A Resolution Strategy for Verifying Cryptographic Protocols with CBC Encryption and Blind Signatures[*]

Véronique Cortier
LORIA, Nancy, France
CNRS & INRIA Project Cassis
cortier@loria.fr

Michael Rusinowitch
LORIA, Nancy, France
INRIA Project Cassis
rusi@loria.fr

Eugen Zălinescu
LORIA, Nancy, France
INRIA Project Cassis
zalinesc@loria.fr

ABSTRACT

Formal methods have proved to be very useful for analyzing cryptographic protocols. However, most existing techniques apply to the case of abstract encryption schemes and pairing. In this paper, we consider more complex, less studied cryptographic primitives like CBC encryption and blind signatures. This leads us to introduce a new fragment of Horn clauses. We show decidability of this fragment using a combination of several resolution strategies.

As a consequence, we obtain a new decidability result for a class of cryptographic protocols (with an unbounded number of sessions and a bounded number of nonces) that may use for example CBC encryption and blind signatures. We apply this result to fix the Needham-Schroeder symmetric key authentication protocol, which is known to be flawed when CBC mode is used.

Categories and Subject Descriptors

F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic; F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Decision Problems*

General Terms

Verification, Security, Theory

Keywords

Verification, Cryptographic Protocols, Horn Clauses, Resolution Strategies

1. INTRODUCTION

Cryptographic protocols are designed to provide certain security guarantees between agents communicating in a hostile environment. Intruders are malicious agents that control the network. They may create security breaches by simply blocking, diverting and, spoofing the messages and modifying the ones they can decrypt. Formal methods have been quite effective [15, 7, 2] in finding these kind of attacks that rely only on the logical structure of the protocols and do not require to *break* the cryptographic primitives. Some decidability results have been derived for unbounded number of sessions [8] and several automatic tools have been developed to verify protocols [4] in this abstract model which is based on the so called *perfect cryptography assumption*: one needs a decryption key to extract the plaintext from the ciphertext, and also, a ciphertext can be generated only with the appropriate key and message (no collision).

However, assuming perfect cryptography is a strong limitation for analyzing protocols, since many cryptographic functions admit simple algebraic properties that can be easily exploited by intruders. These properties may be crucial too for a proper working of the protocol. Therefore, for a sharper analysis, it is important to incorporate them in the protocol modeling.

Among these properties the *prefix property* related to Cipher Bloc Chaining mode (CBC) is important since it is a common encryption mode. Manipulating the encryption blocks, the intruder can get the encrypted prefix of any encrypted message. This yields to attacks [17] that could not be mounted in the case of stronger encryption schemes. The same prefix property holds when weaker encryption schemes are used, like the Electronic Code Book (ECB) encryption mode.

In addition, dealing with the prefix property is a first step towards taking into account homomorphic encryption, which is particularly important since it is employed in the design of several E-voting schemes [3, 11], electronic auctions [5] and privacy-preserving data mining [12]. Also it is known that RSA encryption has homomorphic properties when used with the same modulus.

Also adding new cryptographic primitives for building protocols require to enhance the abstract model and it is unclear whether the known decidability results about security properties are preserved in these extensions. For instance the *blind signature* scheme allows an agent (e.g. a voter) to have a message (e.g. a vote) signed blindly by an another entity (e.g. an administrator). Then in some subsequent protocol steps the message can get unblinded. The blind signature scheme properties offer to the environment new attacks opportunities.

In this paper we propose a resolution strategy for deciding a fragment of first-order logic that allows one to incorporate the prefix property of CBC encryption in our protocol mod-

[*] This work has been partially supported by the ACI-SI Satin and the IST AVISPA project.

PPDP'05, July 11–13, 2005, Lisbon, Portugal.
Copyright 2005 ACM 1-59593-090-6/05/0007 ...$5.00.

eling and to decide the existence of attacks exploiting this property. The same fragment applies to abstract properties of blind signature schemes, which are the main alternative to homomorphic encryption in E-voting [13]. The approach follows the line of [8] but requires a refined strategy in order to eliminate the clauses generated by resolution with the prefix properties. The verification algorithm has been applied to Needham-Schroeder Symmetric Key protocol, which is subject to an attack when implemented using CBC encryption. We show how to fix the protocol and we show the correction of the resulting protocol.

Related works. Recently several procedures for deciding trace-based security properties have been proposed for XOR and Abelian groups operators for a bounded [6] or unbounded [8] number of sessions. In [6] the prefix property which is very similar to homomorphism is handled by a decision procedure for the bounded scenario. In our case the number of sessions is unbounded. Homomorphism theory with AC operators is considered in [14] for the case of a passive intruder (that is one that cannot interact with the protocol execution but can only listen to communications). Here we consider the more complex case of an active intruder participating to communications with honest agents. An electronic voting protocol has been recently analysed in [13]. The protocol relies on a blind signature scheme whose properties have been modeled by equations. Secrecy of votes have been proved automatically using the ProVerif tool by B. Blanchet [4]. The above mentioned works do not address the decidability of secrecy with CBC encryption or blind signature.

Layout of the paper. In Section 2, we explain how protocols can be modeled using Horn clauses, introducing a new fragment of first order clauses. In Section 3, we present our resolution strategy and apply it to this fragment, proving that this strategy is both complete and terminating for this class. This leads to our main contribution: the decidability of satisfiability for this fragment. As a consequence, we obtain that secrecy of cryptographic protocols is decidable for an unbounded number of sessions, in the case for example of CBC encryption and blind signatures, when nonces are abstracted by constant terms and at most one copy is performed at each transition. We apply this result to the Needham-Schroeder symmetric key authentication protocol: this protocol is flawed when the CBC mode is used [17]. We show (in Section 4) how to fix the protocol and we formally prove, using our technique, that the fixed version preserves secrecy. Concluding remarks can be found in Section 6.

2. MODELING PROTOCOLS

The aims of this section is to introduce some notations and to introduce the class of clauses we consider. We also explain how protocols can be modeled using these clauses.

2.1 Notations

Let $\mathcal{F}$ a finite set of function symbols and $\mathcal{V}$ a set of variables. The set of terms on $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F} \cup \mathcal{V})$. If u is a term, $V(u)$ denotes the set of variables occurring in u. A term is *ground* if it has no variable. If u and v are two terms of $\mathcal{T}(\mathcal{F} \cup \mathcal{V})$, the term $u[v/x]$ is the term u where each occurrence of x has been replaced by v. If x is the only variable of u, then we may write $u(x)$ instead of u, in order to emphasize this property. And also, in this case, we simply write $u[v]$ instead of $u[v/x]$. Let I be a unary predicate. *Atoms* A are of the form $I(u)$ where u is a term. *Literals* L are either positive literals $+A$ (or simply A) or negative literals $-A$ where A is an atom. A *clause* is a finite set of literals. If C_1 and C_2 are clauses, $C_1 \vee C_2$ denotes $C_1 \cup C_2$. A *Horn clause* is a clause that contains at most one positive literal. For Horn clauses we may use the alternative notation $A_1, A_2, \ldots, A_{n-1} \rightarrow A_n$ to denote the clause $-A_1 \vee -A_2 \vee \ldots \vee -A_{n-1} \vee A_n$. A *substitution* is a function $\sigma : \mathcal{V} \rightarrow \mathcal{T}(\mathcal{F} \cup \mathcal{V})$. A *ground substitution* maps every variable to a ground term. If M is a term, literal, clause, substitution or, set of such objects, then $M\sigma$ obtained by applying σ to M is defined as usual. If M and N are terms or literals then a *unifier* of M and N is a substitution σ such that $M\sigma = N\sigma$. If such a unifier exists then there exists a *most general unifier (mgu)*, denoted by $\mathrm{mgu}(M, N)$.

If $u \in \mathcal{T}(\mathcal{F} \cup \mathcal{V})$, $|u|$ is the *depth* of u (maximal size of its positions). For $x \in \mathcal{V}$, $|u|_x$ is the maximal depth of an occurrence of x in u. By convention, if $x \in \mathcal{V}$ then $|u|_x = 0$ if $x \notin V(u)$. The definitions of $|\cdot|$ and $|\cdot|_x$ are extended to literals by $|\pm I(u)| = |u|$ and $|\pm I(u)|_x = |u|_x$.

We consider a strict and total order $<_{\mathcal{F}}$ on the function symbols. If u is a term which is not a variable, h_u denotes the head symbol of u. We fix an order on terms. The order is chosen in order to ensure the termination of our resolution procedure.

Definition 1 (order $<$). *Let u and v be two terms. We say that $u < v$ if one of two following conditions holds:*

- *$|u| < |v|$ and $|u|_x < |v|_x$, for every $x \in V(u) \cup V(v)$;*
- *$|u| \leq |v|$, $|u|_x \leq |v|_x$ for every $x \in V(u) \cup V(v)$ and, $h_u <_{\mathcal{F}} h_v$.*

For example, if u is a strict subterm of v then $u < v$. Variables are incomparable. We have $\langle a, x\rangle < \mathrm{h}(\mathrm{h}(x))$ but $\langle \mathrm{h}(\mathrm{h}(a)), x\rangle \not< \mathrm{h}(\mathrm{h}(x))$, where as usual $\langle \_, \_\rangle$ is a function symbol denoting the pairing operation. Later, we may also omit the $\langle\rangle$. By convention, we assume that terms are parsed from left to right, *i.e.* the term $\langle u_1, u_2, \ldots, u_k\rangle$ means the term $\langle u_1, \langle u_2, \langle \ldots, u_k\rangle \ldots\rangle$.

An order is said *liftable* if for any two terms u, v and for any substitution θ, $u < v$ implies $u\theta < v\theta$. This is a crucial property for the completeness of ordered resolution.

