,\ldots,t_n \Vdash r$ and a $\mathcal{R}$-solution of an intruder constraint is a grounding substitution σ such that $r\sigma \in I_{\mathcal{R}}(t_1\sigma,\ldots,t_n\sigma)$.

In the case of a passive intruder, given an instance of the insecurity problem, all the messages are sent by the agents following the given interleaving. Hence, these messages are ground terms, computable from the instance of the problem, and the problem is reducible to the satisfiability of one ground intruder constraints. In other terms, we are reduced to check whether $s \in I_{\mathcal{R}}(t_1,\ldots,t_n)$ where $t_1,\ldots,t_n$ are the messages sent by the agents.

When the the intruder is active, he is able to send some messages to the network during an attack, and these messages cannot be known in advance (from an instance of the insecurity problem). Such intruder messages are represented by variables in right sides of intruder constraints. More precisely, we construct one constraint for each step of the given interleaving I (plus one additional constraint expressing that the secret is compromised) whose left side is the set of the messages sent by the agents at previous steps.

Example 2 *We consider the protocol of Example 1 involving only one agent b. Assume that the intruder's initial knowledge is $T_0 = \{0,a,b,pub(a),pub(b)\}$, and that $\mathcal{R}$ is the standard Dolev-Yao theory defined in Section 3.4.2. There can be only one interleaving of length one. The set of intruder constraints and equations $\mathcal{C}$ associated to this interleaving is:*

$$\left\{T_0 \Vdash x_0^1; T_0, \{s\}_{ad(ad(\pi_2(x_0^1),pub(b)^{-1}),pub(\pi_1(x_0^1)))} \Vdash x; x = s\right\}$$

We can check that $\sigma = \{x_0^1 \mapsto \langle a, \{0\}^a_{pub(b)}\rangle, x \mapsto s\}$ *is a* $\mathcal{R}$*-solution of* $\mathcal{C}$.

This corresponds to the following attack: the intruder, claiming to be a, sends to b the "message" $\langle a, \{0\}^a_{pub(b)}\rangle$*. The answer of b is then:*
$\{s\}_{ad(ad(\pi_2(\langle a,\{0\}^a_{pub(b)}\rangle),pub(b)^{-1}),pub(\pi_1(\langle a,\{0\}^a_{pub(b)}\rangle)))} \xrightarrow{*}_{\mathcal{R}} \{s\}^s_{ad(0,pub(a))}$ *and s is revealed since the encryption key* $ad(0,pub(a))$ *belongs to* $I_{\mathcal{R}}(T_0)$.

3.4.5 Results

We have shown first that the problem $s \in I_{\mathcal{R}}(t_1,\ldots,t_n)$ where $s,t_1,\ldots,t_n$ are ground terms is decidable in polynomial time. The proof is base on a locality result (all the terms involved in the proof of the problem are subterms of $s,t_1,\ldots,t_n$) which permits a reduction to the satisfiability of ground Horn clauses. As a consequence:

Theorem 3 *The insecurity problem is decidable in polynomial time in case of a passive intruder.*

This result can be related to one result presented in Section 3.2. However, the class of equational theories that we consider here is slightly less general than the one of Section 3.2.

We have proposed a non-deterministic polynomial time resolution procedure for the problem of satisfiability of some sets of constraints called *well-formed*. The procedure is based on narrowing techniques with a basic strategy. Its completeness follows from an highly technical lemma which in some sense lift the above locality result from the ground case to the case of constraints with variables.

Theorem 4 *The insecurity problem is NP-complete in case of an active intruder.*

The hardness part is shown by a reduction of 3-SAT.

3.5 Timestamps

Some cryptographic protocols rely upon timestamps that recipients use to verify timeliness of received messages or to identify replays of messages. In some applications where resources are critical, timestamps are used as an approximative implementation of nonces. However, one should be aware that timestamps are ordered and guessable which is not the case of nonces at least in the Dolev-Yao model. More generally, timing constraints can also be used to implement time outs that can be used, e.g. as protection against deny of service attacks.

