A DEXPTIME-complete Dolev-Yao theory with distributive encryption ⋆

A. Baskar[1]⋆⋆, R. Ramanujam[2]⋆⋆⋆, and S.P. Suresh[1]

[1] Chennai Mathematical Institute, Chennai, India. {abaskar, spsuresh}@cmi.ac.in
[2] Institute of Mathematical Sciences Chennai, India. jam@imsc.res.in

Abstract. In the context of modelling cryptographic tools like blind signatures and homomorphic encryption, the Dolev-Yao model is typically extended with an operator over which encryption is distributive. We consider one such theory which lacks any obvious locality property and show that its derivability problem is hard: in fact, it is DEXPTIME-complete. The result holds also when blind pairing is associative. The lower bound contrasts with PTIME decidability for restricted theories of blind signatures, and the upper bound with non-elementary decidability for abelian group operators with distributive encryption.

1 Introduction

Dolev-Yao style term models [DY83] for cryptographic protocols (the so-called "symbolic models") use a term algebra containing operations like pairing, encryption, signatures, hash functions, and nonces to build terms that are sent as messages in the protocol. The adversary against a protocol is modeled as a powerful network, which is only restricted in the way in which messages may be derived from the ones sent by the "honest" principals. Since these models are used for algorithmic analysis, the following *term derivability problem* is of basic interest: given a finite set of terms X and a term t, is there a way for the adversary to derive t from X?

In this paper, we study a security problem for a set of cryptographic primitives in an extension of the Dolev-Yao model which includes a **blind pairing** that commutes over encryption. That is, we can "push" an encryption by key k inside $[t, t']$ and get $[\{t\}_k, \{t'\}_k]$. We can also form a blind pair $[t, t']$ from t and t', and extract t' or t from $[t, t']$, provided we have the other part of the blind pair. We show that the existence of a passive attack (that is, by an attacker who cannot forge messages) is decidable in exponential time.

Though the blind pairing constructor finds natural use in the Dolev-Yao modelling of electronic voting protocols [FOO92], more restricted uses of blind pairing may well suffice in many applications. What then can be interesting about such a result, in a

⋆ We thank the anonymous referees for many helpful comments that helped improve the presentation. Please see http://www.cmi.ac.in/~spsuresh/pdffiles/mfcs2010-tr.pdf for a technical report version with detailed proofs.
⋆⋆ Supported by the Council of Scientific and Industrial Research (CSIR), India.
⋆⋆⋆ I thank NIAS (http://www.nias.knaw.nl) for a Lorentz Fellowship from February to June 2010.

framework with a fixed set of primitives, a weak attacker model and offering an algorithm with such high complexity? Perhaps the fact that the algorithm is presented as an automaton construction; but then it should be noted that the original Dolev-Yao paper used an automaton construction (indeed, a deterministic one) to solve the secrecy problem for a class of protocols called ping-pong protocols.

Indeed the result is of a technical nature and relates to the theoretician's toolkit in the study of Dolev-Yao models. The standard strategy to prove the derivability problem decidable is to prove a so-called **locality property** [RT03,CS03], that if t is derivable from X, then there is a special kind of derivation (a **normal derivation**) π such that every term occurring in π comes from $S(X \cup \{t\})$, where S is a function mapping a finite set of terms to *another finite set of terms*. Typically S is the **subterm function** *st*, but in many cases it is a minor variant. The locality property is used to provide a decision procedure for the derivability problem (which is typically a PTIME algorithm).

As we will show later, our system does not have an obvious locality property, and so we cannot follow the standard route to decidability. In fact, we can construct a set of terms X and a term t such that the set of terms occurring in any derivation of t from X is *exponential* in the size of $X \cup \{t\}$. This suggests that it would be difficult to define a function S of the kind mentioned above such that any term occurring in a normal derivation of t from X comes from $S(X \cup \{t\})$.

The first technical contribution of this paper is to show one way of working around this difficulty. We prove a *weak locality property*: we define a function S which maps every finite set of terms X to an *infinite* set of terms $S(X)$. We then prove that all terms occurring in a normal derivation of t from X are from $S(X \cup \{t\})$, and that the set of terms in $S(X \cup \{t\})$ that are derivable from X is regular. This facilitates an automaton construction and yields a decision procedure for checking whether t is derivable from X.

The second technical contribution is to settle the complexity of the derivability problem by proving a DEXPTIME-hardness result by reduction from the backwards reachability problem for alternating pushdown systems. While many lower bound results for the *active* intruder deduction problem exist in the literature, under various settings, this is one of the few lower bound results for the *passive* intruder deduction problem.

The third technical contribution of the paper is the use (in our decision procedure) of the alternating automaton saturation technique in itself (similar to the one in [BEM97]). In fact, the lower bound reduction shows the close connections to alternating pushdown systems, and so it is no surprise that automaton saturation, one of the standard tools for analysis of pushdown systems, is used for our upper bound proofs. This should also be viewed in the context of the use of tree automata for protocol verification, specifically the idea of representing (an over-approximation of) the set of deducible terms using tree automata. This has been explored in a number of papers [Mon99,Gou00,GK99]. Applications of two-way alternating tree automata to security protocol verification has been touched upon in [CDG+07]. The saturation technique that we use offers yet another tool that may be of use in other contexts.

Where does the high complexity of this problem originate from? It arises from the fact that blind pairing is distributive over encryption. This can be seen in the light of results on closely related constructors.