Proposition 1. *The relation $<$ is a strict liftable ordering.*

Proof. Transitivity and irreflexivity of $<$ are obvious. We have

$$|w\sigma| = \max(|w|, \max_{x \in V(w)}(|w|_x + |x\sigma| - 1))$$

and

$$|w\sigma|_x = \begin{cases} 0, \text{ if } x \notin V(w\sigma), \\ |w|_x, \text{ if } x\sigma = x, \\ \max_{y \in V(w)}(|w|_y + |y\sigma|_x - 1), \text{ otherwise,} \end{cases}$$

Also $h_{w\sigma} = h_w$ if w is not a variable. This shows that $<$ is liftable. □

A term v is said *maximal* in a set S if there is no term $u \in S$ such that $v < u$.

All the definitions are extended on literals by $\pm I(u) < \pm' I(v)$ if and only if $u < v$.

We assume for the rest of the article that $\{\_\}_\_$, $\mathrm{h}(\_)$, $\mathrm{sign}(\_, \_)$ and $\mathrm{blind}(\_, \_)$ are function symbols.

$I(x), I(y) \rightarrow I(\langle x, y\rangle)$	pairing of messages
$I(\langle x, y\rangle) \rightarrow I(x)$	first projection
$I(\langle x, y\rangle) \rightarrow I(y)$	second projection
$I(x), I(y) \rightarrow I(\{x\}_y)$	symmetrical encryption
$I(\{x\}_y), I(y) \rightarrow I(x)$	symmetrical decryption
$I(x) \rightarrow I(\mathrm{h}(x))$	hashing
$I(x), I(y) \rightarrow I(\mathrm{sign}(x, y))$	signing
$I(\mathrm{sign}(x, y)), I(y) \rightarrow I(x)$	verifying the signature
$I(x), I(y) \rightarrow I(\mathrm{blind}(x, y))$	blinding
$I(\mathrm{blind}(x, y)), I(y) \rightarrow I(x)$	undo blinding

Figure 1: Intruder rules: the set $\mathcal{I}$

2.2 Intruder clauses: the class $\mathcal{C}_I$

The intruder analyzes the messages sent on the network. For example, if he sees an encrypted message and if he knows the encryption key, then he can decrypt the message. This can be easily modeled using a very simple Horn clause: $I(\{x\}_y), I(y) \rightarrow I(x)$. The predicate I represents the knowledge of the intruder: $I(m)$ means that the intruder knows the term (or message) m. Thus this clause should be read as "if the intruder knows some message of the form $\{x\}_y$ and if he knows y, then he knows x". Other examples of clauses modeling the intruder power can be found in Figure 1. The set of all the clauses of Figure 1 is denoted by $\mathcal{I}$. Note that we may also consider public encryption by adding for each identity the clauses for encryption and decryption.

Each of these clauses contains at most one function symbol. That is why we consider the following class of clauses.

Definition 2 (class $\mathcal{C}_I$). *Let $\mathcal{C}_I$ be the class of Horn clauses of the form:*

$$\pm I(f(x_1, \ldots, x_n)) \vee \bigvee_{j=1}^{m} \pm I(x_{i_j}).$$

The set of clauses $\mathcal{I}$ (defined in Figure 1) corresponding to the intruder capabilities is clearly in the class $\mathcal{C}_I$.

2.3 Protocol clauses: the class $\mathcal{C}_P$

We now show how the rules of a protocol can also be modeled using Horn clauses. We consider a variant of the Needham-Schroeder symmetric key authentication protocol [16]. The original protocol will be analyzed in Section 4.

The description of the protocol is as follows:

$$\begin{array}{ll} A \Rightarrow S: & A, B, N_a \\ S \Rightarrow A: & \{N_a, B, K_{ab}\}_{K_{as}} \\ S \Rightarrow B: & \{K_{ab}, A\}_{K_{bs}} \\ B \Rightarrow A: & \{N_b\}_{K_{ab}} \end{array}$$

The notations A, B, S represent respectively the roles Alice, Bob and the server. The intruder is a special role that can impersonate other roles and we will denote by $I(A)$ for example a spoofed instance of A. The message field N_a is a *nonce*, meaning a random number freshly generated by A just before it is sent in the first message. In this first message Alice tells the server that she wants to communicate with Bob, and puts a nonce in her message. The server replies with a message containing the same nonce, Bob's name and a fresh session key K_{ab}. The bracketed term $\{N_a, B, K_{ab}\}_{K_{as}}$ represents the encryption of the concatenation of N_a, B and K_{ab} using key K_{as} shared by Alice and the server. In the third message the server forwards to Bob the session key and his name, all encrypted by their common key K_{bs}. Finally Bob can challenge Alice by sending her a nonce N_b encrypted by the session key. The protocol can be extended with the response to this challenge.

The translation in Horn clauses may be found in Figure 2. The intruder controls all the network communications. He can either build and send new messages or forward messages from other agents. Thus we may assume that any message is sent trough the intruder. We explain here the translation of the second rule: each time the server S receives a message of the form $\langle a, b, x\rangle$ (sent be the intruder) where x can be any term, he answers by the message $\{x, b, \mathrm{k}(a, b)\}_{\mathrm{k}(a,s)}$ (intercepted by the intruder). This is expressed by the rule $I(\langle a, b, x\rangle) \rightarrow I(\{x, b, \mathrm{k}(a, b)\}_{\mathrm{k}(a,s)})$. Since every protocol session generates new nonces, for modeling the protocol we have to perform some abstraction by letting the nonces only depend on the agent that has created it and the agent that should receive it and similarly a fresh session key will be parameterized by the agents who share the key. Since the clauses can be applied in any order, any number of times, we also abstract away the order of the rules of the protocol. For instance, in our example, the third rule can be applied even if the first rule has not been played yet. Note that all these abstractions are correct w.r.t. secrecy properties, *i.e.* if a protocol is deemed secure using these abstractions then it is secure without abstractions.

$$\begin{array}{r} \rightarrow I(\langle a, b, n_1(a, b)\rangle) \\ I(\langle a, b, x\rangle) \rightarrow I(\{x, b, \mathrm{k}(a, b)\}_{\mathrm{k}(a,s)}) \\ I(\langle a, b, x\rangle) \rightarrow I(\{\mathrm{k}(a, b), a\}_{\mathrm{k}(b,s)}) \\ I(\{y, a\}_{\mathrm{k}(b,s)}) \rightarrow I(\{n_2(b, a)\}_y) \end{array}$$

Figure 2: Clauses for a variant of the Needham-Schroeder protocol: the set $\mathcal{P}_{\mathsf{ex}}$

We notice that each of the clauses has at most one variable. As noticed in [8], this is the case of protocols *with single blind copying*, *i.e.* protocols for which, at each step of the protocol, at most one part of the message is blindly copied. For example, in the second rule of the Needham-Schroeder protocol, the only blindly copied part is N_a since the other parts (A and B) are names known to the server. Therefore, the second class of clauses we consider is the class $\mathcal{C}_P$ of Horn clauses that contain at most one variable. We have $\mathcal{P}_{\mathsf{ex}} \in \mathcal{C}_P$.

Then to check whether the secrecy of a message m is preserved we add a clause $-I(m)$ to the set of clauses modeling the protocols, the intruder activities and the intruder knowledge (if she knows t we add $I(t)$). The satisfiability of the resulting set of clauses will prove the secrecy property.

Comon and Cortier [8] have shown that satisfiability of a set of clauses of $\mathcal{C}_I \cup \mathcal{C}_P$ is decidable in 3-EXPTIME and Seidl and Verma [18] have shown that satisfiability is in fact DEXPTIME-complete. Since secrecy properties can also be modeled using Horn clauses (of the class $\mathcal{C}_P$), it means that the secrecy preservation of protocols (with at most one blind copy and no nonces) is decidable in 3-EXPTIME.

2.4 Extending the intruder power

The aim of the paper is to extend the decidability result of [8] to a larger class of clauses, in order to model an extended power of the intruder. Indeed, the set of clauses $\mathcal{I}$, described in Figure 1, represents the capabilities of an intruder, assuming *perfect cryptography.* In particular, the intruder cannot learn anything from an encrypted message $\{m\}_k$, except if he has the inverse key. However, depending on the implementation of the cryptographic primitives, the intruder may be able to deduce more messages. We consider here CBC encryption and blind signatures.

2.4.1 Prefix property

Depending on the encryption scheme, an intruder may be able 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 is encoded by the clause:

$$C_{pre} \overset{\text{def}}{=} -I(\{\langle x, y\rangle\}_z) \vee I(\{x\}_z)$$

This is for example the case for Cipher Block Chaining (CBC) encryption. In such a system, the encryption of the 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 $C_0C_1\cdots C_i = \{P_1P_2\cdots P_i\}_K$, 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. This property can be used to mount attacks on several well-known protocols. For example, we explain in Section 4.1 the attack discovered by O. Pereira and J.-J. Quisquater [17] on the Needham-Schroeder symmetric key authentication protocol [16].

This property also holds for homomorphic encryption, *i.e.* encryption schemes that verify that $\{\langle x, y\rangle\}_k = \langle\{x\}_k, \{y\}_k\rangle$. This is the case of the ECB (Electronic Code Book) encryption scheme for example, where the encryption of message block sequence $P_1P_2\cdots P_n$ with the key K is simply the sequence $\{P_1\}_K\{P_2\}_K\cdots\{P_n\}_K$. For such encryption schemes, the clause C_{pre} models only partially the intruder power. Indeed, the intruder is able to recombine messages, which is not modeled by the clause.

2.4.2 Blind signatures

Blind signatures are used in voting protocols like the FOO 92 voting protocol [11, 13]. The idea of the protocol is that the voter first commits its vote v using a blinding function blind and a random blinding factor r: he sends the message $\text{blind}(v, r)$ together with a signature of the message. The administrator A verifies that the voter has the right to vote and has not voted yet. If it is the case, he signs the message, *i.e.* sends the message $\text{sign}(\text{blind}(v, r), \text{ska})$. Note that the administrator does not have access to the vote since it is blinded. Now, the voter can unblind the message, getting $\text{sign}(v, \text{ska})$, using that $\text{unblind}(\text{sign}(\text{blind}(v, r), \text{ska}), r) = \text{sign}(v, \text{ska})$. Then the voter can send its vote to the collector. The "commutativity" property between blinding and signing can be modeled by the clause:

$$C_{sig} \overset{\text{def}}{=} -I(\text{sign}(\text{blind}(x, y), z)) \vee -I(y) \vee I(\text{sign}(x, z)).$$

2.4.3 Definition of the class $\mathcal{C}_S$

First let us note that the clauses C_{pre} and C_{sig} are neither in the class $\mathcal{C}_I$ nor in the class $\mathcal{C}_P$. Therefore they cannot be treated by [8, 18] techniques.

