A Formal Definition of Protocol Indistinguishability and its Verification Using Maude-NPA [⋆]

Sonia Santiago[1], Santiago Escobar[1], Catherine Meadows[2], and José Meseguer[3]

[1] DSIC-ELP, Universitat Politècnica de València, Spain
ssantiago@dsic.upv.es,sescobar@dsic.upv.es
[2] Naval Research Laboratory, Washington DC, USA
meadows@itd.nrl.navy.mil
[3] University of Illinois at Urbana-Champaign, USA
meseguer@illinois.edu

Abstract. Intuitively, two protocols $\mathcal{P}_1$ and $\mathcal{P}_2$ are *indistinguishable* if an attacker cannot tell the difference between interactions with $\mathcal{P}_1$ and with $\mathcal{P}_2$. In this paper we: (i) propose an intuitive notion of indistinguishability in Maude-NPA; (ii) formalize such a notion in terms of state unreachability conditions on their *synchronous product*; (iii) prove theorems showing how —assuming the protocol's algebraic theory has a *finite variant (FV) decomposition*– these conditions can be checked by the Maude-NPA tool; and (iv) illustrate our approach with concrete examples. This provides for the first time a framework for automatic analysis of indistinguishability *modulo* as wide a class of algebraic properties as FV, which includes many associative-commutative theories of interest to cryptographic protocol analysis.

1 Introduction

The security of cryptographic systems has traditionally been reasoned about using two different types of models. The first is the *computational model*, in which the cryptographic system is attacked by a probabilistic polynomial-time adversary. In this model one normally proves some sort of *indistinguishability property* which guarantees, e.g., that an attacker who interacts with two different instances of a protocol involving different data should not be able to tell the difference. The second is commonly known as the *Dolev-Yao model* in which cryptographic operations are modeled as abstract function symbols. This lends itself well to model checking, so it is normally used to verify conditions that can be formulated as state reachability properties, e.g., that the attacker should not be able to obtain a secret in the clear (known as *simple secrecy* in the literature).

⋆ Santiago Escobar and Sonia Santiago have been partially supported by the EU (FEDER) and the Spanish MINECO under grants TIN 2010-21062-C02-02 and TIN 2013-45732-C4-1-P, and by Generalitat Valenciana PROMETEO2011/052. José Meseguer has been partially supported by NSF Grant CNS 13-10109.

Recently the interest in formulating and applying indistinguishability properties for Dolev-Yao models has been growing. There are a number of reasons for this. The first is that cryptography has advanced to the point at which it is not only possible to provide computational proofs of security for algorithms, but also for the protocols that use those algorithms as well. If Dolev-Yao tools can be extended to prove indistinguishability, this increases the likelihood that both approaches can be used together in an effective way to ensure protocol security. The second is that there is a growing class of privacy-protection protocols for which simple secrecy is clearly inadequate. Such protocols protect low-entropy data such as votes, medical records, or network routes; even partial leakage of this information could be harmful. The third is the result of recent work on automatic generation of cryptographic algorithms. In this work, multiple possible algorithms are generated out of a library of components and then checked for security. This may involve the use of Dolev-Yao like tools to weed out insecure algorithms or even verify the security of correct ones, as in [3].

When a Dolev-Yao tool is used to check for subtle properties such as indistinguishability, it is important that it offers as detailed a picture of the properties of the cryptographic operations as possible. This is done by including information about their algebraic properties, that is, the equations obeyed by the function symbols. For example, if a cryptographic system uses exclusive-or, one should be able to take into account the associative-commutative, identity, and self-cancellation properties of exclusive-or. Such algebraic properties have been studied extensively in the literature, although there are still some classes of properties that are not that well understood.

At this point, there are four main problems being explored in relation to Dolev-Yao indistinguishability. The first is how best to formulate in the Dolev-Yao model a property such as indistinguishability that in its broadest sense is *not* a reachability property. The second is how to incorporate equational theories in this reasoning. The third is how to increase the range and complexity of the types of protocols we can reason about. The fourth is when and how to ensure that Dolev-Yao indistinguishability implies computational indistinguishability. In this paper we address the first two of these problems, although we note that the second is closely related to the fourth, and can be used to facilitate its solution. We summarize our contributions below:

Formulating Indistinguishability in Dolev-Yao. We propose an intuitive notion of indistinguishability related to the notion of *uniformity* used in Proverif [6] in that it is defined, not in terms of equivalence between two protocols, but of equivalence between *roles* of two protocols. In this case roles from the two protocol versions $\mathcal{P}_1$ and $\mathcal{P}_2$ are paired together and executed in a synchronous fashion (called a *synchronous product* in our case). We then define our notion as the conjunction of two more basic properties, namely *Indistinguishable Messages* (IM) and *Indistinguishable Attacker Event Sequences* (IAES). Intuitively, the IM property says that the attacker, when performing the same actions for $\mathcal{P}_1$ and for $\mathcal{P}_2$, can never reach two corresponding stages in such action sequences such that it can learn the *same* message from $\mathcal{P}_1$ at both stages, but *different* messages from $\mathcal{P}_2$ at those same stages, or viceversa. The IAES property says that the attacker *can perform the same interaction steps* with $\mathcal{P}_1$ and $\mathcal{P}_2$, which requires a bisimulation between the two protocols. We prove a result with respect to the semantics of the Maude-NPA protocol analysis tool [13], showing that the conjunction of IM and IAES can be formulated in terms of reachability properties in Maude-NPA.

Incorporating Equational Theories. Our approach extends naturally to any algebraic theory E that can be decomposed as $E = E_0 \cup B$, with the equations E_0 oriented as rewrite rules modulo B, and the decomposition (B, E_0) satisfying the *finite variant* (FV) property. In this case we prove theorems showing that the IM and IAES properties can be checked by the Maude-NPA tool. The class of theories with finite variant decompositions contains a large number of theories of interest to cryptographic protocol analysis, including exclusive-or, Abelian groups, and a number of theories describing the properties of modular exponentiation. Thus previous work on analysis of protocols modulo finite variant decompositions is *naturally* extended to the verification of indistinguishability under many possible equational theories.

Finally, we illustrate the IM and IAES properties with two examples.

Example 1. Consider two protocols $\mathcal{P}_1$ and $\mathcal{P}_2$ using the exclusive-or (XOR) operator "$\oplus$". Below we give the exchange of messages for each protocol:

$$(\mathcal{P}_1)\ A \to B : m_1 \oplus m_1 \qquad\qquad (\mathcal{P}_2)\ A \to B : m_1 \oplus m_2$$

where m_1 and m_2 are two constants denoting different messages. Since the attacker can perform the XOR cancellation (i.e. $M \oplus M = 0$), and can generate the XOR unit element 0, then it can distinguish $\mathcal{P}_1$ and $\mathcal{P}_2$ by performing the following actions:

$$(\mathcal{P}_1)\ 1.\ A \to I(B) : m_1 \oplus m_1\ (=_{XOR}\ 0) \qquad (\mathcal{P}_2)\ 1.\ A \to I(B) : m_1 \oplus m_2$$

Thus these protocols do not satisfy the IM property, since in $\mathcal{P}_1$ the attacker generates 0 from two different action sequences, whereas in $\mathcal{P}_2$ it does not.