In [DKR09,BC06,CRZ05], a different way of modelling blind signatures is considered. Two operators, blind and unblind are used, with the following rules:

$$\mathsf{unblind}(\mathsf{blind}(m, r), r) = m$$
$$\mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(m, r), k), r) = \mathsf{sign}(m, k)$$

The restriction (as compared to distributive encryption) here is that the r in the above equations is an atomic term, typically a random number, and whenever a blind pair is signed, the signature gets pushed only to the first component and not the second. Because of this, the system enjoys a locality property, and the basic derivability problem is decidable in PTIME.

In earlier work in [BRS07], we proposed essentially the same system described in this paper, but we imposed a restriction that the second component of blind pairs are always of the form n or $\{n\}_k$ where n is an atomic term (or nonce). And the only rule that involves pushing an encryption inside a blind pair is the derivation of $[\{t\}_k, n]$ from $[t, \{n\}_{inv(k)}]$ and k. This restricted system also satisfies a locality property.

At the other end of the spectrum, a much more powerful system is considered in [LLT07]. They study an abelian group operator $+$ such that $\{t_1 + \cdots + t_n\}_k = \{t_1\}_k + \cdots + \{t_n\}_k$, i.e. encryption is homomorphic over $+$. They employ a very involved argument and prove the derivability problem in the general case to be decidable with a non-elementary upper bound. They also give a DEXPTIME algorithm in the case when the operator is xor, and a PTIME algorithm in the so-called binary case. The blind pair operator we consider has very different characteristics than xor, and the arguments in [LLT07] do not apply here.

2 Extension of the Dolev-Yao model with blind pairs

Assume a set of basic terms $\mathcal{N}$, which includes the set of keys $\mathcal{K}$. Let $inv(k)$ be a function on $\mathcal{K}$ such that $inv(inv(k)) = k$. The set of **terms** $\mathcal{T}$ is defined to be:

$$\mathcal{T} \ ::= \ m \mid (t_1, t_2) \mid [t_1, t_2] \mid \{t\}_k$$

where $m \in \mathcal{N}$, $k \in \mathcal{K}$, and t, t_1, and t_2 range over $\mathcal{T}$.

The set of **subterms** of t, $st(t)$, is the smallest $X \subseteq \mathcal{T}$ such that 1) $t \in X$, 2) if $(t, t') \in X$ or $[t, t'] \in X$, then $\{t, t'\} \subseteq X$, and 3) if $\{t\}_k \in X$ then $\{t, k\} \subseteq X$. $st(X)$ is defined to be $\bigcup_{t \in X} st(t)$. A **keyword** is an element of $\mathcal{K}^*$. Given a term t and a keyword $x = k_1 \cdots k_n$, $\{t\}_x = \{\cdots \{t\}_{k_1} \cdots\}_{k_n}$. If $x = \varepsilon$, $\{t\}_x$ is t itself.

For simplicity, we assume henceforth that all terms are **normal**. These are terms which do not contain a subterm of the form $\{[t_1, t_2]\}_k$. For a term t, we get its normal form $t\!\downarrow$ by "pushing encryptions over blind pairs, all the way inside." Formally, it is defined as follows: $m\!\downarrow = m$ for $m \in \mathcal{N}$; $(t_1, t_2)\!\downarrow = (t_1\!\downarrow, t_2\!\downarrow)$; $[t_1, t_2]\!\downarrow = [t_1\!\downarrow, t_2\!\downarrow]$; and

$$\{t\}_k\!\downarrow = \begin{cases} [\{t_1\}_k\!\downarrow, \{t_2\}_k\!\downarrow] & \text{if } t = [t_1, t_2] \\ \{t\!\downarrow\}_k & \text{otherwise} \end{cases}$$

Definition 1. *A* **derivation** *or a* **proof** π *of* t *from assumptions* X *is a tree whose nodes are labelled by terms, whose root is labelled* t*, whose leaves are instances of the Ax rule and labelled by terms from* X*, and whose internal nodes are instances of one of the analz-rules or synth-rules in Figure 1. We use* $X \vdash t$ *to also denote that there is a proof of* t *from* X*. For a set of terms* X*,* $clos(X) = \{t \mid X \vdash t\}$ *is the* **closure** *of* X*.*

analz-rules		$\dfrac{\{t\}_k\downarrow \quad inv(k)}{t}\ decrypt$	$\dfrac{(t_0,t_1)}{t_i}\ split_i$	$\dfrac{[t_0,t_1]\downarrow \quad t_i\downarrow}{t_{1-i}}\ blindsplit_i$
synth-rules	$\dfrac{}{t}\ Ax\ \ (t \in X)$	$\dfrac{t \quad k}{\{t\}_k\downarrow}\ encrypt$	$\dfrac{t_1 \quad t_2}{(t_1,t_2)}\ pair$	$\dfrac{t_1 \quad t_2}{[t_1,t_2]}\ blindpair$

Fig. 1. Proof system for normal terms (with assumptions from $X \subseteq \mathscr{T}$). In the *decrypt* rule, $\{t\}_k\downarrow$ is the **major premise** and k is the **minor premise**. In the $blindsplit_i$ rule, $[t_0,t_1]\downarrow$ is the **major premise** and t_i is the **minor premise**.

It is significant that the main premise of the *decrypt* rule is $\{t\}_k\downarrow$. This allows us to derive $[t,t']$ from $[\{t\}_k,\{t'\}_k]$ and $inv(k)$, for instance.