In order to extend the intruder power to clauses such as C_{pre} or C_{sig}, we consider the class of *special clauses*, denoted by $\mathcal{C}_S$.

We assume that the set of function symbols $\mathcal{F}$ contains a special symbol f_0 and that this symbol is the smallest symbol of $\mathcal{F}$ for the order $<_{\mathcal{F}}$. This special symbol will stand for encryption in the case of the prefix property or will stand for signing in the case of blind signatures.

Definition 3 (class $\mathcal{C}_S$). *Let $\mathcal{C}_S$ be the set of clauses of the form:*

$$I(f_0(y_j, z)) \vee -I(f_0(u[g(y_1, \dots, y_k)], v)) \vee \\ \vee \bigvee_{i=1}^{p} -I(w_i[g(y_1, \dots, y_k)]) \vee \bigvee_{l=1}^{q} -I(y_{i_l}), \qquad (1)$$

where $\{j, i_1, \dots, i_q\} \subseteq \{1, \dots, k\}$, $p, q \geq 0$, $u, v \in \mathcal{T}(\mathcal{F} \cup \{z\})$ and, $I(f_0(u[g(y_1, \dots, y_k)], v))$ is greater than any other literal of the clause.

For example, the clause C_{pre} is obtained when $u = v = z$, $j = 1$, $p = q = 0$, $f_0 = \{\_\}_\_$ and, $g = \langle \_, \_ \rangle$. The clause C_{sig} is obtained when $u = v = z$, $j = 1$, $p = 0$, $q = 1$, $f_0 = \text{sig}$ and, $g = \text{blind}$. We could also consider for example the clause $-I(\{\langle x, y\rangle\}_z) \vee I(\{y\}_z)$.

Of course, this class could also be used to express more complex protocol clauses.

3. MAIN RESULT

We show that satisfiability of clauses of $\mathcal{C} = \mathcal{C}_I \cup \mathcal{C}_P \cup \mathcal{C}_S$ is still decidable, under a slight semantical assumption. To get this result we consider a variant of ordered resolution where resolution between clauses of a *saturated* set are forbidden. In Section 3.1, we recall the definition of ordered resolution. In Section 3.2, we introduce our variant of ordered resolution. We prove our decidability result in Section 3.3 and show in Section 3.4 that both CBC encryption and blind signatures satisfy the hypotheses of our theorem.

3.1 Ordered resolution

We consider a liftable partial ordering $\prec$, total on closed atoms.

Let A and B be two atoms, C, C_1 and C_2 be clauses and, $\sigma = \text{mgu}(A, B)$. The *resolution rule* is defined by:

$$\frac{C_1' \overset{\text{def}}{=} C_1 \vee A \qquad -B \vee C_2 \overset{\text{def}}{=} C_2'}{C_1\sigma \vee C_2\sigma}$$

The clause $C_1\sigma \vee C_2\sigma$ is called *resolvent* of the clauses C_1' and C_2'. The atom $A\sigma$ is called the *resolved atom.*

The *ordered resolution* (wrt $\prec$) requires that there is no atom in the resolvent greater than the resolved atom.

If $C_1, C_2, \dots, C_n$ are clauses such that their sets of variables are pairwise disjoint then we note the clause $C_1 \vee C_2 \vee \cdots \vee C_n$ by $C_1 \sqcup C_2 \sqcup \cdots \sqcup C_n$, in order to emphasize this property. Considering a set $\mathcal{T}$ whose elements are sets of clauses, the *splitting rule* is defined as follows: $\mathcal{T} \rightarrow_{spl} (\mathcal{T}\backslash\{S\}) \cup \{(S\backslash\{C_1 \sqcup C_2\}) \cup \{C_1\}\} \cup \{(S\backslash\{C_1 \sqcup C_2\}) \cup \{C_2\}\}$, where $S \in \mathcal{T}$ and $C_1 \sqcup C_2 \in S$. We write $\mathcal{T} \Rightarrow_{spl} \mathcal{T}'$ to say that $\mathcal{T} \rightarrow^*_{spl} \mathcal{T}'$ and no application of the splitting rule on $\mathcal{T}'$ is possible anymore. For a binary relation ρ, we denoted by ρ^* its reflexive and transitive closure.

It is well known that ordered resolution with splitting is complete for Horn clauses [1]. However, while ordered resolution was sufficient to prove decidability of satisfiability for clauses of the classes $\mathcal{C}_I \cup \mathcal{C}_P$, it is not the case anymore. Consider for example the order $<$ defined in Section 2.1 (which extends the order considered in [8]). Ordered resolution between the clause C_{pre} and the clause $I(x), I(y) \to I(\{x\}_y)$ yields $I(\langle x, y\rangle), I(z) \to I(\{x\}_z)$. Resolving again this clause with C_{pre} yields $I(\langle\langle x, x'\rangle, y\rangle), I(z) \to I(\{x\}_z)$ and so on. Thus ordered resolution does not terminate. However, we note that deriving the clause $I(\langle x, y\rangle), I(z) \to I(\{x\}_z)$ is useless (w.r.t. the completeness of the resolution) thanks to the clause $I(\langle x, y\rangle) \to I(x)$. This will be formally proved in section 3.4. In terms of resolution theory [1], the set $\mathcal{I} \cup \{C_{pre}\}$ is already saturated. We formalize this notion in the next section.

3.2 Our resolution method

A *partial ordered interpretation* $\mathfrak{I}$ is a set of ground literals such that if $A \in \mathfrak{I}$ then $-A \notin \mathfrak{I}$ and conversely, and if $\pm A \in \mathfrak{I}$ and $B \prec A$ then $\pm B \in \mathfrak{I}$. A ground clause C is *false* in $\mathfrak{I}$ if, for every literal $\pm A$ in C, the opposite literal $\mp A$ belongs to $\mathfrak{I}$. A clause C is *unsatisfiable* in the partial interpretation $\mathfrak{I}$ if there exists a ground substitution θ such that all atoms of $C\theta$ are among those of $\mathfrak{I}$ and $C\theta$ is false in the interpretation. A set of clauses is *unsatisfiable* in the partial interpretation if there is a clause in the set that is unsatisfiable in the partial interpretation.

DEFINITION 4. *A set S of clauses is* saturated *w.r.t. the order $\prec$ if for every resolvent C obtained by ordered resolution from S and for every partial interpretation $\mathfrak{I}$, if C is unsatisfiable in $\mathfrak{I}$ then S is unsatisfiable in $\mathfrak{I}$.*

Let S be a saturated set of clauses. For a set of clauses T such that $S \subseteq T$, we denote by $Res(T)$ the set of clauses derived by ordered resolution method with the restriction that we do not apply resolution if the two premises are clauses of S. We define $R(T) \overset{\text{def}}{=} T \cup Res(T)$. For a class $\mathcal{T}$ of sets of clauses we note by $R(\mathcal{T}) \overset{\text{def}}{=} \{R(T) \mid T \in \mathcal{T}\}$. Also we write $\mathcal{T} \Rightarrow_{\prec,spl} \mathcal{T}'$ to say that $R(\mathcal{T}) \Rightarrow_{spl} \mathcal{T}'$. Remark that $\mathcal{T}'$ is unique. We denote by $\mathcal{R}_S$ the ordered resolution method with splitting together with the mentioned restriction. The following result states the refutational completeness of this method:

PROPOSITION 2. *For any liftable ordering $\prec$, for any sets S and T of clauses, such that S is saturated w.r.t. $\prec$ and $S \subseteq T$, T is unsatisfiable if and only if $\{T\} \Rightarrow^*_{\prec,spl} \mathcal{T}$, for some $\mathcal{T}$ such that every set of clauses in $\mathcal{T}$ contains the empty clause.*

The proof is a direct consequence of the refutational completeness of the standard strategy since, from the hypothesis that S is saturated, all inferences performed between clauses from S are useless.

We extend the presented resolution method with a *tautology elimination rule* and a *subsumption rule*. These rules do not compromise the completeness result of the method.

3.3 A decidable class

Our resolution method is still not sufficient to ensure termination for clauses of $\mathcal{C}$. Thus we consider an additional slight syntactic restriction. For a protocol point of view, this restriction does not reduce the expressivity of the fragment of clauses under consideration.

DEFINITION 5. *We say that a term $f_0(u, v)$ is* well-behaved *if the following two implications are true: if $V(u) \neq \emptyset$ then v is a constant; and, if $V(v) \neq \emptyset$ then u is a ground term.*

We say that a clause of $\mathcal{C}_S$ is well-behaved *if all subterms $f_0(s, t)$ of the terms u, v and w_i, for all i, (see Definition 3) are well-behaved. We say that a clause C not in $\mathcal{C}_S$ is* well-behaved *if for every literal $\pm I(w)$ of C, every subterm $f_0(s, t)$ of w is well-behaved.*

Usually, the terms used in modelling cryptographic protocols are well-behaved. For example, if $\mathcal{S} = \{C_{pre}\}$ (or $\mathcal{S} = \{C_{sig}\}$) (see previous section) then $\mathcal{S} \cup \mathcal{P}_{\mathsf{ex}}$ is well-behaved.

We are now ready to state our main result.

