Effective Symbolic Protocol Analysis via Equational Irreducibility Conditions*

Serdar Erbatur[1], Santiago Escobar[2], Deepak Kapur[3], Zhiqiang Liu[4], Christopher Lynch[4], Catherine Meadows[5], José Meseguer[6], Paliath Narendran[1], Sonia Santiago[2], and Ralf Sasse[6]

[1] University at Albany-SUNY, Albany, NY, USA
se@cs.albany.edu, dran@cs.albany.edu
[2] DSIC-ELP, Universitat Politècnica de València, Spain
sescobar@dsic.upv.es,ssantiago@dsic.upv.es
[3] University of New Mexico, Albuquerque, NM, USA
kapur@cs.unm.edu
[4] Clarkson University, Potsdam, NY, USA
liuzh@clarkson.edu, clynch@clarkson.edu
[5] Naval Research Laboratory, Washington DC, USA
meadows@itd.nrl.navy.mil
[6] University of Illinois at Urbana-Champaign, USA
meseguer@illinois.edu, rsasse@illinois.edu

Abstract. We address a problem that arises in cryptographic protocol analysis when the equational properties of the cryptosystem are taken into account: in many situations it is necessary to guarantee that certain terms generated during a state exploration are in *normal form* with respect to the equational theory. We give a tool-independent methodology for state exploration, based on unification and narrowing, that generates states that obey these irreducibility constraints, called *contextual symbolic reachability analysis*, prove its soundness and completeness, and describe its implementation in the Maude-NPA protocol analysis tool. Contextual symbolic reachability analysis also introduces a new type of unification mechanism, which we call *asymmetric unification*, in which any solution must leave the right side of the solution irreducible. We also present experiments showing the effectiveness of our methodology.

1 Introduction

There has been an increasing amount of research in recent years in building tools for cryptographic protocol analysis where the equational properties of the

* S. Escobar and S. Santiago have been partially supported by the EU (FEDER) and the Spanish MEC/MICINN under grant TIN 2010-21062-C02-02, and by Generalitat Valenciana PROMETEO2011/052. The following authors have been partially supported by NSF: S. Escobar, J. Meseguer and R. Sasse under grants CCF 09-05584, CNS 09-04749, and CNS 09-05584; D. Kapur under grant CNS 09-05222; C. Lynch, Z. Liu, and C. Meadows under grant CNS 09-05378, and P. Narendran and S. Erbatur under grant CNS 09-05286.

cryptosystems are taken into account. This allows one to retain the advantages of a Dolev-Yao style [14] analyzer, such as ease of reasoning about concurrency and ability to construct counterexamples, while allowing for greater expressiveness.

With the above in mind, a number of approaches have been explored in the literature for analyzing protocols when equational theories are involved. These include equational unification techniques for unification-based tools such as Maude-NPA [17], equational constraint solving techniques for constraint based tools, e.g. [12, 11], and equational deducibility procedures for checking whether one term is deducible from a given set of terms, e.g. [2, 5, 9, 13].

In many cases, equational reasoning is integrated with syntactic reasoning. There are a number of reasons for doing this, which we describe in more detail in Section 1.1, but one reason is that optimizations that are done to eliminate redundant or nonsensical states may need to be done via syntactic checking, as in Maude-NPA. We illustrate the issues that can arise with the following protocol, which we will use as a running example. It uses an exclusive-or operator $\oplus$, which is associative and commutative (AC) and self-canceling with identity 0, and a function pk, where $pk(A, X)$ stands for encryption of message X with A's (standing for Alice's) public key; below, B stands for Bob.

Example 1. Upon receiving the final message, Alice verifies that she received $X \oplus N_A$ for some X received in the first message $pk(A, X)$. The protocol is seen differently by Bob and Alice, as shown in the second and third columns.

Alice and Bob	Bob	Alice
1. $B \to A : pk(A, N_B)$	1. $B \to A : pk(A, N_B)$	1. $B \to A : pk(A, X)$
2. $A \to B : pk(B, N_A)$	2. $A \to B : pk(B, Z)$	2. $A \to B : pk(B, N_A)$
3. $B \to A : N_A \oplus N_B$	3. $B \to A : Z \oplus N_B$	3. $B \to A : N_A \oplus X$

We find an instance of the protocol from Alice's perspective by applying the substitution $X \mapsto N_A \oplus Y$ to achieve the left-hand column of Example 2. Maude-NPA could identify this instance as infeasible and discard it, since Alice cannot receive a message $N_A \oplus Y$ before she generates the nonce N_A.

Example 2. But further instantiating Y (perhaps as a result of further unifications elsewhere) to $N_A \oplus N_B$ causes problems.

Alice after $X \mapsto N_A \oplus Y$	Alice after $Y \mapsto N_A \oplus N_B$.
1. $B \to A : pk(A, N_A \oplus Y)$	1. $B \to A : pk(A, N_A \oplus N_A \oplus N_B) = pk(A, N_B)$
2. $A \to B : pk(B, N_A)$	2. $A \to B : pk(B, N_A)$
3. $B \to A : N_A \oplus N_A \oplus Y$	3. $B \to A : N_A \oplus N_A \oplus N_A \oplus N_B = N_A \oplus N_B$

This makes $N_A \oplus Y$ reduce to N_B and $N_A \oplus N_A \oplus Y$ reduce to $N_A \oplus N_B$, giving the right-hand side of Example 2: the intended legal execution of the protocol! Thus, Maude-NPA's syntactic check inadvertently could have ruled out a legal execution.

We avoid this problem as follows. We first decompose the $\oplus$ theory into (R, E), where E is the AC theory and R is a set of rewrite rules for the properties $\{X \oplus 0 = X, X \oplus X = 0\}$. We then divide the possible instantiations of $\{pk(A, X), N_A \oplus X\}$ into two cases, each of which are constrained to remain

irreducible under substitution. One is $\{pk(A, X), N_A \oplus X\}$, and the other is $\{pk(A, Y \oplus N_A), Y\}$ obtained by the substitution $X \mapsto Y \oplus N_A$. Every other reduced instantiation of $N_A \oplus X$ is an instance of either one or the other modulo AC. The case obtained by $X \mapsto Y \oplus N_A$ can now be safely deleted, because due to the irreducibility constraint that Y cannot contain N_A and 0, the N_A will never vanish from $N_A \oplus Y$ under any substitution.

This strategy works for several reasons. One is that Maude-NPA syntactic checks require that irreducibility constraints only be put on received messages. Another, and more important, is that the exclusive-or theory has the *finite variant property* [10] modulo AC. Thus, for every term s there is a finite set $s'_1, \ldots, s'_k$ of reduced instances of s such that any other reduced instance of s is equal modulo AC to a substitution instance of one of the s'_i. These two features mean that it is possible to integrate syntactic checks that are invariant under AC together with unification-based reachability modulo a richer theory, allowing us to improve efficiency without sacrificing soundness and completeness. Indeed, this is vital for Maude-NPA and other tools, because almost all of the checks used for optimization require the received messages to be in normal form.

Another capability that is needed for our strategy to work opens up a new area of research, namely, developing a sound and complete, tool-independent symbolic state exploration algorithm that preserves irreducibility constraints. In Maude-NPA state exploration is implemented via equational unification of sent messages with received messages, which means that the equational unification algorithm used should preserve the irreducibility of the received messages. Indeed, it was experimentation with a unification algorithm that did *not* have this property, the algorithm of [24], that produced the example we described above. Variant narrowing unification (the algorithm currently used by Maude-NPA) has the properties that we need, but our search of the literature has produced no other examples. This has led us to define a class of unification algorithms known as *asymmetric unification algorithms* modulo a theory (R, E), which produce a most general set of unifiers which leave the right hand side irreducible. We are working on techniques for converting standard equational unification algorithms into asymmetric algorithms, and have produced an asymmetric version of the exclusive-or algorithm in [24].