Definition 2. *The* **derivability problem** *(also called the* **passive intruder deduction problem***) is the following: given a finite set* $X \subseteq \mathscr{T}$ *and* $t \in \mathscr{T}$*, determine whether* $X \vdash t$*.*

As we mentioned in the introduction, the standard strategy to prove this problem decidable is to define a notion of **normal proofs**, show that every proof can be transformed to a normal proof, and prove a so-called **locality property**, that every term occurring in a normal proof of $X \vdash t$ comes from $S(X \cup \{t\})$, where $S : 2^{\mathscr{T}} \to 2^{\mathscr{T}}$ is a function mapping a finite set of terms to another finite set of terms. Typically S is the subterm function st, but in many cases it is a minor variant. This typically yields a PTIME algorithm for the derivability problem.

But there is no obvious locality property for the proof system considered here. For instance, to derive the term $\{a\}_k$ from $[a,b]$, $\{b\}_k$ and k, we necessarily need to go via the term $[\{a\}_k,\{b\}_k]$, which is not a subterm of either the premises or the conclusion. In fact, the structure of terms occurring in a proof of $X \vdash t$ can get very complex.

For example, one can code up some kind of a counter – a set X of $O(n)$ terms and another term t, each of size $O(1)$, with $X \vdash t$, but such that every proof of t from X has at least 2^n terms occurring in it. The reader can refer to [BRS10] for details.

3 Normal proofs

Even though our proof system lacks an obvious locality property, we can prove a weak locality property, which will help us derive a decision procedure for the derivability

problem. This section is devoted to a proof of the weak locality property (or **weak subterm property**).

We first define the notion of a normal proof. These are proofs got by applying the transformations of Figure 2 repeatedly. Any subproof that matches a pattern on the left column is meant to be replaced by the proof on the right column in the same row. The idea behind normalization is to perform applications of the *encrypt* and *decrypt* rules as early as possible in the proof.

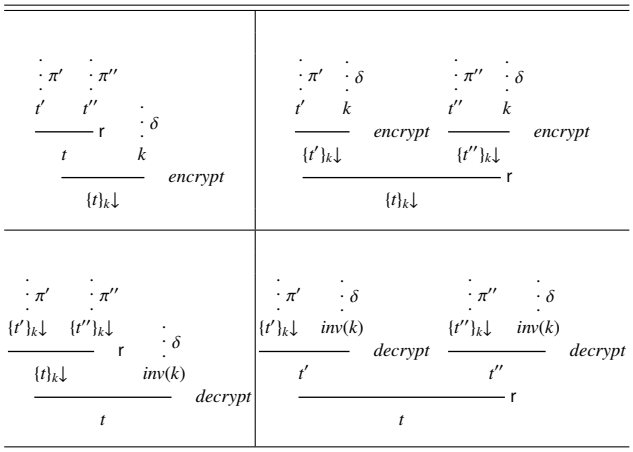

Fig. 2. The normalization rules. Rule r is meant to be either *blindpair* (in which case $t = [t', t'']$), or $blindsplit_0$ (in which case $t' = [t'', t]$), or $blindsplit_1$ (in which case $t' = [t, t'']$).

Definition 3. *A proof π of t from assumptions X is a* **minimal proof** *if t occurs only in the root of the proof.*

A proof π is a **normal proof** *if the following two conditions hold:*

1. *every subproof of π is minimal, and*
2. *the transformations in Figure 2 cannot be applied to π.*

Lemma 1. *Whenever $X \vdash t$, there is a normal proof of t from X.*

We now state the weak locality property for normal proofs. The standard locality property can be viewed as giving a bound on the "width" and encryption depth of terms occurring in a proof of $X \vdash t$. We prove a weaker property, where only the width of terms is bounded. So the set of terms occurring in any normal proof of $X \vdash t$ is got by encrypting terms (perhaps repeatedly) from a "core" set, using keys derivable from X. The core, it turns out, is $st(X \cup \{t\})$. For every $p \in st(X \cup \{t\})$, define $\mathscr{L}_p$ to be $\{x \in (st(X \cup \{t\}) \cap \mathscr{K})^* \mid X \vdash \{p\}_x\}$. We shall show in the next section that $\mathscr{L}_p$ is regular for each p.

We introduce a bit of notation first that will help us conveniently state the weak locality lemma. We say that a proof π of $X \vdash t$ is **purely synthetic** if:

– it ends in an application of the *Ax* or *blindpair* or *pair* rules, or
– it ends in an application of the *encrypt* rule and $t\downarrow$ is not a blind pair.

Lemma 2 (Weak locality property). *Let π be a normal proof of t from X, and let δ be a subproof of π with root labelled r. Then the following hold:*

1. *For every u occurring in δ, there is a term $p \in st(X \cup \{t\})$ and a keyword x such that $u = \{p\}_x$. Moreover, if δ is not a purely synthetic proof then $p \in st(X)$.*
2. *If the last rule of δ is decrypt or split with major premise r_1, then $r_1 \in st(X)$.*

The main difficulty is in coming up with the right statement. The proof itself is a standard induction on derivations, with an exhaustive case analysis, and is presented in full detail in [BRS10].

4 The automaton construction

We recall here some definitions relating to alternating pushdown systems and alternating automata (with ε-moves). The former will be needed for the lower bound argument in the next section, and the latter for the decision procedure to be presented here.