THEOREM 3. *Let $\mathcal{I}, \mathcal{P}, \mathcal{S}$ be finite sets included respectively in the classes $\mathcal{C}_I$, $\mathcal{C}_P$ and $\mathcal{C}_S$. If $\mathcal{I} \cup \mathcal{S}$ is saturated and $\mathcal{P} \cup \mathcal{S}$ is well-behaved then the satisfiability of $\mathcal{I} \cup \mathcal{P} \cup \mathcal{S}$ is decidable.*

The rest of the subsection is devoted to the outline of the proof of the theorem. For the sake of clarity, some proofs of intermediate results are postponed to Section 5.

We consider the order $<$ defined in Section 2.1. By Proposition 1, $<$ is a liftable ordering. We apply the resolution $\mathcal{R}_{\mathcal{I}\cup\mathcal{S}}$ (defined in Section 3.2) to the set $\mathcal{I} \cup \mathcal{S} \cup \mathcal{P}$. Thanks to Proposition 2 this method is refutationally complete. Hence to get decidability we only need to show the termination of the method.

Our resolution method applied to clauses of the class $\mathcal{C}$ may create clauses outside the class $\mathcal{C}$. To obtain an invariant, we introduce the following auxiliary class of clauses. We define $\mathcal{C}_J$ to be the class of clauses of the form:

$$I(f_0(y_j, a)) \vee \bigvee_{i=1}^{r} -I(w_i[g(y_1, \ldots, y_k)]) \vee \bigvee_{l=1}^{s} -I(y_{i_l}),$$

where $r \geq 1$ and $s \geq 0$.

We have that the resolution method $\mathcal{R}_{\mathcal{I}\cup\mathcal{S}}$ applied to any set of clauses of $\mathcal{I} \cup \mathcal{S} \cup \mathcal{C}_P \cup \mathcal{C}_J$ yields a clause in $\mathcal{C}_P$ or $\mathcal{C}_J$.

LEMMA 4. *Let $\mathcal{P}'$ and $\mathcal{J}$ be sets of clauses of respectively $\mathcal{C}_P$ and $\mathcal{C}_J$, such that $\mathcal{P}'$ and $\mathcal{J}$ are well-behaved. The application of $\mathcal{R}_{\mathcal{I}\cup\mathcal{S}}$ resolution on $\mathcal{I} \cup \mathcal{S} \cup \mathcal{P}' \cup \mathcal{J}$ produces clauses in $\mathcal{C}_P$ or $\mathcal{C}_J$. Moreover, the set of resolvents is well-behaved.*

The proof is done in Section 5.1.

We define the *depth* of a clause C to be $||C|| \overset{\text{def}}{=} \max_{L \in C} |L|$. We prove in Section 5.2 that the depth of clauses obtained applying the $\mathcal{R}_{\mathcal{I}\cup\mathcal{S}}$ resolution does not increase except if they are ground, in which case the depth may double.

LEMMA 5. *Let C_1 and C_2 be two clauses of $\mathcal{C}_P$, $\mathcal{C}_I$, $\mathcal{C}_S$ or $\mathcal{C}_J$, such that if C_i, where $i \in 1, 2$, is in $\mathcal{C}_P$, $\mathcal{C}_S$ or $\mathcal{C}_J$ then it is well-behaved. The resolvent C derived by $\mathcal{R}_{\mathcal{I}\cup\mathcal{S}}$ resolution satisfies: $||C|| \leq \max(||C_1||, ||C_2||)$ if C is not ground and $||C|| \leq 2\max(||C_1||, ||C_2||)$ if C is ground.*

These two lemmas allow us to conclude. We denote by T_0 the set $\mathcal{I} \cup \mathcal{S} \cup \mathcal{P}$ and by $\mathcal{T}_0$ the set $\{T_0\}$. For every $i \geq 0$ we define recursively $\mathcal{T}_{i+1}$ to be the set such that $\mathcal{T}_i \Rightarrow_{<,spl} \mathcal{T}_{i+1}$. Due to the application of the splitting rule, the elements of the $\mathcal{T}_i$ are sets of clauses such that either a clause is a ground literal or it does not contain any ground literal.

Using Lemma 4, we obtain by induction that for every i, for every $T \in \mathcal{T}_i$, we can write $T = \mathcal{I} \cup \mathcal{S} \cup \mathcal{P}' \cup \mathcal{J}$, where $\mathcal{P}'$ and $\mathcal{J}$ are well-behaved elements of the classes $\mathcal{C}_P$ and $\mathcal{C}_J$ respectively.

Let $N \overset{\text{def}}{=} \max_{C \in \mathcal{T}_0} ||C||$. Applying now Lemma 5 and induction, we deduce that for every i, for every $T \in \mathcal{T}_i$, for every $C \in T$, we have that $||C|| \leq N$ if C is not ground and $||C|| \leq 2N$ if C is ground.

From the definition of classes $\mathcal{C}_I, \mathcal{C}_S, \mathcal{C}_P$ and $\mathcal{C}_J$ we observe that clauses in sets $T \in \mathcal{T}_i$, for every i, have at most k variables, where k is the maximal arity of function symbols in $\mathcal{F}$. Moreover, in these clauses, the same literal cannot appear twice, thanks to the factorization rule.

Since there is a finite number of sets of clauses of bounded depth (up to variable renaming and repetition of literals), we deduce that the $\mathcal{R}_{\mathcal{I} \cup \mathcal{S}}$ resolution terminates.

With regard to the complexity of this decision procedure we can obtain, using a similar argument as in [10], that the satisfiability of the set $\mathcal{I} \cup \mathcal{P} \cup \mathcal{S}$ is decidable in 3-EXPTIME.

3.4 Examples

In this section we show that the intruder clauses corresponding to our two examples (CBC encryption and blind signatures) are saturated. This means that we can analyze any protocol encoded in $\mathcal{C}_P$ under an extended intruder that has access either to CBC encryption or blind signatures. H. Comon-Lundh and V. Cortier [8] have identified these protocols to be protocols for which, at each transition, at most one part of the message is blindly copied or tested.

We assume a fixed basic intruder power, modelled by the set $\mathcal{I}$ presented in Figure 1.

We first consider the case of the CBC encryption.

Proposition 6. *The set $\mathcal{I} \cup \{C_{pre}\}$ is saturated.*

Proof. Given a partial interpretation $\mathfrak{I}$ and an atom A belonging to $\mathfrak{I}$, we say that A is *true* (resp. *false*) in $\mathfrak{I}$ if A appears with sign $+$ (resp. with sign $-$).

We consider an ordered resolution between clauses of $\mathcal{I} \cup \{C_{pre}\}$. If both premises are clauses of $\mathcal{I}$ then all the resolvents are tautologies. Therefore they are satisfiable for any partial interpretation. The only interesting case is when one of the premise is C_{pre}. In that case, the other premise is necessarily $-I(x) \vee -I(y) \vee +I(\{x\}_y)$. The resolvent of these two clauses is $C \overset{\text{def}}{=} -I(\langle x, y\rangle) \vee -I(z) \vee +I(\{x\}_z)$. We consider an arbitrary partial interpretation $\mathfrak{I}$ such that C is unsatisfiable in $\mathfrak{I}$. By definition, there exists a ground substitution θ such that $C\theta$ is false in $\mathfrak{I}$. The clause $C\theta$ has the form $-I(\langle u, v\rangle) \vee -I(w) \vee +I(\{u\}_w)$, where u, v and w are ground terms. Thus the literals $+I(\langle u, v\rangle)$, $+I(w)$ and $-I(\{u\}_w)$ are in $\mathfrak{I}$. Also, since $u < \{u\}_w$, one of the literals $+I(u)$ or $-I(u)$ must appear in $\mathfrak{I}$. We consider the two cases.

- Either the atom $I(u)$ is true in $\mathfrak{I}$ then the clause $-I(u) \vee -I(w) \vee +I(\{u\}_w)$ is false in $\mathfrak{I}$ and it follows that the clause $-I(x) \vee -I(y) \vee +I(\{x\}_y)$ is unsatisfiable in $\mathfrak{I}$;
- Or the atom $I(u)$ is false in $\mathfrak{I}$ then the clause $-I(\langle u, v\rangle) \vee +I(u)$ is false in $\mathfrak{I}$ so the clause $-I(\langle x, y\rangle) \vee +I(x)$ is unsatisfiable in $\mathfrak{I}$.

In both cases a clause of $\mathcal{I}$ is unsatisfiable in $\mathfrak{I}$. We conclude that the set $\mathcal{I} \cup \{C_{pre}\}$ is saturated. □

The same property is true in the blind signature case.

Proposition 7. *The set $\mathcal{I} \cup \{C_{sig}\}$ is saturated.*

Proof. The proof is analogue to the previous one. It relies on the fact that the clauses $-I(x) \vee -I(z) \vee +I(\text{sign}(x, z))$ and $-I(\text{blind}(x, y)) \vee -I(y) \vee +I(x)$ belong to $\mathcal{I}$. □

As a consequence of these two propositions and applying Theorem 3, we get that for any well-behaved set $\mathcal{P}$ (encoding both a protocol and a security property), the satisfiability of $\mathcal{I} \cup \{C_{pre}\} \cup \mathcal{P}$ (resp. of $\mathcal{I} \cup \{C_{sig}\} \cup \mathcal{P}$) is decidable. Since for example secrecy can be modeled using a ground clause (for example $-I(\text{n}(a, b))$ to express that the intruder should not learn the nonce between a and b), we obtained a procedure for deciding the secrecy of protocols that use the described prefix property or blind signature.

Corollary 8. *The secrecy problem for cryptographic protocols with single blind copying, with bounded number of nonces but unbounded number of sessions, using as primitives CBC encryption or blind signature is decidable.*

In addition, in the case of other extensions of the intruder power leading to other sets $\mathcal{S}$ of clauses in $\mathcal{C}_S$, the saturation of the set $\mathcal{I} \cup \mathcal{S}$ can be easily verified by hand (like in our examples).