3.5.1 The Denning-Sacco key distribution protocol

Let us consider an example of a protocol that relies upon timing constraints. The Denning-Sacco shared key protocol [CJ97] is a protocol for distribution of a shared symmetric key generated by a trusted server. Timestamps are used to ensure the freshness of the supposedly generated shared key. Using the usual notation for cryptographic protocols, the protocol can be described as follows:

$$\begin{array}{lr} A \to S: & A,B \\ S \to A: & \{B,K_{ab},T,\{K_{ab},A,T\}_{K_{bs}}\}_{K_{as}} \\ A \to B: & \{K_{ab},A,T\}_{K_{bs}} \end{array}$$

The keys K_{as} and K_{bs} are shared keys between the participant A, respectively B, and the server S.

Each participant of the protocol may be seen as a sequential process as shown in Figure 1. First, the participant A sends his identity A and the identity of B to the server. Then, A receives back the message $\{B,x,T_1,y\}_{K_{AS}}$. If T_1, is "timely", i.e. the difference between the current time and the value of T_1 is less than the constant parameter δ_1 then A accepts x as session key and forwards the message y to B. We use a special clock *now*. Thus, timeliness of message $\{B,x,T_1,y\}_{K_{AS}}$ is verified by the constraint $now - T_1 < \delta_1$. On the other side, when B receives the message $\{p,u,T_2\}_{K_{BS}}$, he checks that T_2 is "timely" and, if so, it accepts p as session key. The server S, every time it receives a pair of two participants (z,v) it generates a new session key K and sends the message $\{v,K,now,\{K,z,now\}_{K_{vS}}\}_{K_{zS}}$ to the first participant z. In this message, the value of clock *now* is the value of the time at which the message is sent.

A:	S:
	$?(z,v)$;
$!(A,B)$;	$!\{v,K,now,\{K,z,now\}_{K_{vS}}\}_{K_{zS}}$
$now - T_1 < \delta_1 \rightarrow$	
$\quad ?\{B,x,T_1,y\}_{K_{AS}}$;	B :
$!y$	$now - T_2 < \delta_2 \rightarrow ?\{u,p,T_2\}_{K_{BS}}$

Figure 1: Roles as processes

3.5.2 Results

Most of the existing verification methods and decidability results for cryptographic protocols consider time-independent protocols.

In [BEL05], we present a model for time-dependent cryptographic protocols and a corresponding decidability result. Although the model we present only deals with bounded protocols, that is a fixed number of sessions are considered, our model clearly identifies the main ingredients to be included in a general model. It is well-known that the verification problem of unbounded cryptographic protocols is undecidable in the untimed case, and hence, it is so for the timed case. Besides general models for distributed systems that can be used to model security protocols such as Timed CSP and MSR (multi-set rewriting over first-order atomic formulae), we do not know about a model for timed cryptographic protocols.

To model timed cryptographic protocols, we include in our model clocks, time variables, and timestamps. Clocks are variables that range over the time domain and advance with the same rate as time. Each agent has its own set of clocks that it can reset. Clocks can be used to measure the time that elapses between two events, for instance, sending a message and receiving the corresponding response. We allow a global clock that is never reset and that can be read and tested by all participants. Time variables correspond to timestamps in received messages. Such values can be stored and used together with clocks to put conditions on the acceptance of a message.

A second contribution of this paper is a complete and sound symbolic verification algorithm for timed cryptographic protocols that builds upon the results of [BEL04a]. We consider a rich class of reachability properties that allow to specify confidentiality and authentication. In fact, we introduce a logic that allows to describe secrecy, equalities between terms and control points. Then, given a bounded protocol Π and two formulae in this logic Φ and Ψ, the reachability problem we consider is whether there is a run of Π that starts in a configuration that satisfies Φ and reaches a configuration that satisfies Ψ.

We devise a symbolic algorithm that given a property described by a formula Ψ in this logic and given a bounded protocol computes the set of configurations that reaches Ψ. This algorithm uses symbolic constraints (logic formulae) to describe sets of configurations. The logic we introduce combines constraints on the knowledge of the intruder with time constraints on clock values and time variables. To show effectiveness of our verification method we show:

1. that for each action of our model we can express the predecessor configurations of a set of configurations as a formula. We consider input, output, and time actions.