Example 2. Consider a protocol similar to the first step of the Encryption Key Exchange (EKE) protocol [5] in which, unlike the original EKE, the attacker can distinguish whether a decryption succeeds or not. The algebraic properties of this protocol consist of the cancellation of encryption and decryption. In this protocol, Alice sends to Bob her name (A) concatenated with the encryption of her public key *pkey(A)* with a password *pw(A,B)* they have agreed on before.

$$A \to B : A\,;\ \{pkey(A)\}_{pw(A,B)}$$

Consider two cases in which the honest principals perform the same step shown above, but in $\mathcal{P}_1$ the attacker knows the right password *pw(A,B)*, whereas in $\mathcal{P}_2$ it knows a random password *pg(i)*. The intruder can distinguish between $\mathcal{P}_1$ and $\mathcal{P}_2$ by performing the following actions, where steps 2, 3, and 4 denote deductions performed by the intruder:

$(\mathcal{P}_1)$ 1. $A \to I(B) : A\,;\ \{pkey(A)\}_{pw(A,B)}$	$(\mathcal{P}_2)$ 1. $A \to I(B) : A\,;\ \{pkey(A)\}_{pw(A,B)}$
$(\mathcal{P}_1)$ 2. $I : \{pkey(A)\}_{pw(A,B)}$	$(\mathcal{P}_2)$ 2. $I : \{pkey(A)\}_{pw(A,B)}$
$(\mathcal{P}_1)$ 3. $I : pw(A, B)$	$(\mathcal{P}_2)$ 3. $I : pg(i)$
$(\mathcal{P}_1)$ 4. $I :$ *decryption succeeds*	$(\mathcal{P}_2)$ 4. $I :$ *decryption fails*

That is, in step 4 in $\mathcal{P}_1$ the attacker can obtain the message *pkey(A)* in the clear, by decrypting the message sent in step 2 *pw(A,B)*, whereas in $\mathcal{P}_2$ such decryption fails. Hence the protocols are not bisimilar and so fail to satisfy IAES.

Paper Organization. We first discuss related work in Section 2 and recall some technical preliminaries in Section 3. In Section 4 we recall the forwards semantics of Maude-NPA defined in [14]. In Section 5 we present our formal definition of indistinguishability using the fowards semantics given in Section 4 as a reference model and prove that it can be analyzed in Maude-NPA assuming that the equational theory used for $\mathcal{P}_1$ and $\mathcal{P}_2$ has a finite variant decomposition. Section 6 explains how to analyze indistinguishability properties in Maude-NPA. We summarize our conclusions in Section 7.

2 Related Work

Work in extending the Dolev-Yao model to support the definition and verification of indistinguishability properties goes as far back as the early eighties, when Michael Merritt developed a theory of *hidden automorphisms* [19]. The first to apply a tool to analyze protocols for indistinguishability was Gavin Lowe [18], who used the FDR model checker to analyze security of password-base protocols against off-line guessing attacks.

Abadi and Fournet gave in [1] the definition of two kinds of indistinguishability: static equivalence and observational equivalence, in terms of the applied π-calculus presented in that paper. Roughly speaking, static equivalence describes a passive observer's inability to distinguish between two protocols, while observational equivalence describes an active attacker's inability to distinguish between two protocols.

More recently Cortier and Delaune [12] have shown that *trace equivalence* implies observational equivalence for *determinate* applied π-calculus processes. Roughly speaking, a process is determinate if it exhibits no non-deterministic choice points, and two processes are trace equivalent if for any trace produced by one process there is a trace produced by the other process indistinguishable from the first trace by the attacker. In our approach, the IM property assumes the existence of traces in both sides of a protocol pairing and simply compares the messages. Thus, IM is closer to static equivalence than trace equivalence, where you need to prove the existence of such traces. On the other hand, the IAES property ensures the existence of traces with the same behavior. Indeed, IM does not imply IAES, since the latter is tested after the former is proved.

Trace equivalence is decidable in the bounded session model, and a number of algorithms and tools have been developed, covering a wide class of equational theories [4, 8, 9, 7, 10]. However to our knowledge there has been very little work (if any) on trace equivalence involving AC theories.

Although trace equivalence is decidable for the bounded session model, it is not straightforward to implement in search-based tools that are typically used to evaluate cryptographic protocols. This is because trace equivalence is an example of a *hyperproperty* [11] : it is not defined in terms of sets of traces, but sets of pairs of traces, and thus cannot be defined in terms of reachability or unreachability of particular classes of states. However, integration of indistinguishability into state exploration tools has a number of potential benefits, since one automatically obtains support for whatever feature the tool offers, e. g., support for the unbounded session model and equational theories involving AC.

Checking for hyperproperties such as trace or observational equivalence can be implemented in a search-based tool by specifying a stronger property that can be formulated in terms of state reachability. Such an approach was taken by Lowe in [18]; a protocol was secure against guessing attacks if the attacker could not generate certain types of terms. This was later shown in [22] to imply a property similar to the observational equivalence of [1].

The most prominent application of this approach to cryptographic protocol verification has been in the Proverif tool via the notion of *uniformity*, shown to imply observational equivalence in [6]. It is used to define the indistinguishability of two processes that differ only in certain terms. Uniformity requires that the two processes be executed in lock-step as a *bi-process* and projection of the bi-process to each of its components is a bisimulation. The authors prove that uniformity is equivalent to a state unreachability property, and thus can be evaluated using ProVerif. ProVerif can be used to verify uniformity for *subterm convergent* rewrite theories; Arapinis *et al.* [2] have developed methods for extending this to some theories that include a restricted encoding of AC axioms (in particular, some terms must be ground).

The approach we use in our paper is closest to that of ProVerif; we define a pairing between two protocols and define security in terms of reachability conditions on the protocol pairing. One major difference is in the support of AC theories without any encoding restrictions, as long as they have decompositions with the finite variant property. This is inherited from Maude-NPA. There are also differences in the approach we take to specification and implementation of security properties. In ProVerif, the intruder is given the ability to evaluate a predicate that outputs "bad" if there is a violation of uniformity. One then checks for uniformity by proving that no state containing "bad" is reachable. This gives Proverif the ability to reduce everything to just one property.

In our approach we use an unmodified Dolev-Yao intruder with no ability to evaluate predicates. This is motivated by our preference to avoid increasing the complexity of Maude-NPA's Dolev-Yao model unless absolutely necessary, and thus to express our security requirements in the original Maude-NPA framework. In particular, the IM property implicitly includes a test on equality, but it is expressed as a property of the attack state, not as an intruder predicate. We are however, considering developing more general intruder predicate testing functionality as future work, and if we decide to do this we will revisit this issue.

3 Preliminaries

We follow the classical notation and terminology from [23] for term rewriting and from [20, 21] for rewriting logic and order-sorted notions. We assume an *order-sorted signature* $\mathbf{\Sigma} = (\mathsf{S}, \leq, \Sigma)$ with partially ordered set of sorts $(\mathsf{S}, \leq)$. We also assume an S-sorted family $\mathcal{X} = \{\mathcal{X}_\mathsf{s}\}_{\mathsf{s} \in \mathsf{S}}$ of disjoint variable sets with each $\mathcal{X}_\mathsf{s}$ countably infinite. $\mathcal{T}_\Sigma(\mathcal{X})_\mathsf{s}$ is the set of terms of sort s, and $\mathcal{T}_{\Sigma,\mathsf{s}}$ is the set of ground terms of sort s. We write $\mathcal{T}_\Sigma(\mathcal{X})$ and $\mathcal{T}_\Sigma$ for the corresponding order-sorted term algebras. The subterm of t at position p is $t|_p$, and $t[u]_p$ is the result of replacing $t|_p$ by u in t. Application of substitution σ to a term t is denoted $t\sigma$.

