Dolev-Yao theories 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 The 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, and there is an exponential lower bound on the size of derivations. 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

In the use of logic as a tool for analysing security of communication protocols, cryptography is abstracted using a term algebra. In these Dolev-Yao style models [14] for cryptographic protocols (the so-called "symbolic models") we 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 modelled as a powerful intruder who can control the entire network, and can encrypt and decrypt at will; however, the cryptographic means used are assumed to be perfect. Therefore, while the intruder may not have access to actual private keys possessed by agents, he has access to the structural patterns of terms that 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 the basic Dolev-Yao model, the main operators are pairing and encryption, but these two do not interact with each other, in the sense that the encryption of a paired term is no different from that of any other term. The critical use of encrypting a pair such as $\{(t,t')\}_k$ is to ensure that when we see a term t later, we can infer that t' is also "free".

The Dolev-Yao model abstracts away from the details of the encryption schemes used. However, the scheme used by principals would be known to the intruder, who can well make use of this information. In Dolev-Yao theory, the terms $\{t\}_k$ and $\{t'\}_{k'}$ are assumed to be distinct, unless $t = t'$ and $k = k'$. However, this is in general not true of cryptographic schemes such as the RSA. The algebraic properties of the encryption operator may well dictate the use of an equational theory to which the intruder has access. In such a context, interaction between encryption and other operators may be important. The reader is referred to the excellent survey [10] for studies of this kind.

One way of studying such interaction is by considering an extension of the Dolev-Yao term algebra with additional operators that interact in some specific way with encryption. For instance, [18] 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 $+$.

In this paper, we study a tensor operator $\otimes$ such that $\{\otimes t t'\}_k = \otimes\{t\}_k\{t'\}_k$. In other words, encryption is distributive over $\otimes$. (For reasons that will become clearer later, we do not assume commutativity or associativity of $\otimes$.) Tensor is a constructor, whereby we form $\otimes t t'$ from t and t'. In the spirit of the operator being seen as a form of multiplication, its inverse is not given by projection (as in the case of pairing) but by a form of *division*: we can extract t' or t from $\otimes t t'$, provided we have the other of the pair.

For such a theory, we show that the existence of a passive attack (that is, by an attacker who cannot forge messages) is decidable in exponential time. We also study the **proof complexity** of the term derivability problem and use it to show that the decision procedure is indeed optimal.

Is such an extension of the Dolev-Yao model only a mathematical curiosity? In fact, the tensor operator can be seen to be analogous to the blind pairing constructor finds natural use in the Dolev-Yao modelling of electronic voting protocols [15]. However more restricted uses of blind pairing may well suffice in many applications. What then can be interesting about such a result, in a 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 [13].

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 decidability of term derivability is to prove a so-called **locality property** [20, 9], 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 a 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 DEXPTIME-hardness by reduction from the reachability problem for alternating pushdown systems. We also prove an exponential lower bound on the size of derivations. 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. And the proof-size lower bound is one of the first in the study of Dolev-Yao models and its extensions, to our knowledge.

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 [4]). 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 [19, 17, 16]. Applications of two-way alternating tree automata to security protocol verification have been studied in [8]. 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 the tensor is distributive over encryption. This can be seen in the light of results on closely related constructors.

There is a more restricted way of modelling blind signatures: as seen in [12, 3, 11]. This is to consider two operators, blind and unblind 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 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 [1], we proposed essentially the same system described in this paper, but we imposed a restriction that one of the components in the tensor product is always of the form n or $\{n\}_k$ where n is an atomic term. And the only rule that involves distributing an encryption over a tensor is the derivation of $\otimes\{t\}_k n$ from $\otimes t\{n\}_{inv(k)}$ and k. This restricted system also satisfies a locality property. We prove this in Section 2.3, and also outline a PTIME algorithm.

At the other end of the spectrum lies the much more powerful system considered in [18] in which 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 tensor operator we consider has very different characteristics than xor, and the arguments in [18] do not apply here. We postpone a discussion of some technical aspects of [18] and how they relate to our own work to the end.

Organization of the paper

We present the basic definitions related to the Dolev-Yao framework and the term derivability problem in Section 2. In the same section, we present extensions to model distributive encryption, illustrate the difficulties involved, and present a restricted system for which the term derivability is solvable in PTIME. In Section 3, we prove a normalization result and a weak subterm property, which drives all the results that follow later. Section 4 contains details of an automaton-based DEXPTIME decision procedure for the term derivability problem.

The next three sections contain the lower bound proofs. In Section 5, we present some results on sets of terms that we call **rewrite systems**, and normal proofs from these sets. These form the basis for many of the theorems in the next two sections. Section 6 contains the DEXPTIME complexity lower bound, while in Section 7 we present the lower bound on size of derivations. We end with a discussion in the last section.

Many of the results in this paper were already present in the conference version [2]. But the lower bound proof for the size of derivations is completely new in this version. It was only mentioned as a brief example in [2]. We have also significantly simplified the coding for the DEXPTIME lower bound proof in that paper.

2 The Dolev-Yao framework and the term derivability problem

We present some basic definitions and notation. Let Σ be a signature that contains function symbols with appropriate arity ≥ 0.

The set of **terms** over the signature, $\mathcal{T}_\Sigma$, is defined as the smallest set such that:

- if $t_1, t_2, \ldots, t_n \in \mathcal{T}_\Sigma$, and $f \in \Sigma$ with arity $n \geq 0$, then $f(t_1, t_2, \ldots, t_n) \in \mathcal{T}_\Sigma$.

We define the set of **subterms** of term t, denoted by $st(t)$, inductively as follows:

- $st(c) \stackrel{\text{def}}{=} \{c\}$, where $c \in \Sigma$ with arity 0.
- $st(f(t_1, \ldots, t_n)) \stackrel{\text{def}}{=} \bigcup_{1 \leq i \leq n} st(t_i) \cup \{f(t_1, \ldots, t_n)\}$

For a given set of terms, $X, st(X) = \bigcup_{t \in X} st(t)$.

For a given a term t, the **size** of the term, denoted by $|t|$, is defined inductively as follows:

- $|c| \stackrel{\text{def}}{=} 1$, where $c \in \Sigma$ with arity 0.
- $|f(t_1, \ldots, t_n)| \stackrel{\text{def}}{=} |t_1| + \cdots + |t_n| + 1$

For a given a set of terms X, $|X| = \sum_{t \in X} |t|$.

A **proof system** PS is a finite set of rules of the following form.

$$\frac{X \vdash t_1 \quad \ldots \quad X \vdash t_n}{X \vdash t}\,\mathsf{r}$$

In this rule, the sequents $X \vdash t_i$ are the premises, and $X \vdash t$ is the conclusion.

▶ **Definition 1.** A **derivation** or a **proof** π of $X \vdash t$ in the proof system PS is a tree whose nodes are labelled by sequents, whose root is labelled $X \vdash t$, whose leaves are instances of the Ax rule and labelled by sequents of the form $X \vdash r$ (with $r \in X$, and whose internal nodes are instances of one of the rules from PS. We use $X \vdash_{PS} t$ to denote that there is a proof of $X \vdash t$. If PS is clear from the context, we drop the subscript. We say that a sequent occurs in π if it labels the root of one of the subproofs of π. We sometimes say that r occurs in π to mean that $X \vdash r$ occurs in π. We say that r occurs as a subterm in π if there is some t such that $X \vdash t$ occurs in π and $r \in st(t)$.

A **minimal proof** π is one in which no sequent $X \vdash r$ occurs twice on the same branch of π.

▶ **Definition 2.** The **derivability problem** (or the **passive intruder deduction problem**) is the following: given a finite set $X \subseteq \mathcal{T}$ and $t \in \mathcal{T}$, determine whether $X \vdash_{PS} t$.

2.1 The basic model

Assume a set of basic terms $\mathcal{N}$, which also includes the set of keys $\mathcal{K}$. Let $inv(k)$ be a function on $\mathcal{K}$ such that $inv(inv(k)) = k$. The signature for this term algebra contains $\mathcal{N}$ a set of nullary operators, a binary pairing operator, and a binary encryption operator.

▶ **Definition 3.** The set of **terms** $\mathcal{T}$ is defined to be:

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

where $m \in \mathcal{N}$, $k \in \mathcal{K}$, and t, t_1, and t_2 range over $\mathcal{T}$. A **keyword** is a sequence of keys $x \in \mathcal{K}^*$. For a keyword $x = k_0 \cdots k_{n-1}$ and term t, $\{t\}_x$ is defined to be $\{\{t\}_{k_0} \cdots\}_{k_{n-1}}$.

The Dolev-Yao proof system is given in figure 1, and we use DY to denote the proof system.

The decidability of the derivability problem for DY is based on the **subterm property** for minimal proofs. An important property of minimal proofs is the following, which is proved easily by induction on the structure of proofs.

analz-rules		$\dfrac{X \vdash \{t\}_k \quad X \vdash inv(k)}{X \vdash t}\, decrypt$	$\dfrac{X \vdash (t_0, t_1)}{X \vdash t_i}\, split_i$
synth-rules	$\dfrac{}{X \vdash t}\, Ax\ (t \in X)$	$\dfrac{X \vdash t \quad X \vdash k}{X \vdash \{t\}_k}\, encrypt$	$\dfrac{X \vdash t_1 \quad X \vdash t_2}{X \vdash (t_1, t_2)}\, pair$

Figure 1 Dolev-Yao theory

▶ **Lemma 4** (Subterm property or locality)**.** *Let π be a minimal proof of t from X. Then every r occurring in π belongs to $st(X \cup \{t\})$.*

▶ **Theorem 5.** *Given a finite set of terms X and a term t, checking whether $X \vdash_{DY} t$ is decidable in time polynomial in size of $X \cup \{t\}$.*

Proof. Suppose there is a proof of $X \vdash t$. Then there is a minimal proof of $X \vdash t$. Also, all the terms occurring in this proof are subterms of $X \cup \{t\}$. Further, along every branch of a minimal proof, the same sequent cannot occur twice. Thus the height of a normal proof of $X \vdash t$ is bounded by the size of $st(T \cup \{t\})$, say D. Therefore it suffices to check if there exists a proof of $X \vdash t$ of height D.

Here is a procedure to do this, which runs in time polynomial in D:

$X' := X \cup \{t\}$
`repeat` D `times:`
  $X'' := \{r'' \mid \exists r, r' \in X'$ `such that` r'' `is derived from` r `and` r'
  `by the application of one synth- or analz-rule`$\}$;
  $X' := X'' \cap \mathtt{st}(X \cup \{t\}$;

Thus the problem of checking whether $X \vdash_{DY} t$ is decidable in polynomial time. ◀

2.2 Distributive encryption in the Dolev-Yao framework

We extend the basic Dolev-Yao system to include a tensor term: that is, a binary tensor operator is added to the signature. We choose to present it as a **blind pair** operator, because it was originally studied in the context of modelling blind signatures in voting protocols [1], among other applications.

The set of **terms** $\mathscr{T}$ is defined to be:

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

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

As we have said in the introduction, the new feature of the blind pair operator is the interaction with the encryption operator. We can push the encryption operator inside the blind pair operator, and this is modelled by the equational theory in Figure 2.

The proof system is now a deduction system modulo the equational theory, and is presented in Figure 3.

But for most of our analysis of deducibility, it is easier to work with a pure proof system that does not involve equations. The standard way to achieve this is to define a notion of normal terms (induced by the equational theory), and present a proof system involving only normal terms. We present the definition of normal terms next and define a proof system based on it in Figure 4. We refer to it as the **extended Dolev-Yao system**, and we let *EDY* denote it.

$$\{[t,t']\}_k \equiv [\{t\}_k, \{t'\}_k]$$

$$\frac{}{t \equiv t} \qquad \frac{t \equiv t'}{t' \equiv t} \qquad \frac{t \equiv t' \quad t' \equiv t''}{t' \equiv t''}$$

$$\frac{t_1 \equiv t_1' \quad t_2 \equiv t_2'}{(t_1,t_2) \equiv (t_1',t_2')} \qquad \frac{t_1 \equiv t_1' \quad t_2 \equiv t_2'}{[t_1,t_2] \equiv [t_1',t_2']} \qquad \frac{t \equiv t'}{\{t\}_k \equiv \{t'\}_k}$$

Figure 2 Equational theory for the distributive encryption

$$\frac{}{X \vdash t}\,Ax\ (t \in X) \qquad \frac{X \vdash t}{X \vdash t'}\ (t \equiv t')$$

$$\frac{X \vdash \{t\}_k \quad X \vdash inv(k)}{X \vdash t}\,decrypt \qquad \frac{X \vdash t \quad X \vdash k}{X \vdash \{t\}_k}\,encrypt$$

$$\frac{X \vdash (t_0,t_1)}{X \vdash t_i}\,split_i \qquad \frac{X \vdash t_1 \quad X \vdash t_2}{X \vdash (t_1,t_2)}\,pair$$

$$\frac{X \vdash [t_0,t_1] \quad X \vdash t_i}{X \vdash t_{1-i}}\,blindsplit_i \qquad \frac{X \vdash t_1 \quad X \vdash t_2}{X \vdash [t_1,t_2]}\,blindpair$$

Figure 3 Dolev-Yao theory modulo equations

▶ **Definition 6.** **Normal terms** 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 $m \in \mathcal{N}$
- $(t_1,t_2)\!\downarrow = (t_1\!\downarrow, t_2\!\downarrow)$
- $[t_1,t_2]\!\downarrow = [t_1\!\downarrow, t_2\!\downarrow]$
- $\{t\}_k\!\downarrow = \begin{cases} [\{t_1\}_k\!\downarrow, \{t_2\}_k\!\downarrow] & \text{if } t = [t_1,t_2] \\ \{t\!\downarrow\}_k & \text{otherwise} \end{cases}$

Observe that if X is a set of normal terms and there is a proof of $X \vdash t$, then t is also a normal term. So we can assume for the rest of the paper that we are working only with normal terms.