4. APPLICATION TO A CRYPTOGRAPHIC PROTOCOL

4.1 Presentation of the protocol

We consider the Needham-Schroeder symmetric key authentication protocol [16] as an example of application of our result. The goal of the protocol is the key exchange between two parties, which we call Alice and Bob, and the mutual conviction of the possession of the key by each other. The key is created by a trusted server which shares the secret keys K_{as} and K_{bs} with Alice and Bob respectively. The description of the protocol is as follows:

$$P_{\mathsf{NS}} : \left\{ \begin{array}{ll} A \Rightarrow S : & A, B, N_a \\ S \Rightarrow A : & \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}} \\ A \Rightarrow B : & \{K_{ab}, A\}_{K_{bs}} \\ B \Rightarrow A : & \{N_b\}_{K_{ab}} \\ A \Rightarrow B : & \{N_b - 1\}_{K_{ab}} \end{array} \right.$$

Here we concentrate on the key exchange goal, rather than on the authentication of the two parties. The key exchange goal can be expressed as the secrecy of the nonce N_b. Intuitively, if N_b remains secret, it means that the key K_{ab} used by B has also been kept secret.

If the cryptosystem used to implement this protocol uses for example CBC then the following attack [17] is possible. In a first session (1), an intruder can listen to the message $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$ and then, using the CBC property, he can compute $\{N_a, B\}_{K_{as}}$. In another session of the protocol, he can send it to Alice in the third round. Alice thinks that Bob has started a session (2) with her: Bob plays the role of the initiator and Alice the role of the second participant. And so Alice would use N_a as the shared

$$\begin{array}{lll}
(1).1 & A \Rightarrow S: & A, B, N_a \\
(1).2 & S \Rightarrow A: & \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}} \\
(2).3 & I(B) \Rightarrow A: & \{N_a, B\}_{K_{as}} \\
(2).4 & A \Rightarrow I(B): & \{N'_a\}_{N_a}
\end{array}$$

Figure 3: Attack on the Needham-Schroeder protocol, using the prefix property

key, while it is a publicly known message. This attack is summarized in Figure 3.

The clauses that model the protocol rules are the following ones:

$$\begin{aligned}
&\rightarrow I(\langle \alpha, \beta, \mathrm{n}_1(\alpha, \beta)\rangle) \\
I(\langle \alpha, \beta, x\rangle) &\rightarrow I(\{x, \beta, \mathrm{k}(\alpha, \beta), \\
&\qquad\qquad \{\mathrm{k}(\alpha, \beta), \alpha\}_{\mathrm{k}(\beta, s)}\}_{\mathrm{k}(\alpha, s)}) \\
I(\{\mathrm{n}_1(\alpha, \beta), \beta, y, z\}_{\mathrm{k}(\alpha, s)}) &\rightarrow I(z) \\
I(\{y, \alpha\}_{\mathrm{k}(\beta, s)}) &\rightarrow I(\{\mathrm{n}_2(\beta, \alpha)\}_y) \\
I(\{x\}_y) &\rightarrow I(\{\mathrm{pred}(x)\}_y)
\end{aligned}$$

where α and β range over constants that denote the identities of the involved parties. Comon and Cortier [9] have shown that, for secrecy properties, it is sufficient to verify the correctness of a protocol for only three parties: two honest and one dishonest participants. Hence we consider three agents having their identities represented by the constants a, b and i, where a and b stand for the honest participants while i stands for the dishonest participant. We denote by $C_{\mathsf{NS}_j}(\alpha, \beta)$, for j from 1 to 5, the clauses listed above, corresponding to the rules of the protocol P_{NS} when the initiator is α and the responder is β. The set of clauses that model the rules of the protocol is

$$\mathcal{P}_{\mathsf{NS}} \stackrel{\text{def}}{=} \bigcup_{1 \leq j \leq 5,\ \alpha, \beta \in \{a, b, i\}, \alpha \neq \beta} \{C_{\mathsf{NS}_j}(\alpha, \beta)\}.$$

The intruder has also some initial knowledge. He knows the identities of the participating parties, he can create nonces and, he knows the secret key of the compromised agent. This initial knowledge is modeled by the following clauses:

$$\begin{array}{ll}
\rightarrow I(a) & \rightarrow I(\mathrm{k}(i, s)) \\
\rightarrow I(b) & \rightarrow I(\mathrm{n}_1(i, x)) \\
\rightarrow I(i) & \rightarrow I(\mathrm{n}_2(i, x))
\end{array}$$

We denote this set of clauses, corresponding to the initial knowledge of the intruder, by $\mathcal{P}_0$. We remark that these clauses are either ground or with a single variable thus belong to $\mathcal{C}_P$.

In addition, we enrich the set $\mathcal{I}$ (defined in Section 2.2) with he clause $I(x) \rightarrow I(\mathrm{pred}(x))$ that models the ability of the intruder to compute the predecessor of a message (seen as a number).

4.2 Correcting the protocol

We remark that the attack comes from the fact that the intruder, using the second rule of the protocol together with the CBC property, can get the encryption of any message by the key K_{as}: replacing the nonce N_a by any plaintext m of its choice in the first message, he obtains a message of the form $\{m, \ldots\}_{K_{as}}$ from the server and using the CBC property he gets $\{m\}_{K_{as}}$.

To avoid this, we interchange the place of N_a and B in the message sent in the second round. But a similar attack is still possible since the intruder can modify the first message of Alice and send $\langle A, B, B\rangle$ to the server. Then the shared key would be the identity B. Such an attack is possible only if identities can be confused with keys.

To avoid such a type flaw attack, we add a hash of the shared key as the first component of the message sent by the server to Alice and then to Bob. Note that this second transformation is not sufficient by itself since the intruder has also the ability to produce hashes. The obtained protocol is described below. We refer to this version as the corrected protocol.

$$P_{\mathsf{NSc}} : \left\{ \begin{array}{ll}
A \Rightarrow S: & A, B, N_a \\
S \Rightarrow A: & \{B, N_a, K_{ab}, \{h(K_{ab}), K_{ab}, A\}_{K_{bs}}\}_{K_{as}} \\
A \Rightarrow B: & \{h(K_{ab}), K_{ab}, A\}_{K_{bs}} \\
B \Rightarrow A: & \{N_b\}_{K_{ab}} \\
A \Rightarrow B: & \{N_b - 1\}_{K_{ab}}
\end{array} \right.$$

The clauses that model the rules of this protocol are the following ones:

$$\begin{aligned}
&\rightarrow I(\langle \alpha, \beta, \mathrm{n}_1(\alpha, \beta)\rangle) \\
I(\langle \alpha, \beta, x\rangle) &\rightarrow I(\{\beta, x, \mathrm{k}(\alpha, \beta), \{\mathrm{h}(\mathrm{k}(\alpha, \beta)), \\
&\qquad\qquad \mathrm{k}(\alpha, \beta), \alpha\}_{\mathrm{k}(\beta, s)}\}_{\mathrm{k}(\alpha, s)}) \\
I(\{\beta, \mathrm{n}_1(\alpha, \beta), y, z\}_{\mathrm{k}(\alpha, s)}) &\rightarrow I(z) \\
I(\{\mathrm{h}(y), y, \alpha\}_{\mathrm{k}(\beta, s)}) &\rightarrow I(\{\mathrm{n}_2(\beta, \alpha)\}_y) \\
I(\{x\}_y) &\rightarrow I(\{\mathrm{pred}(x)\}_y)
\end{aligned}$$

As for the protocol P_{NS}, we denote by $\mathcal{P}_{\mathsf{NSc}}$ the set of clauses obtained from the ones presented above by instantiating α and β ($\alpha \neq \beta$) by the constants a, b and i.

The aim of the rest of the section is to prove that the corrected protocol preserves the secrecy of N_b.

4.3 A transformation preserving secrecy

We observe that the clauses corresponding to the third round and fifth round of the protocol P_{NSc} are not in $\mathcal{C}_P$ since they have two variables. Therefore we cannot apply directly our result and we are led to an additional modification of the protocol.

We remark that the server sends to Alice in the second round an encrypted message that Alice cannot decrypt. This message could be directly sent to Bob by the server. In addition, the last rule of the protocol does not seem to be able to compromise the secrecy of N_b, thus we remove it. These modifications yield the following protocol:

$$P_{\mathsf{NSv}} : \left\{ \begin{array}{ll}
A \Rightarrow S: & A, B, N_a \\
S \Rightarrow A: & \{B, N_a, K_{ab}\}_{K_{as}} \\
S \Rightarrow B: & \{h(K_{ab}), K_{ab}, A\}_{K_{bs}} \\
B \Rightarrow A: & \{N_b\}_{K_{ab}}
\end{array} \right.$$

The set of clauses that model the protocol are listed below:

$$\begin{aligned}
&\rightarrow I(\langle \alpha, \beta, \mathrm{n}_1(\alpha, \beta)\rangle) \\
I(\langle \alpha, \beta, x\rangle) &\rightarrow I(\{\beta, x, \mathrm{k}(\alpha, \beta)\}_{\mathrm{k}(\alpha, s)}) \\
I(\langle \alpha, \beta, x\rangle) &\rightarrow I(\{\mathrm{h}(\mathrm{k}(\alpha, \beta)), \mathrm{k}(\alpha, \beta), \alpha\}_{\mathrm{k}(\beta, s)}) \\
I(\{\mathrm{h}(y), y, \alpha\}_{\mathrm{k}(\beta, s)}) &\rightarrow I(\{\mathrm{n}_2(\beta, \alpha)\}_y)
\end{aligned}$$

As before, we denote by $\mathcal{P}_{\mathsf{NSv}}$ the set of clauses obtained by instantiating α and β ($\alpha \neq \beta$) with the constants a, b and i in the clauses presented above.

We emphasize that we do not pretend that this new version is a realistic protocol. It should be view as a toy protocol, used to prove the secrecy of the corrected protocol. Our approach is as follows: we prove that this version is a weaker version than the corrected protocol, *i.e.* that its correctness implies the correctness of the corrected version. Then, since this version fits our class, we apply our resolution method to prove that this version preserves the secrecy of N_b, which allow us to conclude that the corrected version also preserves the secrecy of N_b.