We are not the only ones to use an approach that integrates syntactic and equational reasoning: this has also been done by other researchers for other reasons, as we describe in Section 1.1. However, most work in this area has concentrated on specific applications of this approach, and not on how to implement the approach itself. This paper is devoted to providing a general procedure for doing this, called *contextual symbolic reachability analysis* modulo a theory (R, E), where R is a set of rewrite rules. This employs a technique called *contextual unification* in which some subterms of the two terms being unified are constrained to be irreducible. In Maude-NPA these are input terms, which, since they are unified with output terms, create the opportunity for exploiting asymmetric unification. However, this is not the only way contextual symbolic reachability analysis could be implemented. For example, we could follow the

approach of OFMC [4] which requires that both input and output terms are irreducible. Thus, our tool-independent framework should have many applications beyond Maude-NPA, allowing for experimentation with different techniques.

The rest of the paper is organized as follows. In Section 2 we give some preliminary definitions used in rewriting and unification. In Section 3 we give a general procedure for symbolic reachability via narrowing. In Section 4 we introduce contextual symbolic reachability analysis, prove its soundness and completeness, and illustrate its use in Maude-NPA. In Section 5, we show experiments illustrating the benefits, in Maude-NPA, of using contextual symbolic reachability and asymmetric unification to integrate reachability analysis modulo exclusive-or with optimizations based on syntactic checks. In Section 6 we discuss some future directions.

1.1 Related Work

Although our specific approach has not, to the best of our knowledge, been exploited in cryptographic protocol analysis tools outside of Maude-NPA, there are a number of similar cases. For example, ProVerif [6] (detail in [8, Sec. 5]) and OFMC [4] (detail in [29, Sec. 10]) both compute the variants of intruder and/or protocol rules, modulo the free theory for ProVerif, and modulo the free theory or AC for OFMC. This has the effect of computing the variants of both sides of the unification problem. More recently, variants have been applied to expanding the capacity of ProVerif to deal with AC theories. Thus, in [23], Küsters and Truderung implement a special case of the exclusive-or theory in the ProVerif tool by expressing it as a rewrite theory with the finite variant property with respect to the free theory ($E = \emptyset$) and computing variants that are unified syntactically. This requires some restrictions on the syntax of the protocol, however. Similar approaches have been applied by Küsters and Truderung for modular exponentiation [22], and Arapinis et al. [3] for commuting encryption and AC theories.

The main differences between this work and what we propose here are twofold. First of all, unlike [8, 23, 22, 3] we do not restrict ourselves to the case in which E is the free theory ($E = \emptyset$), but allow it to be AC, or, potentially, any other theory for which finitary unification algorithms exist. Secondly, unlike ProVerif, OFMC, and [23, 22, 3] we do not necessarily require that irreducible variants be computed for both sides of a unification problem, but we allow for example the possibility that variants are computed for only one side, allowing for potentially more efficient special-purpose asymmetric unification algorithms.

2 Preliminaries

We follow the classical notation and terminology from [32] for term rewriting, and from [27] for rewriting logic and order-sorted notions. We assume an order-sorted signature $\mathbf{\Sigma} = (S, \leq, \Sigma)$ with poset of sorts $(S, \leq)$. 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. For a term t, $Var(t)$ denotes the set of variables in t.

A *substitution* $\sigma \in \mathcal{S}ubst(\Sigma, \mathcal{X})$ is a sorted mapping from a finite subset of $\mathcal{X}$ to $\mathcal{T}_\Sigma(\mathcal{X})$. Substitutions are written as $\sigma = \{X_1 \mapsto t_1, \ldots, X_n \mapsto t_n\}$ where the domain of σ is $Dom(\sigma) = \{X_1, \ldots, X_n\}$ and the set of variables introduced by terms $t_1, \ldots, t_n$ is written $Ran(\sigma)$. The identity substitution is *id*. Substitutions are homomorphically extended to $\mathcal{T}_\Sigma(\mathcal{X})$. The application of a substitution σ to a term t is denoted by $t\sigma$ or $\sigma(t)$.

A *Σ-equation* is an unoriented pair $t = t'$, where $t, t' \in \mathcal{T}_\Sigma(\mathcal{X})_\mathsf{s}$ for some sort $\mathsf{s} \in \mathsf{S}$. An *equational theory* (Σ, E) is a pair with Σ an order-sorted signature and E a set of Σ-equations.

An *E-unifier* for a Σ-equation $t = t'$ is a substitution σ such that $t\sigma =_E t'\sigma$. For $Var(t) \cup Var(t') \subseteq W$, a set of substitutions $CSU_E^W(t = t')$ is said to be a *complete* set of unifiers for the equality $t = t'$ modulo E away from W iff: (i) each $\sigma \in CSU_E^W(t = t')$ is an E-unifier of $t = t'$; (ii) for any E-unifier ρ of $t = t'$ there is a $\sigma \in CSU_E^W(t = t')$ such that $\sigma|_W \sqsupseteq_E \rho|_W$ (i.e., there is a substitution η such that $(\sigma\eta)|_W =_E \rho|_W$); and (iii) for all $\sigma \in CSU_E^W(t = t')$, $Dom(\sigma) \subseteq (Var(t) \cup Var(t'))$ and $Ran(\sigma) \cap W = \emptyset$.

A *rewrite rule* is an oriented pair $l \to r$, where $l \notin \mathcal{X}$ and $l, r \in \mathcal{T}_\Sigma(\mathcal{X})_\mathsf{s}$ for some sort $\mathsf{s} \in \mathsf{S}$. 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 rewriting relation on $\mathcal{T}_\Sigma(\mathcal{X})$, written $t \to_R t'$ or $t \to_{p,R} t'$ holds between t and t' iff there exist $p \in Pos_\Sigma(t)$, $l \to r \in R$ and a substitution σ, such that $t|_p = l\sigma$, and $t' = t[r\sigma]_p$. The relation $\to_{R/E}$ on $\mathcal{T}_\Sigma(\mathcal{X})$ is $=_E; \to_R; =_E$. The transitive (resp. transitive and reflexive) closure of $\to_{R/E}$ is denoted $\to^+_{R/E}$ (resp. $\to^*_{R/E}$). A term t is called $\to_{R/E}$-irreducible (or just R/E-irreducible) if there is no term t' such that $t \to_{R/E} t'$. For $\to_{R/E}$ confluent and terminating, the irreducible version of a term t is denoted by $t{\downarrow_{R/E}}$.

A relation $\to_{R,E}$ on $\mathcal{T}_\Sigma(\mathcal{X})$ is defined as: $t \to_{p,R,E} t'$ (or just $t \to_{R,E} t'$) iff there is a non-variable position $p \in Pos_\Sigma(t)$, a rule $l \to r$ in R, and a substitution σ such that $t|_p =_E l\sigma$ and $t' = t[r\sigma]_p$. $\to_{R/E}$-reducibility is undecidable in general since E-congruence classes can be arbitrarily large. Therefore, R/E-rewriting is usually implemented [21] by R, E-rewriting under some conditions on R and E such as confluence, termination, and coherence (see [21]). We call (Σ, E, R) a *decomposition* of an order-sorted equational theory (Σ, G) if $G = R \uplus E$ and R and E satisfy the conditions for $\to_{R,E}$ to implement $\to_{R/E}$.

Given a decomposition (Σ, E, R) of an equational theory, (t', θ) is an *R, E-variant* [19] (or just a variant) of term t if $t\theta{\downarrow_{R,E}} =_E t'$ and $\theta{\downarrow_{R,E}} =_E \theta$. A *complete set of R, E-variants* [19] (up to renaming) of a term t is a subset, denoted by $[\![t]\!]_{R,E}$, of the set of all R, E-variants of t such that, for each R, E-variant (t', σ) of t, there is an R, E-variant $(t'', \theta) \in [\![t]\!]_{R,E}$ such that $(t'', \theta) \sqsupseteq_{R,E} (t', \sigma)$, i.e., there is a substitution ρ such that $t' =_E t''\rho$ and $\sigma|_{Var(t)} =_E (\theta\rho)|_{Var(t)}$. A decomposition (Σ, E, R) has the *finite variant property* [19] (also called a *finite*

variant decomposition) iff for each Σ-term t, a complete set $[\![t]\!]_{R,E}$ of its most general variants is finite.