As we have remarked above, the standard route to decision procedure for term derivability is via the locality property: we define a set $S(X)$ and ensure that any minimal proof only uses terms from this set. Unfortunately, our system does not have any obvious locality property. We illustrate this by giving examples which violate the locality property for some simple choices of $S(X)$.

▶ **Example 7.** Suppose we choose $S(X)$ to be $st(X)$. If we take X to be the set $\{[a,b], \{b\}_k, k\}$, and t to be $\{a\}_k$, then the following derivation shows that $X \vdash t$, but it is intuitively clear (and can in fact be rigorously proved using the methods in Section 5) that the term $[\{a\}_k, \{b\}_k]$, which does not belong to $st(X \cup \{t\})$, has to occur in all derivations of $X \vdash t$.

$$\frac{}{X \vdash t}\,Ax\ (t \in X)$$

$$\frac{X \vdash \{t\}_k{\downarrow} \quad X \vdash inv(k)}{X \vdash t}\,decrypt \qquad\qquad \frac{X \vdash t \quad X \vdash k}{X \vdash \{t\}_k{\downarrow}}\,encrypt$$

$$\frac{X \vdash (t_0, t_1)}{X \vdash t_i}\,split_i \qquad\qquad \frac{X \vdash t_1 \quad X \vdash t_2}{X \vdash (t_1, t_2)}\,pair$$

$$\frac{X \vdash [t_0, t_1] \quad X \vdash t_i}{X \vdash t_{1-i}}\,blindsplit_i \qquad\qquad \frac{X \vdash t_1 \quad X \vdash t_2}{X \vdash [t_1, t_2]}\,blindpair$$

analz-rules *synth*-rules

Figure 4 Extended Dolev-Yao theory. In the rule *decrypt*, $\{t\}_k{\downarrow}$ is the major premise and $inv(()k)$ the minor premise. In the rule *encrypt*, t is hte major premise and k is the minor premise. In the $blindsplit_i$ rule, $[t_0, t_1]$ is the major premise and t_i is the minor premise.

$$\frac{\dfrac{X \vdash [a,b] \quad X \vdash k}{X \vdash [\{a\}_k, \{b\}_k]}\,encrypt \qquad X \vdash \{b\}_k}{X \vdash \{a\}_k}\,blindsplit$$

▶ **Example 8.** The previous example suggests that we may need to extend the definition of $S(X)$ by including all terms of the same encryption depth as the ones mentioned in X. More formally,

$$S(X) = \{\{t\}_x \mid t \in st(X), x \in (st(X) \cap \mathscr{K})^*, |x| \leq d\}$$

where d is the maximum nested encryption depth of terms in X.

We now give an example X and t such that the maximum encryption depth of terms in $X \cup \{t\}$ is one, but the most natural derivation contains a term of encryption depth 3. We take X to be the set

$$\{k, [a, \{b\}_k], [b, \{c\}_k], [c, \{d\}_k], [e, \{d\}_k], [f, \{e\}_k], \{f\}_k\}$$

and t to be the term a. A derivation of $X \vdash t$ is shown below, but it can be shown that every derivation of $X \vdash t$ has to contain the term $\{d\}_{kkk}$, which clearly does not belong to $S(X \cup \{t\})$.

We give the promised derivation below. To reduce clutter, we display only the terms on the right hand side of sequents. We also present the proof in two parts. Let π' be the derivation of $\{d\}_{kkk}$ from X, given in Figure 5. The actual derivation of $X \vdash a$ is given in Figure 6, and it uses π'.

$$\frac{\dfrac{\dfrac{\dfrac{}{[e,\{d\}_k]}\,Ax \quad \dfrac{}{k}\,Ax}{[\{e\}_k, \{d\}_{kk}]}\,encrypt \quad \dfrac{}{k}\,Ax}{[\{e\}_{kk}, \{d\}_{kkk}]}\,encrypt \qquad \dfrac{\dfrac{\dfrac{}{[f,\{e\}_k]} \quad \dfrac{}{k}\,Ax}{[\{f\}_k, \{e\}_{kk}]}\,encrypt \quad \dfrac{}{\{f\}_k}\,Ax}{\{e\}_{kk}}\,blindsplit}{\{d\}_{kkk}}\,blindsplit$$

Figure 5 Proof π' of $\{d\}_{kkk}$

$$\dfrac{\dfrac{}{[a,\{b\}_k]}\,\mathit{Ax} \qquad \dfrac{\dfrac{\dfrac{}{[b,\{c\}_k]}\,\mathit{Ax} \quad \dfrac{}{k}\,\mathit{Ax}}{[\{b\}_k,\{c\}_{kk}]}\,\mathit{encrypt} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{[c,\{d\}_k]}\,\mathit{Ax} \quad \dfrac{}{k}\,\mathit{Ax}}{[\{c\}_k,\{d\}_{kk}]}\,\mathit{encrypt} \quad \dfrac{}{k}\,\mathit{Ax}}{[\{c\}_{kk},\{d\}_{kkk}]}\,\mathit{encrypt} \quad \begin{matrix}\vdots\,\pi'\\ \{d\}_{kkk}\end{matrix}}{\{c\}_{kk}}\,\mathit{blindsplit}}{\{b\}_k}\,\mathit{blindsplit}}{a}\,\mathit{blindsplit}$$

Figure 6 Proof π of a

The second example might suggest yet another definition of $S(X)$ which takes into account the number of terms in X, their "width", and their encryption depth. But it is not clear what a natural definition would be, and it would be very difficult to prove the locality property with respect to complicated definitions. In fact, we shall show in Section 7 that for every n, there exist X and t of size $O(n)$, such that every derivation of $X \vdash t$ is of exponential size.

In light of this, there are two directions to proceed in. In the next subsection, we restrict the interaction between the blind pair operator and encryption to arrive at a more well-behaved system which enjoys a locality property. Though the interaction is restricted, it still suffices for some common applications, and is hence a usable system.

But what about the system *EDY*? We show that we can settle for a weak locality property, where $S(X)$ is not necessarily a finite set, but which is still useful in the analysis of proofs. These results are proved in Section 3.

2.3 Restricted proof system

In this section, we place a restriction on the interaction between the blind pair and encryption operators to get a PTIME algorithm for the term derivability problem. Consider the following restricted interaction: if we encrypt the blind pair term with the key k, then we push in the encryption if at least one part of the blind pair term is of the form $\{m\}_{inv(k)}$. Here m is an atomic term. In other words, we will have the following equations in the equational theory instead of the more general equation which we saw earlier.

$$\{[t,\{m\}_{inv(k)}]\}_k = [\{t\}_k, m]$$
$$\{[\{m\}_{inv(k)}, t]\}_k = [m, \{t\}_k]$$

We again consider a deduction system modulo the equational theory generated by the above equations, and its simplification using normal terms. The proof system is the same as *EDY*, except that we use the following definition of $t\downarrow$. We use *RDY* to refer to the new system.

▶ **Definition 9.** We define $t\downarrow$ inductively as follows.

- $m\downarrow = m$ for $m \in \mathscr{N}$
- $(t_1, t_2)\downarrow = (t_1\downarrow, t_2\downarrow)$
- $[t_1, t_2]\downarrow = [t_1\downarrow, t_2\downarrow]$
- $\{t\}_k\downarrow = \begin{cases} [\{t'\}_k\downarrow, m] & \text{if } t = [t', \{m\}_{inv(k)}] \\ [m, \{t'\}_k\downarrow] & \text{if } t = [\{m\}_{inv(k)}, t'] \\ \{t\downarrow\}_k & \text{otherwise} \end{cases}$

Despite the restricted interaction, the system can still be used to model cryptographic operations like blind signatures, which have applications in electronic voting protocols.

▶ **Example 10** (Blind signatures)**.** Suppose Alice wants to get Bob to sign a message t for her, without revealing t to Bob. This can be done as follows. Alice sends $[t, \{r\}_{public(B)}]$ to Bob, where r is a some random number chosen by Alice . Now Bob signs this message to get $\{[m, \{r\}_{public(B)}]\}_{private(B)}\downarrow$, which is the same as $[\{m\}_{private(B)}, r]$. From this, Bob cannot get m as he does not know r. But on receiving this message, Alice can get $\{m\}_{private(B)}$ using the *blindsplit* rule (since she has r).

We now prove that the term derivability problem for this restricted proof system is in PTIME. We follow the standard strategy to prove this claim. First, we define the notion of a normal proof.

▶ **Definition 11.** 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 7 can not be applied to π.

$$\dfrac{\dfrac{\dfrac{\overset{\vdots\,\pi'}{X\vdash t} \quad \overset{\vdots\,\pi''}{X\vdash \{m\}_{inv(k)}}}{X\vdash [t,\{m\}_{inv(k)}]}\,\textit{blindpair} \quad \overset{\vdots\,\pi'''}{X\vdash k}}{X\vdash [\{t\}_k, m]}\,\textit{encrypt} \quad \overset{\vdots\,\delta}{X\vdash m}}{X\vdash \{t\}_k}\,\textit{blindsplit} \quad \rightsquigarrow \quad \dfrac{\overset{\vdots\,\pi'}{X\vdash t} \quad \overset{\vdots\,\pi'''}{X\vdash k}}{X\vdash \{t\}_k}\,\textit{encrypt}$$

$$\dfrac{\dfrac{\dfrac{\overset{\vdots\,\pi'}{X\vdash \{m\}_{inv(k)}} \quad \overset{\vdots\,\pi''}{X\vdash t}}{X\vdash [\{m\}_{inv(k)}, t]}\,\textit{blindpair} \quad \overset{\vdots\,\pi'''}{X\vdash k}}{X\vdash [m, \{t\}_k]}\,\textit{encrypt} \quad \overset{\vdots\,\delta}{X\vdash m}}{X\vdash \{t\}_k}\,\textit{blindsplit} \quad \rightsquigarrow \quad \dfrac{\overset{\vdots\,\pi''}{X\vdash t} \quad \overset{\vdots\,\pi'''}{X\vdash k}}{X\vdash \{t\}_k}\,\textit{encrypt}$$

Figure 7 Transformation rules for *RDY*. The rules correspond to the equation $\mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(m,r),k),r) = \mathsf{sign}(m,k)$ from [12].

Thus in a normal proof, there cannot be an application of a *blindpair* rule followed by an application of a *encrypt* rule followed by an application of a *blindsplit* rule. It can be easily shown that any proof can be transformed to a normal proof. As in the basic Dolev-Yao theory, normal proofs in the theory *RDY* also enjoy a locality property, where we take $S(X)$ to be $est(X)$ (defined below) rather than $st(X)$.

▶ **Definition 12.** The set of **extended subterms** of t, $est(t)$, for any term t is defined as follows:
- $est(m) \stackrel{\text{def}}{=} \{m\}$ for $m \in \mathcal{N}$,
- $est((t_1, t_2)) \stackrel{\text{def}}{=} \{(t_1, t_2)\} \cup est(t_1) \cup est(t_2)$,
- $est(\{t\}_k) \stackrel{\text{def}}{=} \{\{t_1\}_k\} \cup est(t_1) \cup \{k\}$,
- $est([t_1, t_2]) \stackrel{\text{def}}{=} \{t_1, t_2\} \cup est(t_1) \cup est(t_2)$ if $[t_1, t_2]$ is not one of $[t, \{m\}_k], [\{m\}_k, t]$,
- $est([t, \{m\}_k]) \stackrel{\text{def}}{=} \{[t, \{m\}_k], [\{t\}_{inv(k)}, m]\} \cup est(\{m\}_k) \cup est(\{t\}_{inv(k)})$, and
- $est([\{m\}_k], t) \stackrel{\text{def}}{=} \{[\{m\}_k, t], [m, \{t\}_{inv(k)}]\} \cup est(\{m\}_k) \cup est(\{t\}_{inv(k)})$.

For a set of terms X, *est*(X) is defined to be $\bigcup_{t \in T}$ *est*(t). It is easy to see that $|est(t)| \leq 7 \cdot |t|$, and that $|est(X)| \leq 7 \cdot |X|$.

▶ **Proposition 13.** *Let π be a normal proof of $X \vdash t$. Then $r \in est(X \cup \{t\})$ for all terms r occurring in π. Moreover, if π ends in an application of an analz rule, $r \in est(X)$.*

Proof. We prove this by induction on the structure of proofs. We will use the fact that subproofs of normal proofs are also normal. So the induction hypothesis is always available to us. We present only the most important case.

Suppose π is of the following form and r is a term occurring in π:

$$\dfrac{\begin{array}{c}\pi_1 \\ \vdots \\ X \vdash [t, t']\end{array} \quad \begin{array}{c}\pi_2 \\ \vdots \\ X \vdash t'\end{array}}{X \vdash t}\ \textit{blindsplit}$$

We first prove that $[t, t'] \in est(X)$.