For each protocol P_l, where l is NS, NSc or NSv, we note by $T_l \overset{\text{def}}{=} \mathcal{I} \cup \mathcal{P}_0 \cup \mathcal{P}_l \cup \{C_{pre}\}$ the entire set of clauses that model the protocol ($\mathcal{P}_l$ is the set of clauses representing only the rounds of the protocol). The secrecy property of the protocol P_l can be formulated as the satisfiability of the set of clauses $T_l \cup \{-I(\mathrm{n}_2(b,a))\}$.

We have already seen that $T_{\mathsf{NS}} \cup \{-I(\mathrm{n}_2(b,a))\}$ is not satisfiable. We prove that the satisfiability of $T_{\mathsf{NSc}} \cup \{-I(\mathrm{n}_2(b,a))\}$ can be reduced to the satisfiability of $T_{\mathsf{NSv}} \cup \{-I(\mathrm{n}_2(b,a))\}$.

Proposition 9. *If the set of clauses $T_{\mathsf{NSv}} \cup \{-I(\mathrm{n}_2(b,a))\}$ is satisfiable then the set of clauses $T_{\mathsf{NSc}} \cup \{-I(\mathrm{n}_2(b,a))\}$ is also satisfiable.*

To prove this proposition, we use another variant of the resolution method, the *positive* resolution [1], which requires that one of the premise is a positive clause. The method is also refutationally complete. Since we consider Horn clauses, the set $T_l \cup \{-I(\mathrm{n}_2(b,a))\}$ is unsatisfiable if and only if there is a deduction of the clause $+I(\mathrm{n}_2(b,a))$ by positive resolution on T_l. We denote by $P_l \vdash I(m)$ the fact that the clause $+I(m)$ can be obtained by positive resolution on T_l.

The following property ensures that the transformation of protocol P_{NSc} in P_{NSv} preserves the secrecy. In other words, if there is an attack in P_{NSc} then there is a corresponding attack in P_{NSv}.

Proposition 10. *If $P_{\mathsf{NSc}} \vdash I(\mathrm{n}_2(b,a))$ then $P_{\mathsf{NSv}} \vdash I(\mathrm{n}_2(b,a))$.*

Proof. To prove the proposition, it is sufficient to find an application $t \mapsto \overline{t}$ on the set of ground terms such that $\overline{\mathrm{n}_2(b,a)} = \mathrm{n}_2(b,a)$ and, for all message m, if $P_{\mathsf{NSc}} \vdash I(m)$ then $P_{\mathsf{NSv}} \vdash I(\overline{m})$. We show that the following application satisfies the required properties.

$$\begin{aligned}
\overline{a} =& \quad a, \text{ for all constant } a \\
\overline{x} =& \quad x, \text{ for all variable } x \\
\overline{\{u\}_v} =& \quad \begin{cases} \langle \{a, \mathrm{n}_1(a,b), \overline{r}\}_{\mathrm{k}(a,s)}, \overline{t}\rangle \\ \qquad \text{if } \{u\}_v = \{b, \mathrm{n}_1(a,b), r, t\}_{\mathrm{k}(a,s)}, \\ \mathrm{n}_1(i,i) \quad \text{if } u = \mathrm{pred}(r), \\ \{\overline{u}\}_{\overline{v}} \quad \text{otherwise.} \end{cases} \\
\overline{\mathrm{pred}(u)} =& \quad \mathrm{n}_1(i,i) \\
\overline{f(u_1,\ldots,u_n)} =& \quad f(\overline{u_1},\ldots,\overline{u_n}), \forall f \in \mathcal{F}, f \neq \{\_\}_\_, f \neq \mathrm{pred}
\end{aligned}$$

In what follows, a, b are arbitrary constants, r, t are arbitrary terms, while i and s are fixed constants, standing for the intruder and server identities.

We restrict ourselves to deductions on the form:

$$\frac{C \overset{\text{def}}{=} \bigvee_{i=1}^{n} -I(m_i) \vee +I(m) \quad +I(m_1) \;\cdots\; +I(m_n)}{+I(m)}$$

where C is an instance of a clause of $\mathcal{P}_{\mathsf{NSc}}$. Thus we are reduced to show that, for each clause $\bigvee_{i=1}^{n} -I(m_i) \vee +I(m)$ that is a ground instance of a clause C of $\mathcal{P}_{\mathsf{NSc}}$, if $P_{\mathsf{NSv}} \vdash I(\overline{m_i})$, for every i, then $P_{\mathsf{NSv}} \vdash I(\overline{m})$. We only present here the more difficult cases.

- $C = -I(x) \vee -I(y) \vee +I(\{x\}_y)$. We have to verify that if $P_{\mathsf{NSv}} \vdash I(\overline{u})$ and $P_{\mathsf{NSv}} \vdash I(\overline{v})$, where u and v are two ground terms, then $P_{\mathsf{NSv}} \vdash I(\overline{\{u\}_v})$.

  Suppose that $\{u\}_v$ is of the form $\{b, \mathrm{n}_1(a,b), r, t\}_{\mathrm{k}(a,s)}$. Then we have that $P_{\mathsf{NSv}} \vdash I(\langle b, \mathrm{n}_1(a,b), \overline{r}, \overline{t}\rangle)$ and $P_{\mathsf{NSv}} \vdash I(\mathrm{k}(a,s))$. The projection clauses are in T_{NSv}. Using them with the first relation we obtain $P_{\mathsf{NSv}} \vdash I(\langle b, \mathrm{n}_1(a, b), \overline{r}\rangle)$ and $P_{\mathsf{NSv}} \vdash I(\overline{t})$. Now using the encryption clause and then the pairing clause we obtain that $P_{\mathsf{NSv}} \vdash I(\langle\{b, \mathrm{n}_1(a,b), \overline{r}\}_{\mathrm{k}(a,s)}, \overline{t}\rangle)$, which is what we needed.

  Suppose now that $u = \mathrm{pred}(r)$. Then we have $\overline{\{u\}_v} = \mathrm{n}_1(i,i)$. But $P_{\mathsf{NSv}} \vdash \overline{\mathrm{n}_1(i,i)}$, as $+I(\mathrm{n}_1(i,i))$ is a clause from $\mathcal{P}_0$.

  If we are in none of these two special cases then it is sufficient to use the encryption clause in order to obtain that $P_{\mathsf{NSv}} \vdash I(\overline{\{u\}_v})$.

- $C = -I(\{x\}_y) \vee -I(y) \vee +I(x)$. We have to show that if $P_{\mathsf{NSv}} \vdash I(\overline{\{u\}_v})$ and $P_{\mathsf{NSv}} \vdash I(\overline{v})$, where u and v are two ground terms, then $P_{\mathsf{NSv}} \vdash I(\overline{u})$.

  If $\{u\}_v = \{b, \mathrm{n}_1(a,b), r, t\}_{\mathrm{k}(a,s)}$ then we have $P_{\mathsf{NSv}} \vdash I(\langle\{b, \mathrm{n}_1(a,b), \overline{r}\}_{\mathrm{k}(a,s)}, \overline{t}\rangle))$. Therefore we obtain $P_{\mathsf{NSv}} \vdash I(\{b, \mathrm{n}_1(a,b), \overline{r}\}_{\mathrm{k}(a,s)})$ and $P_{\mathsf{NSv}} \vdash I(\overline{t})$. But we also have $P_{\mathsf{NSv}} \vdash I(\mathrm{k}(a,s))$. And, as the decryption clause is in the model of P_{NSv}, we obtain $P_{\mathsf{NSv}} \vdash I(\langle b, \mathrm{n}_1(a,b), \overline{r}\rangle)$. From which we arrive at the desired relation $P_{\mathsf{NSv}} \vdash I(\langle b, \mathrm{n}_1(a,b), \overline{r}, \overline{t}\rangle)$.

  If $u = \mathrm{pred}(r)$ then there is nothing to prove because $\overline{\mathrm{pred}(u)} = \mathrm{n}_1(i,i)$ and $P_{\mathsf{NSv}} \vdash I(\mathrm{n}_1(i,i))$. Otherwise the proof is direct.

- $C = -I(\langle a,b,x\rangle) \vee +I(\{b, x, \mathrm{k}(a,b), \{\mathrm{h}(\mathrm{k}(a,b)), \mathrm{k}(a,b), a\}_{\mathrm{k}(b,s)}\}_{\mathrm{k}(a,s)})$. Knowing that $P_{\mathsf{NSv}} \vdash I(\overline{\langle a,b,u\rangle})$, where u is a ground term, we must obtain that the transformed positive literal of C is deductible from T_{NSv}.

  The second clause of $\mathcal{P}_{\mathsf{NSv}}$ assures that we have $P_{\mathsf{NSv}} \vdash I(\{b, \overline{u}, \mathrm{k}(a,b)\}_{\mathrm{k}(a,s)})$. Applying the pairing clause and the third clause of $\mathcal{P}_{\mathsf{NSv}}$, we obtain what we needed, *i.e.* $P_{\mathsf{NSv}} \vdash I(\langle\{b, \overline{u}, \mathrm{k}(a,b)\}_{\mathrm{k}(a,s)}, \{\mathrm{h}(\mathrm{k}(a,b)), \mathrm{k}(a,b), a\}_{\mathrm{k}(b,s)}\rangle)$.

- $C = -I(\{b, \mathrm{n}_1(a,b), y, z\}_{\mathrm{k}(b,s)}) \vee +I(z)$. For any two ground terms u and v, we must prove that if $P_{\mathsf{NSv}} \vdash I(\overline{\{b, \mathrm{n}_1(a,b), u, v\}_{\mathrm{k}(b,s)}})$ then $P_{\mathsf{NSv}} \vdash I(\overline{v})$. But this is immediate, from the definition of the application and by using the projection on the second component.