3 Symbolic Reachability Analysis by Narrowing

In this section we recall basic facts about narrowing modulo equations of [28] using topmost rewriting as a tool-independent semantic framework for symbolic reachability analysis of protocols under algebraic properties. We first define reachability goals.

Definition 1 (Reachability goal). *Given an order-sorted rewrite theory (Σ, G, T), a* reachability goal *is defined as a pair $t \stackrel{?}{\rightarrow}^*_{T/G} t'$, where $t, t' \in \mathcal{T}_\Sigma(\mathcal{X})_\mathsf{s}$. It is abbreviated as $t \stackrel{?}{\rightarrow}^* t'$ when the theory is clear from the context; t is the* source *of the goal and t' is the* target*. A substitution σ is a T/G-*solution *of the reachability goal (or just a solution for short) iff there is a sequence $\sigma(t) \rightarrow_{T/G} \sigma(u_1) \rightarrow_{T/G} \cdots \rightarrow_{T/G} \sigma(u_{k-1}) \rightarrow_{T/G} \sigma(t')$.*

A set Γ of substitutions is said to be a complete set of solutions *of $t \stackrel{?}{\rightarrow}^*_{T/G} t'$ iff (i) every substitution $\sigma \in \Gamma$ is a solution of $t \stackrel{?}{\rightarrow}^*_{T/G} t'$, and (ii) for any solution ρ of $t \stackrel{?}{\rightarrow}^*_{T/G} t'$, there is a substitution $\sigma \in \Gamma$ more general than ρ modulo G, i.e., $\sigma|_{Var(t) \cup Var(t')} \sqsupseteq_G \rho|_{Var(t) \cup Var(t')}$.*

If in a goal $t \stackrel{?}{\rightarrow}^*_{T/G} t'$, terms t and t' are ground, then goal solving becomes a standard rewriting reachability problem. However, since we allow terms t, t' with variables, we need a mechanism more general than standard rewriting to find solutions of reachability goals. *Narrowing* generalizes rewriting by performing *unification* at non-variable positions instead of the usual matching. Specifically, narrowing instantiates the variables in a term by a G-unifier that enables a rewrite modulo G with a given rule and a term position.

Definition 2 (Narrowing modulo G). *Given an order-sorted rewrite theory (Σ, G, T), the narrowing relation on $\mathcal{T}_\Sigma(\mathcal{X})$ modulo G is defined as $t \stackrel{\sigma}{\rightsquigarrow}_{T,G} t'$ (or $\stackrel{\sigma}{\rightsquigarrow}$ if T, G is understood) iff there is $p \in Pos_\Sigma(t)$, a rule $l \rightarrow r$ in T such that $Var(t) \cap (Var(l) \cup Var(r)) = \emptyset$, and $\sigma \in CSU^V_G(t|_p = l)$ for a set V of variables containing $Var(t)$, $Var(l)$, and $Var(r)$, such that $t' = \sigma(t[r]_p)$.*

The reflexive and transitive closure of narrowing is defined as $t \stackrel{\sigma}{\rightsquigarrow}^*_{T,G} t'$ iff either $t = t'$ and $\sigma = id$, or there are terms $u_1, \ldots, u_n$, $n \geq 1$, and substitutions $\sigma_1, \ldots, \sigma_{n+1}$ s.t. $t \stackrel{\sigma_1}{\rightsquigarrow}_{T,G} u_1 \stackrel{\sigma_2}{\rightsquigarrow}_{T,G} u_2 \cdots u_n \stackrel{\sigma_{n+1}}{\rightsquigarrow}_{T,G} t'$ and $\sigma = \sigma_1 \cdots \sigma_{n+1}$.

Soundness and completeness of narrowing for solving reachability goals is proved in [21, 28] for order-sorted *topmost* rewrite theories, i.e., rewrite theories were all the rewrite steps happened at the top of terms.

3.1 Search in Maude-NPA

In this section we give a high-level summary of the general narrowing-based approach implemented in Maude-NPA. For further information, please see [15, 17]. Note that our treatment of symbolic reachability analysis modulo equations by narrowing is completely general and *tool-independent*. We only use Maude-NPA for illustration purposes to give examples, and also because it supports the irreducibility conditions discussed in this paper. Multiset rewrite rules, used as a model for protocol analysis [30, 7], is another example of topmost rewrite theories where reachability properties are checked.

Given a protocol $\mathcal{P}$, states are modeled as elements of an initial algebra $T_{\Sigma_\mathcal{P}/E_\mathcal{P}}$, 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 $[t] \in T_{\Sigma_\mathcal{P}/E_\mathcal{P}}$ with t a ground $\Sigma_\mathcal{P}$-term. However, we explore *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 sorted variables X.

In Maude-NPA [15, 17], a *state pattern* in a protocol execution is a term t of sort State, $t \in T_{\Sigma_\mathcal{P}/E_\mathcal{P}}(X)_{\mathsf{State}}$, which is a term of the form $\{S_1 \& \cdots \& S_n \& \{IK\}\}$ where & is an associative-commutative union operator with identity symbol $\emptyset$. Each element in the set is either a *strand* S_i or the *intruder knowledge* $\{IK\}$ at that state.

The *intruder knowledge* $\{IK\}$ also belongs to the state and is represented as a set of facts. 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.

A *strand* [20] represents the sequence of messages sent and received by a principal executing the protocol and is represented as a sequence of messages $[msg_1^-, msg_2^+, msg_3^-, \ldots, msg_{k-1}^-, msg_k^+]$ such that msg_i is a term of sort Msg, msg^- (also written $-msg$) represents an *input* message, and msg^+ (also written $+msg$) represents an *output* message. 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 specified for each intruder action). In Maude-NPA, strands evolve over time; the symbol | is used to divide past and future. Also, we keep track of all the variables of sort `Fresh` generated by a concrete strand. That is, all the variables $r_1, \ldots, r_j$ of sort `Fresh` generated by a strand are made explicit right before the strand, as follows: $:: r_1, \ldots, r_j :: [\ {m_1}^\pm, \ \ldots, \ {m_i}^\pm \mid {m_{i+1}}^\pm, \ \ldots, \ {m_k}^\pm \ ]$ where $msg_1^\pm, \ldots, msg_i^\pm$ are the past messages, and $msg_{i+1}^\pm, \ldots, msg_k^\pm$ are the future messages ($msg_{i+1}^\pm$ is the immediate future message). The nils are present so that the bar may be placed at the beginning or end of the strand if necessary, but we often remove them, except when there is nothing else between the vertical bar and the beginning or end of a strand. A strand $:: r_1, \ldots, r_j :: [msg_1^\pm, \ldots, msg_k^\pm]$ is a shorthand for $:: r_1, \ldots, r_j :: [nil \mid msg_1^\pm, \ldots, msg_k^\pm, nil]$.

Example 3. For the protocol of Example 1, the strand specification of the protocol is as follows:

$$\begin{array}{ll} \text{(Bob)} & :: r_1 :: [\ +(pk(A, n(B, r_1))),\ -(pk(B, Y)),\ \ \ +(Y \oplus n(B, r_1))\ ] \\ \text{(Alice)} & :: r_2 :: [\ -(pk(A, X)),\ \ \ +(pk(B, n(A, r_2))),\ -(n(A, r_2) \oplus X)\ ] \end{array}$$

Intruder strands are also included for each function. For example, application of exclusive-or by the intruder is described by the strand $[(X)^-, (Y)^-, (X \oplus Y)^+]$.