An **alternating pushdown system** (APDS) is a triple $\mathscr{P} = (P, \Gamma, \hookrightarrow)$, where P is a *finite set of control locations*, Γ is a *finite stack alphabet*, and $\hookrightarrow \subseteq P \times \Gamma^* \times 2^{(P \times \Gamma^*)}$ is a *finite set of transition rules*. We write transitions as $(a, x) \hookrightarrow \{(b_1, x_1), \ldots, (b_n, x_n)\}$. A *configuration* is a pair (a, x) where $a \in P$ and $x \in \Gamma^*$. Given a set of configurations C, a configuration (a, x), and $i \geq 0$, we say that $(a, x) \Rightarrow_{\mathscr{P},i} C$ iff:

– $(a, x) \in C$ and $i = 0$, or
– there is a transition $(a, y) \hookrightarrow \{(b_1, y_1), \ldots, (b_n, y_n)\}$ of $\mathscr{P}$, $z \in \Gamma^*$, and $i_1, \ldots, i_1 \geq 0$ such that $i = i_1 + \cdots + i_n + 1$ and $x = yz$ and for all $j \in \{1, \ldots, n\}$, $(b_j, y_j z) \Rightarrow_{\mathscr{P},i_j} C$.

We say that $(a, x) \Rightarrow_{\mathscr{P}} C$ iff $(a, x) \Rightarrow_{\mathscr{P},i} C$ for some $i \geq 0$.

An **alternating automaton** is an APDS $\mathscr{P} = (Q, \Sigma, \hookrightarrow)$ such that $\hookrightarrow \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times 2^{(Q \times \{\varepsilon\})}$. For $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, and $C \subseteq Q$, we use $q \overset{a}{\hookrightarrow} C$ to denote the fact that $(q, a, C \times \{\varepsilon\}) \in \hookrightarrow$. For ease of notation, we will also write $q \overset{a}{\hookrightarrow} q'$ to mean $q \overset{a}{\hookrightarrow} \{q'\}$. Given $C \subseteq Q$, and $x \in \Sigma^*$, we use the notation $q \overset{x}{\Rightarrow}_{\mathscr{P},i} C$ to mean that $(q, x) \Rightarrow_{\mathscr{P},i} C \times \{\varepsilon\}$. For $C = \{q_1, \ldots, q_m\}$ and $C' \subseteq Q$, we use the notation $C \overset{x}{\Rightarrow}_{\mathscr{P},i} C'$ to mean that for all $j \leq m$, there exists i_j such that $q_j \overset{x}{\Rightarrow}_{\mathscr{P},i_j} C'$, and $i = i_1 + \cdots + i_m$. We also say $q \overset{x}{\Rightarrow}_{\mathscr{P}} C$ and $C \overset{x}{\Rightarrow}_{\mathscr{P}} C'$ to mean that there is some i such that $q \overset{x}{\Rightarrow}_{\mathscr{P},i} C$ and $C \overset{x}{\Rightarrow}_{\mathscr{P},i} C'$, respectively.

We typically drop the superscript $\mathscr{P}$ if it is clear from the context which APDS is referred to.

Fix a finite set of terms X_0 and a term t_0. We let Y_0 denote $st(X_0 \cup \{t_0\})$ and $K_0 = Y_0 \cap \mathscr{K}$. In this section, we address the question of whether there exists a normal proof of t_0 from X_0. Lemma 2 provides a key to the solution – every term occurring in such a proof is of the form $\{p\}_x$ for $p \in Y_0$ and $x \in K_0^*$. Therefore it is easy to see that the different $\mathscr{L}_p$ (for $p \in Y_0$) satisfy the following equations (among others):

$kx \in \mathscr{L}_p$ iff $x \in \mathscr{L}_{\{p\}_k}$
if $x \in \mathscr{L}_p$ and $x \in \mathscr{L}_{p'}$ then $x \in \mathscr{L}_{[p,p']}$
if $x \in \mathscr{L}_p$ and $x \in \mathscr{L}_{[p,p']}$ then $x \in \mathscr{L}_{p'}$
if $x \in \mathscr{L}_{p'}$ and $x \in \mathscr{L}_{[p,p']}$ then $x \in \mathscr{L}_p$
if $x \in \mathscr{L}_p$ and $\varepsilon \in \mathscr{L}_k$ then $xk \in \mathscr{L}_p$
if $\varepsilon \in \mathscr{L}_{\{p\}_k}$ and $\varepsilon \in \mathscr{L}_{inv(k)}$ then $\varepsilon \in \mathscr{L}_p$

This immediately suggests the construction of an alternating automaton $\mathscr{A}$ such that for every $t \in Y_0$ and keyword x, $x \in \mathscr{L}_t$ if and only if there is a run of $\mathscr{A}$ on the word x from the state t to a designated "final state" f. Then checking whether $X \vdash t_0$ (or in other words, $\varepsilon \in \mathscr{L}_{t_0}$) is simply a matter of checking if there is a run of $\mathscr{A}$ on ε from the state t_0 to the state f.

The states of the automaton are terms from Y_0 and the transitions are a direct transcription of the above equations. For instance there is an edge $t \overset{k}{\hookrightarrow} \{t\}_k$, and there is an edge $t \overset{\varepsilon}{\hookrightarrow} \{[t, t'], t'\}$. In the construction, we wish every $x \in \mathscr{L}_t$ to be witnessed by a run $t \overset{x}{\Rightarrow} \{f\}$ (f is a designated final state). This forces us to apply a saturation construction. For instance, suppose that $kx \in \mathscr{L}_t$ and this fact is witnessed by a run $t \overset{kx}{\Rightarrow} \{f\}$ (at some stage of the automaton construction). It is also the case that $x \in \mathscr{L}_{\{t\}_k}$, and this ought to be witnessed by a run $\{t\}_k \overset{x}{\Rightarrow} \{f\}$. To achieve this, we introduce a new transition $\{t\}_k \overset{\varepsilon}{\hookrightarrow} C$ whenever $t \overset{k}{\Rightarrow} C$. In fact, it does not suffice to stop after revising the automaton once. The procedure has to be repeated till no more new transitions can be added.