- Suppose π_1 ends in an *analz*-rule.
  By induction hypothesis, for every r occurring in π_1, $r \in est(X)$. In particular, $[t, t'] \in est(X)$, and we are done.
- Suppose the last rule of π_1 is a *synth*-rule.
  Let us look at the last rule of π_1. It can be either *blindpair* or *encrypt*. If it is *blindpair*, then $X \vdash t$ should be one of the premises. But this is not possible since π_1 is a normal proof. Hence the last rule of π_1 should be *encrypt*. Clearly $[t, t']$ should be of the form $[\{u\}_k, m]$ or $[m, \{u\}_k]$ for some $k \in \mathscr{K}$ and $m \in \mathscr{B}$. We will consider only the first case, and the second case can be argued similarly. Now π looks as follows.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ \pi_1' \\ X \vdash [u, \{m\}_{inv(k)}]\end{array} \quad \begin{array}{c}\vdots\ \pi_1'' \\ X \vdash k\end{array}}{X \vdash [\{u\}_k, m]}\ \textit{encrypt} \quad \begin{array}{c}\vdots\ \pi_2 \\ X \vdash m\end{array}}{X \vdash \{u\}_k}\ \textit{blindsplit}$$

Now we look at the last rule of π_1'. It can be *blindpair*, *encrypt* or an *analz*-rule.

- Suppose if it is a *blindpair* rule, then we can apply the transformation in figure 7. But this is a contradiction to π being a normal proof. Hence, π_1' cannot end in *blindpair* rule.
- Suppose π' ends with an *encrypt* rule. Then the major premise of π_1' is $[\{u\}_k, m]$. This term occurs twice in a branch of π, and this contradicts the fact that π is a normal proof. Hence the last rule of π_1' can not be *encrypt*.
- Suppose π_1' ends in an *analz*-rule. By induction hypothesis, $[u, \{m\}_{inv(k)}]$ belongs to *est*(X), and hence so does $[\{u\}_k, m]$ and $\{u\}_k$ (by our definition of *est*(X)).

We have proved that $[t, t'] \in est(X)$, whence $est(X \cup \{[t, t']\}) = est(X \cup \{t'\}) = est(X)$. Now by the induction hypothesis, any r occurring in π_1 belongs to $est(X \cup \{[t, t']\}) = est(X)$, and any r occuring in π_2 belongs to $est(X \cup t') = est(X)$. Hence, any r occurring in π either occurs in π_1, π_2 or is the same as t. In all cases, $r \in est(X)$. ◀

Now we can prove that the derivability problem for *RDY* is decidable in polynomial time. The proof is similar to the proof of Theorem 5, except that we use *est*(X) instead of *st*(X).

▶ **Theorem 14.** *Given a finite set of terms X and a term t, checking whether $X \vdash_{RDY} t$ is decidable in time polynomial in size of X.*

3 Normal proofs and weak locality

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 8 repeatedly. Any subproof that matches the 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.

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ t'\end{matrix} \quad \begin{matrix}\vdots\,\pi'' \\ t''\end{matrix}}{[t,t']}\,\mathit{blindpair} \quad \begin{matrix}\vdots\,\delta \\ k\end{matrix}}{[\{t'\}_k{\downarrow},\{t''\}_k{\downarrow}]}\,\mathit{encrypt} \quad \rightsquigarrow \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ t'\end{matrix} \quad \begin{matrix}\vdots\,\delta \\ k\end{matrix}}{\{t'\}_k{\downarrow}}\,\mathit{encrypt} \quad \dfrac{\begin{matrix}\vdots\,\pi'' \\ t''\end{matrix} \quad \begin{matrix}\vdots\,\delta \\ k\end{matrix}}{\{t''\}_k{\downarrow}}\,\mathit{encrypt}}{[\{t'\}_k{\downarrow},\{t'\}_k{\downarrow}]}\,\mathit{blindpair}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ \{t'\}_k{\downarrow}\end{matrix} \quad \begin{matrix}\vdots\,\pi'' \\ \{t''\}_k{\downarrow}\end{matrix}}{[\{t'\}_k{\downarrow},\{t''\}_k{\downarrow}]}\,\mathit{blindpair} \quad \begin{matrix}\vdots\,\delta \\ \mathit{inv}(k)\end{matrix}}{[t',t'']}\,\mathit{decrypt} \quad \rightsquigarrow \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ \{t'\}_k{\downarrow}\end{matrix} \quad \begin{matrix}\vdots\,\delta \\ \mathit{inv}(k)\end{matrix}}{t'}\,\mathit{decrypt} \quad \dfrac{\begin{matrix}\vdots\,\pi'' \\ \{t''\}_k{\downarrow}\end{matrix} \quad \begin{matrix}\vdots\,\delta \\ \mathit{inv}(k)\end{matrix}}{t''}\,\mathit{decrypt}}{[t',t'']}\,\mathit{blindpair}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ [\{t'\}_k{\downarrow},\{t''\}_k{\downarrow}]\end{matrix} \quad \begin{matrix}\vdots\,\pi'' \\ \{t''\}_k{\downarrow}\end{matrix}}{\{t'\}_k{\downarrow}}\,\mathit{blindsplit} \quad \begin{matrix}\vdots\,\delta \\ \mathit{inv}(k)\end{matrix}}{t'}\,\mathit{decrypt} \quad \rightsquigarrow \quad \dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ [\{t'\}_k{\downarrow},\{t''\}_k{\downarrow}]\end{matrix} \quad \begin{matrix}\vdots\,\delta \\ \mathit{inv}(k)\end{matrix}}{[t',t'']}\,\mathit{decrypt} \quad \dfrac{\begin{matrix}\vdots\,\pi'' \\ \{t''\}_k{\downarrow}\end{matrix} \quad \begin{matrix}\vdots\,\delta \\ \mathit{inv}(k)\end{matrix}}{t''}\,\mathit{decrypt}}{t'}\,\mathit{blindsplit}$$

Figure 8 The normalization rules

▶ **Definition 15.** A proof π of $X \vdash t$ is a **minimal proof** if $X \vdash t$ occurs only in the root of the proof. A proof π is a **normal proof** if:

1. π is minimal, and
2. the transformations in Figure 8 cannot be applied to π.

The following lemma highlights the centrality of normal proofs.

▶ **Lemma 16.** *Whenever $X \vdash t$, there is a normal proof of t from X.*

Proof. For every proof π, we define a measure $d(\pi)$ recursively as follows:

- if π ends in an *Ax* rule, $d(\pi) = 1$,
- if π has immediate subproofs π' and π'' and ends in an application of a rule other than *encrypt* or *decrypt*, then $d(\pi) = d(\pi') + d(\pi'') + 1$, and
- if π ends in an application of either *encrypt* or *decrypt* and has immediate subproofs π' and π'', then $d(\pi) = 2^{d(\pi')+d(\pi'')}$.

We can view normal proofs as the result of repeatedly applying the reduction steps in Figure 8 and a reduction step which replaces proofs by subproofs which have the same root. And it suffices to show that for each of these reduction steps that transforms π to π', $d(\pi') < d(\pi)$. This immediately proves that the normalization procedure terminates.

The non-trivial cases are the reductions in Figure 8. For these, we observe that the measure of the proof on the left is $2^{d(\pi')+d(\pi'')+d(\delta)+1}$, while the measure of the proof on the right is $2^{d(\pi')+d(\delta)} + 2^{d(\pi'')+d(\delta)} + 1$. Let $d(\pi') = m$, $d(\pi'') = n$, and $d(\delta) = p$, and assume without loss of generality that $m \geq n$. Then—since $m, n, p > 0$—$2^{m+n+p+1} > 2^{m+p+1} + 1 \geq 2^{m+p} + 2^{n+p} + 1$. This concludes the proof. ◀

Also important is the following lemma, which is used vitally to prove lower bounds on the size of proofs in Section 7. The lemma is easily seen to be true by inspecting the normalization rules.

▶ **Lemma 17.** *If a proof π reduces to another proof π', then for any term t that occurs in π', a term of the form $\{t\}_x$ occurs in π, for some keyword x. Furthermore, if t is not a blind pair, then t itself occurs in π.*

The above lemma (whose truth is easily seen by inspecting the normalization rules) is extremely important, since it allows us to prove lower bounds on the number of terms occurring in any proof of $X \vdash t$ (and hence on the size of any proof of $X \vdash t$) by proving a corresponding lower bound for any *normal proof* of $X \vdash t$. Since normal proofs are likely to have much more structure than non-normal proofs, it is to be expected that they are more amenable to non-trivial analysis. We will witness this phenomenon both in the weak locality property (Lemma 18) and the upper bound results that follow from it (Section 4), and in the complexity lower bound (Section 6) and lower bound on proof size (Section 7).

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 $\{x \in (st(X \cup \{t\}) \cap \mathscr{K})^* \mid X \vdash \{p\}_x\downarrow\}$. 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 is not a blind pair.

▶ **Lemma 18** (Weak locality property)**.** *Let π be a normal proof of $X \vdash t$, and let δ be a subproof of π with root labelled r. Let u be a term occurring in δ. Then:*
1. *If δ is a purely synthetic proof, then either $u \in st(r)$ or there is a term $p \in st(X)$ and a keyword x such that $u = \{p\}_x\downarrow$.*
2. *If δ is not a purely synthetic proof, then there is a term $p \in st(X)$ and a keyword x such that $u = \{p\}_x\downarrow$.*
3. *If the last rule of δ is decrypt or split with major premise r_1, then $r_1 \in st(X)$.*

Proof. We do an induction on the structure of proofs. We assume the claim for every proper subproof δ' of δ, and prove it for δ itself.
- Suppose δ is of the following form:

$$\frac{}{X \vdash r}\,Ax$$

Then $r \in X \subseteq st(X)$, and we are done.

- Suppose δ is the following form (and $r = (r', r'')$):

$$\dfrac{\begin{array}{cc} \vdots\,\delta' & \vdots\,\delta'' \\ X \vdash r' & X \vdash r'' \end{array}}{X \vdash r}\ \textit{pair}$$

In this case, δ is a purely synthetic proof, and we aim to prove that for every u occurring in δ, either $u \in st(r)$ or there is $p \in st(X)$ and keyword x such that $u = \{p\}_x\!\downarrow$. But any such u either occurs in δ' or δ'' or is the same as r. In the first case, by induction hypothesis, $u \in st(r')$ or there exists $p \in st(X)$ and keyword x such that $u = \{p\}_x\!\downarrow$. But since $r' \in st(r)$, $u \in st(r)$ or $u = \{p\}_x\!\downarrow$, and we are done. We argue similarly in the second case. Finally $r \in st(r)$, and so we are done in the third case as well.

- Suppose δ is of the following form:

$$\dfrac{\begin{array}{c} \vdots\,\delta' \\ X \vdash (r, r') \end{array}}{X \vdash r}\ \textit{split}$$

We have to consider the following cases:

1. Suppose δ' is not a purely synthetic proof and for every u occurring in δ' there is a $p' \in st(X)$ and keyword x' such that $u = \{p'\}_{x'}\!\downarrow$. In particular, there is a $p \in st(X)$ and keyword x such that $(r, r') = \{p\}_x\!\downarrow$. But this means that $x = \varepsilon$ and $(r, r') = p \in st(X)$. So $r \in st(X)$ as well. Thus we have proved that for every u occurring in δ, there is a $p \in st(X)$ and keyword x such that $u = \{p\}_x\!\downarrow$. We have also proved that the major premise of the last rule is in $st(X)$.
2. Suppose δ' is a purely synthetic proof. But then δ' has to end in an application of the *pair* rule, and therefore one of the premises of the last rule of δ' has to be r, and this contradicts minimality of δ. So this case is not possible.

- Suppose δ is the following form (and $r = [r', r'']$):

$$\dfrac{\begin{array}{cc} \vdots\,\delta' & \vdots\,\delta'' \\ X \vdash r' & X \vdash r'' \end{array}}{X \vdash r}\ \textit{blindpair}$$

We argue exactly as in the case when the last rule is *pair*.

- Suppose δ is of the following form:

$$\dfrac{\begin{array}{cc} \vdots\,\delta' & \vdots\,\delta'' \\ X \vdash [r, s] & X \vdash s \end{array}}{X \vdash r}\ \textit{blindsplit}_1$$

We have to consider the following cases:

1. Suppose δ' is not a purely synthetic proof and for every u occurring in δ' there is a $p' \in st(X)$ and keyword x' such that $u = \{p'\}_{x'}\!\downarrow$. In particular, there is a $p \in st(X)$ and keyword x such that $[r, s] = \{p\}_x\!\downarrow$.
   Turning our attention to u occurring in δ'', either $u \in st(s)$ or there is $v \in st(X)$ and keyword y such that $u = \{v\}_y\!\downarrow$. But recall that $s \in st([r, s])$ and there is $p \in st(X)$ and keyword x such that $[r, s] = \{p\}_x$. Therefore if $u \in st(s)$, clearly there is $v' \in st(X)$ such that $u = \{v'\}_x$. It also immediately follows that $r = \{q\}_x\!\downarrow$ for some $q \in st(X)$. Thus we have proved that for every u occurring in δ, there is a $p \in st(X)$ and keyword x such that $u = \{p\}_x\!\downarrow$.

2. Suppose δ' is a purely synthetic proof. But then δ' does not end with an instance of the *encrypt* rule, and hence ends with an instance of the *blindpair* rule. But that contradicts the minimality of δ, as we can see by reasoning similar to the case when δ ends with a *split*. So this case is not possible.