The protocol analysis methodology of Maude-NPA is then based on the idea of *backward reachability analysis*, where we begin with one or more state patterns corresponding to *attack states*, and want to prove or disprove that they are *unreachable* from the set of initial protocol states. In order to perform such a reachability analysis we must describe how states change as a consequence of principals performing protocol steps and of the intruder actions. This can be done by describing such state changes by means of a set $T_{\mathcal{P}}$ of *rewrite rules*, so that the rewrite theory $(\Sigma_{\mathcal{P}}, G_{\mathcal{P}}, T_{\mathcal{P}})$ characterizes the behavior of protocol $\mathcal{P}$ modulo the equations $G_{\mathcal{P}}$.

The following rewrite rules describe the general state transitions, where each state transition implies moving the vertical bar of one strand:

$$\{SS\ \&\ [L \mid M^-, L']\ \&\ \{M{\in}\mathcal{I}, IK\}\} \rightarrow \{SS\ \&\ [L, M^- \mid L']\ \&\ \{IK\}\} \quad (1)$$

$$\{SS\ \&\ [L \mid M^+, L']\ \&\ \{IK\}\} \rightarrow \{SS\ \&\ [L, M^+ \mid L']\ \&\ \{IK\}\} \quad (2)$$

$$\{SS\ \&\ [L \mid M^+, L']\ \&\ \{M{\notin}\mathcal{I}, IK\}\} \rightarrow \{SS\ \&\ [L, M^+ \mid L']\ \&\ \{M{\in}\mathcal{I}, IK\}\} \quad (3)$$

where variables L, L' denote lists of input and output messages of the form m^+ or m^- within a strand, IK denotes a set of intruder facts ($m{\in}\mathcal{I}$,$m{\notin}\mathcal{I}$), and SS denotes a set of strands. In a *forward execution* of the protocol strands, Rule (1) synchronizes an input message with a message already learned by the intruder, Rule (2) accepts output messages but the intruder's knowledge is not increased, and Rule (3) accepts output messages and the intruder's knowledge is positively increased. For an unbounded number of sessions, we have extra rewrite rules (one for each positive message in a protocol or intruder strand) that dynamically introduce additional strands into a state.

The way to analyze *backwards* reachability is then relatively easy, namely, to run the protocol "in reverse." This can be achieved by using the set of rules $T_{\mathcal{P}}^{-1}$, where $v \longrightarrow u$ is in $T_{\mathcal{P}}^{-1}$ iff $u \longrightarrow v$ is in $T_{\mathcal{P}}$.

Example 4. The protocol of Example 1 can be modeled as a rewrite theory (Σ, G, T) where T is the reversed version of the generic rewrite rules (1)–(3) plus the rewrite rules for introducing new strands. The final pattern used as an input to the backwards symbolic reachability analysis could, for example, be as follows:

$$\begin{array}{l} \{\ :: r_2 :: [nil, -(pk(A, X)), +(pk(B, n(A, r_2))), -(X \oplus n(A, r_2)) \mid nil]\ \& \\ \quad :: r_1 :: [nil, +(pk(A, n(B, r_1))), -(pk(B, Y)), +(Y \oplus n(B, r_1)) \mid nil]\ \& SS\ \&\ \{IK\}\} \end{array}$$

This pattern does not require the intruder to have learnt anything, so it is very general and could lead to a regular execution and to an attack. Indeed, this

protocol has the following attack reachable from that final pattern, where the intruder starts a protocol session with B but uses B's nonce to start a protocol session with A, so finally the intruder is able to learn both B's nonce and A's nonce:

1. $B \to I : pk(i, N_B)$
2. $I \to A : pk(a, N_B)$
3. $A \to B : pk(B, N_A)$
4. $B \to A, I : N_A \oplus N_B$

4 Contextual Symbolic Reachability Analysis

As we have explained in the Introduction, the symbolic reachability approach presented in the previous section does not really work in practice, since the particular way that a representative is chosen for each equivalence class may be crucial for the correct behavior, and in many cases the termination of a tool crucially depends on state space reduction techniques based on checking such representatives, as we illustrated for the case of nonces that *cannot* have been generated yet at a given point. Therefore, we now present a general, tool-independent framework for symbolic reachability analysis which refines narrowing modulo equations by imposing *irreducibility conditions* on representatives of equivalence classes. First, we give a way of imposing these irreducibility conditions on a rewrite theory, expressed by the notion of *contextual rewrite theory.*[7]

Definition 3 (Contextual Rewrite Theory). *A* contextual rewrite theory *is a tuple (Σ, E, R, T, ϕ) where $(\Sigma, E \cup R, T)$ is an order-sorted topmost rewrite theory, (Σ, E, R) is a decomposition of the equational theory $(\Sigma, E \cup R)$, and ϕ, called the* irreducibility requirements*, is a function mapping each $f \in \Sigma$ to a set of its arguments, i.e., $\phi(f) \subseteq \{1, \ldots, ar(f)\}$, where $ar(f)$ is the number of arguments of f. The set of maximal irreducible positions of a term t is denoted by $\phi(t)$.*

*A term t is called ϕ, R, E-*irreducible *(or just ϕ-irreducible) if for each $p \in \phi(t)$, $t|_p \downarrow_{R,E} =_E t|_p$, and* strongly ϕ-irreducible *if for any R, E-normalized substitution σ, $t\sigma$ is ϕ-irreducible.*

Example 5. For the protocol of Examples 1 and 3, the contextual rewrite theory (Σ, E, R, T, ϕ) is formed of T containing the reversed version of the generic rewrite rules (1)–(3) plus the rewrite rules for introducing new strands, and the equational theory $(\Sigma, E \cup R)$ for exclusive-or is decomposed into (Σ, E, R) where E is the associativity and commutativity axioms for $\oplus$ and R is as follows:[8]

$$X \oplus 0 \to X \qquad X \oplus X \to 0 \qquad X \oplus X \oplus Y \to Y$$

[7] Our use of "contextual" should be distinguished from : (i) "contextual rewriting," e.g., [34], and (ii) "context-sensitive rewriting," e.g., [26]. Our use is unrelated to contextual rewriting, which is a form of conditional rewriting with constraints, but is closely related to context-sensitive rewriting, where the rewritable argument positions of a function symbol f are specified by a function $\mu(f) \subseteq \{1, \ldots, ar(f)\}$ similar to our irreducibility requirements function $\phi(f) \subseteq \{1, \ldots, ar(f)\}$. However, ϕ-irreducibility is a *strictly stronger* requirement than μ-irreducibility when $\phi = \mu$.

[8] Note that the two first equations are not AC-coherent, but adding the last equation is sufficient to recover that property (see [33]).

The irreducibility requirements ϕ are imposed on two operators: $-(\_)$ for *input messages* in a strand, and $\_\in\mathcal{I}$ for each *positive fact* in the intruder knowledge. That is, $\phi(-(\_)) = \{1\}$, $\phi(\_\in\mathcal{I}) = \{1\}$, and $\phi(f) = \emptyset$ otherwise.

We extend the notion of a reachability goal to the contextual case.

Definition 4 (Contextual Reachability goal). *Given a contextual rewrite theory (Σ, E, R, T, ϕ), we define a contextual reachability goal as $t \stackrel{?}{\rightarrow}^*_{T,R,E,\phi} t'$, where $t, t' \in \mathcal{T}_\Sigma(\mathcal{X})_{\mathsf{s}}$. We write $t \stackrel{?}{\rightarrow}^*_{\phi} t'$ when the theory is clear. A substitution σ is a* solution *of the contextual reachability goal $t \stackrel{?}{\rightarrow}^*_{T,R,E,\phi} t'$ iff there is a sequence $\sigma(t) \rightarrow_{T,(E\cup R)} \sigma(u_1) \rightarrow_{T,(E\cup R)} \cdots \rightarrow_{T,(E\cup R)} \sigma(u_{k-1}) \rightarrow_{T,(E\cup R)} \sigma(t')$ such that $\sigma(t), \sigma(u_1), \ldots, \sigma(u_{k-1}), \sigma(t')$ are all ϕ, R, E-irreducible.*

As for reachability goals, a contextual version of narrowing provides a mechanism to find solutions to contextual reachability goals. However, we have to first define a new equational unification mechanism, called *contextual unification*, as the basis for contextual narrowing, where the $E \cup R$-unification is extended to the contextual case, which has some asymmetry due to the irreducibility restrictions only on the right hand side.