Thus we define a sequence of alternating automata $\mathscr{A}_0, \mathscr{A}_1, \ldots, \mathscr{A}_i, \ldots$, each of which adds transitions to the previous one, as given by the definition in Figure 3.

We would like to emphasize that saturating an alternating automaton fits in very naturally with our problem. For example, $X \vdash m$ where $X = \{[\{t\}_k, m], t, k\}$. To detect this, we need to test if $m \overset{\varepsilon}{\hookrightarrow}_i \{f\}$ for some i. This test turns out to be true for $i = 4$, as witnessed by the following sequence of edges and paths. Other constructions like two-way automata do not seem immediately applicable to this situation.

$m \overset{\varepsilon}{\hookrightarrow}_0 \{[\{t\}_k, m], \{t\}_k\}$.
$t \overset{\varepsilon}{\hookrightarrow}_1 \{f\}$, $k \overset{\varepsilon}{\hookrightarrow}_1 \{f\}$, $[\{t\}_k, m] \overset{\varepsilon}{\hookrightarrow}_1 \{f\}$.
$f \overset{k}{\hookrightarrow}_2 \{f\}$, $t \overset{k}{\Rightarrow}_2 \{f\}$.
$\{t\}_k \overset{\varepsilon}{\hookrightarrow}_3 \{f\}$ (this is a crucial use of saturation), $m \overset{\varepsilon}{\Rightarrow}_3 \{f\}$.
$m \overset{\varepsilon}{\hookrightarrow}_4 \{f\}$.

The following lemma essentially shows that the saturation procedure terminates in exponential time.

Lemma 3. *1. For all $i \geq 0$ and all $a \in \Sigma \cup \{\varepsilon\}$, the relation $\overset{a}{\Rightarrow}_i$ is constructible from $\hookrightarrow_i$ in time $2^{O(d)}$, where $d = |Q|$.*
2. For all $i \geq 0$ and all $a \in \Sigma$, the relation $\overset{a}{\hookrightarrow}_{i+1}$ is constructible from $\Rightarrow_i$ in time $2^{O(d)}$.
3. There exists $d' \leq d^2 \cdot 2^d$ such that for all $i \geq d'$, $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, and $C \subseteq Q$, $q \overset{a}{\hookrightarrow}_i C$ if and only if $q \overset{a}{\hookrightarrow}_{d'} C$.

For each $i \geq 0$, $\mathscr{A}_i$ is given by $(Q, \Sigma, \hookrightarrow_i)$ where $Q = Y_0 \cup \{f\}$ ($f \notin Y_0$) and $\Sigma = K_0$. We define $\hookrightarrow_i$ by induction.

- $\hookrightarrow_0$ is the smallest subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ that satisfies the following:
  1. if $t \in Y_0, k \in K_0$ such that $\{t\}_k\downarrow \in Y_0$, then $t \overset{k}{\hookrightarrow}_0 \{\{t\}_k\downarrow\}$.
  2. if $t, t', t'' \in Y_0$ such that t is the conclusion of an instance of the *blindpair* or *blindsplit*$_i$ rules with premises t' and t'', then $t \overset{\varepsilon}{\hookrightarrow}_0 \{t', t''\}$.
- $\hookrightarrow_{i+1}$ is the smallest subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ such that:
  1. if $q \overset{a}{\Rightarrow}_i C$, then $q \overset{a}{\hookrightarrow}_{i+1} C$.
  2. if $\{t\}_k\downarrow \in Y_0$ and $t \overset{k}{\Rightarrow}_i C$, then $\{t\}_k\downarrow \overset{\varepsilon}{\hookrightarrow}_{i+1} C$.
  3. if $k \in K_0$ and $k \overset{\varepsilon}{\Rightarrow}_i \{f\}$, then $f \overset{k}{\hookrightarrow}_{i+1} \{f\}$.
  4. if $\Gamma \subseteq Y_0$, $t \in Y_0$, and if there is an instance r of one of the rules of Figure 1 (nullary, unary or binary) whose set of premises is (exactly) Γ and conclusion is t—note that Ax is a nullary rule, and hence this clause covers all $t \in X_0$—the following holds:

$$\text{if } u \overset{\varepsilon}{\Rightarrow}_i \{f\} \text{ for every } u \in \Gamma, \text{ then } t \overset{\varepsilon}{\hookrightarrow}_{i+1} \{f\}.$$

Fig. 3. The sequence of automata for analyzing $X_0 \vdash t_0$, with $Y_0 = st(X_0 \cup \{t_0\})$ and $K_0 = Y_0 \cap \mathscr{K}$. We use $\hookrightarrow_i$ for $\hookrightarrow_{\mathscr{A}_i}$ and $\Rightarrow_i$ for $\Rightarrow_{\mathscr{A}_i}$.