2. Then, we show decidability of the satisfiability problem for our logic.

Our model is clearly inspired by timed automata and our verification method influenced by the work on symbolic verification of timed automata and temporal logics for real-time systems (e.g.[AD94, HNSY92, AFH91, BL95]).

The developed results provide an algorithm for checking security properties (confidentiality and authentication) of timed cryptographic protocols. It has several interesting aspects:

1. it covers other properties than confidentiality (secrecy),

2. as initial configuration are described by formulae of the introduced logic, it can deal with infinite non-regular sets of messages initially known by the intruder.

3. we believe that our method is more easily amenable to extended intruder models.

Handling time constraints, unbounded message size symbolically and automatically is the distinguishing feature of our verification method. Most of the work on timed cryptographic protocols uses theorem-provers or finite-state model-checking [BP98, Coh00, ES00b]. While the first needs human help, the second relies on typing assumptions and assumption on the time window to bound the search space. An exception is the work by Delzanno and Ganty who present in [DG04] a verification method of unbounded timed protocols.

3.6 Verification techniques parametrized by equational theories

The following section presents an on-going work. Some parts of the proofs of the theorems are still missing.

The results mentioned in sections 2.3, 3.1 and 3.4 show how to relax the perfect cryptography assumption in a number of particular situations. The aim of this work is to bring together such results. We give general conditions on the equational theory and the deduction system under which we get a deduction procedure. Such a procedure will terminate for a bounded number of sessions.

The main ideas are:

1. Reduce equational theories to simpler ones using *variants*.

2. Keep apart the instantiation component of the protocol rules using an equational constraint. Therefore, we design constrained deduction rules.

3. Rely on a proof normalization result which allows to reduce the search space. It should become finite for a bounded number of sessions.

3.6.1 Locality properties and variants

$\mathcal{S}$ is any intruder deduction system (*e.g* those described in section 2). We use sequents $T \vdash s$ (meaning s can be deduced from knowledge T) and deduction rules. For instance:

$$\frac{T \vdash u \quad T \vdash v}{T \vdash \{u\}_v}$$

expressing that whenever the intruder knows the terms u and v, he can build the term $\{u\}_v$. This is a construction rule.

We assume that there is a mapping F from finite sets of terms to finite sets of terms such that:

locality property 1 . For every proof of $T \vdash s$ in $\mathcal{S}$,

- either the last proof rule is a construction rule and there is a proof Π of $T \vdash s$ in $\mathcal{S}$ such that every sequent $T \vdash u$ occurring in Π is such that $u \in F(T \cup \{s\})$
- or the last proof rule is not a construction rule and there is a proof Π of $T \vdash s$ in $\mathcal{S}$ in which every sequent $T \vdash u$ occurring in Π is such that $u \in F(T)$.

locality property 2 . There is a reduction ordering on terms (monotone, well-founded, compatible with the rewrite system) with minimal element 0, such that, for any proof Π in $\mathcal{S}$, and any non-zero term t in normal form, if t does not occur on the right-hand side of a sequent of Π, then there is a subterm u of t and a term $v < u$ such that $\Pi[u \mapsto v]$ is again a proof in $\mathcal{S}$. This also means that each construction is performed step by step. This is inspired by the oracle condition in [RT01].

iteration of F : for any T, $T \subseteq F(T)$ and $F(F(T)) = F(T)$.

In addition, we assume that an arbitrary equational theory can be decomposed as follows; this is the *finite variant property*.

$\mathcal{E}$ **finite variant property** : We assume that $\mathcal{E} = \mathcal{E}_1 \uplus \mathcal{E}_2$ in such a way that

1. Equivalence classes modulo $\mathcal{E}_2$ are finite
2. $\mathcal{E}_1$ can be turned into a finite convergent rewrite system modulo $\mathcal{E}_2$; we write $t\downarrow$ the normal form of a term t
3. For every term t, it is possible to compute a finite set of terms $t_1, \ldots, t_k$ such that $\{t\sigma\downarrow \mid \sigma \in \Sigma\} = \bigcup_{i=1}^{k} \{t_i\sigma \mid \sigma \in \Sigma\}$.