Definition 5 (Contextual Unification). *Given a contextual rewrite theory (Σ, E, R, T, ϕ), a substitution σ is a* contextual R, E-unifier *of a set P of* contextual equations *of the form $P = \{t_1 =_{\downarrow_\phi} t'_1, \ldots, t_n =_{\downarrow_\phi} t'_n\}$ iff for every contextual equation $t_i =_{\downarrow_\phi} t'_i$ in P, the substitution σ is an $(R \cup E)$-unifier of the equation $t_i = t'_i$ and, furthermore, $\sigma(t'_i)$ is ϕ,R,E-irreducible.*

A set of substitutions Ω is a complete set of contextual R, E-unifiers *of P, denoted by $CSU_{R,E,\phi}(P)$, iff: (i) every member of Ω is a contextual R, E-unifier of P, and (ii) for every contextual R, E-unifier θ of P there exists $\sigma \in \Omega$ such that $\sigma \sqsupseteq_E \theta$.*

Example 6. Consider the protocol of Example 1. The contextual unification problem found by Maude-NPA is $t =_{\downarrow_\phi} t'$ where t is $\{SS \;\&\; [L, M^+ \mid L'] \;\&\; \{M \in \mathcal{I}, IK\}\}$ i.e., the right-hand side of Rule (3), and t' is the following state, found by Maude-NPA after one backwards narrowing step from the state pattern of Example 4:

$$\begin{aligned}&\{ :: r_2 :: [nil, -(pk(A, X)), +(pk(B, n(A, r_2))) \mid -(X \oplus n(A, r_2)), nil] \;\&\\ &\quad :: r_1 :: [nil, +(pk(A, n(B, r_1))), -(pk(B, Y)), +(Y \oplus n(B, r_1)) \mid nil] \;\&\\ &\quad SS \;\&\; \{(X \oplus n(A, r_2)) \in \mathcal{I}, IK\}\}\end{aligned}$$

The two key terms are $Y \oplus n(B, r_1)$ and $X \oplus n(A, r_2)$. Note that term $X \oplus n(A, r_2)$ appears in two positions in t', under symbols $-(\_)$ and $\_\in\mathcal{I}$, both required to be irreducible by ϕ. The singleton most general contextual unifier is $\sigma_1 = \{Y \mapsto X \oplus n(B, r_1) \oplus n(A, r_2)\}$, whereas the substitution $\sigma_2 = \{X \mapsto Y \oplus n(B, r_1) \oplus n(A, r_2)\}$ is *not* a valid contextual unifier: term $X \oplus n(A, r_2)$ is under the irreducibility condition of symbol $-(\_)$ and the substitution σ_2 would make it reducible, whereas term $Y \oplus n(B, r_1)$ is under symbol $+(\_)$, which does not have any irreducibility condition and the substitution σ_1 makes it reducible.

Contextual unification can be reduced to the simpler notion of *asymmetric unification*.

Definition 6 (Asymmetric Unification). *Given a decomposition (Σ, E, R) of an equational theory $(\Sigma, E \cup R)$, a substitution σ is an* asymmetric R,E-unifier *of a set P of* asymmetric equations $\{t_1 =_{\downarrow} t'_1, \ldots, t_n =_{\downarrow} t'_n\}$ *iff for every asymmetric equation $t_i =_{\downarrow} t'_i$ in P, σ is an $(E \cup R)$-unifier of the equation $t_i = t'_i$ and $(t'_i{\downarrow}_{R,E})\sigma$ is in R,E-normal form.*

A set of substitutions Ω is a complete set of asymmetric R,E-unifiers *of P iff: (i) every member of Ω is an asymmetric R,E-unifier of P, and (ii) for every asymmetric R,E-unifier θ of P there exists a $\sigma \in \Omega$ such that $\sigma \sqsupseteq_E \theta$ (over $Var(P)$).*

A special-purpose asymmetric unification algorithm for exclusive-or has been developed for this paper and is used in the experiments reported in Section 5. A detailed discussion of this algorithm will be presented elsewhere. The reduction of contextual unification to the simpler asymmetric unification is provided by the following lemma.

Lemma 1. *Given a contextual rewrite theory (Σ, E, R, T, ϕ) and a set of contextual equations $P = \{t_1 =_{\downarrow_\phi} t'_1, \ldots, t_n =_{\downarrow_\phi} t'_n\}$, σ is a contextual R, E-unifier of P iff there is a substitution θ such that θ is an asymmetric R, E-unifier of $\Gamma(P)$ and $\sigma =_E \theta|_{Var(P)}$, where*

$$\begin{aligned}\Gamma(P) =& \{t_i =_{\downarrow} X, t'_i =_{\downarrow} X \mid t_i =_{\downarrow_\phi} t'_i \in P, X \text{ fresh variable}\} \cup \\ & \{t'_i|_{p.j} =_{\downarrow} t'_i|_{p.j} \mid t_i =_{\downarrow_\phi} t'_i \in P, f \in \Sigma, p \in Pos_f(t'_i), j \in \phi(f)\}\end{aligned}$$

Using a contextual unification algorithm, we can modify the standard notion of narrowing so that it uses contextual unification to solve symbolic contextual reachability goals. Note that the following definition differs from Definition 2 only in using contextual unification $CSU_{R,E,\phi}(l =_{\downarrow_\phi} t|_p)$ instead of regular unification $CSU_{R \cup E}(l = t|_p)$ and and carrying a set of irreducible terms Π passed to the contextual unification algorithm, where Π is the set of irreducible terms that have been computed earlier in the narrowing sequence.

Definition 7 (Contextual Narrowing modulo R, E). *Given a contextual rewrite theory (Σ, E, R, T, ϕ), the contextual narrowing relation modulo R, E on pairs $\langle t, \Pi \rangle$ for t a term and Π a set of irreducible terms is defined as $\langle t, \Pi \rangle \overset{\sigma}{\rightsquigarrow}_{T,R,E,\phi} \langle t', \sigma(\Pi) \rangle$ (or $\overset{\sigma}{\rightsquigarrow}_\phi$ if T, R, E are understood) iff there is $p \in Pos_\Sigma(t)$, a rule $l \to r$ in T such that $Var(t) \cap (Var(l) \cup Var(r)) = \emptyset$, a substitution $\sigma \in CSU^V_{R,E,\phi}(P)$ for $P = \{l =_{\downarrow_\phi} t|_p\} \cup \{u =_{\downarrow_\phi} u \mid u \in \Pi\}$ and a set V of variables containing $Var(t)$, $Var(l)$, and $Var(r)$, and $t' = \sigma(t[r]_p)$.*

The essential equivalence between contextual reachability analysis and standard narrowing-based reachability analysis is proved as follows: given a standard goal $t \overset{?}{\to}^*_{T,R \cup E} t'$, *any* solution to it can be computed by contextual narrowing $\rightsquigarrow_{T,R,E,\phi}$ under some extra conditions involving *variants*. Let us motivate the issues involved by an example.

Example 7. Let us consider the state pattern shown in Example 4 with an extra requirement that the intruder learns $n(A, r_2)$:

$$\{ :: r_2 :: [nil, -(pk(A, X)), +(pk(B, n(A, r_2))), -(X \oplus n(A, r_2)) \mid nil] \;\&$$
$$:: r_1 :: [nil, +(pk(A, n(B, r_1))), -(pk(B, Y)), +(Y \oplus n(B, r_1)) \mid nil] \;\&$$
$$SS \;\&\; \{n(A, r_2) \in \mathcal{I}, IK\}\}$$

This attack pattern should be possible in Maude-NPA by just applying the substitution $X \mapsto 0$, where 0 is the identity symbol of $\oplus$. However, the term $X \oplus n(A, r_2)$ becomes reducible under such substitution and the attack would not be reachable because of our irreducibility condition on $X \oplus n(A, r_2)$. To solve this problem, the key idea is that the pattern $X \oplus n(A, r_2)$ should be replaced by its *variants* before each contextual narrowing step, i.e., by the possible instance patterns of it which are irreducible, namely: (i) the pattern $X \oplus n(A, r_2)$ itself, (ii) the pattern Y, which is the normal form after applying substitution $X \mapsto Y \oplus n(A, r_2)$, (iii) the pattern 0, which is the normal form after applying substitution $X \mapsto n(A, r_2)$, and (iv) the pattern $n(A, r_2)$, which is the normal form after applying substitution $X \mapsto 0$. Only after replacement of the original term by these variants, can we impose the irreducibility conditions for reducing the search space. That is, for contextual reachability analysis, we need to first compute what we call the ϕ, R, E-variants of a term.