A *Σ-equation* is an unoriented pair $t = t'$, where $t, t' \in \mathcal{T}_\Sigma(\mathcal{X})$ have a common typing $\mathsf{t} : \mathsf{s}$, $\mathsf{t}' : \mathsf{s}$, $\mathsf{s} \in \mathsf{S}$. Given a set E of Σ-equations, order-sorted equational logic induces a congruence relation $=_E$ on terms $t, t' \in \mathcal{T}_\Sigma(\mathcal{X})$; see [21]. We write $CSU_E(t = t')$ for a complete set of unifiers of t and t' modulo E.

A *rewrite rule* is an oriented pair $l \to r$, where $l \notin \mathcal{X}$, $Var(r) \subseteq Var(l)$, and $l, r \in \mathcal{T}_\Sigma(\mathcal{X})$ have a common typing $\mathsf{l} : \mathsf{s}$, $\mathsf{r} : \mathsf{s}$, $\mathsf{s} \in \mathsf{S}$. An *(unconditional) order-sorted rewrite theory* is a triple (Σ, E, R) with Σ an order-sorted signature, E a set of Σ-equations, and R a set of rewrite rules. The relation $\to_{R,E}$ on $\mathcal{T}_\Sigma(\mathcal{X})$ is defined as: $t \xrightarrow{p}_{R,E} t'$ (or $\to_{R,E}$) if p is a position of t, $l \to r \in R$, $t|_p =_E l\sigma$, and $t' = t[r\sigma]_p$ for some σ. A term u is in $\to_{R,E}$-normal form if there is no v such that $u \to_{R,E} v$.

Given an (unconditional) order-sorted rewrite theory (Σ, E, R) such that E has a finitary and complete unification algorithm, the narrowing relation $\rightsquigarrow_{R,E}$ on $\mathcal{T}_\Sigma(\mathcal{X})$ is defined as $t \overset{p}{\rightsquigarrow}_{\sigma,R,E} t'$ if p is a non-variable position of t, $l \to r \in R$, $\sigma \in CSU_E(t|_p = l)$, and $t' = (t[r]_p)\sigma$.

A *decomposition* (Σ, B, E_0) of an equational theory E is a rewrite theory that satisfies the following properties: (i) B is regular, sort-preserving and uses top-sort variables, (ii) B has a finitary unification algorithm, and (iii) the rules E_0 are *convergent* modulo B, i.e., sort-decreasing, confluent, terminating, and coherent modulo B. Given a decomposition $E = (\Sigma, B, E_0)$, the normal form of a term t is denoted by $t{\downarrow_{E_0,B}}$. Given a decomposition $E = (\Sigma, B, E_0)$, a variant of a term t is a pair (t', θ) such that $t' =_B (t\theta){\downarrow_{E_0,B}}$. A decomposition (Σ, B, E_0) has the *finite variant (FV) property* if there is a complete and finite set of variants for each term (see [15] for details). If a decomposition (Σ, B, E_0) of an equational theory E has the *finite variant property*, there is an algorithm to compute a finite complete set $CSU_E(t = t')$ of E-unifiers [15].

A transition system is written $\mathcal{A} = (A, \to)$, where A is a set of states, and $\to$ is a transition relation between states, i.e., $\to \subseteq A \times A$. A rewrite theory $\mathcal{R} = (\Sigma, B, R)$ specifies a transition system $\mathcal{T}_\mathcal{R}$ whose states are elements of the initial algebra $\mathcal{T}_{\Sigma/B}$, and whose transitions are specified by the set of rewrite rules R. Given two transition systems $\mathcal{A} = (A, \to_\mathcal{A})$ and $\mathcal{B} = (B, \to_\mathcal{B})$, a *simulation* from $\mathcal{A}$ to $\mathcal{B}$, written $\mathcal{A}\, H\, \mathcal{B}$, is a relation $H \subseteq A \times B$ such that $a\, H\, b$ and $a \to_\mathcal{A} a'$ implies that there exists $b' \in B$ such that $a'\, H\, b'$ and $b \to_\mathcal{B} b'$. A simulation H from $(A, \to_\mathcal{A})$ to $(B, \to_\mathcal{B})$ is a *bisimulation* if H^{-1} is a simulation from $(B, \to_\mathcal{B})$ to $(A, \to_\mathcal{A})$.

4 Forwards Semantics for Maude-NPA

Maude-NPA [13] performs a *backwards narrowing-based reachability analysis*. However, our notion of indistinguishability is more naturally defined in Maude-NPA in terms of a *rewriting-based forwards analysis*. In this section, we briefly recall the forwards semantics and refer the reader to [14] for further information.

4.1 Maude-NPA's Strand Space Model

In this section we give an overview of Maude-NPA's use of the strand space model to specify protocols and states. This model, with some minor differences, is the same as that used in the backwards semantics.

Given a protocol $\mathcal{P}$, states are modeled as elements of an initial algebra $\mathcal{T}_{\Sigma_{\mathcal{P}}/E_{\mathcal{P}}}$, where $\Sigma_{\mathcal{P}}$ is the signature defining the sorts and function symbols (for the cryptographic functions and for all the state constructor symbols) and $E_{\mathcal{P}}$ is a set of equations specifying the *algebraic properties* of the cryptographic functions and the state constructors. Therefore, a state is an $E_{\mathcal{P}}$-equivalence class $[t] \in \mathcal{T}_{\Sigma_{\mathcal{P}}/E_{\mathcal{P}}}$ with t a ground $\Sigma_{\mathcal{P}}$-term.

In Maude-NPA, a state consists of a multiset of partially executed *strands* S_i and a set of terms in the intruder's knowledge, i.e. a state is a term of the form $\{S_1 \,\&\, \cdots \,\&\, S_n \,\&\, \{IK\}\}$ where & is an associative-commutative union operator.

The *intruder knowledge* $\{IK\}$ is represented as a set of facts using the comma as an associative-commutative union operator with identity operator *empty*. In the forwards semantics knowledge facts are all of the from $m{\in}\mathcal{I}$ (the intruder knows m) where m is a message expression.

A *strand* [16] specifies the sequence of messages sent and received by a principal executing the protocol and is represented as a sequence of {+,-}–labeled messages $[msg_1^{\pm}, \ldots, msg_{k-1}^{\pm}, msg_k^{\pm}]$ such that msg_i^- (also written $-msg_i$) represents an *input* message, msg_i^+ (also written $+msg_i$) represents an *output* message, and each msg_i is a term of a special sort Msg. Strands are used to represent both the actions of honest principals (with a strand specified for each protocol role) and the actions of an intruder (with a strand for each operation an intruder is able to perform on terms).

In the following, a protocol specification is defined as a collection of honest and intruder strand *patterns* together with the algebraic theory of the protocol.