The finite variant property allows to compute in advance the redexes, hence not having to consider reductions any more.

Such properties are satisfied by *exclusive or* theory, Abelian groups theory, Dolev-Yao theory, etc., described in section 2.3.

3.6.2 Constrained deductions

Given an intruder deduction system $\mathcal{S}$ as above, we consider the protocol as an oracle which can be used by the intruder. Hence $\mathcal{S}$ is extended, including deduction rules which reflect this capability. In addition, the rules are lifted to terms with variables, using constrained sequents of the form $T \vdash u \quad [\![E]\!]$ where E is a record of the constraints on variables.

Equations in the constraint part are interpreted modulo $\mathcal{E}_2$. The constraints also contain the control states of the protocol: variables $x_{R,k,s,v}$ ranging over $\{0,1\}$ indicating that the session s of role R reached stage k, using the variant v of the protocol. Those constraints are in solved form, i.e they are conjunctions of equations $x_1 = u_1 \wedge \ldots \wedge x_n = u_n$ where $x_1,\ldots,x_n$ are the variables of E and, for every $1 \le i \le j \le n$, x_i does not occur in u_j.

We also lift the normalization operator $\downarrow$ to a constrained version $\downarrow_E$ using the rewrite rule $\rightarrow_E$.

An offline intruder rule is lifted; for instance, deducing $f(t_1,\ldots,t_k)$ from $t_1,\ldots,t_k$ becomes:

$$\frac{T \vdash u_1 \quad [\![E_1]\!] \ \ldots \ T \vdash u_k \quad [\![E_k]\!]}{T \vdash f(u_1,\ldots,u_k)\downarrow_{E_1\wedge\ldots\wedge E_k} \quad [\![E_1 \wedge \ldots \wedge E_k]\!]}$$

We add the following inference rules:

Instantiation rule:

$$\frac{T \vdash x \quad [\![E \wedge x = u]\!]}{T \vdash u\downarrow_{E\wedge x=u} \quad [\![E \wedge x = u]\!]}$$

Instantiation is only allowed at top level.

Weakening rule:

$$\frac{T \vdash u_1 \quad [\![E_1]\!] \qquad T \vdash u_2 \quad [\![E_2]\!]}{T \vdash u_1\downarrow_{E_1\wedge E_2} \quad [\![E_1 \wedge E_2]\!]}$$

Deducing u_2 is a way to move the control point and nothing else.

The protocol rule for session progression or opening

$$\frac{T \vdash u_1 \quad [\![E_1]\!] \qquad \cdots \qquad T \vdash u_n \quad [\![E_n]\!]}{T \vdash w\downarrow_E \quad [\![E]\!]}$$

where $v \Rightarrow w$ is the rule $k > 1$ of role R, session s, E is one of the solved forms of the conjunction

$$v = u_0\{x_1 \mapsto u_1; \ldots; x_n \mapsto u_n\} \wedge E_1 \wedge \ldots \wedge E_n \wedge \sigma_s \wedge x_{R,k,s,v} = 1$$

Moreover, u_0 can be built using construction rules and terms from $T \cup \{x_1,\ldots,x_k\}$. It is a term in $\widehat{v}$ which is the set of linear terms u_0 whose free variables are $x_1,\ldots x_k$ and such that v matches u_0.

One condition to apply this rule is that $E \models_{\mathcal{E}} x_{R,k-1,s,v} = 1$ but $E \not\models_{\mathcal{E}} x_{R,k,s,v} = 1$.

The rule expresses that, whenever some (instance of) v has been deduced and the protocol as reached stage $k-1$ of session s and role R, then the intruder may use the kth rule of R as an oracle: he gets the corresponding instance w and the control point moves to k for that session. In addition, we want to keep the instances of variables in the constraint, so, instead of v itself, we let u as a premise of the role, however imposing that the instance defined by the constraint is a unifier or u, v.

For session opening only, σ_s is the parameters bindings for session s and k is equal to 0.

Compromised agents

$$\frac{T \vdash u \quad [\![E]\!]}{T \vdash N_i(s) \quad [\![E]\!]}$$

If

1. $E \models_{\mathcal{E}} x_{R,1,s,v} = 1$
2. the main actor of session s is compromised
3. $N_i(s)$ is one of the nonces generated in session s.