Definition 8 (ϕ, R, E-variants). *Given a contextual rewrite theory (Σ, E, R, T, ϕ), the set of R,E,ϕ-variants of a pair $\langle t, \Pi \rangle$ for t a term and Π a set of irreducible terms is defined as $[\![\langle t, \Pi \rangle]\!]^{\phi}_{R,E} = \{(\sigma(t)[v_1, \ldots, v_n]_{p_1,\ldots,p_n}, \sigma) \mid (g(v_1, \ldots, v_n), \sigma) \in [\![g(t|_{p_1}, \ldots, t|_{p_n})]\!]_{R,E} \wedge \forall u \in \Pi : \sigma(u)$ is ϕ, R, E-irreducible$\}$ where $\phi(t) = \{p_1, \ldots, p_n\}$ and g is an auxiliary function symbol not appearing in R and E. For readability, we write $\langle t, \Pi \rangle \twoheadrightarrow^{\theta}_{R,E} \langle w, \overline{\Pi} \rangle$ to denote that $(w, \theta) \in [\![\langle t, \Pi \rangle]\!]^{\phi}_{R,E}$ and $\overline{\Pi} = \theta(\Pi) \cup \{w\}$.*

Example 8. Let us consider the state t' shown in Example 6:

$$\{ :: r_2 :: [nil, -(pk(A, X)), +(pk(B, n(A, r_2))) \mid -(X \oplus n(A, r_2)), nil] \;\&$$
$$:: r_1 :: [nil, +(pk(A, n(B, r_1))), -(pk(B, Y)), +(Y \oplus n(B, r_1)) \mid nil] \;\&$$
$$SS \;\&\; \{(X \oplus n(A, r_2)) \in \mathcal{I}, IK\}\}$$

We generate the four variants associated to $X \oplus n(A, r_2)$ in subterms rooted by $-(\_)$ and $\_ \in \mathcal{I}$, since these are the symbols with irreducibility constraints: (i) the original one but with the assumption that X will never contain either $n(A, r_2)$ or 0, (ii) the pattern $n(A, r_2)$ where $X \oplus n(A, r_2)$ has been collapsed into the nonce, (iii) the pattern Z where $X \oplus n(A, r_2)$ has been collapsed into a new variable Z by assuming $X \mapsto Z \oplus n(A, r_2)$, and (iv) the term 0 where $X \oplus n(A, r_2)$ has been collapsed into 0 by assuming $X \mapsto n(A, r_2)$:

$$\{ :: r_2 :: [nil, -(pk(A, X)), +(pk(B, n(A, r_2))) \mid -(X \oplus n(A, r_2)), nil] \;\&$$
$$:: r_1 :: [nil, +(pk(A, n(B, r_1))), -(pk(B, Y)), +(Y \oplus n(B, r_1)) \mid nil] \;\&$$
$$SS \;\&\; \{(X \oplus n(A, r_2)) \in \mathcal{I}, IK\}\}$$

$$\{ :: r_2 :: [nil, -(pk(A,0)), +(pk(B, n(A,r_2))) \mid -(n(A,r_2)), nil] \;\&$$
$$:: r_1 :: [nil, +(pk(A, n(B,r_1))), -(pk(B,Y)), +(Y \oplus n(B,r_1)) \mid nil] \;\&$$
$$SS \;\&\; \{n(A,r_2) \in \mathcal{I}, IK\}\}$$

$$\{ :: r_2 :: [nil, -(pk(A, Z \oplus n(A,r_2))), +(pk(B, n(A,r_2))) \mid -(Z), nil] \;\&$$
$$:: r_1 :: [nil, +(pk(A, n(B,r_1))), -(pk(B,Y)), +(Y \oplus n(B,r_1)) \mid nil] \;\&$$
$$SS \;\&\; \{Z \in \mathcal{I}, IK\}\}$$

$$\{ :: r_2 :: [nil, -(pk(A, n(A,r_2))), +(pk(B, n(A,r_2))) \mid -(0), nil] \;\&$$
$$:: r_1 :: [nil, +(pk(A, n(B,r_1))), -(pk(B,Y)), +(Y \oplus n(B,r_1)) \mid nil] \;\&$$
$$SS \;\&\; \{0 \in \mathcal{I}, IK\}\}$$

The reader can check that only the variants of the terms in the intruder knowledge (which are indeed coming from messages of the form $-(M)$) are generated.

The key idea to achieve the desired semantic equivalence between contextual narrowing and ordinary narrowing is to precede each contextual narrowing step by a ϕ-variant computation step.

Theorem 1 (Contextual Soundness and Completeness). *Given a contextual rewrite theory (Σ, E, R, T, ϕ), a reachability goal $t \stackrel{?}{\rightarrow}^* t'$, and a solution σ of it, there are a set of terms $u_1, \ldots, u_n, w_1, \ldots, w_{n+1}, t''$ and a set of substitutions $\theta_1, \ldots, \theta_{n+1}, \theta'_1, \ldots, \theta'_{n+1}$ such that*

$$\begin{array}{lll}
\langle t, \Pi_0 \rangle \rightarrow\!\!\!\!\rightarrow^{\theta_1}_{R,E} \langle w_1, \Pi_1 \rangle & & \stackrel{\theta'_1}{\rightsquigarrow}_{T,R,E,\phi} \langle u_1, \overline{\Pi_1} \rangle \\
\rightarrow\!\!\!\!\rightarrow^{\theta_2}_{R,E} \langle w_2, \Pi_2 \rangle & & \stackrel{\theta'_2}{\rightsquigarrow}_{T,R,E,\phi} \langle u_2, \overline{\Pi_2} \rangle \\
\quad\vdots & & \\
\rightarrow\!\!\!\!\rightarrow^{\theta_n}_{R,E} \langle w_n, \Pi_n \rangle & & \stackrel{\theta'_n}{\rightsquigarrow}_{T,R,E,\phi} \langle u_n, \overline{\Pi_n} \rangle \\
\rightarrow\!\!\!\!\rightarrow^{\theta_{n+1}}_{R,E} \langle w_{n+1}, \Pi_{n+1} \rangle & & \stackrel{\theta'_{n+1}}{\rightsquigarrow}_{T,R,E,\phi} \langle t'', \overline{\Pi_{n+1}} \rangle
\end{array}$$

and also: (i) $\Pi_0 = \emptyset$, $\Pi_1 = \{w_1\}$, $\overline{\Pi_1} = \theta'_1(\Pi_1)$, $\Pi_2 = \theta_2(\overline{\Pi_1}) \cup \{w_2\}$, $\overline{\Pi_2} = \theta'_2(\Pi_2)$, ..., $\Pi_{n+1} = \overline{\Pi_n} \cup \{w_{n+1}\}$, $\overline{\Pi_{n+1}} = \theta'_{n+1}(\Pi_{n+1})$, (ii) for each $i \in \{1, \ldots, n+1\}$, the term $w_i \theta'_i \theta_{i+1} \theta'_{i+1} \cdots \theta_{n+1} \theta'_{n+1}$ is ϕ, R, E-irreducible, (iii) there is a substitution τ such that $\sigma =_E \theta_1 \theta'_1 \theta_2 \theta'_2 \cdots \theta_{n+1} \theta'_{n+1} \tau$, and (iv) $t' =_E t'' \tau$.