Definition 1 (Protocol Specification). *Given a protocol $\mathcal{P}$, its protocol specification is a triple $((\Sigma_{\mathcal{P}}, E_{\mathcal{P}}), HPS_{\mathcal{P}}, IS_{\mathcal{P}})$ where $(\Sigma_{\mathcal{P}}, E_{\mathcal{P}})$ is the algebraic theory of $\mathcal{P}$, and $HPS_{\mathcal{P}}$ and $IS_{\mathcal{P}}$ denote the set of honest and intruder strands of $\mathcal{P}$, respectively.*

4.2 Forwards Semantics

In a forward reachability analysis, state changes are defined by means of a set $R_{F\mathcal{P}}$ of *rewrite rules*, so that the rewrite theory $(\Sigma_{\mathcal{P}}, E_{\mathcal{P}}, R_{F\mathcal{P}})$ characterizes the behavior of protocol $\mathcal{P}$ modulo the equations $E_{\mathcal{P}}$ [4] . These rewrite rules are generated from principal and intruder strands as explained below.

The forwards execution of a protocol begins with an "empty" initial state, that is a state with the empty set of strands and intruder knowledge. The protocol is executed to determine whether or not an *attack state* can be reached, where an attack state is a ground instance of a state pattern defined by the user. One progresses by applying *rewrite rules* to states.

For each different kind of principal and intruder strand a set of rewrite rules is associated as follows. Given a strand $[m_1{}^{\pm}, \ldots, m_n{}^{\pm}]$, n different rewrite rules are applied to incrementallly "build" the strand step by step. See [14] for more details and a formal definition of the forwards semantics.

For example, the intruder encryption capability denoted by the strand $[-(K), -(M), +(e(K, M)]$ has the following associated rewrite rules:

$$\{SS \,\&\, \{K{\in}\mathcal{I}, IK\}\,\} \to \{SS \,\&\, [-(K)] \,\&\, \{K{\in}\mathcal{I}, IK\}\,\}$$
$$\{SS \,\&\, [-(K)] \,\&\, \{M{\in}\mathcal{I}, IK\}\,\} \to \{SS \,\&\, [-(K), -(M)] \,\&\, \{M{\in}\mathcal{I}, IK\}\,\}$$
$$\{SS \,\&\, [-(K), -(M)] \,\&\, \{IK\}\,\} \to \{SS \,\&\, [-(K), -(M), +(e(K, M)]\,\{e(K, M){\in}\mathcal{I}, IK\}\,\}$$

where SS denotes a set of strands and IK a set of intruder knowledge facts.

The forwards execution of a protocol induces a transition system as follows.

Definition 2 (Transition System induced by a Protocol). *Given a protocol $\mathcal{P}$ characterized by the forward rewrite theory $(\Sigma_{\mathcal{P}}, E_{\mathcal{P}}, R_{F\mathcal{P}})$ such that $(\Sigma_{\mathcal{P}}, B, E_0)$ is a decomposition of $(\Sigma, E_{\mathcal{P}})$, we can associate to it a transition system $\mathcal{L}_{\mathcal{P}}$ whose states are B-equivalence classes of terms in E_0, B-canonical form and whose transitions are of the form:*

$$[t]_B \to [t']_B$$

where $t \to_{R_{F\mathcal{P}},B} u$ and $t' =_B u{\downarrow_{E_0,B}}$.

Unlike the case with process calculi, no information is removed from a state and the history of previous actions can be recovered from a state. Therefore there is no need to record this information through labels in order to obtain a labeled transition system. However, labels can be added if desired (e.g. as a compact way of encoding essential information).

5 Formalization of Indistinguishability in Maude-NPA

Intuitively, two protocols are indistinguishable if an intruder cannot tell the difference between them. In this section we provide the framework that will allow the definition and verification of indistinguishability in Maude-NPA. To define such a framework, in Section 5.1 we first formalize, by the concept of a *protocol pairing*, the notion of pairs of protocols that are similar enough to

[4] The rules $R_{F\mathcal{P}}$ model *protocol transitions*. In addition, *equations* $E_{\mathcal{P}}$ are assumed to have a *decomposition* $(\Sigma_{\mathcal{P}}, B, E_0)$ with $E_{\mathcal{P}} = B \cup E_0$ which has the FV property, where E_0 are *not* transitions, but *oriented equations*.

each other so that the issue of their indistinguishability can arise. Then, in Section 5.2 we formalize the idea of *similar interactions* of the attacker with a protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ by the concept of the *synchronous product* $\mathcal{P}_1 \otimes \mathcal{P}_2$. Finally, in Section 5.3 we define the indistinguishability of a protocol pairing in Maude-NPA as the conjunction of two simpler properties called IAES and IM.

5.1 Protocol Pairing

The notion of indistinguishability implies comparing two protocols $\mathcal{P}_1$ and $\mathcal{P}_2$ to ensure that an intruder *cannot distinguish* the behaviors of $\mathcal{P}_1$ and $\mathcal{P}_2$. In practical applications $\mathcal{P}_1$ and $\mathcal{P}_2$ are somewhat different *versions* of a given protocol with some significant *differences*. In this section, we formalize, by the concept of a *protocol pairing*, the notion of such pair of protocols in Maude-NPA.

Definition 3 (Protocol Pairing). *A* protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ *is a pair of protocol specifications of the form* $((\Sigma_{\mathcal{P}_1}, E_{0_{\mathcal{P}_1}} \cup B_{\mathcal{P}_1}), \mathit{HPS}_{\mathcal{P}_1}, \mathit{IS}_{\mathcal{P}_1}), ((\Sigma_{\mathcal{P}_2}, E_{0_{\mathcal{P}_2}} \cup B_{\mathcal{P}_2}), \mathit{HPS}_{\mathcal{P}_2}, \mathit{IS}_{\mathcal{P}_2})$ *such that:*

1. *$\mathcal{P}_1$ and $\mathcal{P}_2$ share the same algebraic signature and equations, i.e.* $(\Sigma_{\mathcal{P}_1}, E_{0_{\mathcal{P}_1}} \cup B_{\mathcal{P}_1}) = (\Sigma_{\mathcal{P}_2}, E_{0_{\mathcal{P}_2}} \cup B_{\mathcal{P}_2}) = (\Sigma, E_0 \cup B)$ *having a decomposition* (Σ, B, E_0).
2. *$\mathit{HPS}_{\mathcal{P}_1}$ and $\mathit{HPS}_{\mathcal{P}_2}$ have strands for the same roles, with the same length and the same polarities ($+$ or $-$) at each position in the strand.*
3. *$\mathit{IS}_{\mathcal{P}_1}$ and $\mathit{IS}_{\mathcal{P}_2}$ have strands for the same operations, with the same length and the same polarities ($+$ or $-$) at each position in the strand.*

We assume both $\mathit{HPS}_{\mathcal{P}_1}$ and $\mathit{HPS}_{\mathcal{P}_2}$, and $\mathit{IS}_{\mathcal{P}_1}$ and $\mathit{IS}_{\mathcal{P}_2}$ have disjoint variables.

We will also require that strands in certain pairs both be identical up to change of variables, depending on the indistinguishability model used. For example, in the standard model based on cryptographic definitions of indistinguishability the same attacker interacts with two different versions of the protocol, so any two paired intruder strands must be identical up to change of variables. In Lowe's password guessing model the same attacker with different password guesses interacts with the same version of the protocol, so we require that any two paired strands be identical up to change of variables, except that describing the attacker's guess of the password.