- Suppose δ is of the following form (and $r = \{r'\}_k\downarrow$):

$$\dfrac{\begin{array}{cc} \vdots\,\delta' & \vdots\,\delta'' \\ X \vdash r' & X \vdash k \end{array}}{X \vdash r}\ \textit{encrypt}$$

We have to consider the following cases:

1. Suppose r is not a blind pair, and hence δ is a purely synthetic proof. Then we aim to prove that for every u occurring in δ, either $u \in st(r)$ or there is $p \in st(X)$ and keyword x such that $u = \{p\}_x\downarrow$. But any such u either occurs in δ' or occurs in δ'' or is the same as r. In the first case, by induction hypothesis, either $u \in st(r')$ or there exists $p \in st(X)$ and keyword x such that $u = \{p\}_x\downarrow$. But since $r' \in st(r)$, the desired conclusion follows. We argue similarly in the second case, when u occurs in δ''. Finally $r \in st(r)$, and so we are done in the third case as well.
2. Suppose r is a blind pair, and hence δ is not a purely synthetic proof. We aim to prove that for every u occurring in δ, there is $p \in st(X)$ and keyword x such that $u = \{p\}_x\downarrow$. We consider the following subcases:
   a. Suppose δ' is not a purely synthetic proof and for every u occurring in δ' there is a $p' \in st(X)$ and keyword x' such that $u = \{p'\}_{x'}\downarrow$. In particular, there is a $p \in st(X)$ and keyword x such that $r' = \{p\}_x\downarrow$. But this means that $r = \{p\}_{xk}\downarrow$. If u occurs in δ'', then since k is atomic, δ'' ends in an *analz* rule, and so there is a $q \in st(X)$ and keyword y such that $u = \{q\}_y\downarrow$. Thus we have proved that for every u occurring in δ, there is a $p \in st(X)$ and keyword x such that $u = \{p\}_x\downarrow$.
   b. Suppose δ' is a purely synthetic proof. We note that r' is a blind pair, and hence the last rule of δ' is not *encrypt* (since δ' is purely synthetic). The only other possibility is that the last rule of δ' is *blindpair*, but that would violate the normality of δ, as one of the transformations specified by the first row of Figure 8 would apply to δ. So this case is not possible.

- Suppose δ is of the following form:

$$\dfrac{\begin{array}{cc} \vdots\,\delta' & \vdots\,\delta'' \\ X \vdash \{r\}_k & X \vdash inv(k) \end{array}}{X \vdash r}\ \textit{decrypt}$$

We first note that $inv(k)$ is an atomic key and hence δ'' should end with the *analz* rule. Hence for every u occurring in δ'', there exists $p \in st(X)$ and a keyword x such that $u = \{p\}_x\downarrow$.
We now consider δ'. It cannot end in a *blindpair* rule, since the transformation rule in the second row of Figure 8 would apply to δ, thereby contradicting normality of δ. Nor can δ' end in an *encrypt* rule, since then the major premise of the last rule of δ' would be r, and this contradicts the minimality of δ. The only possibilities therefore are that δ' ends in an application of *split* or *decrypt* or *blindsplit*. In the first two cases, we know by induction hypothesis that the major premise r_1 of the last rule of δ' is in $st(X)$. Hence $\{r\}_k$, as well as r, are in $st(X)$ as well. We now consider the case when the last rule of δ' is *blindsplit*$_1$. Let r_1 be the major premise of this rule, and r_2 the minor premise. Now it cannot be the case that r_1 is of the form $[\{r\}_k, \{r'\}_k]$. For, in that case r_2 would have been $\{r'\}_k$, and the transformation rule in the second row of Figure 8 would apply to δ, and this contradicts its normality.

We also know from the induction hypothesis (applied to δ') that there is a $p \in st(X)$ and a keyword x such that $r_1 = \{p\}_x$. But since r_1 is $[\{r\}_k, r_2]$, where r_2 is not of the form $\{r'\}_k$ for any r', we conclude that $x = \varepsilon$ and $r_1 = p \in st(X)$. It follows that $r \in st(X)$ as well.

◀

4 The automaton construction

We recall here some definitions relating to alternating pushdown systems (APDSs) 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.

▶ **Definition 19.** An **alternating pushdown system** 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 $i_1,\ldots,i_n$ such that $i = i_1 + \cdots + i_n + 1$ and $x = yz$ and $(b_j, y_j z) \Rightarrow_{\mathscr{P},i_j} C$ for all $j \in \{1,\ldots,n\}$.

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

▶ **Definition 20.** 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 18 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):

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

This immediately suggests the construction of an alternating automaton $\mathscr{A}$ such that for every $t \in Y$ and keyword x, $x \in \mathscr{L}_t$ if and only if there is an accepting run of $\mathscr{A}$ on the word x from the state t. 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 an accepting run of $\mathscr{A}$ on ε from the state t_0.

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 labelled k from t to $\{t\}_k$, and there is an (AND-)edge labelled ε from t to the set $\{[t, t'], t'\}$. We introduce a final state f and introduce an ε-labelled edge from t to f whenever $\varepsilon \in \mathscr{L}_t$. But notice that if $kx \in \mathscr{L}_t$ then $x \in \mathscr{L}_{\{t\}_k}$, and this cannot be represented directly by a transition in the automaton. Thus we define a revised automaton by adding an edge labelled ε from $\{t\}_k$ to q whenever the original automaton has an edge labelled k from t to q. In fact, it does not suffice to stop after revising the automaton once. The procedure has to be repeated till no more new edges 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 9. Some examples that illustrate the saturation procedure are presented in Appendix A.

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

- $\hookrightarrow_0$ is the smallest subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ such that:
  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 4 (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:

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

Figure 9 The sequence of automata for analysing $X_0 \vdash t_0$, 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 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.

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

The sequence of automata for this example is given in the appendix.

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

▶ **Lemma 21.** 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 $q \overset{a}{\hookrightarrow}_{d'} C$.*

Proof. 1. We first compute $\overset{\varepsilon}{\Rightarrow}_i$ inductively as follows:
   - $q \overset{\varepsilon}{\Rightarrow}_{i,0} C$ if and only if $C = \{q\}$,
   - $q \overset{\varepsilon}{\Rightarrow}_{i,j+1} C$ if and only if either $q \overset{\varepsilon}{\Rightarrow}_{i,j} C$ or there is $C' \subseteq Q$ such that $q \overset{\varepsilon}{\hookrightarrow}_i C'$ and $C' \overset{\varepsilon}{\Rightarrow}_{i,j} C$.

   It is clear that $\overset{\varepsilon}{\Rightarrow}_i$ is computable in $d \cdot 2^d$ iterations of the above induction, each step taking at most $d \cdot 2^d$ time. Once $q \overset{\varepsilon}{\Rightarrow}_i$ is computed, $q \overset{a}{\Rightarrow}_i$ is computed inductively as follows (for $a \in \Sigma$):
   - $q \overset{a}{\Rightarrow}_{i,1} C$ if and only if $q \overset{a}{\hookrightarrow}_i C$,
   - $q \overset{a}{\Rightarrow}_{i,j+1} C$ if and only if $q \overset{a}{\Rightarrow}_{i,j} C$ or there exist $C', C'' \subseteq Q$ and k, ℓ such that $k + \ell = j$ and $q \overset{\varepsilon}{\hookrightarrow}_i C'$ and $C' \overset{a}{\Rightarrow}_{i,k} C'' \overset{\varepsilon}{\Rightarrow}_{i,\ell} C$.

   Again it is clear that $\overset{a}{\Rightarrow}_i$ is computed in time $2^{O(d)}$, once $\overset{\varepsilon}{\Rightarrow}_i$ has been computed. Thus the overall time needed is $2^{O(d)}$.
2. This is easily seen from the construction.
3. Observe that whenever $q \overset{a}{\Rightarrow}_i C$, it is also the case that $q \overset{a}{\Rightarrow}_{i+1} C$, and the number of possible triples in any $\Rightarrow_j$ is $d^2 \cdot 2^d$. Thus the desired statement follows.

◀

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 notation:

- 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 22** (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 23.** *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 path from x to C, but the difficulty with the proof is that in a run over x from t to C, each path 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.

Proof. **Case $i = 0$:** Suppose $t \overset{x}{\Rightarrow}_{0,d} C$, and either $C \subseteq Y_0$ or $X_0 \vdash y$. Now if $t \overset{x}{\Rightarrow}_{0,0} C$, it has to be the case that $x = \varepsilon$ and $C = D = \{t\}$. Then it is immediate that $X_0 \cup \{D\}_y \vdash \{t\}_{xy}$.
So suppose $x = ax'$ for some $a \in \Sigma \cup \{\varepsilon\}$, and there is a $C' \subseteq Q$ (with $D' = C' \cap Y_0$) such that $t \overset{a}{\hookrightarrow}_0 C' \overset{x'}{\Rightarrow}_{0,d'} C$ for some $d' < d$. Then by induction hypothesis (on d), $X_0 \cup \{D\}_y \vdash \{u\}_{x'y}$ for every $u \in D'$. So it suffices to prove that $X_0 \cup \{D'\}_{x'y} \vdash \{t\}_{ax'y}$. Now there are two main cases to consider:

- Suppose $a = k$ and $C' = D' = \{\{t\}_k\}$. Then it is clear that $\{D'\}_{x'y} \vdash \{\{t\}_k\}_{x'y}$.
- Suppose $a = \varepsilon$ and $C' = D' = \{[t, t'], t'\}$. Again it is immediate that $\{D'\}_{x'y} \vdash \{t\}_{x'y}$. The *blindsplit*$_0$ and *blindpair* cases are similar.

Case $i = j + 1$: Suppose $t \overset{x}{\Rightarrow}_{j+1,d} C$ and either $C \subseteq Y_0$ or $X \vdash y$. Either $t \overset{x}{\Rightarrow}_j C$ in which case we are done (by the induction hypothesis on i), or $d > 1$. In the second case, suppose $x = ax'$ for some $a \in \Sigma \cup \{\varepsilon\}$ and there is a $C' \subseteq Q$ (with $D' = C' \cap Q$) such that $t \overset{a}{\hookrightarrow}_{j+1} C' \overset{x'}{\Rightarrow}_{j+1,d'} C$ for some $d' < d$. Then by induction hypothesis (on d), $X_0 \cup \{D\}_y \vdash \{u\}_{x'y}$ for every $u \in D'$. So it suffices to prove that $X_0 \cup \{D'\}_{x'y} \vdash \{t\}_{ax'y}$.

We note that if f is in C', f is also in C, and that $f \overset{x'}{\Rightarrow}_{j+1} \{f\}$ (since $C' \overset{x'}{\Rightarrow}_{j+1,d'} C$), and $X \vdash y$ (since $C \nsubseteq Y_0$). But if $f \overset{x'}{\Rightarrow}_{j+1} \{f\}$, by definition of $\hookrightarrow_{j+1}$, it means that $k \overset{\varepsilon}{\hookrightarrow}_j \{f\}$ for every k occurring in x'. By induction hypothesis (on i), $X_0 \vdash k$ for each such k, and hence $X_0 \vdash x'$. Thus either $C' \subseteq Y_0$ or $X_0 \vdash x'y$. Now there are three cases to consider:

- Suppose $t \overset{a}{\Rightarrow}_j C'$. By induction hypothesis (on i), $X_0 \cup \{D'\}_{x'y} \vdash \{t\}_{ax'y}$.
- Suppose $t = \{t'\}_k$ and $a = \varepsilon$ and $t' \overset{k}{\Rightarrow}_j C'$. It follows that $X_0 \cup \{D'\}_{x'y} \vdash \{t'\}_{kx'y}$, by induction hypothesis (on i). Thus $X_0 \cup \{D'\}_{x'y} \vdash \{t\}_{x'y}$.
- Suppose $a = \varepsilon$, $C' = \{f\}$, $t \in Y_0$ is the conclusion of some rule with premises $\Gamma \subseteq Y_0$, and $p \overset{\varepsilon}{\Rightarrow}_j \{f\}$ for every $p \in \Gamma$. Since $p \overset{\varepsilon}{\Rightarrow}_j \{f\}$, we can apply the induction hypothesis (on i) taking $x = y = \varepsilon$ and conclude that $X_0 \vdash p$, for all $p \in \Gamma$. It follows that $X_0 \vdash t$. But since $C' \nsubseteq Y_0$, $X_0 \vdash x'y$. So $X_0 \vdash \{t\}_{x'y}$.

◀

▶ **Lemma 24.** *For all $t, t' \in Y_0$, $C \subseteq Y_0$, and keywords x, x' such that $\{t\}_x\downarrow = \{t'\}_{x'}\downarrow$, if $t \overset{x}{\Rightarrow}_i C$ for some i, then there is a $j \geq i$ such that $t' \overset{x'}{\Rightarrow}_j C$.*

Proof. There are two cases to consider.

- Suppose $x' = k_1 \cdots k_n x$, and thus $t = \{t'\}_{k_1 \cdots k_n}$. Then it is easy to see that:

$$t' \overset{k_1}{\hookrightarrow}_i \{t'\}_{k_1} \overset{k_2}{\hookrightarrow}_i \cdots \overset{k_n}{\hookrightarrow}_i \{t'\}_{k_1 \cdots k_n} \overset{x}{\Rightarrow}_i C.$$

- Suppose $x = k_1 \cdots k_n x'$, and thus $t' = \{t\}_{k_1 \cdots k_n}$. Suppose that

$$t \overset{k_1}{\Rightarrow}_i D_1 \overset{k_2}{\Rightarrow}_i D_2 \cdots D_{n-1} \overset{k_n}{\Rightarrow}_i D_n \overset{x'}{\Rightarrow}_i C.$$

Then, it is also the case that

$$\{t\}_{k_1} \overset{\varepsilon}{\hookrightarrow}_{i+1} D_1 \overset{k_2}{\Rightarrow}_i D_2 \cdots D_{n-1} \overset{k_n}{\Rightarrow}_i D_n \overset{x'}{\Rightarrow}_i C.$$

But then $\{t\}_{k_1} \overset{k_2}{\Rightarrow}_{i+1} D_2$ and so

$$\{t\}_{k_1 k_2} \overset{\varepsilon}{\hookrightarrow}_{i+2} D_2 \cdots D_{n-1} \overset{k_n}{\Rightarrow}_i D_n \overset{x'}{\Rightarrow}_i C.$$

Arguing likewise, we have

$$\{t\}_{k_1\cdots k_n} \overset{\varepsilon}{\hookrightarrow}_{i+n} D_n \overset{x'}{\Rightarrow}_i C.$$

Hence $t' \overset{x'}{\Rightarrow}_{i+n} C$, and we are done.