Conversely, any substitution σ for which there is a sequence as above satisfying conditions (i)-(iv) is a solution of $t \stackrel{?}{\rightarrow}^* t'$.

Example 9. Continuing Example 8, we have four state patterns after variant generation. Contextual narrowing follows from the first variant state pattern as described in Example 10 below. The second variant state pattern will lead to an initial state where the intruder provides message $pk(A, 0)$ and the vertical bar of Bob's strand is never touched. And the third and the fourth variant state patterns will be discarded by Maude-NPA, since they do not satisfy the syntactic check explained in the Introduction discarding states sending a nonce before it is

generated. The state space reduction achieved in Maude-NPA is huge by using the irreducibility conditions on symbols $-(\_)$ and $\_\in\mathcal{I}$ and other state space reduction techniques based on such conditions (we further discuss experiments on this topic in Section 5).

Condition (ii) in Theorem 1 for terms w_i to be (ϕ-)irreducible after substitution application ensures that variants are not computed more than once for each irreducible subterm in term t or irreducible subterms introduced by right-hand sides of rules. This is very important to further reduce the search space.

Example 10. Let us consider the state t' shown in Example 6. After several variant generation and contextual narrowing steps using the reversed form of rewrite rules (1)–(3), the following state is found

$$\begin{aligned}
&\{ :: r_2 :: [nil, -(pk(A,X)), +(pk(B,n(A,r_2))) \mid -(X \oplus n(A,r_2)), nil] \ \&\\
&\quad :: r_1 :: [nil, +(pk(A,n(B,r_1))) \mid -(pk(B, X \oplus n(A,r_2) \oplus n(B,r_1))),\\
&\qquad\qquad\qquad\qquad +(X \oplus n(A,r_2) \oplus n(B,r_1) \oplus n(B,r_1), nil] \ \&\\
&\quad SS \ \& \ \{ \ pk(B, X \oplus n(A,r_2) \oplus n(B,r_1)) \in \mathcal{I}, \ (X \oplus n(A,r_2)) \notin \mathcal{I}, \ IK \ \}\}
\end{aligned}$$

We can check that there is no contextual unifier that allows terms $pk(B, n(A, r_2))$ and $pk(B, X \oplus n(A, r_2) \oplus n(B, r_1))$ to be unifiable according to the reversed form of rewrite rule (3), since the second term is under a symbol with irreducibility restrictions and the substitution $X \mapsto n(B, r_1)$ would make it reducible.

However, another protocol session can be used, since the term $pk(B, X \oplus n(A, r_2) \oplus n(B, r_1))$ can be unified with term $pk(B, n(A', r_2'))$ coming from another session, using the contextual unifier $X \mapsto n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1)$. The resulting state is as follows

$$\begin{aligned}
&\{ :: r_2' :: [nil, -(pk(A', X')) \mid +(pk(B, n(A', r_2'))), -(X' \oplus n(A', r_2')), nil] \ \&\\
&\quad :: r_2 :: [nil, \ -(pk(A, n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1))),\\
&\qquad\qquad +(pk(B, n(A, r_2))) \mid\\
&\qquad\qquad -(n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1) \oplus n(A, r_2)), nil] \ \&\\
&\quad :: r_1 :: [nil, \ +(pk(A, n(B, r_1))) \mid\\
&\qquad\qquad -(pk(B, n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1) \oplus n(A, r_2) \oplus n(B, r_1))),\\
&\qquad\qquad +(n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1) \oplus n(A, r_2) \oplus n(B, r_1) \oplus n(B, r_1), nil] \ \&\\
&\quad SS \ \& \ \{ \ pk(B, n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1) \oplus n(A, r_2) \oplus n(B, r_1)) \notin \mathcal{I},\\
&\qquad (n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1) \oplus n(A, r_2)) \notin \mathcal{I}, \ IK \ \}\}
\end{aligned}$$

However, although the two contextual narrowing steps have computed contextual unifiers, the combination of both unifiers does not satisfy the irreducibility conditions of the original term $-(X \oplus n(A, r_2))$, since now it is reducible, i.e., the term $-(n(A', r_2') \oplus n(A, r_2) \oplus n(B, r_1) \oplus n(A, r_2))$ is reducible. Therefore, this narrowing sequence is discarded, since it does not fulfill the conditions for solutions of contextual reachability goals given in Theorem 1, further reducing search.

states/seconds	1 step	2 steps	3 steps	4 steps	5 steps
RP - Standard	2/0.08	5/0.16	13/0.86	49/3.09	267/17.41
RP - Contextual	1/0.03	45/1.08	114/2.26	1175/37.25	13906/4144.30
WEPP - Standard	5/0.09	9/0.42	26/1.27	106/5.80	503/ 34.76
WEPP - Contextual	4/0.05	9/0.12	26/0.64	257/144.65	2454/612.08
TMN - Standard	5/0.11	15/ 0.55	99/3.82	469/ 25.68	timeout
TMN - Contextual	4/0.06	24/0.53	174/3.63	1079/170.29	9737/1372.55

Table 1. Experiments with standard reachability analysis using regular XOR unification algorithm vs contextual reachability analysis using asymmetric XOR unification algorithm. A pair n/t means: n = number of states, and t = time in seconds.

5 Experiments

We have performed several experiments to compare the contextual symbolic reachability approach presented in this paper with other approaches. We have used three protocols using exclusive-or: (i) the running protocol (RP) of Example 1, (ii) the Wired Equivalent Privacy Protocol (WEPP) of [1], and (iii) the TMN protocol of [31, 25]. For all three protocols, we are able to find the associated attacks in Table 2 below. We have run the experiments in this Section in an Intel Xeon machine with 4 cores and 24GB of memory, using Maude 2.7.

In Table 1, we compare the standard reachability analysis of Section 3, which uses the XOR unification algorithm developed in [24], and the contextual reachability analysis of Section 4, which uses the asymmetric XOR unification algorithm developed for this paper. A detailed discussion of this asymmetric XOR unification algorithm will be presented elsewhere. We show the number of states generated from one level to the next one of the backwards reachability tree with the indicated number of steps as the maximum depth. We also include the execution time from one level to the next one. We write "timeout" when the tool did not finish within a time interval of two hours.

As shown in Table 1, contextual reachability analysis is not better than the standard reachability analysis because of variant generation, which creates many more states than may be necessary for rule application. However, although typically many more states are created, the use of variants and irreducibility constraints is crucial (as explained in the Introduction) for further optimizations of the search space, as shown in Table 2, which shows that contextual reachability analysis enables several Maude-NPA optimizations, including grammars (see [16, 18] for details) and drastically reduces the search space.

Table 2 shows that, although, due to the extra computations needed for the optimization, the execution time without optimization is sometimes better than with optimizations, this only happens up to Step 3. The important point is that from Step 2 on, the total number of states is *drastically reduced* when optimizations are added (the only exception at Step 1 is RP, due to some differences on how variants are generated). In fact, the crucial point is not just the great reduction in the number of states, but the *finiteness of the analysis* for all the examples with optimization, whereas no such finiteness is even theoretically possible without optimizations. This is particularly important when an attack does *not* exist, since then finiteness of the analysis *proves* that the protocol is secure

states/seconds	1 step	2 steps	3 steps	4 steps	5 steps	Finite Analysis?
RP - w/o Opt.	1/0.03	45/1.08	114/2.26	1175/37.25	13906/4144.30	No, timeout with 6 steps
RP - with Opt.	4/0.59	7/0.59	7/1.92	7/1.89	7/3.02	Yes, at step 10
WEPP - w/o Opt.	4/0.05	9/0.12	26/0.64	257/144.65	2454/612.08	No, timeout with 7 steps
WEPP - with Opt.	2/0.36	2/0.20	1/0.80	2/1.42	1/0.03	Yes at step 5
TMN - w/o Opt.	4/0.06	24/0.53	174/3.63	1079/170.29	9737/1372.55	No, timeout with 7 steps
TMN - with Opt.	3/0.42	6/9.85	9/1.78	9/4.43	8/3.20	Yes, at step 21

Table 2. Experiments for contextual reachability analysis using asymmetric XOR unification algorithm with and without optimizations

against such an attack. Therefore, the above performance results validate experimentally the main thesis of this paper, namely that: (i) support of irreducibility conditions in symbolic reachability is essential for *effective* protocol analysis, since crucial optimizations depend on such conditions; and (ii) contextual reachability analysis supports irreducibility conditions in a sound and complete way and makes such optimizations possible.