This expresses that all data generated by compromised (or dishonest) agents are available to the intruder.

We can show that $\overline{\mathcal{S}}$ is correct and complete with respect to the chosen model.

3.6.3 A normal proof result

The main result is the following theorem:

Theorem 5 *If there is a proof of* $T \vdash s \quad [\![E]\!]$ *in* $\overline{\mathcal{S}}$*, using the set* V *of protocol rules, where* E *is solvable, then there is a proof of* $T \vdash s \quad [\![E']\!]$ *in* $\overline{\mathcal{S}}$ *such that*

1. *E' is solvable.*
2. *For all sequents* $T \vdash u \quad [\![E'']\!]$ *occurring in the proof, u belongs to $F(V,T,s)$*
3. *All constraints E'' are conjunctions of equations between terms in $F(V,T,\xi)$.*
4. *Each time the instantiation rule is applied, $u \in F(V,T)$*

We apply theorem 5 to the case of a bounded number of sessions. In this case, studied in [RT01, CKRT03, CLS03, CKR$^+$03b] for instance, the number of applications of the protocol rule is bounded on each branch. However, there is no a priori restriction on the sizes of messages, which are forged by the intruder; the transition system is bounded in depth but it is infinitely branching. The following theorem generalizes the previous results (except [CKR$^+$03b]):

Theorem 6 *Assuming that $\mathcal{S}$ has the locality properties and $\mathcal{E}$ has the finite variant property, then the insecurity problem for a bounded number of sessions is decidable. Moreover, in case $\mathcal{E}$ is the theory of inverse key, or* exclusive or*, or Abelian Group, or their combination, this problem is in NP.*

4 Conclusion

We have studied the verification problem for protocols for several extended models of the intruder: models with equational theories like homomorphism, AC, *exclusive or*, models with guessing capabilities for the intruder, or models with timestamps. For each of the extended models, we have provided decidability and often complexity results.

References

[AC02] R. Amadio and W. Charatonik. On name generation and set-based analysis in the dolev-yao model. In *Proc. of the 13th International Conference on Concurrency Theory (CONCUR'02)*, LNCS, pages 499–514. Springer Verlag, 2002.

[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)*, Turku, Finland, April 2004.

[AD94] R. Alur and D. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126, 1994.

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

[AFH91] R. Alur, T. Feder, and T.A. Henzinger. The benefits of relaxing punctuality. In *Proceedings of the 10th ACM Symposium on Principles of Distributed Computing*, pages 139–152. ACM Press, 1991.

[ALV02] R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theoretical Computer Science*, 290(1):695–740, 2002.

[AST00] Giuseppe Ateniese, Michael Steiner, and Gene Tsudik. New multiparty authentication services and key agreement protocols. *IEEE Journal on Selected Areas in Communications*, 18(4):628–639, April 2000.

[BB03] M. Boreale and M.G. Buscemi. On the symbolic analysis of low-level cryptographic primitives: Modular exponentiation and the diffie-hellman protol. In *Proc. Workshop of Foundations of Computer Security (FCS'03)*, 2003.

[BCLM03] S. Bistarelli, I. Cervesato, G. Lenzini, and F. Martinelli. Relating Process Algebras and Multiset Rewriting for Security Protocol Analysis. In *Proc. Workshop on Issues in the Theory of Security (WITS'03)*, Warsaw (Poland), April 2003.

[BEL04a] L. Bozga, C. Ene, and Y. Lakhnech. On the existence of an effective and complete proof system for bounded security protocols. In *Proceedings of Foundations of Software Science and Computation Structures (FOSSACS'04)*, volume 2987 of *Lecture Notes in Computer Science*, 2004.

[BEL04b] L. Bozga, C. Ene, and Y. Lakhnech. A symbolic decision procedure for cryptographic protocols with time stamps. In *Proc. of the 15th International Conference on Concurrency Theory (CONCUR' 04)*, London (England), 2004.

[BEL05] L. Bozga, C. Ene, and Y. Lakhnech. A symbolic decision procedure for cryptographic protocols with time stamps. *Journal of Logic and Algebraic Programming*, To appear, 2005.