◀

▶ **Theorem 25** (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 $t \overset{x}{\Rightarrow}_i \{f\}$.*

Proof. The proof is by induction on the structure of (normal) proofs. Let π be a normal proof of $\{t\}_x\downarrow$ from X. The following cases need to be considered:

- Suppose the last rule r of π has premises $\Gamma \subseteq Y_0$ and conclusion $\{t\}_x\downarrow \in Y_0$. By induction hypothesis, there is an i such that for all $u \in \Gamma$, $u \overset{\varepsilon}{\Rightarrow}_i \{f\}$. But our construction guarantees that $\{t\}_x\downarrow \overset{\varepsilon}{\hookrightarrow}_{i+1} \{f\}$. By Lemma 24, this means that $t \overset{x}{\Rightarrow}_j \{f\}$ for some $j > i$.
  It follows by weak locality of normal proofs that this subsumes the cases where π ends in an application of the *Ax*, *pair*, *split*, and *decrypt* rules.
- Suppose π is the following proof:

$$\dfrac{\begin{array}{cc} \vdots\, \pi' & \vdots\, \pi'' \\ X \vdash \{t\}_{x'}\downarrow & X \vdash k \end{array}}{X \vdash \{t\}_{x'k}\downarrow}\; encrypt$$

  By induction hypothesis, there is an i such that $t \overset{x'}{\Rightarrow}_i \{f\}$ and $k \overset{\varepsilon}{\hookrightarrow}_i \{f\}$. Hence $f \overset{k}{\hookrightarrow}_{i+1} \{f\}$, and thus $t \overset{x'k}{\Rightarrow}_{i+1} \{f\}$.
- Suppose π ends in a $blindsplit_i$ rule or a *blindpair* rule. The reasoning in all three cases is similar. We consider the case when π has the following form:

$$\dfrac{\begin{array}{cc} \vdots\, \pi' & \vdots\, \pi'' \\ X \vdash [\{t\}_x\downarrow, t'] & X \vdash t' \end{array}}{X \vdash \{t\}_x\downarrow}\; blindsplit_1$$

  By Lemma 18, we know that $[\{t\}_x\downarrow, t']$ is of the form $\{r\}_y\downarrow$ for some $r \in Y_0$. But this r has to be of the form $[u, u']$. And therefore $t' = \{u'\}_y\downarrow$. Now by induction hypothesis, there is i such that $[u, u'] \overset{y}{\Rightarrow}_i \{f\}$ and $u' \overset{y}{\Rightarrow}_i \{f\}$. But by construction, $u \overset{\varepsilon}{\hookrightarrow}_0 \{[u, u'], u'\}$, and thus $u \overset{\varepsilon}{\hookrightarrow}_i \{[u, u'], u'\}$. Therefore $u \overset{y}{\Rightarrow}_i \{f\}$. But now $\{t\}_x\downarrow = \{u\}_y\downarrow$, and hence by Lemma 24, $t \overset{x}{\Rightarrow}_j \{f\}$ for some j.

◀

▶ **Theorem 26.** *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 21, 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 22), 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 25), 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 Normal proofs and lower bounds

In the last section, we saw that normal proofs play a crucial role in proving an upper bound for the term derivability problem. They turn out to play a crucial role in the proofs of both the complexity lower bound and proof size lower bound for the problem. In this section, we bring together a few notions and facts that prove useful in both lower bound proofs.

▶ **Definition 27.** Let $b_1, \ldots, b_n, b$ be nonces and let $y_1, \ldots, y_n, y$ be keywords. We define

$$\{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \Longrightarrow \{b\}_y$$

to be the following term:

$$[[\cdots[[\{b\}_y, \{b_1\}_{y_1}], \{b_1\}_{y_1}], \cdots, \{b_n\}_{y_n}], \{b_n\}_{y_n}].$$

We call such terms **rewrite terms**.

We define a very useful bit of notation that will be used in many places later:

For any rewrite term $t = \{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \Longrightarrow \{b\}_y$, we define t_i for $i \in \{0, \ldots, n\}$ and t_i' for $i \in \{1, \ldots, n\}$ as follows:

- $t_i = [[\cdots[[\{b\}_y, \{b_1\}_{y_1}], \{b_1\}_{y_1}], \cdots, \{b_i\}_{y_i}], \{b_i\}_{y_i}]$.
- $t_i' = [[\cdots[[\{b\}_y, \{b_1\}_{y_1}], \{b_1\}_{y_1}], \cdots, \{b_{i-1}\}_{y_{i-1}}], \{b_i\}_{y_i}]$.

In particular, $t_n = t$ and $t_0 = \{b\}_y$.

▶ **Definition 28.** A **rewrite system** is a tuple (N, K, e, X_1, X_2) where:

- N is a finite set of nonces,
- $K \cup \{e\}$ is a finite set of keys,
- $N \cap K = \varnothing$, $e \notin N \cup K$,
- for $k, k' \in N \cup K$, it is not the case that $inv(k) = k'$,
- X_1 is a finite set of terms, all of the form $\{a\}_{xe}$ with $a \in N$ and $x \in K^*$, and
- X_2 is a finite set of terms, all of the form $\{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \Longrightarrow \{b\}_y$ where $b, b_1, \ldots, b_n \in N$ and $y, y_1, \ldots, y_n \in K^*$.

▶ **Lemma 29.** *Suppose (N, K, e, X_1, X_2) is a rewrite system, with $X = X_1 \cup X_2 \cup K \cup \{e\}$. Suppose also that $\{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \Longrightarrow \{b\}_y$ is a rewrite term in X_2 and $z \in K^*$ such that*

$$\text{for all } i \leq n : X \vdash \{b_i\}_{y_i ze}.$$

Then $X \vdash \{b\}_{yze}$.

Proof. Suppose π_i is a derivation of $X \vdash \{b_i\}_{y_i ze}$, for $i \leq n$. The required derivation is given in Figure 10. ◀

We next seek to prove the converse of Lemma 29 – that is, whenever $\{a\}_{xe}$ is provable from X, it is either in X_1 or there is a rewrite term $\{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \Longrightarrow \{a\}_y$ in X_2, and $z \in K^*$ such that:

1. $x = yz$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{t_n}\,Ax \quad \dfrac{}{ze}\,Ax}{\{t_n\}_{ze}}\,encrypt \qquad \begin{array}{c}\vdots\ \pi_n\\ \{b_n\}_{y_n ze}\end{array}}{\{t_n'\}_{ze}}\,blindsplit \qquad \begin{array}{c}\vdots\ \pi_n\\ \{b_n\}_{y_n ze}\end{array}}{\{t_{n-1}\}_{ze}}\,blindsplit}{\begin{array}{c}\vdots\\ \dfrac{\{t_1\}_{ze} \qquad \begin{array}{c}\vdots\ \pi_1\\ \{b_1\}_{y_1 ze}\end{array}}{\{t_1'\}_{ze}}\,blindsplit \qquad \begin{array}{c}\vdots\ \pi_1\\ \{b_1\}_{y_1 ze}\end{array}\end{array}}
$$

$$
\dfrac{\{t_1'\}_{ze} \qquad \{b_1\}_{y_1 ze}}{\{b\}_{yze}}\,blindsplit
$$

Figure 10 Derivation of $X \vdash \{b\}_{yze}$

2. for all $i \leq n$, $\{b_i\}_{y_i ze}$ occurs in π.

Towards that, we first prove a strengthening of the weak locality theorem for rewrite systems.

▶ **Lemma 30.** *Suppose (N, K, e, X_1, X_2) is a rewrite system, and let $X = X_1 \cup X_2 \cup K \cup \{e\}$. Let π be a normal proof of $X \vdash \{a\}_{xe}$, for $a \in N$ and $x \in K^*$. Then any term u occurring in π is of the form $\{p\}_w$, for $p \in st(X)$ and $w \in K^* \cup K^*e$.*

Proof. Call a term t **short** if it is of the form $\{p\}_w$, where $p \in st(X)$ and $w \in K^* \cup K^*e$, and **long** if it is of the form $\{p\}_{wew'}$ where $w' \neq \varepsilon$. The weak locality property (and the fact that $\{a\}_{xe}$ is short) guarantees that every term occurring in π is either long or short. We wish to prove that every such u is short.

Since all terms are either long or short and since there is no pair term in $st(X \cup \{t\})$, the *pair* and *split* cannot occur in π. Nor can the *decrypt* occur, since none of the keys in K is an inverse of another key in K.

Now suppose there is a long term u occurring in π. Then, since the root of π is short, there is some subproof δ of π of the following form, where r is short, and at least one of r' and r'' is long.

$$
\dfrac{\begin{array}{c}\vdots\ \delta'\\ X \vdash r'\end{array} \quad \begin{array}{c}\vdots\ \delta''\\ X \vdash r''\end{array}}{X \vdash r}
$$

There are three cases to consider.

- Suppose δ is of the following form:

$$
\dfrac{\begin{array}{c}\vdots\ \delta'\\ X \vdash r'\end{array} \quad \begin{array}{c}\vdots\ \delta''\\ X \vdash r''\end{array}}{X \vdash [r', r'']}\,blindpair
$$

  But it can be easily seen that if one of r' and r'' is long, then so is $r = [r', r'']$, which contradicts our assumption that r is short.

- Suppose δ is of the following form:

$$
\dfrac{\begin{array}{c}\vdots\ \delta_1\\ X \vdash r'\end{array} \quad \begin{array}{c}\vdots\ \delta_2\\ X \vdash k\end{array}}{X \vdash \{r'\}_k{\downarrow}}\,encrypt
$$

Since $r'' = k$ is short, it has to be the case that r' is long, but in that case it is easily seen that $r = \{r'\}_k$ is also long, which is a contradiction.

- Suppose δ is of the following form:

$$\dfrac{\begin{array}{cc}\vdots\,\delta' & \vdots\,\delta'' \\ X \vdash [r, r''] & X \vdash r''\end{array}}{X \vdash r}\;\textit{blindsplit}$$

Suppose r'' is short, then $r' = [r, r'']$ is long. Suppose on the other hand that r'' is long. Then also it is easily seen that $[r, r'']$ is long, since no short term contains a long term as a subterm. Hence $[r, r'']$ is of the form $[\{p\}_{wew'}, \{p''\}_{wew'}]$, for $[p, p''] \in st(X)$ and $w' \neq \varepsilon$. But then it follows $r = \{p\}_{wew'}$ is itself long, which is again a contradiction.

◀

The following lemma constrains the structure of rules that occur in any normal proof of $X \vdash \{a\}_{xe}$.

▶ **Lemma 31.** *Suppose (N, K, e, X_1, X_2) is a rewrite system, and let $X = X_1 \cup X_2 \cup K \cup \{e\}$. Let π be a normal proof of $X \vdash \{a\}_{xe}$, for $a \in N$ and $x \in K^*$. Let δ be a subproof of π with root labelled r.*

1. *If δ ends with the encrypt rule, then $r = \{p\}_w$ for some $p \in X$ and keyword $w \in K^* \cup K^*e$.*
2. *If δ ends with the blindsplit rule, then*
   a. *its minor premise is not a blind pair term, and*
   b. *$r = \{p\}_{we}$, where $p \in st(X)$ and $w \in K^*$.*

Proof. Let π be a normal proof of $X \vdash \{a\}_{xe}$, and let δ be a subproof of π with root labelled r. We assume both parts of the lemma for all proper subproofs δ' of δ, and prove it for δ.

1. Suppose δ ends with the *encrypt* rule, and has the following structure:

$$\dfrac{\begin{array}{cc}\vdots\,\delta' & \vdots\,\delta'' \\ X \vdash r' & X \vdash \ell\end{array}}{X \vdash r}\;\textit{encrypt}$$

If δ' ends with the *encrypt* rule, then $r' = \{p\}_w$ for some $p \in X$. In that case we are done, since $r = \{p\}_{w\ell}$. δ' cannot end with the *blindpair* rule, since that violates normality of π. The other option is that δ' ends with the *blindsplit* rule, in which case r' is of the form $\{p\}_{we}$ (by part 2 of this lemma applied to δ'). But then $r = \{p\}_{we\ell}$, and that violates Lemma 30, so this case cannot arise.

2. Suppose δ ends with the *blindsplit* rule and has the following form:

$$\dfrac{\begin{array}{cc}\vdots\,\delta' & \vdots\,\delta'' \\ X \vdash r' & X \vdash r''\end{array}}{r}\;\textit{blindsplit}$$

   a. Suppose, towards a contradiction, that r'' is a blind pair term. Clearly r' is a blind pair term too, and is a subterm of $\{t\}_w$, for $t = \{b_1\}_{y_1} \wedge \cdots \{b_n\}_{y_n} \Longrightarrow \{b\}_y$ from X_2. Now there are two cases to consider based on the form of r'.