We now present theorems that assert the correctness of the above construction. It is **sound**, i.e. none of the automata accept an x starting from r where $\{r\}_x$ is not derivable from X_0; and that it is **complete**, i.e. whenever $\{r\}_x$ is derived from X_0, one of the $\mathscr{A}_i$'s has an accepting run over x starting from r. To simplify the statement and proof in the rest of this section, we first introduce the following notations:

- for $X \subseteq \mathscr{T}$ and keyword x, we use $X \vdash x$ to mean that $X \vdash k$ for every k occurring in x.
- for $C \subseteq Y_0$ and keyword y, $\{C\}_y = \{\{t\}_y\downarrow \mid t \in C\}$.
- for $q \in Q, C \subseteq Q$, $q \overset{x}{\Rightarrow}_{i,d} C$ iff $q \overset{x}{\Rightarrow}_{\mathscr{A}_i,d} C$.
- for $C, C' \subseteq Q$, $C \overset{x}{\Rightarrow}_{i,d} C'$ iff $C \overset{x}{\Rightarrow}_{\mathscr{A}_i,d} C'$.

Theorem 1 (Soundness). *For any i, any $t \in Y_0$, and any keyword x, if $t \overset{x}{\Rightarrow}_i \{f\}$, then $X_0 \vdash \{t\}_x\downarrow$.*

Soundness is an immediate consequence of the following lemma, taking $C = \{f\}$ and $y = \varepsilon$.

Lemma 4. *Suppose $i, d \geq 0$, $t \in Y_0, x, y \in K_0^*$, and $C \subseteq Q$ (with $D = C \cap Y_0$). Suppose the following also hold: 1) $t \overset{x}{\Rightarrow}_{i,d} C$, and 2) $C \subseteq Y_0$ or $X_0 \vdash y$. Then $X_0 \cup \{D\}_y \vdash \{t\}_{xy}$.*

As one may expect, the proof is by induction on the size of the run labelled x from t to C, but the difficulty with the proof is that in a run over x from t to C, each branch may hit f after reading a different prefix of x. Hence the inductive statement is subtle and this is why the statement of the Lemma is complex. In fact, formulating Lemma 4

precisely turned out to be the trickiest part of the upper bound proof. A detailed proof is presented in the technical report [BRS10].

Theorem 2 (Completeness). *For any $t \in Y_0$ and any keyword x, if $X_0 \vdash \{t\}_x\!\downarrow$, then there exists $i \geq 0$ such that $t \overset{x}{\Rightarrow}_i \{f\}$.*

The proof is by induction on derivations, and is reasonably straightforward.

Theorem 3. *Given $X_0 \subseteq \mathscr{T}$ and $t_0 \in \mathscr{T}$, it is decidable in* DEXPTIME *whether $X_0 \vdash t_0$.*

Proof. Let X_0 and t_0 be given, and let $Y_0 = st(X_0 \cup \{t_0\})$.

By Lemma 3, there is d' such that for all $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, and $C \subseteq Q$, and any $i \geq 0$,

$$\text{if } q \overset{a}{\hookrightarrow}_i C \text{ then } q \overset{a}{\hookrightarrow}_{d'} C.$$

Further $\hookrightarrow_{d'}$ is computable in time $2^{O(d)}$, where $d = |Y_0|$.

By the soundness theorem (Theorem 1), for all i, any $t \in Y_0$ and any keyword x, if $t \overset{x}{\Rightarrow}_i \{f\}$, then $X_0 \vdash \{t\}_x\!\downarrow$. In particular, this holds for $i = d'$. On the other hand, by the completeness theorem (Theorem 2), whenever $X_0 \vdash \{t\}_x\!\downarrow$ for $t \in Y_0$ and keyword x, there is an i such that $t \overset{x}{\Rightarrow}_i \{f\}$, and hence $t \overset{x}{\Rightarrow}_{d'} \{f\}$. Thus to check whether $X_0 \vdash t_0$, it suffices to check if $t_0 \overset{\varepsilon}{\Rightarrow}_{d'} \{f\}$. But by construction, if $t_0 \overset{\varepsilon}{\Rightarrow}_{d'} \{f\}$, then $t_0 \overset{\varepsilon}{\hookrightarrow}_{d'+1} \{f\}$, but this means that $t_0 \overset{\varepsilon}{\hookrightarrow}_{d'} \{f\}$.

Thus one only needs to check—in constant time—whether $t_0 \overset{\varepsilon}{\hookrightarrow}_{d'} \{f\}$. Thus the derivability problem is solvable in DEXPTIME. □

5 A DEXPTIME lower bound for the derivability problem

We recall the following fact about alternating pushdown systems.

Fact 4 *The* **backwards-reachability problem for alternating pushdown systems**, *which asks, given an APDS $\mathscr{P}$ and two configurations (s, x_s) and (f, x_f), whether $(s, x_s) \Rightarrow_{\mathscr{P}} (f, x_f)$, is* DEXPTIME*-complete [SSE06].*

We reduce this problem to the problem of checking whether $X \vdash t$ in our proof system, given $X \subseteq \mathscr{T}$ and $t \in \mathscr{T}$.

Assume that we are given an APDS $\mathscr{P} = (P, \Gamma, \hookrightarrow)$, and two configurations (s, x_s) and (f, x_f). Let us assume that the rules in $\hookrightarrow$ are numbered 1 to ℓ.

We will take $M = P \cup \{\mathsf{c}_m \mid 1 \leq m \leq \ell\}$ to be a set of atomic terms, and $K = \Gamma \cup \{d, e\}$ to be a set of *non-symmetric keys* (such that none of them is the inverse of another, and such that $d, e \notin \Gamma$).