[BL95] A. Bouajjani and Y. Lakhnech. Temporal logic + timed automata: expressiveness and decidability. In I. Lee and S. A. Smolka, editors, *CONCUR'95: Concurrency Theory*, volume 962 of *Lecture Notes in Computer Science*, pages 531–546. Springer, 1995.

[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.

[Bla04a] B. Blanchet. Automatic proof of strong secrecy for security protocols. In *IEEE Symposium on Security and Privacy*, pages 86–100, Oakland, California, May 2004.

[Bla04b] B. Blanchet. Cryptographic protocol verifier user manual. http://www.di.ens.fr/˜blanchet/crypto/proverif-manual.ps.gz, July 2004.

[Bor01] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proc. of the 28th Int. Coll. Automata, Languages, and Programming (ICALP'01)*. Springer Verlag, July 2001.

[BP98] G. Bella and L. C. Paulson. Mechanizing BAN Kerberos by the inductive method. In A. J. Hu and M. Y. Vardi, editors, *Proceedings of the 10th International Conference on Computer-Aided Verification (CAV'98)*, volume 1427 of *Lecture Notes in Computer Science*, pages 416–427, Vancouver, B.C., Canada, June 1998. Springer.

[BP03] Bruno Blanchet and Andreas Podelski. Verification of cryptographic protocols: Tagging enforces termination. In Andrew Gordon, editor, *Foundations of Software Science and Computation Structures (FoSSaCS'03)*, volume 2620 of *LNCS*, April 2003.

[CC01] H. Comon and V. Cortier. Tree automata with one memory, set constraints and cryptographic protocols. In *Research Report LSV-01-13*, december 2001.

[CDL04] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. Technical Report LSV-04-15, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2004. 36 pages.

[Che03] Y. Chevalier. *Résolution de probèmes d'accessibilité pour la compilation et la validation de protocoles cryptographiques*. PhD thesis, Université Henri Poincaré, Nancy, France, 2003.

[CJ97] John Clark and Jeremy Jacob. A survey of authentication protocol literature. `http://www.cs.york.ac.uk/˜jac/ papers/drareviewps.ps`, November 1997.

[CKR+03a] 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.

[CKR+03b] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, Mathieu Turuani, and Laurent Vigneron. Extending the dolev-yao intruder for analyzing an unbounded number of sessions. In *Proc. of the 17th International Workshop in Computer Science Logic (CSL'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.

[CKRT04] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, and Mathieu Turuani. Deciding the security of protocols with commuting public key encryption. In *Proc. of Automated Reasoning for Security Protocol Analysis (ARSPA '04)*, pages 53–63, Cork, Ireland, 2004.

[CL04] Hubert Comon-Lundh. Intruder theories (ongoing work). In *Foundations of Software Science and Computation Structures (FoSSaCS'04)*, volume 2987 of *Lecture Notes in Computer Science*, pages 1–4. Springer-Verlag, 2004.

[CLC03] H. Comon-Lundh and V. Cortier. New decidability results for fragments of first-order logic and application to cryptographic protocols. In *Proc. of the 14th Int. Conf. on Rewriting Techniques and Applications (RTA'2003)*, volume 2706 of *LNCS*, pages 148–164. Springer-Verlag, June 2003.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS '03)*, pages 271–280, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[CLT03] Hubert Comon-Lundh and Ralf Treinen. Easy intruder deductions. In Nachum Dershowitz, editor, *Verification: Theory & Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *Lecture Notes in Computer Science*, pages 225–242. Springer Verlag, 2003.

[Coh00] Ernie Cohen. Taps: A first-order verifier for cryptographic protocols. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop (CSFW'00)*, page 144. IEEE Computer Society, 2000.

[DEK83] D. Dolev, S. Even, and R.M. Karp. On the security of ping-pong protocols. In R.L. Rivest, A. Sherman, and D. Chaum, editors, *Proc. of CRYPTO 82*, pages 177–186. Plenum Press, 1983.

[DG04] G. Delzanno and P. Ganty. Automatic verification of time sensitive cryptographic protocols. In *Proc. of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04)*, volume 2988 of *Lecture Notes in Computer Science*, Barcelone (Spain), 2004.