The differences between $\mathcal{P}_1$ and $\mathcal{P}_2$ can be specified by having different messages in the same j-th strands of $\mathcal{P}_1$ and $\mathcal{P}_2$, at the same positions in the strands. Below we illustrate the notion of protocol pairing using the following example.

Example 3. Let us consider the two protocols $\mathcal{P}_1$ and $\mathcal{P}_2$ of Example 1 shown above. Note that both $\mathcal{P}_1$ and $\mathcal{P}_2$ share the same algebraic signature and equations (XOR), and have the same set of intruder strands, and differ only in the set of honest principal strands, as explained below. More specifically, the sets $\mathit{HPS}_{\mathcal{P}_1}$ and $\mathit{HPS}_{\mathcal{P}_2}$ of honest strands of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, are as follows:

$$\mathit{HPS}_{\mathcal{P}_1} = \{\, [\ (m_1 \oplus m_1)^+\ ]\} \qquad \mathit{HPS}_{\mathcal{P}_2} = \{\, [\ (m_1 \oplus m_2)^+\ ]\}$$

where m_1 and m_2 are two constants denoting different messages, and $\oplus$ is the exclusive-or operator. Therefore, $\mathcal{P}_1, \mathcal{P}_2$ is a protocol pairing.

5.2 Synchronous Product of Protocols

Given a protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ as explained above, the analysis of its indistinguishability assumes that the attacker interacts in an analogous way with both protocols at each step. That is, if it performs an action a in $\mathcal{P}_1$, then it does so in $\mathcal{P}_2$ too. In this section we formalize the idea of *similar interactions* of the attacker with a protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ by the concept of the *synchronous product* $\mathcal{P}_1 \otimes \mathcal{P}_2$. Intuitively, a synchronous product is a new protocol obtained from a protocol pairing in which both protocols from the pairing are executed in a synchronous manner.

In order to provide a formal definition of a synchronous product of protocols, we first define the synchronous product of strands.

Definition 4 (Synchronous Product of Strands). *Given two strands* $Str_1 = [{m_1}^{\pm}, \ldots, {m_n}^{\pm}]$, *and* $Str_2 = [{m'_1}^{\pm}, \ldots, {m'_n}^{\pm}]$ *corresponding to the same protocol role or intruder action, and with the same polarities at each position in the strand, the synchronous product of* Str_1 *and* Str_2*, written* $Str_1 \otimes Str_2$*, is a strand of the form* $[(m_1 \otimes m'_1)^{\pm}, \ldots, (m_n \otimes m'_n)^{\pm}]$*, with* $\otimes$ *a new operator.*

Let SS_1 *and* SS_2 *be two sets of strands that have* n *strands corresponding to the same protocol roles or intruder actions. The synchronous product of* SS_1 *and* SS_2*, written* $SS_1 \otimes SS_2$*, is a set of strands of the form* $\{Str_{1_i} \otimes Str_{2_j}\}_{0 \leq i,j \leq n}$*, such that* $Str_{1_i} \in SS_1$*,* $Str_{2_j} \in SS_2$*, and* Str_{1_i} *and* Str_{2_j} *correspond to the same protocol role or intruder action.*

Let us now define the synchronous product of protocols as follows.

Definition 5 (Synchronous Product of Protocols). *Given a protocol pairing* $\mathcal{P}_1, \mathcal{P}_2$*, its* synchronous product*, denoted by* $\mathcal{P}_1 \otimes \mathcal{P}_2$*, is a new protocol as explained below. Basically, the signature is extended with new sorts and symbols to support the specification of a pair of protocols.*

- *First, the theory decomposition* (Σ, B, E_0) *shared by* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *is renamed to* $(\widehat{\Sigma}, \widehat{B}, \widehat{E_0})$*, just by a renaming* $\mathsf{s} \mapsto \widehat{\mathsf{s}}$ *(where* $\mathsf{s} \in \mathsf{S}$ *and* $\widehat{\mathsf{s}} \in \widehat{\mathsf{S}}$*, of the partially ordered set of sorts so that:* $\widehat{\mathsf{Msg}} = \mathsf{SingleMsg}$*, and* $\widehat{\mathsf{s}} = \mathsf{s}$ *otherwise, and with* $\mathsf{s} < \mathsf{s}'$ *iff* $\widehat{\mathsf{s}} < \widehat{\mathsf{s}'}$*. The operators* $\widehat{\Sigma}$ *are renamed accordingly, so that* $f : \mathsf{s_1} \cdots \mathsf{s_n} \to \mathsf{s}$ *is renamed to* $f : \widehat{\mathsf{s_1}} \cdots \widehat{\mathsf{s_n}} \to \widehat{\mathsf{s}}$*, and the equations* B *and* E_0 *are renamed to* $\widehat{B}$ *and* $\widehat{E_0}$ *just by renaming the sorts of their variables by the mapping* $\mathsf{s} \mapsto \widehat{\mathsf{s}}$*.*
- *A new sort* Msg *is added as the new top sort of the connected component for messages, so that* $\mathsf{SingleMsg} < \mathsf{Msg}$*.*
- *A new operator* $\_ \otimes \_ : \mathsf{SingleMsg}\ \mathsf{SingleMsg} \to \mathsf{Msg}$ *is added to* $\Sigma_{\otimes}$*.*
- *Its protocol specification is the triple* $\mathcal{P}_1 \otimes \mathcal{P}_2 = ((\Sigma_{\otimes}, \widehat{E_0} \cup \widehat{B}), HPS_{\mathcal{P}_1} \otimes HPS_{\mathcal{P}_2}, IS_{\mathcal{P}_1} \otimes IS_{\mathcal{P}_2})$*, where* $HPS_{\mathcal{P}_1}$*,* $HPS_{\mathcal{P}_2}$*,* $IS_{\mathcal{P}_1}$*, and* $IS_{\mathcal{P}_2}$ *are renamed to have disjoint variables.*

Therefore, if (Σ, B, E_0) *is the original theory decomposition, then the theory of the synchronous product is* $(\Sigma_{\otimes}, \widehat{B}, \widehat{E_0})$*.*

Example 4. For example, given the protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ of Example 3 the set $HPS_{\mathcal{P}_1 \otimes \mathcal{P}_2}$ of protocol strands of the synchronous product $\mathcal{P}_1 \otimes \mathcal{P}_2$ is as follows:

$$HPS_{\mathcal{P}_1 \otimes \mathcal{P}_2} = \{\ [\ ((m_1 \oplus m_1) \otimes (m_1 \oplus m_2))^+\ ]\ \}$$

The indistinguishability of a protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ in Maude-NPA is characterized in terms of the synchronous product $\mathcal{P}_1 \otimes \mathcal{P}_2$, as we explain in Section 5.3. To be analyzed in Maude-NPA the equational theory of the synchronous product $\mathcal{P}_1 \otimes \mathcal{P}_2$ should have a finite variant decomposition. The following result states that, if the rewrite theory used by two protocols $\mathcal{P}_1$ and $\mathcal{P}_2$ has a finite variant decomposition, then so does the rewrite theory of $\mathcal{P}_1 \otimes \mathcal{P}_2$.