   Case 1 Suppose $r' = t_m$ for some $m \in \{1, \ldots, n\}$. Then δ is of the following form:

$$\dfrac{\begin{array}{cc}\vdots\,\delta' & \vdots\,\delta'' \\ X \vdash \{t_m\}_w & X \vdash \{t'_m\}_w\end{array}}{X \vdash \{b_m\}_{y_m w}}\;\textit{blindsplit}$$

Clearly $t'_m \notin X$ and hence δ'' does not end in an encrypt rule (by part 1 of this lemma applied to δ''). So it ends in a *blindpair* rule or a *blindsplit* rule. If it ends in a *blindpair*, the minor premise of the rule is $\{b_m\}_{y_m w}$, and this violates the normality of π. If it ends in a *blindsplit* rule, then by induction hypothesis, the minor premise is not a blind pair term, and so has to be $\{b_m\}_{y_m w}$ again, which violates the normality of π.

Case 2 Suppose $r' = \{t'_m\}_w$ for some $m \in \{1,\ldots,n\}$. Then δ is of the following form:

$$\dfrac{\dfrac{\vdots\, \delta'}{X \vdash \{t'_m\}_w} \quad \dfrac{\vdots\, \delta''}{X \vdash \{t_{m-1}\}_w}}{\{b_m\}_{y_m w}}\,\textit{blindsplit}$$

Clearly $t'_m \notin X$ and hence δ' does not end in an *encrypt* rule (again by part 1 of this lemma, applied to δ' this time). It cannot end in a *blindpair* rule, since that violates normality, and hence ends in a *blindsplit* rule. But then the minor premise of that rule is not a blind pair term (by induction hypothesis), and hence has to be $\{b_m\}_{y_m w}$, violating the normality of π again.

Thus the minor premise of δ cannot be a blind pair term.

b. We now seek to show that $r = \{p\}_{we}$ for some $p \in st(X)$ and $w \in K^*$. Observe that δ' has to end with the *blindsplit* or the *encrypt* rule, and also that δ'' has to either end with the *encrypt* or *blindsplit* rule, since it cannot end with the *blindpair* rule (by what we just proved above). If δ'' ends with the *blindsplit* rule, then by induction hypothesis, $r'' = \{p\}_{w''e}$ for $p \in st(X)$ and $w'' \in K^*$. If, on the other hand, δ'' ends with the *encrypt* rule, then $r'' = \{p''\}_z$ for $p'' \in X$. But p'' is not a blind pair term, so $p'' \in X_1$. But then $p = \{a\}_{w''e}$ for $a \in N$ and $w'' \in K^*$, and hence $z = \varepsilon$, since otherwise Lemma 30 is violated. So in either case r'' is of the form $\{p''\}_{w''e}$.

But now e is a subterm of the blind pair term r', and since e is not a subterm of any blind pair term in X, r' has to be of the form $\{p'\}_{w'e}$. Thus r, the result of the *blindsplit* rule, is of the form $\{p\}_{we}$.

◀

In particular, it follows from the above that such proofs never have an application of the *blindpair* rule. Since the conclusion is not a blind pair term, the last rule of π is either *blindsplit* or *encrypt*. Then there has to be a *blindpair* rule whose conclusion is either the major premise of a *blindsplit* or *encrypt* rule, or a minor premise of a *blindsplit* rule. But the first case violates normality of π, and the second has just been proved above to be impossible. So we can assume that the only rules in such proofs are *Ax*, *blindsplit*, and *encrypt*.

We now prove an important property of normal proofs from rewrite systems – namely that whenever the "conclusion" of a rewrite term is provable, all the "premises" are provable too.

▶ **Lemma 32.** *Suppose (N,K,e,X_1,X_2) is a rewrite system, and let $X = X_1 \cup X_2 \cup K \cup \{e\}$. Let π be a normal proof of $X \vdash \{a\}_{xe}$, for $a \in N$ and $x \in K^*$. Then either $\{a\}_{xe} \in X_1$ or there is a rewrite term $\{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \implies \{a\}_y$ in X_2, and $z \in K^*$ such that:*

1. $x = yz$
2. *for all $i \leq n$, $\{b_i\}_{y_i ze}$ occurs in π.*

Proof. Let π be a normal proof of $X \vdash \{a\}_{xe}$ and suppose that $\{a\}_{xe} \notin X_1$.

For any term $t = \{b_1\}_{y_1} \wedge \cdots, \{b_n\}_{y_n} \implies \{a\}_y$ from X_2 and $r \in st(t)$, define *residues*(t,r) as follows:

- If $r = t_m$ for $m \in \{0,\ldots,n\}$, *residues*$(t,r) = \{\{b_{m+1}\}_{y_{m+1}},\ldots,\{b_n\}_{y_n}\}$. It is easy to see that *residues*$(t,t) = \emptyset$ and *residues*$(t,\{a\}_y) = \{\{b_1\}_{y_1},\ldots,\{b_n\}_{y_n}\}$.

- If $r = t'_m$ for $m \in \{1,\ldots,n\}$, $\mathit{residues}(t,r) = \{\{b_m\}_{y_m},\ldots,\{b_n\}_{y_n}\}$.

We prove that whenever $\{r\}_{ze}$ occurs as the root of a subproof δ of π, for any $r \in \mathit{st}(X_2)$, there is $t \in X_2$ such that $\{c\}_{yze}$ occurs in a proper subproof of δ for every $\{c\}_y \in \mathit{residues}(t,r)$. The statement of the lemma follows immediately. We distinguish the following three cases:

- Suppose δ ends in an *Ax* rule. Then $\{r\}_{ze} \in X$, which is a contradiction.
- Suppose δ ends in an *encrypt* rule. Then $\{r\}_{ze} = \{p\}_{we}$ for some $p \in X$. But then it has to be the case that $r \in X_2$. Notice that $\mathit{residues}(r,r) = \emptyset$, and our statement is vacuously true.
- Suppose δ ends in an *blindsplit* rule. Let $\{u\}_{ze}$ be the major premise and $\{b_m\}_{y_m ze}$ be the minor premise (by the previous lemma, the minor premise is not a blind pair and therefore has to be a term of this form). By induction hypothesis, there is $t \in X_2$ such that $\{c\}_{yze}$ occurs in δ for every $\{c\}_y \in \mathit{residues}(t,u)$. We distinguish two cases now.
  - $\mathit{residues}(t,r) = \mathit{residues}(t,u)$. In this case we are done.
  - $\mathit{residues}(t,r) = \mathit{residues}(t,u) \cup \{\{b_m\}_{y_m}\}$. In this case also we are done, since $\{b_m\}_{y_m ze}$ occurs as a minor premise.

◀

6 A dexptime lower bound for the derivability problem

We recall the following fact about alternating pushdown systems.

▶ **Theorem 33** ([21]). *The* **reachability problem for alternating pushdown systems**, *which asks, given an APDS $\mathscr{P}$ and configurations (s,x_s) and (f,x_f), whether $(s,x_s) \Rightarrow_{\mathscr{P}} (f,x_f)$, is* DEXPTIME*-complete.*

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

▶ **Definition 34.** Suppose $\mathscr{P} = (P,\Gamma,\hookrightarrow)$ is an APDS, and (s,x_s) and (f,x_f) are two configuration of $\mathscr{P}$. Then the corresponding rewrite system is given by (P,Γ,e,X_1,X_2) where:

- P is taken to be a set of nonces, which are not keys,
- $\Gamma \cup \{e\}$ is taken to be a set of non-symmetric keys, such that $e \notin \Gamma$, and none of the keys in $\Gamma \cup \{e\}$ is an inverse of another,
- $X_1 = \{\{f\}_{x_f e}\}$, and
- $X_2 = \{\{b_1\}_{x_1} \wedge \cdots \wedge \{b_n\}_{x_n} \Longrightarrow \{p\}_x \mid (a,x) \hookrightarrow_{\mathscr{P}} \{(b_1,x_1),\ldots,(b_n,x_n)\}\}$.

$X = X_1 \cup X_2 \cup \Gamma \cup \{e\}$.

We claim that $(s,x_s) \Rightarrow_{\mathscr{P}} (f,x_f)$ iff $X \vdash \{s\}_{x_s e}$. This follows from the following two lemmas.

▶ **Lemma 35.** *For all configurations (a,x) and all $i \geq 0$, if $(a,x) \Rightarrow_i \{(f,x_f)\}$ then $X \vdash \{a\}_{xe}$.*

Proof. We prove this by induction on i. If $i = 0$ then $(a,x) = (f,x_f)$ and thus $X \vdash \{a\}_{xe}$, since $\{f\}_{x_f e} \in X$. If $i > 0$, there is a rule of $\mathscr{P}$, $(a,y) \hookrightarrow \{(b_1,y_1),\ldots,(b_n,y_n)\}$, $z \in \Gamma^*$, and $i_1,\ldots,i_n \geq 0$ such that $x = yz$ and $(c_j,y_jz) \Rightarrow_{i_j} \{(f,x_f)\}$ for all $j \in \{1,\ldots,n\}$, and such that $i = i_1 + \cdots + i_n + 1$. By induction we know that $X \vdash \{b_j\}_{y_j ze}$ for all j. It immediately follows from the definition of X_2 and Lemma 29 that $X \vdash \{a\}_{yze}$. Since $x = yz$, $X \vdash \{a\}_{xe}$. ◀

▶ **Lemma 36.** *For any configuration (a,x), if there is a normal proof of $X \vdash \{a\}_{xe}$, then*

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

Proof. By Lemma 32, $X \vdash \{a\}_{xe}$ means that either $\{a\}_{xe} \in X_1$ or there is a $z \in K^*$ and a rewrite term $\{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \Longrightarrow \{a\}_y$ in X_2 such that:

1. $x = yz$
2. for all $i \leq n$, $\{b_i\}_{y_i ze}$ occurs in π.

In the first case, $a = f$ and $x = x_f$, and it follows that $(a,x) \Rightarrow_{\mathscr{P}} (f, x_f)$.

In the second case, by induction hypothesis, $(b_i, y_i z) \Rightarrow_{\mathscr{P}} (f, x_f)$, for all $i \leq n$. Combined with $(a,y) \hookrightarrow \{(b_1,y_1),\ldots,(b_n,y_n)\}$, it follows that $(a,x) = (a,yz) \Rightarrow_{\mathscr{P}} (f, x_f)$. ◀

And the following theorem is the end result.

▶ **Theorem 37.** *The passive intruder deduction problem is* DEXPTIME*-hard.*

7 Exponential lower bounds for proof size

▶ **Theorem 38.** *For every n, there exist X_n, t_n such that:*
1. *$|X_n \cup \{t_n\}|$ is $O(n)$*
2. *$X_n \vdash t_n$*
3. *Any proof of $X_n \vdash t_n$ is of size at least 2^n.*

The idea is to show that every normal proof of $X_n \vdash t_n$ has to contain a term with an encryption chain of length at least 2^n. Since one requires an exponential number of applications of the encrypt rule to generate this term, the proof itself is of exponential size. Further, since any proof can be converted to a normal proof all of whose terms occur in the original proof, all proofs of $X_n \vdash t_n$ are of exponential size. In what follows, we fix n and refer to X_n as just X, to reduce clutter. X consists of three parts: X_0 consists of the *counters*, which seek to “count” by building a list containing all n-bit numbers, and X_2 consists of the *validators*, which check that the count is correct by verifying whether each pair of adjacent numbers differ by one. To help in this process, X_2 needs the help of terms in X_1, which are the *bit-verifiers*. The list is built as a sequence of encryptions using the keys K. There are exactly four keys, of which k_0 and k_1 stand for the bits 0 and 1, k acts as a marker between adjacent numbers, and k' acts as an endmarker. The coding here is similar in spirit to the simulation of an ASPACE(n) Turing machine by an alternating pushdown automaton [6].

For $m \in \{0,\ldots,2^n - 1\}$, we use $\underline{m}$ to denote the **reverse** of the n-bit representation of m. For example, $\underline{5}$ is 1010^{n-3} and $\underline{23}$ is 111010^{n-5}.

▶ **Definition 39.** The **exponential counter** is given by the set of terms $X = X_0 \cup X_1 \cup X_2 \cup K$ where:
- $K = \{k_0, k_1, k, k'\}$. (We use L to denote $K \setminus \{k'\}$.)
- X_0, the set of **counters**, consists of the following terms (here and in what follows, we use $\{t\}_0$ as shorthand for $\{t\}_{k_0}$ and $\{t\}_1$ as shorthand for $\{t\}_{k_1}$):
  - $\{a\}_{k'}$
  - $a \Longrightarrow \{b_1\}_0, b_1 \Longrightarrow \{b_2\}_0, \ldots, b_{n-1} \Longrightarrow \{b_n\}_0$
  - $a \Longrightarrow \{b_1\}_1, b_1 \Longrightarrow \{b_2\}_1, \ldots, b_{n-1} \Longrightarrow \{b_n\}_1$
  - $b_n \Longrightarrow \{a\}_k, a \Longrightarrow \{c\}_{\underline{2^n-1}}$
- X_1, the set of **bit-verifiers**, consists of the following terms (where $\ell \in L$):
  - $\{e\}_{k'}, e \Longrightarrow \{e\}_\ell$
  - $e \Longrightarrow \{f_0\}_0, f_0 \Longrightarrow \{f_1\}_\ell, \ldots, f_{n-1} \Longrightarrow \{f_n\}_\ell$
  - $e \Longrightarrow \{g_0\}_1, g_0 \Longrightarrow \{g_1\}_\ell, \ldots, g_{n-1} \Longrightarrow \{g_n\}_\ell$
- X_2, the set of **validators**, consists of the following terms:
  - $\{d\}_k \Longrightarrow c$
  - $\{c\}_0 \wedge g_n \Longrightarrow c, \{c\}_1 \wedge f_n \Longrightarrow d$
  - $\{d\}_0 \wedge f_n \Longrightarrow d, \{d\}_1 \wedge g_n \Longrightarrow d$

▶ Remark. Note that $ExpCount = (N, L, k', Y_1, Y_2)$ is a rewrite system, where:

- $N = \{a, b_1, \ldots, b_n, c, d, e, f_0, \ldots, f_n, g_0, \ldots, g_n\}$
- $L = \{k_0, k_1, k\}$
- $Y_1 = \{\{a\}_{k'}, \{e\}_{k'}\}$
- $Y_2 = (X_0 \cup X_1 \cup X_2) \setminus Y_1$.