- $C = -I(\{x\}_y) \vee +I(\{\mathrm{pred}(x)\}_y)$. As we have $P_{\mathsf{NSv}} \vdash I(\mathrm{n}_1(i,i))$ and, for all ground terms u and v, $\overline{\{\mathrm{pred}(u)\}_v} = \mathrm{n}_1(i,i)$, this case is trivial.

The conclusion in the remaining cases follows directly from the definition of the application $t \mapsto \overline{t}$. □

4.4 Secrecy of the corrected protocol

We have verified using our resolution method that the transformed protocol P_{NSv} has no attack. Since the clauses of T_{NSv} verify the hypotheses of our main result, this protocol could be verified using an automatic tool. We have checked it by hand, because an implementation has not been done yet; though it is conceivable.

Proposition 11. *The set of clauses $T_{\mathsf{NSv}} \cup \{-I(\mathrm{n}_2(b,a))\}$ is satisfiable.*

We can state now the correctness of the protocol P_{NSc}.

Corollary 12. *The set of clauses $T_{\mathsf{NSc}} \cup \{-I(\mathrm{n}_2(b,a))\}$ is satisfiable.*

Proof. Immediate, by propositions 9 and 11. □

5. PROOFS OF INTERMEDIATE RESULTS

5.1 Invariance under resolution

We show in this subsection that our resolution method on a set of clauses in the class $\mathcal{C} \cup \mathcal{C}_J$ produces a set of resolvents that remains in this class.

Lemma 4. *Let $\mathcal{I}$, $\mathcal{P}'$, $\mathcal{S}$ and $\mathcal{J}$ be sets of clauses of respectively $\mathcal{C}_I$, $\mathcal{C}_P$, $\mathcal{C}_S$ and, $\mathcal{C}_J$, such that $\mathcal{I} \cup \mathcal{S}$ is saturated and $\mathcal{P}'$, $\mathcal{S}$ and $\mathcal{J}$ are well-behaved. The application of $\mathcal{R}_{\mathcal{I}\cup\mathcal{S}}$ resolution on $\mathcal{I} \cup \mathcal{S} \cup \mathcal{P}' \cup \mathcal{J}$ produces clauses in $\mathcal{C}_P$ or $\mathcal{C}_J$. Moreover, the set of resolvents is well-behaved.*

Proof. Let C_1 and C_2 be clauses in $\mathcal{I} \cup \mathcal{S} \cup \mathcal{P}' \cup \mathcal{J}$. We write $C_1 = C_1' \vee L_1$ and $C_2 = C_2' \vee L_2$. Let C be the resolvent of C_1 and C_2 and $\sigma = \mathrm{mgu}(L_1, L_2)$. We have to prove that the clause C is in the class $\mathcal{C}_P$ or $\mathcal{C}_J$. In order to obtain this, we examine all possible cases according to the membership of C_1 and C_2 to the sets $\mathcal{I}$, $\mathcal{S}$, $\mathcal{P}'$ and $\mathcal{J}$.

For $l \in \{0,1\}$, if C_l belongs to $\mathcal{I}$, $\mathcal{S}$ or $\mathcal{J}$ then C_l is written as in the definition of classes $\mathcal{C}_I$, $\mathcal{C}_S$ and $\mathcal{C}_J$, respectively. If $C_l \in \mathcal{P}'$ and C_l is ground then the resolvent is also ground and hence in $\mathcal{C}_P$. Therefore, in what follows, we suppose that for $C_l \in \mathcal{P}'$, C_l is not ground. Hence we write $C_l = \pm I(s(x)) \vee \bigvee_{i=1}^{m} \pm I(t_i(x))$, where $m \geq 0$, and we assume that the resolution inference is performed on the literal $\pm I(s(x))$. If $C_l \in \mathcal{J}$ then we assume that the literal of C_l upon which resolution is performed is $\pm I(w_1[g(y_1, \ldots, y_k)])$.

The case study follows:

- $C_1, C_2 \in \mathcal{P}'$: The resolvent C has at most one variable hence C is a clause of $\mathcal{C}_P$.

- $C_1 \in \mathcal{P}'$ and $C_2 \in \mathcal{S}$: By maximality of L_2 in C_2 we deduce that $L_2 = -I(f_0(u[g(y_1, \ldots, y_k)], v))$. The literal L_1 is $+I(s(x))$. The following two cases are possible:

  – $s(x) = x$. Then $C_1 = +I(x)$ and so the resolvent is an instance of C_1. Since subsumed clauses can be eliminated then this case does not produce a new clause.

  – $s(x) = f_0(s_1, s_2)$. By hypothesis, $s(x)$ is well-behaved.

    Suppose that $V(s_1) \neq \emptyset$. Then $s_2 = a$, for some constant a. We deduce that $v = z$ and $z\sigma = a$. The following two cases are possible: $y_i\sigma = s_i'$, where each s_i' is a subterm of s_1 and by consequence the resolvent is in $\mathcal{C}_P$; or, $x\sigma = u'[g(y_1, \ldots, y_k)]$, where u' is a subterm of u, and, in this case, the resolvent is in $\mathcal{C}_J$.

    Suppose now that $V(s_2) \neq \emptyset$. Then s_1 is a ground term. As before, two cases are possible, and in both cases the resolvent is a clause of $\mathcal{C}_P$.

    Observe that, in all the cases, $f_0(y_j, z)\sigma$ is well-behaved. Since u, v and for each i, w_i are well-behaved we obtain, following the same line of proof as in Lemma 13, that the resolvent is also well-behaved.

- $C_1 \in \mathcal{P}'$ and $C_2 \in \mathcal{I}$: We have $L_1 = \pm I(s(x))$ and $L_2 = \mp I(f(x_1, \ldots, x_n))$. The following two cases are possible:

  – $s(x) = x$. Then, by maximality of $s(x)$ we have $C_1 = \pm I(x)$. Therefore C is subsumed by C_1.

  – $s(x) = f(s_1, \ldots, s_n)$, where, for all i, s_i is a subterm of $s(x)$. Hence, for all i, $x_i\sigma = s_i$. So the resolvent is in $\mathcal{C}_P$.

- $C_1 \in \mathcal{P}'$ and $C_2 \in \mathcal{J}$: We have $L_1 = \pm I(s(x))$ and $L_2 = \mp I(w_1[g(y_1, \ldots, y_k)])$. Again, two cases are possible with regard to the form of σ:

  – σ is ground. Then the resolvent is in $\mathcal{C}_P$.

  – $x\sigma = x$ and for all i, $y_i\sigma = s_i(x)$, where s_i is a subterm of $s(x)$. In this case C is in $\mathcal{C}_P$.

  – for all i, $y_i\sigma = y_i$ and $x\sigma = w'[g(y_1, \ldots, y_k)]$, where w' is a subterm of w_1. Hence in this case C is in $\mathcal{C}_J$.

- $C_1, C_2 \in \mathcal{I} \cup \mathcal{S}$: the strategy forbids any resolution in this case.

- $C_1 \in \mathcal{I}$ and $C_2 \in \mathcal{J}$: We have $L_1 = \pm I(f(x_1, \ldots, x_n))$ and $L_2 = \mp I(w_1[g(y_1, \ldots, y_k)])$. As before, two cases are possible:

  – $w_1(z) = z$ and $g = f$. The clauses derived by resolution (and possibly splitting) belong to $\mathcal{C}_P$.

  – for all i, $x_i\sigma = w_i'$, where w_i' is a subterm of $w_1[g(y_1, \ldots, y_k)]$. In this case the resolvent is in $\mathcal{C}_J$ if the substitution is not ground and is in $\mathcal{C}_P$ otherwise.

- $C_1, C_2 \in \mathcal{S} \cup \mathcal{J}$: Since none of the positive literals in C_1, C_2 is maximal in its clause, the resolution inferences are blocked.

To finish the proof of the lemma we have to show that the resolvent C is well-behaved. This is a consequence of the invariance of the well-behavior property under resolution, fact proven in the following lemma. □

Lemma 13. *The resolvent of two well-behaved clauses of $\mathcal{C}_P \cup \mathcal{C}_J$ is well-behaved.*

Proof. Let $\sigma = \mathrm{mgu}(w_1, w_2)$, where $C_1 = C_1' \vee +I(w_1)$ and $C_2 = C_2' \vee -I(w_2)$ are the two premisses. Suppose that there is a literal $I(w)$ in $C_1' \vee C_2'$ and a subterm $f_0(u, v)$ of $w\sigma$ such that $f_0(u, v)$ is not well-behaved.

We have the following possibilities for $f_0(u, v)$: either it is a subterm of w, either $f_0(u, v) = f_0(u', v')\sigma$ where $f_0(u', v')$ is a subterm of w, either it is a subterm of $x\sigma$, where $x \in V(w)$. In the first two cases we obtain that w is not well-behaved as we can easily see a subterm that doesn't respect the conditions. In the third case we can show that there must be a subterm $f_0(u'', v'')$ of a w_1 or w_2 and a substitution θ such that $f_0(u'', v'')\theta = x\sigma$. Therefore w_1 or w_2 is not well-behaved. Hence we obtained a contradiction and so the assumption was false. □

5.2 Termination of the resolution method

We now show that every clause derived by our resolution strategy has its size bounded by the maximum of the sizes of its premises. This will imply that only a finitely many different clauses can be derived by this strategy.

Lemma 5. *Let C_1 and C_2 be two clauses of $\mathcal{C}_P$, $\mathcal{C}_I$, $\mathcal{C}_S$ or $\mathcal{C}_J$, such that if C_i, where $i \in 1, 2$, is in $\mathcal{C}_P$, $\mathcal{C}_S$ or $\mathcal{C}_J$ then it is well-behaved. The resolvent C derived by ordered resolution satisfies: $||C|| \leq \max(||C_1||, ||C_2||)$ if C is not ground and $||C|| \leq 2\max(||C_1||, ||C_2||)$ if C is ground.*

Proof. Let $C_1 = I(u_1) \vee C_1'$, $C_2 = -I(u_2) \vee C_2'$ and $\sigma = \mathrm{mgu}(u_1, u_2)$. It suffices to show that for every term w such that $\pm I(w) \in C_1 \vee C_2$ we have $|w\sigma| \leq \max(|w|, |u_1|, |u_2|)$ if σ is not ground and $|w\sigma| \leq 2\max(|w|, |u_1|, |u_2|)$ otherwise. We observe that $|w\sigma| = \max(|w|, |w|_x + |x\sigma| - 1)$. We have to consider the same cases as in the proof of Lemma 4 to show these properties.