[DJ04] S. Delaune and F. Jacquemard. A decision procedure for the verification of security protocols with explicit destructors. In *Proc. 11th ACM Conference on Computer and Communications Security (CCS'04)*, pages 278–287, Washington (Washington, USA), 2004. ACM Press.

[DLMS99] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. of the Workshop on Formal Methods and Security Protocols*, 1999.

[DS81] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8):533–536, 1981.

[DW83] M. D. Davis and E. J. Weyuker. *Computability, complexity and languages*, chapter 7, pages 128–132. Computer Science and Applied Mathematics. Academic Press, 1983.

[DY83] D. Dolev and A. Yao. On the security of public key protocols. *Proc. of IEEE Transactions on Information Theory*, IT-29(2):198–208, March 1983.

[EG83] S. Even and O. Goldreich. On the security of multi-party ping-pong protocols. In *Technical Report*. IEEE Computer Society Press, 1983.

[EGS86] S. Even, O. Goldreich, and A. Shamir. On the security of ping-pong protocols when implemented using the rsa. In *Advances in cryptology—CRYPTO 85*, volume 218 of *Lecture Notes in Computer Sciences*, pages 58–72, Santa Barbara, California, United States, 1986.

[ES00a] N. Evans and S. Schneider. Analysing time dependent security properties in csp using pvs. In *Proc of the 6th European Symposium on Research in Computer Security (EUSORICS '00)*, Toulouse, France, 2000.

[ES00b] Neil Evans and Steve Schneider. Analyzing time dependent security properties in CSP using PVS. In *ESORICS*, pages 222–237, 2000.

[GLNS93] L. Gong, M. A. Lomas, R. M. Needham, and J. H. Saltzer. Protecting poorly chosen secrets from guessing attacks. *IEEE Journal on Selected Areas in Communications*, 11(5):648–656, 1993.

[HNSY92] T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model-checking for real-time systems. In *Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 394–406. IEEE Computer Society Press, 1992.

[Hut02] H. Huttel. Deciding framed bisimulation. In *4th International Workshop on Verification of Infinite State Systems INFINITY'02*, pages 1–20, 2002.

[JGLV04] M. Roger J. Goubault-Larrecq and K. N. Verma. Abstraction and resolution modulo AC: How to verify diffie-hellman-like protocols automatically. *Journal of Logic and Algebraic Programming*, 2004. To appear.

[JRV00] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and verifying security protocols. In *Logic for Programming and Automated Reasoning (LPAR'00)*, volume 1955 of *LNCS*, November 2000.

[KKS87] E. Kaltofen, M. S. Krishnamoorthy, and B. D. Saunders. Fast parallel computation of hermite and smith forms of polynomial matrices. *SIAM J. Algebraic Discrete Methods*, 8(4):683–690, 1987.

[KNW02] D. Kapur, P. Narendran, and L. Wang. A unification algorithm for analysis of protocols with blinded signatures. Technical report, Department of Computer Science, SUNY Albany, 2002.

[KNW03] D. Kapur, P. Narendran, and L. Wang. An E-unification algorithm for analyzing protocols that use modular exponentiation. In *Proc. Intl. Conf on Rewriting Techniques and Applications (RTA-2003)*, pages 165–179, Valencia, Spain, 2003.

[LLT04] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for *ac*-like equational theories with homomorphisms. Research Report LSV-04-16, LSV, ENS de Cachan, November 2004. Available at `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rapports-year-2004-list.php`.

[LM04] C. Lynch and C. Meadows. On the Relative Soundness of the Free Algebra Model for Public Key Encryption. In *Proc. of the 4th Workshop on Issues in the Theory of Security (WITS'04)*, April 2004.

[Low96] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96)*, volume 1055, pages 147–166. Springer-Verlag, Berlin Germany, march 1996.

[Low97] G. Lowe. Casper: A compiler for the analysis of security protocols. In *Proc. of 10th Computer Security Foundations Workshop (CSFW'97)*. IEEE Computer Society Press, 1997.