Theorem 1. *For a synchronous product $\mathcal{P}_1 \otimes \mathcal{P}_2$, the rewrite theory $(\Sigma_\otimes, \widehat{B}, \widehat{E_0})$ has the finite variant property iff the rewrite theory (Σ, B, E_0) does also.*

Proof. Since $(\widehat{\Sigma}, \widehat{B}, \widehat{E_0})$ is just a sort-renamed copy of (Σ, B, E_0) it has the FV property. Note that $(\widehat{\Sigma}, \widehat{B}, \widehat{E_0}) \subseteq (\Sigma_\otimes, \widehat{B}, \widehat{E_0})$ is a theory inclusion, and $\Sigma_\otimes$ terms are either $\widehat{\Sigma}$-terms, which have all a finite set of variants, or terms in $\mathcal{T}_{\Sigma_\otimes}(\mathcal{X}) - \mathcal{T}_{\widehat{\Sigma}}(\mathcal{X})$, which are either variables, whose only variant is itself, or terms of the form $t \otimes t'$, with $t, t' \in \mathcal{T}_{\widehat{\Sigma}}(\mathcal{X})$. Let $bd(t)$ and $bd(t')$ be the bounds (see [15]) on reduction sequences to their normal forms for t and t', respectively. Then $bd(t \otimes t') \leq bd(t) + bd(t')$, and thus $(\Sigma_\otimes, \widehat{B}, \widehat{E})$ has the FV property. $\square$

Using Definition 2, the forward analysis of a synchronous product $\mathcal{P}_1 \otimes \mathcal{P}_2$ induces a transition system $\mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2}$.

In the following, we define two projection functions that will be used below to connect the behavior of a synchronous product of protocols with the behavior of each protocol.

Definition 6 (Projection Functions). *Let Σ and $\Sigma_\otimes$ be as in Definition 5, and let $\widehat{\mathcal{X}} = \{\mathcal{X}_{\widehat{\mathsf{s}}}\}_{\widehat{\mathsf{s}} \in \widehat{\mathsf{S}}}$, and $\mathcal{X} = \{\mathcal{X}_{\mathsf{s}}\}_{\mathsf{s} \in \mathsf{S}}$. We then define functions $\pi_1, \pi_2 : \mathcal{T}_{\Sigma_\otimes}(\widehat{\mathcal{X}}) \to \mathcal{T}_\Sigma(\mathcal{X})$ recursively as follows:*

$$\begin{aligned}
&\pi_1(x{:}\widehat{\mathsf{s}}) = \pi_2(x{:}\widehat{\mathsf{s}}) = x{:}\mathsf{s}\\
&\pi_1(t_1 \otimes t_2) = t_1 \qquad \pi_2(t_1 \otimes t_2) = t_2\\
&\pi_1(f(t_1, \ldots, t_n)) = f(\pi_1(t_1), \ldots, \pi_1(t_n)) \textit{ for } f \neq \otimes\\
&\pi_2(f(t_1, \ldots, t_n)) = f(\pi_2(t_1), \ldots, \pi_2(t_n)) \textit{ for } f \neq \otimes
\end{aligned}$$

These projection functions are homomorphically extended to states, transitions and transition systems.

Proposition 1. *Given two protocols $\mathcal{P}_1$ and $\mathcal{P}_2$, its synchronous product $\mathcal{P}_1 \otimes \mathcal{P}_2$ and their associated transition systems $\mathcal{L}_{\mathcal{P}_1}$, $\mathcal{L}_{\mathcal{P}_2}$, and $\mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2}$ as defined above, the projection functions in Definition 6, $\pi_1 : \mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2} \to \mathcal{L}_{\mathcal{P}_1}$, and $\pi_2 : \mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2} \to \mathcal{L}_{\mathcal{P}_2}$ are both simulations.*

Proof. Easy from analysis of the rules $R_{F\mathcal{P}_1}$, $R_{F\mathcal{P}_2}$, and $R_{F(\mathcal{P}_1 \otimes \mathcal{P}_2)}$. $\square$

5.3 Indistinguishability in Maude-NPA

We now define IM and IAES as follows.

Definition 7 (Indistinguishable Messages (IM)). *A protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ with underlying equational theory $(\Sigma, E_0 \cup B)$ satisfies the* indistinguishable messages (IM) property *iff for any initial state St_0 of $\mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2}$, there exists no sequence of transitions $St_0 \rightarrow St_1 \cdots St_{n-1} \rightarrow St_n$ such that the intruder knowledge in St_n contains two facts $(m_1 \otimes m_2) \in \mathcal{I}$ and $(m'_1 \otimes m'_2) \in \mathcal{I}$ such that either (i) $m_1 =_E m'_1$ but $m_2 \neq_E m'_2$, or (ii) $m_2 =_E m'_2$ but $m_1 \neq_E m'_1$, where $E = E_0 \cup B$.*

Definition 8 (Indistinguishable Attack Event Sequences (IAES)). *Given a protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ and the two projections $\pi_1 : \mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2} \rightarrow \mathcal{L}_{\mathcal{P}_1}$ and $\pi_2 : \mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2} \rightarrow \mathcal{L}_{\mathcal{P}_2}$, we say that $\mathcal{P}_1, \mathcal{P}_2$ have* indistinguishable attack event sequences (IAES) *iff π_1 and π_2 are bisimulations.*

Corollary 1. *Given a protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ and the two projections $\pi_1 : \mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2} \rightarrow \mathcal{L}_{\mathcal{P}_1}$ and $\pi_2 : \mathcal{L}_{\mathcal{P}_1 \otimes \mathcal{P}_2} \rightarrow \mathcal{L}_{\mathcal{P}_2}$, if π_1 and π_2 are bisimulations then protocols $\mathcal{P}_1$ and $\mathcal{P}_2$ are bisimilar.*

In Section 6 we explain in detail how the IAES and IM properties explained above are analyzed in practice in Maude-NPA.

6 Indistinguishability Verification in Maude-NPA

In this section we explain how the theoretical framework for indistinguishability verification presented in Section 5 can be implemented in Maude-NPA. In Section 4 we presented the rewriting-based forwards operational semantics of Maude-NPA, which is used as a reference model to define our notion of indistinguishability in Maude-NPA, as explained in Section 5. However, Maude-NPA performs a *backward narrowing-based reachability analysis* that we briefly recall in Section 6.1 (see [13] for further details). Thus the IM and IAES properties are analyzed in Maude-NPA by performing backward narrowing-based reachability analysis from certain attack patterns, as explained in Section 6.2 below.

6.1 Backwards Reachability Analysis in Maude-NPA

Maude-NPA explores *symbolic state patterns* $[t(x_1, \ldots, x_n)] \in T_{\Sigma_{\mathcal{P}}/E_{\mathcal{P}}}(X)$ on the free $(\Sigma_{\mathcal{P}}, E_{\mathcal{P}})$-algebra over a set of variables X. In this way, a state pattern $[t(x_1, \ldots, x_n)]$ represents not a single concrete state but a possibly infinite set of such states, namely, all the instances of the pattern $[t(x_1, \ldots, x_n)]$ where the variables $x_1, \ldots, x_n$ have been instantiated by concrete ground terms.

As in Section 4, a state is represented as a set of strands and the intruder knowledge. However, in the backwards semantics there are two kinds of intruder facts: *positive* knowledge facts (the intruder knows m, i.e., $m \in \mathcal{I}$), and *negative* knowledge facts (the intruder *does not yet know* m but *will know it in a future state*, i.e., $m \notin \mathcal{I}$), where m is a message expression.