First let us take $C_1, C_2 \in \mathcal{C}_P$. If one of the premises is ground then the conclusion is immediate. Then we can assume $w, u_1 \in \mathcal{T}(\mathcal{F} \cup \{x\})$ and $u_2 \in \mathcal{T}(\mathcal{F} \cup \{y\})$. Since the literal that is resolved is maximal in its clause $u_1\sigma \not< w\sigma$ and $u_2\sigma \not< w\sigma$. Since $<$ is liftable we have $u_1 \not< w$ and $u_2 \not< w$. As in Proposition 8.5 of [10] we have to examine 4 cases:

- $x\sigma = y\sigma = z$. Then $w\sigma = w$.

- $x\sigma = u_2'$, $y\sigma = y$, where u_2' is a subterm of u_2. We have $u_1\sigma = u_2\sigma = u_2$. If $|w|_x \leq |u_1|_x$ then we have

  $$|w\sigma| = \max(|w|, |w|_x + |x\sigma| - 1) \leq \max(|w|, |u_1|_x + |x\sigma| - 1) \leq \max(|w|, |u_1\sigma|) = \max(|w|, |u_1|, |u_2|).$$

  And if $|w|_x > |u_1|_x$ then $|w\sigma|_y > |u_1\sigma|_y$, therefore $|w\sigma| < |u_1\sigma|$, since otherwise $u_1\sigma < w\sigma$.

- $x\sigma = x$, $y\sigma = u_1'$, where u_1' is a subterm of u_1. Then $w\sigma = w$.

- σ is ground. Then $x\sigma$ is a ground subterm of u_2 or $y\sigma$ is a ground subterm of u_1. In the first case (the other one is similar) we also have $|w\sigma| \leq |u_1\sigma|$ since otherwise $w\sigma > u_1\sigma$. Hence

  $$|w\sigma| \leq |u_1\sigma| = \max(|u_1|, |u_1|_x + |x\sigma| - 1) \leq \leq |u_1| + |u_2| \leq 2\max(|u_1|, |u_2|).$$

We consider now the case where $C_1 \in \mathcal{C}_P$ and $C_2 \in \mathcal{C}_S$. Then $u_2 = f_0(u[g(y_1, \ldots, y_k)], v)$, where u and v are as in the Definition 3. Firstly we prove that $|w\sigma| \leq \max(|w|, |u_1\sigma|, |u_2\sigma|)$ (recall that $u_1\sigma = u_2\sigma$). Indeed, if $\pm I(w)$ is a literal of C_2 then $w\sigma < u_2\sigma$, as $-I(u_2)$ is the greatest literal among the literals of C_2. Suppose now that $\pm I(w)$ is a literal of C_1. We compare the depth of w with that of u_1.

- Case $|w| > |u_1|$. Then $|w|_x \leq |u_1|_x$, since $u_1\sigma$ is maximal. We have that

  $$|w\sigma| = \max(|w|, |w|_x + |x\sigma| - 1) \leq \leq \max(|w|, |u_1|_x + |x\sigma| - 1) \leq \max(|w|, |u_1\sigma|).$$

- Case $|w| \leq |u_1|$. If $|w|_x \leq |u_1|_x$ then $|w\sigma| \leq |u_1\sigma|$. If $|w|_x > |u_1|_x$ then for all $y \in V(x\sigma)$, $|w\sigma|_y > |u_1\sigma|_y$. Therefore $|w\sigma| \leq |u_1\sigma|$ because otherwise $u_1\sigma < w\sigma$.

Since we have proved that $|w\sigma| \leq \max(|w|, |u_1\sigma|, |u_2\sigma|)$ it is now sufficient to show that

$$|u_1\sigma| = |u_2\sigma| = \max(|u_1|, |u_2|).$$

If $x\sigma = x$ then it is indeed the case because we have $u_1\sigma = u_1$. Suppose now $x\sigma \neq x$. We write $u_1 = f_0(s_1, s_2)$. By hypothesis, u_1 is well-behaved.

- If $V(s_1) \neq \emptyset$ then $s_2 = a$, for some constant a. In this case we have $z\sigma = a$ and $y_i\sigma = s_i'$, where each s_i' is a subterm of s_1. So $u_2\sigma = u_2$.

- If $V(s_2) \neq \emptyset$ then s_1 is ground. We have then that s_1 is a ground term, $z\sigma = z$, $x\sigma = v'(z)$, where v' is a subterm of v, and for all i, $y_i\sigma$ is some ground subterm of s_1. We note by u' the term $u[g(y_1, \ldots, y_k)]$. We have the following relations:

  $$\max(|u_1|, |u_2|) = 1 + \max(|s_1|, |s_2|, |u'|, |v|) = = 1 + \max(|s_1|, |v|),$$

  since $u'\sigma = s_1\sigma = s_1$ and $s_2\sigma = v\sigma = v$. We obtain the desired equality by observing that $|u_1\sigma| = 1 + \max(|s_1|, |s_2\sigma|)$ and $|u_2\sigma| = 1 + \max(|u'\sigma|, |v|)$.

Thus we have obtained that $|w\sigma| \leq \max(|w|, |u_1|, |u_2|)$. The other cases that we have to consider (as in the proof of Lemma 4) are easy and left to the reader. □

6. CONCLUSIONS

We have obtained new decidability results for the secrecy of cryptographic protocols that employ encryption primitives satisfying properties that could not be treated by previous decision procedures. The results followed from the termination of a resolution strategy on a class of Horn clauses. This resolution strategy might be useful for larger classes of protocols and more encryption properties. Indeed, while termination is no more ensured for larger classes, completeness is still guaranteed.

We have applied our technique to the debugging of a protocol under a more realistic threat model than the one usually considered. We have transformed this protocol so that it falls into the scope of our Horn class. This transformation preserves the attacks and therefore the correctness of the target protocol ensures the correctness of the initial one. The transformation is interesting by itself. It would be interesting to study further this type of transformations and to characterize the protocols to which they can be safely applied.

7. REFERENCES

[1] L. Bachmair and H. Ganzinger. Resolution theorem proving. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 19–99. Elsevier and MIT Press, 2001.

[2] D. Basin, S. Mödersheim, and L. Viganò. An on-the-fly model-checker for security protocol analysis. In E. Snekkenes and D. Gollmann, editors, *Proc. of ESORICS'03*, volume 2808 of *Lecture Notes on Computer Science*, pages 253–270. Springer-Verlag, 2003.

[3] J. C. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC*, pages 544–553, 1994.

[4] B. Blanchet. Abstracting Cryptographic Protocols by Prolog Rules (invited talk). In P. Cousot, editor, *8th International Static Analysis Symposium (SAS'2001)*, volume 2126 of *Lecture Notes on Computer Science*, pages 433–436, Paris (France), July 2001. Springer-Verlag.

[5] X. Chen, B. Lee, and K. Kim. Receipt-free electronic auction schemes using homomorphic encryption. In J. I. Lim and D. H. Lee, editors, *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 259–273, Seoul (Korea), November 2003. Springer-Verlag.

[6] Y. Chevalier, R. Kuesters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In *Proc. of the Logic In Computer Science Conference LICS'03*, June 2003.

[7] Y. Chevalier and L. Vigneron. Automated unbounded verification of security protocols. In E. Brinksma and K. Guldstrand Larsen, editors, *14th International Conference on Computer Aided Verification, CAV'2002*, volume 2404 of *Lecture Notes in Computer Science*, pages 324–337, Copenhagen (Denmark), July 2002. Springer-Verlag.

[8] 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 *Lecture Notes in Computer Science*, pages 148–164, Valencia (Spain), June 2003. Springer-Verlag.

[9] H. Comon-Lundh and V. Cortier. Security properties: two agents are sufficient. In *Proc. of the 12th European Symposium On Programming (ESOP'03)*, volume 2618 of *Lecture Notes in Computer Science*, pages 99–113, Warsaw (Poland), April 2003. Springer-Verlag.

[10] V. Cortier. *Vérification automatique des protocoles cryptographiques.* PhD thesis, École Normale Supérieure de Cachan, Cachan (France), March 2003.

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

[12] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikinen. On secure scalar product computation for privacy-preserving data mining. In C. Park and S. Chee, editors, *The 7th Annual International Conference in Information Security and Cryptology (ICISC 2004)*, Lecture Notes in Computer Science, Seoul (Korea), December 2004. Springer-Verlag.

[13] S. Kremer and M. Ryan. Analysis of an Electronic Voting Protocol in the Applied Pi-Calculus. In M. Sagiv, editor, *Proceedings of the 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 186–200, Edinburgh, U.K., April 2005. Springer-Verlag.

[14] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for *AC*-like equational theories with homomorphisms. In *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322, Nara (Japan), April 2005. Springer-Verlag.

[15] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In Margaria and Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055 of *Lecture Notes on Computer Science*, pages 147–166, 1996.

[16] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communication of the ACM*, 21(12):993–999, 1978.

[17] O. Pereira and J.-J. Quisquater. On the perfect encryption assumption. In *Proc. of the 1st Workshop on Issues in the Theory of Security (WITS'00)*, pages 42–45, Geneva (Switzerland), 2000.

[18] H. Seidl and K. N. Verma. Flat and one-variable clauses: Complexity of verifying cryptographic protocols with single blind copying. In *Proc. of 11th International Conference on Logic for Programming and Automated Reasoning (LPAR'04)*, volume 3452 of *Lecture Notes in Computer Science*, pages 79–94, Montevideo (Uruguay), 2005. Springer-Verlag.