▶ **Lemma 40.** $X \vdash \{c\}_{\underline{0}k'}$.

Proof. The derivation is huge, and involves repeated use of Lemma 29. We give the overall structure of the derivation below.

Step 1 We first show that for any $m \in \{0, \ldots, 2^n - 1\}$ and any $x \in L^*$,

$$\text{if } X \vdash \{a\}_{xk'} \text{ then } X \vdash \{a\}_{k\underline{m}xk'}.$$

Suppose $\underline{m} = \alpha_0 \cdots \alpha_{n-1}$. Here is the required derivation. The fist line is what is assumed, and all the other lines follow from the previous ones by Lemma 29.

$$\begin{array}{l}
X \vdash \{a\}_{xk'} \\
X \vdash \{b_1\}_{\alpha_{n-1}xk'} \\
X \vdash \{b_2\}_{\alpha_{n-2}\alpha_{n-1}xk'} \\
\cdots \\
X \vdash \{b_n\}_{\alpha_0 \cdots \alpha_{n-1}xk'} (= \{b_n\}_{\underline{m}xk'}) \\
X \vdash \{a\}_{k\underline{m}k'}
\end{array}$$

Step 2 We now show that $X \vdash \{c\}_{\underline{2^n-1}k\underline{2^n-2}k\cdots k\underline{1}k\underline{0}k'}$. Here is a derivation. The first line is by the *Ax* rule, and the rest follow by from the preceding lines by Step 1.

$$\begin{array}{l}
X \vdash \{a\}_{k'} \\
X \vdash \{a\}_{k\underline{0}k'} \\
X \vdash \{a\}_{k\underline{1}k\underline{0}k'} \\
\cdots \\
X \vdash \{a\}_{k\underline{2^n-2}k\cdots k\underline{1}k\underline{0}k'} \\
X \vdash \{c\}_{\underline{2^n-1}k\underline{2^n-2}k\cdots k\underline{1}k\underline{0}k'}
\end{array}$$

Step 3 We next show that for all $i \in \{0, \ldots, n-1\}$, $x = \ell_0 \cdots \ell_{m-1} \in L^*$, and $y, z \in \{0, 1\}^*$ such that $|y| = n - 1 - i$ and $|z| = i$,

$$X \vdash \{f_n\}_{ykz0xk'} \text{ and } X \vdash \{g_n\}_{ykz1xk'}.$$

Here is a derivation of $\{f_n\}_{ykz0xk'}$. The other derivation is similar.

$$\begin{array}{l}
X \vdash \{e\}_{k'} \\
X \vdash \{e\}_{\ell_{m-1}k'} \\
X \vdash \{e\}_{\ell_{m-2}\ell_{m-1}k'} \\
\cdots \\
X \vdash \{e\}_{xk'} \\
X \vdash \{f_0\}_{0xk'} \\
\cdots \\
X \vdash \{f_i\}_{z0xk'} \\
X \vdash \{f_{i+1}\}_{kz0xk'} \\
X \vdash \{f_n\}_{ykz0xk'}
\end{array}$$

Step 4 We next show that for $r \in \{0, \ldots, n-1\}$ and any $y \in \{0, 1\}^*$ with $|y| = n - r - 1$, and any $x \in L^*$,

$$\text{if } X \vdash \{c\}_{0^r 1yk1^r 0ykxk'}, \text{ then } X \vdash \{c\}_{1^r 0ykxk'}.$$

We illustrate the derivation with an example. Suppose $n = 6$, $r = 2$, and $y = 010$. Suppose also that $X \vdash \{c\}_{001010k110010kxk'}$. Here is a derivation of $\{c\}_{110010kxk'}$.

Line 1.	$X \vdash \{c\}_{001010k110010kxk'}$	(by assumption)
Line 2.	$X \vdash \{g_6\}_{01010k110010kxk'}$	(from Step 2)
Line 3.	$X \vdash \{c\}_{01010k110010kxk'}$	(by Lemma 29 and lines 1 and 2)
Line 4.	$X \vdash \{g_6\}_{1010k110010kxk'}$	(from Step 2)
Line 5.	$X \vdash \{c\}_{1010k110010kxk'}$	(by Lemma 29 and lines 3 and 4)
Line 6.	$X \vdash \{f_6\}_{010k110010kxk'}$	(from Step 2)
Line 7.	$X \vdash \{d\}_{010k110010kxk'}$	(by Lemma 29 and lines 5 and 6)
Line 8.	$X \vdash \{f_6\}_{10k110010kxk'}$	(from Step 2)
Line 9.	$X \vdash \{d\}_{10k110010kxk'}$	(by Lemma 29 and lines 7 and 8)
Line 10.	$X \vdash \{g_6\}_{0k110010kxk'}$	(from Step 2)
Line 11.	$X \vdash \{d\}_{0k110010kxk'}$	(by Lemma 29 and lines 9 and 10)
Line 12.	$X \vdash \{f_6\}_{k110010kxk'}$	(from Step 2)
Line 13.	$X \vdash \{d\}_{k110010kxk'}$	(by Lemma 29 and lines 11 and 12)
Line 14.	$X \vdash \{c\}_{110010kxk'}$	(by Lemma 29 and line 13)

Step 5 We derive $\{c\}_{\underline{2^n-1}k\underline{2^n-2}k\cdots k\underline{1}k\underline{0}k'}$ and repeat Step 4 $2^n - 1$ times to derive $\{c\}_{\underline{0}k'}$. ◀

We now prove the lower bound on the size of any normal proof of $X \vdash \{c\}_{\underline{0}k'}$. This is a long proof, which we break down into a sequence of lemmas. The key lemmas involve showing that if a term t occurs in such a proof, then a different term t' of some desired kind also occurs in the same proof. Eventually we prove that a term with an exponentially long encryption sequence occurs in the proof, and that will do the job.

The next lemma summarizes the important structural constraint imposed by the counters in X – there are exactly n bits between any two occurrences of the marker k in any keyword occurring in a normal proof.

▶ **Lemma 41.** *Let π be a normal proof of $X \vdash \{c\}_{\underline{0}k'}$.*

1. *If $\{a\}_{xk'}$ occurs in π then either $x = \varepsilon$ or x is of the form $ky_0k\cdots ky_r$, where 1) $r \geq 0$, 2) each $y_i \in \{0,1\}^*$, and 3) $|y_i| = n$ for each $i \geq 0$.*
2. *For $j \in \{1,\dots,n\}$, if $\{b_j\}_{xk'}$ occurs in π then x is of the form $y_0ky_1\cdots ky_r$, where 1) $r \geq 0$, 2) each $y_i \in \{0,1\}^*$, 3) $|y_i| = n$ for each $i \geq 1$, and 4) $|y_0| = j$.*
3. *For $p \in \{c,d\}$, if $\{p\}_{xk'}$ occurs in π then x is of the form $y_0ky_1\cdots ky_r$, where 1) $r \geq 0$, 2) each $y_i \in \{0,1\}^*$, 3) $|y_i| = n$ for each $i \geq 1$, and 4) $|y_0| \leq n$.*

Proof. Let δ be a subproof of π. We assume the statement of the lemma for all terms occurring in all proper subproofs of δ and prove it for δ itself. Clearly we only need to consider the last rule of δ.

1. Suppose the root of δ is labelled $\{a\}_{xk'}$. It follows from Lemma 32 (specialized to the rewrite system *ExpCount*) that either $x = \varepsilon$ or $x = ky$ and $\{b_n\}_{yk'}$ occurs in a proper subproof of δ. Hence the statement of the lemma applies to $\{b_n\}_{yk'}$. Therefore $y = y_0ky_1\cdots ky_r$, where 1) $r \geq 0$, 2) each $y_i \in \{0,1\}^*$, 3) $|y_i| = n$ for each $i \geq 1$, and 4) $|y_0| = n$. The corresponding statement for x follows immediately.
2. Suppose the root of δ is labelled $\{b_1\}_{xk'}$. It follows from Lemma 32 that $x = \alpha y$ and $\{a\}_{yk'}$ occurs in a proper subproof of δ. Hence the statement of the lemma applies to $\{a\}_{yk'}$. Therefore $y = ky_0ky_1\cdots ky_r$, where 1) $r \geq 0$, 2) each $y_i \in \{0,1\}^*$, 3) $|y_i| = n$ for each $i \geq 0$. The corresponding for x follows immediately.
   We make a similar argument for $j > 1$.
3. Suppose the root of δ is labelled $\{c\}_{xk'}$. It follows from Lemma 32 that the following cases can arise:

Case 1: $x = \underline{2^n-1}y$ and $\{a\}_{yk'}$ occurs in a proper subproof of δ. Hence the statement of the lemma applies to it. Therefore $y = ky_0ky_1\cdots ky_r$, where 1) $r \geq 0$, 2) each $y_i \in \{0,1\}^*$, 3) $|y_i| = n$ for each $i \geq 0$. The corresponding statement for x follows immediately.

Case 2: $\{d\}_{kxk'}$ occurs in a proper subproof of δ. Now kx has the structure as stated in the lemma, and the corresponding statement for x follows.

Case 3: $\{c\}_{0x}$ occurs in a proper subproof of δ. Now $0x$ has the structure as stated in the lemma, and the corresponding statement for x follows.

The corresponding statement for $\{d\}_x$ is proved similarly.

◀

The next two lemmas summarize the exact conditions on x such that $X \vdash \{p\}_{xk'}$ occurs in π, for $p \in \{c,d,f_0,\ldots,f_n,g_0,\ldots,g_n\}$.

▶ **Lemma 42.** *Let π be a normal proof of $X \vdash \{c\}_{\underline{0}k'}$.*

1. *For $j \in \{0,\ldots,n\}$, if $\{f_j\}_{xk'}$ occurs in π, then $x = y0z$, where $y \in L^*$ such that $|y| = j$.*
2. *For $j \in \{0,\ldots,n\}$, if $\{g_j\}_{xk'}$ occurs in π, then $x = y1z$, where $y \in L^*$ such that $|y| = j$.*

Proof. Let δ be a subproof of π with root labelled r. We assume the statement of the lemma for all terms occurring in all proper subproofs of δ and prove it for δ itself. Clearly we only need to consider the last rule of δ. We prove the claim for terms of the form $\{f_j\}_{xk'}$. The proof for $\{g_j\}_{xk'}$ are identical.

Suppose $r = \{f_0\}_{xk'}$. Then by Lemma 32 (specialized to the rewrite system *ExpCount*), $x = 0z$ for $z \in K^*$, as desired.

Suppose $r = \{f_{j+1}\}_{xk'}$ for $j \in \{0,\ldots,n-2\}$. Then by Lemma 32, $x = \ell x'$ (for $\ell \in L$), and $\{f_j\}_{x'k'}$ occurs in a proper subproof of δ. Hence the statement of this lemma tells us that $x' = y0z$ with $y \in L^*$ and $|y| = j$. . Thus $x = \ell y0z$ is also of the desired form. ◀

▶ **Lemma 43.** *Let π be a normal proof of $X \vdash \{c\}_{\underline{0}k'}$. Suppose $x = ykzw$ where $y,z \in \{0,1\}^*$ such that $|z| = i$ and $|y| = n-i$ for some $i \in \{0,\ldots,n-1\}$, and $w \in L^*$. Then:*

1. *if $\{c\}_{xk'}$ occurs in π, then $z = 1^i$.*
2. *if $\{d\}_{xk'}$ occurs in π, then $z \neq 1^i$.*

Proof. As usual, we prove by induction on the size of the subproof of π in which the term occurs. So suppose δ is a subproof of π with root labelled r. We will consider the case when $r = \{c\}_{xk'}$. The case when $r = \{d\}_{xk'}$ is handled along the same lines.

By Lemma 32 (specialized to the rewrite system *ExpCount*), there are three cases.

Case 1: $y = \underline{2^n-1}$. In this case, $z = \varepsilon$ and the statement is vacuously true.

Case 2: $\{d\}_{kxk'}$ occurs in a proper subproof of δ. In this case too, it follows that $|y| = n$ and $z = \varepsilon$, and the statement holds vacuously.

Case 3: $\{c\}_{0xk'}$ and $\{g_n\}_{xk'}$ occur in proper subproofs of δ. Since $\{g_n\}_{xk'}$ occurs in δ, it has to be the case that $x = w'1w$ with $|w'| = n$. Thus, it follows that $z = z'1$ (since $|ykz| = |w'1|$ and they are both prefixes of x). But we also know that $\{c\}_{0x}$ occurs in a proper subproof of δ. So the lemma applies to $0x = 0ykz'1w$, and hence $z' = 1^{i-1}$. Thus $z = 1^i$.