The integration of this framework into Maude-NPA is still under testing and optimization, and further work is needed to increase performance. Indeed, the current experiments have been performed with a version of the contextual narrowing simpler than the conditions of Theorem 1 (irreducibility constraints on Π are not enforced), but is still valid for the benchmarked protocols, i.e., in these protocols, each strand contains only one expression using the xor operator, and thus Π remains irreducible by default.

6 Conclusions and Future Directions

We are only at the beginning of exploring contextual symbolic reachability analysis as a general approach, and there are many paths that can be followed. One is exploring the different types of irreducibility constraints and their effect on efficiency. It would appear that an approach that requires fewer constraints would be more efficient than one that applies more; e.g. that modifying a tool such as OFMC that requires constraints on both sent and received messages to use constraints only on input messages, as does Maude-NPA, would lead to reduced state space size and greater efficiency, but this needs to be verified.

Using one-sided constraints also potentially allows us to gain greater efficiency through special-purpose asymmetric unification algorithms. We are now investigating this with respect to asymmetric exclusive-or unification, and plan to develop and investigate other such algorithms in the future. Asymmetric unification is a subject about which currently very little is known; as it is explored further, we expect to find out a lot more about it and how it can be optimized.

Finally, we believe that cryptographic protocol analysis is not the only potential application for symbolic contextual reachability analysis. Indeed, it should be applicable to any state exploration problem in which symbolic states obey equational theories. Future work in this area should involve an investigation of these other problems and the ways in which contextual symbolic reachability analysis could be applied to them.

References

1. *IEEE 802.11 Local and Metropolitan Area Networks: Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications*. 1999.
2. Martín Abadi and Véronique Cortier. Deciding knowledge in security protocols under equational theories. *Theor. Comput. Sci.*, 367(1-2):2–32, 2006.
3. Myrto Arapinis, Sergiu Bursuc, and Mark Ryan. Privacy supporting cloud computing: Confichair, a case study. In Pierpaolo Degano and Joshua D. Guttman, editors, *Principles of Security and Trust - First International Conference, POST 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012, Proceedings*, volume 7215 of *Lecture Notes in Computer Science*, pages 89–108. Springer, 2012.
4. David Basin, Sebastian Mödersheim, and Luca Viganò. An on-the-fly model-checker for security protocol analysis. In *In Proceedings of Esorics'03, LNCS 2808*, pages 253–270. Springer-Verlag, 2003.
5. Mathieu Baudet, Véronique Cortier, and Stéphanie Delaune. YAPA: A generic tool for computing intruder knowledge. In *Proc. RTA'09*, volume 5595 of *LNCS*, pages 148–163, Brasília, Brazil, June-July 2009. Springer.
6. Bruno Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *CSFW*, pages 82–96. IEEE Computer Society, 2001.
7. Bruno Blanchet. Using horn clauses for analyzing security protocols. In Veronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*. IOS Press, 2011.
8. Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. *J. Log. Algebr. Program.*, 75(1):3–51, 2008.
9. Ştefan Ciobâcă, Stéphanie Delaune, and Steve Kremer. Computing knowledge in security protocols under convergent equational theories. In *Proc. CADE'09*, LNAI, pages 355–370, Montreal, Canada, 2009. Springer.
10. Hubert Comon-Lundh and Stéphanie Delaune. The finite variant property: How to get rid of some algebraic properties. In Jürgen Giesl, editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2005.
11. Hubert Comon-Lundh, Stéphanie Delaune, and Jonathan Millen. Constraint solving techniques and enriching the model with equational theories. In Véronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, volume 5 of *Cryptology and Information Security Series*, pages 35–61. IOS Press, 2011.
12. Hubert Comon-Lundh and Vitaly Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *LICS*, pages 271–280. IEEE Computer Society, 2003.
13. Ştefan Ciobâcă. Knowledge in security protocols.
14. Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols (extended abstract). In *FOCS*, pages 350–357, 1981.
15. Santiago Escobar, Catherine Meadows, and José Meseguer. A rewriting-based inference system for the NRL protocol analyzer and its meta-logical properties. *Theoretical Computer Science*, 367(1-2):162–202, 2006.
16. Santiago Escobar, Catherine Meadows, and José Meseguer. State space reduction in the Maude-NRL protocol analyzer. In Sushil Jajodia and Javier López, editors,

Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings, volume 5283 of *Lecture Notes in Computer Science*, pages 548–562. Springer, 2008.
17. Santiago Escobar, Catherine Meadows, and José Meseguer. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In *Foundations of Security Analysis and Design V, FOSAD 2007/2008/2009 Tutorial Lectures*, LNCS vol. 5705, pages 1–50. Springer, 2009.
18. Santiago Escobar, Catherine Meadows, José Meseguer, and Sonia Santiago. State space reduction in the maude-nrl protocol analyzer. *Information and Computation*, 2012. In Press.
19. Santiago Escobar, Ralf Sasse, and José Meseguer. Folding variant narrowing and optimal variant termination. *J. Log. Algebr. Program.*, 2012. In Press.
20. 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.
21. Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. Comput.*, 15(4):1155–1194, 1986.
22. Ralf Küsters and Tomasz Truderung. Using ProVerif to analyze protocols with Diffie-Hellman exponentiation. In *CSF*, pages 157–171. IEEE Computer Society, 2009.
23. Ralf Küsters and Tomasz Truderung. Reducing protocol analysis with xor to the xor-free case in the horn theory based approach. *Journal of Automated Reasoning*, 46(3-4):325–352, 2011.
24. Zhiqiang Liu and Christopher Lynch. Efficient general unification for xor with homomorphism. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *Lecture Notes in Computer Science*, pages 407–421. Springer, 2011.
25. Gavin Lowe and Bill Roscoe. Using csp to detect errors in the tmn protocol. *IEEE Transactions on Software Engineering*, 23:659–669, 1997.
26. S. Lucas. Context-sensitive computations in functional and functional logic programs. *J. Functl. and Log. Progr.*, 1(4):446–453, 1998.
27. José Meseguer. Conditional rewriting logic as a united model of concurrency. *Theor. Comput. Sci.*, 96(1):73–155, 1992.
28. José Meseguer and Prasanna Thati. Symbolic reachability analysis using narrowing and its application to verification of cryptographic protocols. *Higher-Order and Symbolic Computation*, 20(1-2):123–160, 2007.
29. Sebastian Mödersheim. *Models and methods for the automated analysis of security protocols*. PhD thesis, ETH Zurich, 2007.
30. Sebastian Mödersheim, Luca Viganò, and David A. Basin. Constraint differentiation: Search-space reduction for the constraint-based analysis of security protocols. *Journal of Computer Security*, 18(4):575–618, 2010.
31. Makoto Tatebayashi, Natsume Matsuzaki, and David Newman. Key distribution protocol for digital mobile communication systems. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO'89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 324–334. Springer Berlin / Heidelberg, 1990.
32. TeReSe, editor. *Term Rewriting Systems*. Cambridge University Press, Cambridge, 2003.
33. Patrick Viry. Equational rules for rewriting logic. *Theor. Comput. Sci.*, 285(2):487–517, 2002.
34. Hantao Zhang and Jean-Luc Remy. Contextual rewriting. In Jean-Pierre Jouannaud, editor, *RTA*, volume 202 of *Lecture Notes in Computer Science*, pages 46–62. Springer, 1985.