In the symbolic backwards semantics *strands* evolve over time; the symbol $|$ is used to divide past and future. That is, in a strand $[\ {m_1}^{\pm}, \ldots, {m_i}^{\pm}\ |\, {m_{i+1}}^{\pm}, \ldots,\ {m_k}^{\pm}\ ]$, messages $m_1^{\pm}, \ldots, m_i^{\pm}$ are the *past messages*, and messages $m_{i+1}^{\pm}, \ldots, m_k^{\pm}$ are the *future messages* ($m_{i+1}^{\pm}$ is the immediate future message).

State changes are described by means of a set $R_{B\mathcal{P}}$ of *rewrite rules*, so that the rewrite theory $(\Sigma_{\mathcal{P}}, E_{\mathcal{P}}, R_{B\mathcal{P}})$ characterizes the behavior of protocol $\mathcal{P}$ modulo the equations $E_{\mathcal{P}}$ for backwards execution. The rules $R_{B\mathcal{P}}$ are generic rules plus rules generated for each principal and intruder strand in the given protocol. See [14] or [13] for more detail and a formal definition of the backwards semantics.

The *backwards* reachability analysis performed by Maude-NPA consists in symbolically running the protocol "in reverse" by narrowing modulo the equations $E_{\mathcal{P}}$. This can be achieved by using the set of rules $R_{B\mathcal{P}}^{-1}$ (where $v \longrightarrow u$ is in $R_{B\mathcal{P}}^{-1}$ iff $u \longrightarrow v$ is in $R_{B\mathcal{P}}$), and performing backwards *narrowing* steps of the form $S \rightsquigarrow_{R_{B\mathcal{P}}^{-1}, E_{\mathcal{P}}} S'$ to search for an *initial state pattern* from an *attack pattern* symbolically describing an attack situation, If Maude-NPA finds an initial state, then the protocol is insecure, and if it terminates without finding an initial state, then it proves the protocol is secure w.r.t. the attack pattern.

The following result, proved in [14], allows us to use reachability in the backwards semantics to prove reachability in the forwards semantics.

Theorem 2 (Soundness and Completeness). ([14], Theorems 1 and 2) *The backwards narrowing-based reachability analysis performed by Maude-NPA is* sound *and* complete *with respect to the forwards rewriting-based reachability analysis presented in Section 4.2.*

6.2 Analysis of Indistinguishability properties in Maude-NPA

In Section 5.3 we defined our notion of indistinguishability as the conjunction of two more basic properties, namely, *Indistinguishable Messages* (IM) and *Indistinguishable Attacker Event Sequences* (IAES), using the forwards operational semantics of Maude-NPA explained in Section 4. In this section we prove a result with respect to the symbolic backwards semantics of the Maude-NPA protocol analysis tool presented in 6.1, showing that the conjunction of IM and IAES of a protocol pairing $\mathcal{P}_1, \mathcal{P}_2$ can be expressed as the unreachability of several different attack patterns for the synchronous product $\mathcal{P}_1 \otimes \mathcal{P}_2$.

In the following, for each property we describe a set of attack patterns in Maude-NPA's syntax denoting states violating that property. If Maude-NPA proves that at least one of the patterns is reachable from an initial state, then it proves that the protocol pairing violates the property and, therefore, the intruder can distinguish between both protocol variants. If the tool proves that no attack pattern is reachable (by terminating without finding an initial state), then it proves that the protocol pairing satisfies the property. Necessary and sufficient conditions for IM and IAES respectively are given in the two theorems below.

Theorem 3. *Let $\mathcal{P}_1, \mathcal{P}_2$ be a protocol pairing. Then $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy the IM property iff no initial state can be symbolically backwards reached from an attack state of $\mathcal{P}_1 \otimes \mathcal{P}_2$ of either one of the following forms:*

(1) $\{SS \,\&\, \{(m_1 \otimes m_2){\in}\mathcal{I},\ (m_1 \otimes m_2'){\in}\mathcal{I}, (m_2 \neq_E m_2'), IK\}\}$, *or*
(2) $\{SS \,\&\, \{(m_1 \otimes m_2){\in}\mathcal{I},\ (m_1' \otimes m_2){\in}\mathcal{I}, (m_1 \neq_E m_1'), IK\}\}$

Proof. Trivial by the soundness and completeness of the backwards operational semantics w.r.t. the forwards operational semantics (see Theorem 2). □

Theorem 4. *Let $\mathcal{P}_1, \mathcal{P}_2$ be a protocol pairing satisfying the IM property. Then $\mathcal{P}_1, \mathcal{P}_2$ satisfy the IAES property iff no initial state can be symbolically backwards reached from any attack state of either of the forms:*

(1) $\{\ SS \,\&\, [\ L\ \mid\ -(m_1 \otimes m_2),\ L'\ ] \,\&\, \{(m_1 \otimes m_2'){\in}\mathcal{I},\ (m_2 \neq_E m_2'),\ IK\}\}$, *or*
(2) $\{\ SS \,\&\, [\ L\ \mid\ -(m_1 \otimes m_2),\ L'\ ] \,\&\, \{(m_1' \otimes m_2){\in}\mathcal{I},\ (m_1 \neq_E m_1'),\ IK\}\}$

Proof. ($\Rightarrow$) We reason by contradiction. Suppose $\mathcal{P}_1, \mathcal{P}_2$ satisfy IAES and an attack of type (1) exists. By the soundness and completeness of the backwards narrowing performed by Maude-NPA there is a ground substitution θ such that from an initial state of $\mathcal{P}_1 \otimes \mathcal{P}_2$ we can reach a ground state, via the forwards semantics, of the form $t = \{\widetilde{SS}\ \theta \,\&\, [L\theta] \,\&\, \{(m_1\theta \otimes m_2'\theta){\in}\mathcal{I}, \widetilde{IK}\ \theta\}\}$, where $\widetilde{SS}$ are instances of *already executed* strand fragments in SS, and $\widetilde{IK}\ \theta$ are the instances of *positive* knowledge facts in IK, and with $m_2'\theta \neq_E m_2\theta$.

Because $(m_1\theta \otimes m_2\theta){\in}\mathcal{I}$, $\mathcal{P}_1$ can make a transition

$$\pi_1(t) \to \{\pi_1(\widetilde{SS}\ \theta) \,\&\, [\pi_1(L\theta), -m_1\theta] \,\&\, \{m_1\theta{\in}\mathcal{I}, \pi_1(\widetilde{IK}\ \theta)\}\} \qquad (\dagger)$$

But, since π_1 is a bisimulation, this means that $\mathcal{P}_1 \otimes \mathcal{P}_2$ can make a transition

$$t \to \{SS\theta \,\&\, [L, -(m_1\theta \otimes m_2\theta)] \,\&\, \{(m_1\theta \otimes m_2\theta){\in}\mathcal{I}, IK\}\} \qquad (\ddagger)$$

which is only possible if there is a fact $(m_1\theta \otimes m_2\theta){\in}\mathcal{I}$ in IK, violating the IM assumption. The proof for an attack of type (2) is entirely similar.