We translate each rule to a term as follows. Suppose the m^{th} rule is:

$$(a, x) \hookrightarrow \{(b_1, x_1), \ldots, (b_n, x_n)\}.$$

This gets translated to the following term r_m:

$\mathsf{r}_m = [[\cdots[[\mathsf{r}'_m, \{b_1\}_{x_1}], \{b_2\}_{x_2}], \cdots, \{b_{n-1}\}_{x_{n-1}}], \{b_n\}_{x_n}]$, where
$\mathsf{r}'_m = [[\cdots[[\{\mathsf{c}_m\}_d, \{a\}_x], \{b_1\}_{x_1}], \cdots, \{b_{n-1}\}_{x_{n-1}}], \{b_n\}_{x_n}]$.

We take X to be the set $\{\mathsf{r}_m \mid 1 \leq m \leq \ell\} \cup \{\{f\}_{x_f e}\} \cup \{\{\mathsf{c}_m\}_d \mid 1 \leq m \leq \ell\} \cup \Gamma \cup \{e\}$.

The reduction is almost a straight transcription of the APDS rules. But we need to take some care because given a blind pair $[t, t']$, we can split it using either t or t'. Further, we have to avoid an accidental split of r_m using a part of r_n, for distinct $m, n \leq \ell$. This explains the need for the "tags" c_m ($m \leq \ell$).

We claim that $(s, x_s) \Rightarrow_{\mathscr{P}} (f, x_f)$ iff $X \vdash \{s\}_{x_s e}$. A detailed proof for both directions is presented in [BRS10]. Here we just present a high-level sketch of the proof. We prove the harder direction, that if there is a normal proof of $X \vdash \{a\}_{xe}$ then $(a, x) \Rightarrow_{\mathscr{P}} (f, x_f)$. The overall strategy is to prove that whenever a term of the form $\{a\}_{xe}$ is proved, there has to be a rule of $\mathscr{P}$ of which (a, y) is the left side, $x = yz$, and there is a shorter proof of $\{b\}_{y_i z}$, for every (b_i, y_i) on the right side of that rule. This requires to do a careful analysis of the proof of $X \vdash \{a\}_{xe}$. Here it is crucial to consider normal proofs, since the weak locality property (Lemma 2) imposes some structure on the terms occurring in such proofs. For instance, throughout the following we will use the fact that the *pair* rule will never be used in normal derivations that we encounter in the following proof.

We now introduce the following bit of notation, for conveniently presenting the argument. For any term t whose normal form is $[t_1, \ldots, t_n]$, we define *comps*(t) to be the set $\{t_1, \ldots, t_n\}$. If $t \in st(X)$ such that $\{\mathsf{c}_m\}_d \in st(t)$, then *residues*$(t)$ is defined by the following:

- *residues*$(\mathsf{r}_m) = \emptyset$
- if $t \neq \mathsf{r}_m$, then *residues*(t) = *residues*$([t, t'])$ $\cup \{t'\}$, where t' is the unique term such that $[t, t'] \in st(\mathsf{r}_m)$.

Lemma 5. *For any configuration (a, x), if there is a normal proof of $X \vdash \{a\}_{xe}$, then*

$$(a, x) \Rightarrow_{\mathscr{P}} (f, x_f)$$

The lemma follows easily, by induction on the size of normal proofs, from the next assertion.

Lemma 6. *If there is a normal proof π of $X \vdash \{a\}_{xe}$, then either $(a, x) = (f, x_f)$ or there is a rule of $\mathscr{P}$, $(a, y) \hookrightarrow \{(b_1, y_1), \ldots, (b_n, y_n)\}$, and $z \in \Gamma^*$ such that $x = yz$, and for each $j \leq n$, a* subproof *π_j of π with conclusion $X \vdash \{b_j\}_{y_j z e}$.*

Proof. The observation that drives the proof of this lemma is the following.

For any normal proof π of $X \vdash \{a\}_{xe}$ and any subproof δ of π with conclusion $\{p\}_{we}$, and any $m \leq \ell$:

1. if the last rule of δ is an application of *blindpair*, and if $\{\mathsf{c}_m\}_d \in st(p)$, then $X \vdash \{r\}_{we}$ is the conclusion of some subproof of δ, for every $\{r\}_{we} \in comps(\{p\}_{we})$.
2. if the last rule of δ is an application of *blindsplit*, and if $\{\mathsf{c}_m\}_d \in st(p)$, then $X \vdash \{r\}_{we}$ is the conclusion of some subproof of δ, for every $r \in residues(p)$.

Let π be a normal proof of $X \vdash \{a\}_{xe}$ and suppose that $(a, x) \neq (f, x_f)$. Then it is clear that for all prefixes y of xe, $\{a\}_y \notin X$. Thus π does not end in an application of *encrypt* (an easy consequence of the structure of X). It obviously cannot end in an application of *blindpair*. So it is clear that the last rule is an application of *blindsplit*, with major premise t and minor premise t'. Now t is a blind pair, and hence there is a unique $p \in st(X)$ and $z \in \Gamma^*$ such that $t = \{p\}_{ze}$ (again a consequence of the structure of X). It can be seen that $\{\mathsf{c}_m\}_d \in st(p)$ for some $m \leq \ell$. If t is obtained as the result of an application of *encrypt*, then it can be seen that $p = \mathsf{r}_m$ and thus p has no residues, and hence it is vacuously true that $\{r\}_{ze}$ occurs in δ for all $r \in residues(p)$. Otherwise, t is the result of a blind split, and hence, by the observation at the start of the proof, $\{r\}_{ze}$ occurs in δ for all $r \in residues(p)$.