[Low98] G. Lowe. Towards a completeness result for model checking of security protocols. In *Proc. of the 11th Computer Security Foundations Workshop (CSFW'98)*. IEEE Computer Society Press, 1998.

[Low02] Gavin Lowe. Analyzing protocols subject to guessing attacks. In *Proc. of the Workshop on Issues in the Theory of Security (WITS'02)*, 2002.

[Maz04a] L. Mazaré. Decidability of opacity with non-atomic keys. In *Proc. of the second international Workshop on Formal Aspects in Security and Trust (FAST2004), Toulouse, France*, 2004.

[Maz04b] L. Mazaré. Satisfiability of Dolev-Yao constraints. In *Proc. of Automated Reasoning for Security Protocol Analysis (ARSPA'04)*, 2004.

[Maz04c] L. Mazaré. Using unification for opacity properties. In *Proc. of the Workshop on Issues in the Theory of Security (WITS'04)*, 2004.

[McA93] David A. McAllester. Automatic recognition of tractability in inference relations. *J. ACM*, 40(2):284–303, 1993.

[Mea96] Catherine Meadows. Language generation and verification in the NRl protocol analyzer. In *Proc. of the 9th Computer Seurity Foundation Workshop (CSFW'96)*. IEEE Computer Society Press, 1996.

[Mea00] C. Meadows. Extending formal cryptographic protocol analysis techniques for group protocols and low-level cryptographic primitives. In P. Degano, editor, *Proceedings of the First Workshop on Issues in the Theory of Security (WITS'00)*, pages 87–92, Geneva, Switzerland, July 2000.

[Mea03] C. Meadows. Towards a hierarchy of cryptographic protocol models. In *Proc. of Formal Methods in Security Engineering (FMSE'03)*, 2003.

[Mil03] J. Millen. On the freedom of decryption. *Information Processing Letters*, 86(6):329–333, June 2003.

[MS01] J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proc. of the 8th ACM Conference on Computer and Communications Security (CCS'01)*, 2001.

[MS03] J. Millen and V. Shmatikov. Symbolic protocol analysis with products and diffie-hellman exponentiation. In *Proc. of the 16th IEE Computer Seurity Foundation Workshop (CSFW'03)*, 2003.

[Nar96] Paliath Narendran. Solving linear equations over polynomial semirings. In *Logic in Computer Science*, pages 466–472, 1996.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pau97] L. Paulson. Mechanized proofs for a recursive authentication protocol. In *Proc. of the 10th Computer Security Foundations Workshop*, pages 84–95. IEEE Computer Society Press, 1997.

[RS98] P. Y. A. Ryan and S. A. Schneider. An attack on a recursive authentication protocol: a cautionary tale. *Information Processing Letters*, 65(1):7–10, 1998.

[RS03] R. Ramanujam and S.P.Suresh. Tagging makes secrecy decidable for unbounded nonces as well. In *Proc. of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'03)*, Mumbai, 2003.

[RT01] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc. of the 14th Computer Security Foundations Workshop (CSFW'01)*, pages 174–190. IEEE Computer Society Press, 2001.

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

[Sch96] S. Schneider. Security properties and CSP. In *IEEE Symposium on Security and Privacy*, pages 174–187, 1996.

[Shm04] V. Shmatikov. Decidable analysis of cryptographic protocols with products and modular exponentiation. In *Proc. of the 13th European Symposium On Programming (ESOP'04)*, LNCS, pages 355–369. Springer Verlag, April 2004.

[TMN89] M. Tatebayashi, N. Matsuzaki, and D.B. Newman. Key distribution protocol for digital mobile communication systems. In *Advance in Cryptology — CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 324–333. Springer-Verlag, 1989.

[Tur03] M. Turuani. *Sécurié des protocoles cryptographiques: décidabilité et complexité*. PhD thesis, Université Henri Poincaré, Nancy, France, 2003.

[Ver03] K. N. Verma. Two-way equational tree automata for ac-like theories: Decidability and closure properties. In R. Nieuwenhuis, editor, *Proc. of the 14th International Conference on Rewriting Techniques and Applications (RTA'03)*, LNCS, pages 180–196, Valencia, Spain, June 2003. Springer.