($\Leftarrow$) Suppose no attacks of type (1) or (2) exist but, say $\pi_1 : \mathcal{P}_1 \otimes \mathcal{P}_2 \to \mathcal{P}_1$ (the case for π_2 is similar) is *not* a bisimulation. Thus there is a ground state t reachable from the initial state via the forwards semantics such that $\mathcal{P}_1$ can make a transition $\pi_1(t) \to u$ but there is no transition $t \to v$ in $\mathcal{P}_1 \otimes \mathcal{P}_2$ with $\pi_1(v) = u$. This can only happen for a *message receive* transition in a *user strand*. Therefore, the transition must be of the form (†) above, but there is *no* transition of the form (‡) above. Thus the received message $m_1\theta{\in}\mathcal{I}$ in $\pi_1(t)$ comes from a pair $(m_1\theta \otimes m_2'\theta){\in}\mathcal{I}$ in t such that $m_2'\theta \neq_E m_2\theta$, contradicting the assumption that no attack of type (1) exists. □

6.3 Experimental evaluation

We have begun exploring the implementation of indistinguishability verification in Maude-NPA following the method presented in this paper, and performed a preliminary evaluation (source files and output available at `http://www.dsic.upv.es/~ssantiago/indist.html`). Although we were unable to achieve termination of some analyses, we have proved in Maude-NPA the non-indistinguishability of some pairs of protocols such as the protocol involving an

XOR operator of Example 1 and a complete version of the EKE protocol following the style of Example 2. We also investigated reasons for non-termination due to state space explosion. One is that the attack states used are quite general. As future work, we plan to study sound and complete ways to replace them by more specific atttack states. Another is that these analyses make a heavy use of inequalities modulo equational theories. ProVerif includes methods for solving inequality constraints early in the search, but they do not apply to theories with AC operators. Thus we hold off on evaluating inequalities until the end of the search. However, work on evaluating inequality constraints for theories with AC is beginning to appear (see e.g. [17]). We expect to explore this issue further in our future work.

7 Conclusions

We have formalized an intuitive notion of indistinguishability as the conjunction of the IM and IAES properties defined in terms of the synchronous products of two protocols and shown it to be *checkable* automatically by Maude-NPA. This is a significant step forward in indistinguishability research, because, for the first time, there is a tool that can perform such automatic checks *modulo* a very wide class of theories, namely, all theories with the finite variant property that can have axioms B such as AC or C, which include theories like Abelian Groups, several theories of homomorphic encryption, exclusive-or, and modular exponentiations essential for many privacy-preserving protocols. We have also illustrated with concrete examples how this kind of indistinguishability analysis can be performed by Maude-NPA.

As usual, much work remains ahead. First of all, our indistinguishability notion can be further strengthened in various ways that should also be formalized and mechanized in Maude-NPA. For example, it can be extended to cover other issues that may allow the intruder to distinguish two protocols, such as the sorts of the messages. Second, an interesting line of future work in this sense is the handling of protocols with branching, which would allow us to check indistinguishability properties for a wider class of protocols. Third, a more detailed comparison with other indistinguishability notions should be carried out. This is nontrivial, since they depend on different protocol models. Fourth, much more experimentation is needed to test and improve our capacity to prove indistinguishability for protocol pairings.

References

1. M. Abadi and C.Fournet. Mobile values, new names, and secure communication. In *POPL*, pp. 104–115, 2001.
2. M. Arapinis, S. Bursuc, and M. D. Ryan. Reduction of equational theories for verification of trace equivalence: Re-encryption, associativity and commutativity. In *POST*, pp. 169–188, 2012.

3. G. B. Barthe, J. M. Crespo, B. Grégoire, C.Kunz, Y. Lakhnech, B. Schmidt, and S. Z. Béguelin. Fully automated analysis of padding-based encryption in the computational model. In *ACM Conference on Computer and Communications Security*, pp. 1247–1260, 2013.
4. M. Baudet. Deciding security of protocols against off-line guessing attacks. In *Proc. ACM CCS 2005*, pp. 16–25. ACM, 2005.
5. S. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *Proceedings of the 1992 Symposium on Research in Security and Privacy*, pp. 72–84. IEEE, 1992.
6. B. Blanchet, M. Abadi, and C.Fournet. Automated verification of selected equivalences for security protocols. *J. Log. Algebr. Program.*, 75(1):3–51, 2008.
7. R. Chadha, Ş. Ciobâcă, and S. Kremer. Automated verification of equivalence properties of cryptographic protocols. In *ESOP*, pp. 108–127, 2012.
8. V. Cheval, H. Comon-Lundh, and S. Delaune. Automating security analysis: symbolic equivalence of constraint systems. In *Proceedings of IJCAR'10*, *LNAI* vol. 6173, pp. 412–426, Edinburgh, Scotland, UK, July 2010. Springer-Verlag.
9. V. Cheval, H. Comon-Lundh, and S. Delaune. Trace equivalence decision: negative tests and non-determinism. In *Proc. ACM CCS 2011*, pp. 321–330, 2011.
10. V. Cheval, V. Cortier, and A. Plet. Lengths may break privacy - or how to check for equivalences with length. In *CAV*, *LNCS* vol. 8044, pp. 708–723. Springer, 2013.
11. M. Clarkson and F. Schneider. Hyperproperties. *J. Computer Security*, 18(6):1157-1210, 2010.
12. V. Cortier and S. Delaune. A method for proving observational equivalence. In *CSF*, pp. 266–276. IEEE Computer Society, 2009.
13. S. Escobar, C. Meadows, and J. Meseguer. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In *FOSAD 2007/2008/2009 Tutorial Lectures*, LNCS vol. 5705, pp. 1–50. Springer, 2009.
14. S. Escobar, C. Meadows, J. Meseguer, and S. Santiago. A rewriting-based forwards semantics for Maude-NPA. In *Proc. HotSoS*, 2014. to appear (Preliminary version available at: `http://www.dsic.upv.es/~sescobar/papers/HotSoS2014.pdf`).
15. S. Escobar, R. Sasse, and J. Meseguer. Folding variant narrowing and optimal variant termination. *J. Log. Algebr. Program.*, 81(7-8):898–928, 2012.
16. F. J. Thayer Fabrega, J. Herzog, and J. Guttman. Strand Spaces: What Makes a Security Protocol Correct? *Journal of Computer Security*, 7:191–230, 1999.
17. R. Gutiérrez, J. Meseguer, and C. Rocha. Order-sorted equality enrichments modulo axioms. In *WRLA*, pp. 162–181, 2012.
18. G. Lowe. Analysings protocol subject to guessing attacks. *Journal of Computer Security*, 12(1):83–98, 2004.
19. M.Merritt. *Cryptographic Protocols.* PhD thesis, Georgia Inst. of Technology, 1984.
20. J. Meseguer. Conditional rewriting logic as a united model of concurrency. *Theor. Comput. Sci.*, 96(1):73–155, 1992.
21. J. Meseguer. Membership algebra as a logical framework for equational specification. In *Proc. WITS'06*, *LNCS* vol. 1376, pp. 18–61. Springer, 1997.
22. T. Newcomb and G. Lowe A computational justification for guessing attack formalisms Technical report No. RR-05-05. Oxford University Computing Laboratory. October, 2005
23. TeReSe, ed. *Term Rewriting Systems.* Cambridge Univ. Press, Cambridge, 2003.