Now if $p \in st(\mathsf{r}'_m)$, then among the residues of p will be found $\{b_j\}_{y_j}$ for every (b_j, y_j) on the right hand side of the rule numbered m. So by what has been proved above, there is a subproof π_j of π whose conclusion is $X \vdash \{b_j\}_{y_j ze}$, and we are done.

Suppose $p \notin st(\mathsf{r}'_m)$. Then, it can be seen that $t' = \{p'\}_{ze}$ for some $p' \in st(X)$ such that $\mathsf{r}'_m \in st(p')$. Now clearly $p' \notin X$ (since it is a proper subterm of r_m, missing a component of the form $\{a\}_w$ as it does) and hence t' is not the result of an application of *encrypt* (again an easy consequence of the structure of X). It cannot also be the result of an application of *blindsplit*, since then one of the premises has to be $\{a\}_{xe}$, contradicting minimality. Thus t' is the result of an application of *blindpair*, but the observation at the beginning of this proof tells us that $X \vdash \{r\}_{ze}$ for all $\{r\}_{ze} \in comps(\{p'\}_{ze})$. But notice that $\mathsf{r}'_m \in st(p')$, and hence we can conclude that among $comps(p')$ will be found $\{b_j\}_{y_j}$ for every (b_j, y_j) on the right hand side of the rule numbered m. So by the observation at the start of the proof, we can conclude that for each j, there is a subproof π_j of π whose conclusion is $X \vdash \{b_j\}_{y_j ze}$, and we are done. □

And the following theorem is the end result.

Theorem 4. *The passive intruder deduction problem is* DEXPTIME*-hard.*

6 Discussion

We can think of a number of extensions of our system by considering more algebraic properties of the blind pair operator, like associativity, commutativity, unitariness, etc. It then becomes more convenient to treat an extension of the Dolev-Yao model with a polyadic + operator, over which encryption distributes. In this framework, a very powerful system is studied in [LLT07], where + is treated as an abelian group operator.

The decidability results in [LLT07] are driven by a set of normalization rules whose effect is drastically different from ours. Our rules ensure that the "width" of terms occurring in a normal proof of $X \vdash t$ is bounded by $X \cup \{t\}$. But their normalization rules ensure that the encryption depth of terms occurring in a normal proof of $X \vdash t$ is bounded by $X \cup \{t\}$. On the other hand, the width of terms, represented by coefficients in the +-terms, can grow unboundedly. The rest of their decidability proof is an involved argument using algebraic methods.

The techniques of our paper do not seem to extend to the system with an abelian group operator, nor for slightly weaker systems where + is associative and commutative,

or when + is a (not necessarily commutative) group operator and the term syntax allows terms of the form $-t$. But the techniques for our upper bound proofs extends to the case when + is just an associative operator (not necessarily commutative, or has inverses). Another extension that is usually considered is encryption with constructed keys rather than atomic keys. The upper bound results go through for this system as well, with much of the hard work lying in extending the weak locality theorem. A sketch of the proofs is presented in the technical report [BRS10], and will be developed further in a companion paper.

References

[BC06] V. Bernat and H. Comon-Lundh. Normal proofs in intruder theories. In *ASIAN*, pp. 151–166, 2006.

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model-checking. In *CONCUR'97*, pp. 135–150, 1997.

[BRS07] A. Baskar, R. Ramanujam, and S.P. Suresh. Knowledge-based modelling of voting protocols. In *Proc. of TARK XI*, pp. 62–71, 2007.

[BRS10] A. Baskar, R. Ramanujam, and S.P. Suresh. A DEXPTIME-complete Dolev-Yao theory with distributive encryption. Technical report, May 2010. Available at: `http://www.cmi.ac.in/~spsuresh/pdffiles/mfcs2010-tr.pdf`.

[CDG+07] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. *Tree Automata Techniques and Applications*. 2007. Available at: `http://www.grappa.univ-lille3.fr/tata`.

[CS03] H. Comon-Lundh and V. Shmatikov. Intruder Deductions, Constraint Solving and Insecurity Decisions in Presence of Exclusive or. In *Proc. LICS 18*, pp. 271–280, June 2003.

[CRZ05] V. Cortier, M. Rusinowitch, and E. Zalinescu. A resolution strategy for verifying cryptographic protocols with cbc encryption and blind signatures. In *PPDP*, pp. 12–22, 2005.

[DKR09] S. Delaune, S. Kremer, and M.D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, July 2009.

[DY83] D. Dolev and A. Yao. On the Security of public-key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *ASIACRYPT*, pp. 244–251, 1992.

[GK99] T. Genet and F. Klay. Rewriting for cryptographic protocol verification. Technical report, CNET-France Telecom, 1999.

[Gou00] J. Goubault-Larrecq. A method for automatic cryptographic protocol verification. In *15th IPDPS*, *LNCS*, vol. 1800, pp. 977–984, 2000.

[LLT07] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for the equational theory of abelian groups with distributive encryption. *Information and Computation*, 205(4):581–623, April 2007.

[Mon99] David Monniaux. Abstracting cryptographic protocols with tree automata. In *Static analysis symposium*, *LNCS*, vol. 1694, pp. 149–163, 1999.

[RT03] M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete. *TCS*, 299:451–475, 2003.

[SSE06] D. Suwimonteerabuth, S. Schwoon, and J. Esparza. Efficient algorithms for alternating pushdown systems with an application to the computation of certificate chains. In *ATVA 4*, *LNCS*, vol. 4218, pp. 141–153, 2006.