◀

Till now, we just constrained the structure of terms occurring in a normal proof of $X \vdash \{c\}_{\underline{0}k'}$. Now, we come to the crucial statement, which says that if a term with a certain keyword occurs in π, then another term with a longer keyword occurs in π.

▶ **Lemma 44.** *Let π be a normal proof of $X \vdash \{c\}_{\underline{0}k'}$. Then*

1. *If* $\{c\}_{xk'}$ *occurs in* π *for* $x = \alpha_0 \cdots \alpha_{n-1} k y$ *and* $\alpha_i = 0$ *for some* $i \in \{0, \ldots, n-1\}$*, then* $\{d\}_{kx}$ *occurs in* π*.*
2. *Suppose* $x = \alpha_{i+1} \cdots \alpha_{n-1} k \beta_0 \cdots \beta_{n-1} k y$ *for* $i \in \{0, \ldots, n-1\}$*, and* $\beta_j = 0$ *for some* $j \in \{0, \ldots, n-1\}$*. Then:*
   a. *if* $\{c\}_{xk'}$ *occurs in* π*, then* $\{c\}_{0xk'}$ *also occurs in* π*.*
   b. *if* $\{d\}_{xk'}$ *occurs in* π *and* $\beta_j = 1$ *for all* $j < i$*, then* $\{c\}_{1xk'}$ *also occurs in* π*.*
   c. *if* $\{d\}_{xk'}$ *occurs in* π *and* $\beta_j = 0$ *for some* $j < i$*, then* $\{d\}_{\beta_i x k'}$ *also occurs in* π*.*

Proof. 1. It is given that $\alpha_0 \cdots \alpha_{n-1} \neq \underline{2^n - 1}$. So the only cases that can arise are that $\{d\}_{kxk'}$ occurs earlier in the proof, or that $\{c\}_{0xk'}$ occurs earlier. But $\{c\}_{0x}$ cannot occur, since $0\alpha_0 \cdots \alpha_{n-1}$ is of length $n+1$, and that is a contradiction. Therefore $\{d\}_{kx}$ occurs in π.

2. a. Suppose $\{c\}_{xk'}$ occurs in π, for $i \in \{0, \ldots, n-1\}$ and $x = \alpha_{i+1} \cdots \alpha_{n-1} k \beta_0 \cdots \beta_{n-1} k y k'$, such that $\beta_j = 0$ for some $j \in \{0, \ldots, n-1\}$. Since $\alpha_{i+1} \cdots \alpha_{n-1}$ is of length smaller than n, neither $\{d\}_{kxk'}$ nor $\{a\}_{kyk'}$ can occur in δ. It then follows that $\{c\}_{0xk'}$ occurs in a proper subproof of π.
   b. Suppose $\{d\}_{xk'}$ occurs in π, for x as above, and suppose $\beta_j = 1$ for all $j < i$. Then it cannot be the case that $\{d\}_{\alpha_i x k'}$ occurs in π, as that would violate the previous lemma. Therefore it has to be the case that $\{c\}_{1xk'}$ and $\{f_n\}_{xk'}$ occur in proper subproofs of π.
   c. Suppose $\{d\}_{xk'}$ is the conclusion of a *blindsplit* in π, for x as above, and suppose $\beta_j = 0$ for some $j < i$. Then it cannot be the case that $\{c\}_{\alpha_i x k'}$ occurs in π, as that would violate the previous lemma. Therefore it has to be the case that either $\{d\}_{0xk'}$ and $\{f_n\}_{xk'}$ occur in proper subproofs of π, or that $\{d\}_{1xk'}$ and $\{g_n\}_{xk'}$ occur in proper subproofs of π. If $\{f_n\}_{xk'}$ occurs, then $\beta_i = 0$, but then $\alpha_i = 0$ as well. If $\{g_n\}_x$ occurs, then $\beta_i = 1$, but then $\alpha_i = 1$ as well. Thus $\{d\}_{\beta_i x k'}$ occurs in π.

◀

▶ **Lemma 45.** *Let* π *be a normal proof of* $X \vdash \{c\}_{\underline{0}k'}$*. Suppose* $\{c\}_{\underline{m}xk'}$ *occurs in* π*, and* $m \neq 2^n - 1$*. Then* $\{c\}_{\underline{m+1}k\underline{m}xk'}$ *also occurs in* π*.*

Proof. Let $\underline{m} = 1^r 0 \beta_{r+1} \cdots \beta_{n-1}$, so that $\underline{m+1} = 0^r 1 \beta_{r+1} \cdots \beta_{n-1}$.

Firstly, by item (1) of the previous lemma, $\{d\}_{k\underline{m}xk'}$ occurs in π. Now by applying item (2c) of the previous lemma repeatedly, we can conclude that

$$\{d\}_{\beta_{r+1} \cdots \beta_{n-1} k 1^r 0 \beta_{r+1} \cdots \beta_{n-1} x k'}$$

occurs in π. Applying item (2b) of the previous lemma now gives us that

$$\{c\}_{1 \beta_{r+1} \cdots \beta_{n-1} k 1^r 0 \beta_{r+1} \cdots \beta_{n-1} x k'}$$

occurs in π. We now apply item (2a) of the previous lemma repeatedly to get that

$$\{c\}_{0^r 1 \beta_{r+1} \cdots \beta_{n-1} k 1^r 0 \beta_{r+1} \cdots \beta_{n-1} x k'}$$

occurs in π.

But $0^r 1 \beta_{r+1} \cdots \beta_{n-1}$ is precisely $\underline{m+1}$. Hence the lemma is proved. ◀

The following lemma, which is our main goal, is an immediate consequence of the above.

▶ **Lemma 46.** *Let* π *be a normal proof of* $X \vdash \{c\}_{\underline{0}k'}$*. Then* $\{c\}_{\underline{2^n-1}k\underline{2^n-2}k \cdots k\underline{1}k\underline{0}k'}$ *occurs in* π*.*

Since $\{c\}_{\underline{2^n-1}k\underline{2^n-2}k \cdots k\underline{1}k\underline{0}k'}$ it occurs in any proof whose normalization is π. Thus any proof of $X \vdash \{c\}_{underline0k'}$ contains a term which has an exponentially long chain of encryption. Since building such a term involves exponentially many applications of the encryption rule, any proof of $X \vdash \{c\}_{\underline{0}k'}$ contains exponentially many terms.

8 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 [18], where $+$ is treated as an abelian group operator.

The decidability results in [18] 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 extend 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.

To illustrate the difference between our system and the system in [18], consider the derivation in Example 8. In the notation of their paper, the set X of that example is:

$$\{k, a+\{b\}_k, b+\{c\}_k, c+\{d\}_k, e+\{d\}_k, f+\{e\}_k, \{f\}_k\}$$

The following is a derivation of a from X.

$$
\begin{prooftree}
\AxiomC{}\UnaryInfC{$a+\{b\}_k$}
\AxiomC{}\UnaryInfC{$b+\{c\}_k$}
\AxiomC{}\UnaryInfC{$c+\{d\}_k$}
\AxiomC{}\UnaryInfC{$e+\{d\}_k$}
\BinaryInfC{$c-e$}
\AxiomC{}\UnaryInfC{k}
\BinaryInfC{$\{c\}_k-\{e\}_k$}
\AxiomC{}\UnaryInfC{$f+\{e\}_k$}
\BinaryInfC{$\{c\}_k+f$}
\BinaryInfC{$b-f$}
\AxiomC{}\UnaryInfC{k}
\BinaryInfC{$\{b\}_k-\{f\}_k$}
\AxiomC{}\UnaryInfC{$\{f\}_k$}
\BinaryInfC{$\{b\}_k$}
\BinaryInfC{a}
\end{prooftree}
$$

The key ingredient in the above proof is the ability to form linear combinations of arbitrary terms. This allows one to make a clever use of linear combinations to avoid the blow-up in the encryption depth. In fact, every term occurring in this derivation is a linear combination of subterms of $X \cup \{a\}$. One can devise normalization rules which always ensure this property for normal proofs, as done in [18]. But even though in the above simple example, the coefficients of the terms were all from $\{-1, 0, 1\}$, one can conceive of examples where the coefficients cannot be bounded simply. In fact, one has to use algebraic methods to impose some structure on the derivations despite there being no obvious bounds on the coefficients.

Our methods do not extend to a system as above, which allows arbitrary linear combinations, since the automaton construction seems to depend on the existence of a finite "core", as we observed earlier. But the relationships between the two techniques need to be studied in more depth. We leave this for future work.

We have concentrated on the passive intruder derivability problem in this paper. It is interesting to consider the active intruder deduction problem for these systems, in the spirit of [7]. It would also be interesting to investigate techniques for decidability of the secrecy problem when we do not necessarily have a locality property for passive intruder deductions but only an automaton-based decision procedure. This would be in the spirit of the work [5], which studies conditions under which a locality-based decidability for passive intruder deducibility can be lifted to a decision procedure for the active intruder deducibility. We leave that too for future work.

References

1 A. Baskar, R. Ramanujam, and S.P. Suresh. Knowledge-based modelling of voting protocols. In Dov Samet, editor, *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 62–71, 2007.

2 A. Baskar, R. Ramanujam, and S.P. Suresh. A DEXPTIME-complete Dolev-Yao theory with distributive encryption. In *Proceedings of MFCS 2010*, volume 6281 of *Lecture Notes in Computer Science*, pages 102–113, August 2010.

3 Vincent Bernat and Hubert Comon-Lundh. Normal proofs in intruder theories. In *ASIAN*, pages 151–166, 2006.

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

5 Sergiu Bursuc, Stéphanie Delaune, and Hubert Comon-Lundh. Deducibility constraints. In Anupam Datta, editor, *Proceedings of ASIAN 2009*, volume 5913 of *Lecture Notes in Computer Science*, pages 24–38. Springer, December 2009.

6 Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the ACM*, 28:114–133, January 1981.

7 Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. An NP decision procedure for protocol insecurity with XOR. *Theoretical Computer Science*, 338(1–3):247–274, 2005.

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

9 Hubert Comon-Lundh and Vitaly Shmatikov. Intruder Deductions, Constraint Solving and Insecurity Decisions in Presence of Exclusive or. In *Proceedings of the 18th IEEE Synposium on Logic in Computer Science (LICS)*, pages 271–280, June 2003.

10 Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.

11 Véronique Cortier, Michaël Rusinowitch, and Eugen Zalinescu. A resolution strategy for verifying cryptographic protocols with cbc encryption and blind signatures. In *PPDP*, pages 12–22, 2005.

12 Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, July 2009.

13 Danny Dolev, Shimon Even, and Richard M. Karp. On the Security of Ping-Pong Protocols. *Information and Control*, 55:57–68, 1982.

14 Danny Dolev and Andrew Yao. On the Security of public-key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.

15 Atsushi Fujioka, Tatsuaki Okamoto, and Kaazuo Ohta. A practical secret voting scheme for large scale elections. In *ASIACRYPT*, pages 244–251, 1992.

16 Thomas Genet and Francis Klay. Rewriting for cryptographic protocol verification. Technical report, CNET-France Telecom, 1999.

17 Jean Goubault Larrecq. A method for automatic cryptographic protocol verification. In *Proceedings of the 15th IPDPS Workshops 2000*, volume 1800 of *Lecture Notes in Computer Science*, pages 977–984, 2000.

18 Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for the equational theory of abelian groups with distributive encryption. *Information and Computation*, 205(4):581–623, April 2007.

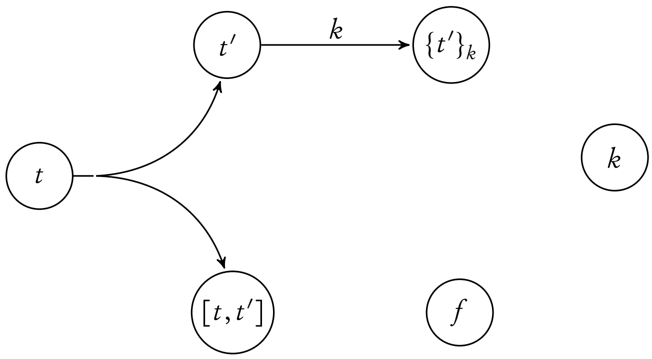
19 David Monniaux. Abstracting cryptographic protocols with tree automata. In *Static analysis symposium*, volume 1694 of *Lecture Notes in Computer Science*, pages 149–163, 1999.

20 Michaël Rusinowitch and Mathieu Turuani. Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete. *Theoretical Computer Science*, 299:451–475, 2003.

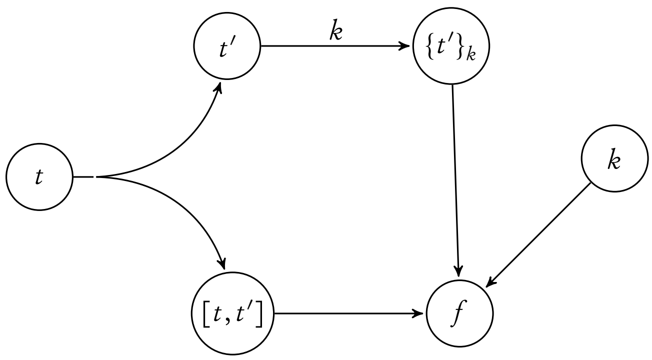
21 Dejvuth Suwimonteerabuth, Stefan Schwoon, and Javier Esparza. Efficient algorithms for alternating pushdown systems with an application to the computation of certificate chains. In Susanne Graf and Wenhui Zhang, editors, *4th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 4218 of *Lecture Notes in Computer Science*, pages 141–153, Beijing, China, October 2006. Springer.

A Examples of the automaton construction

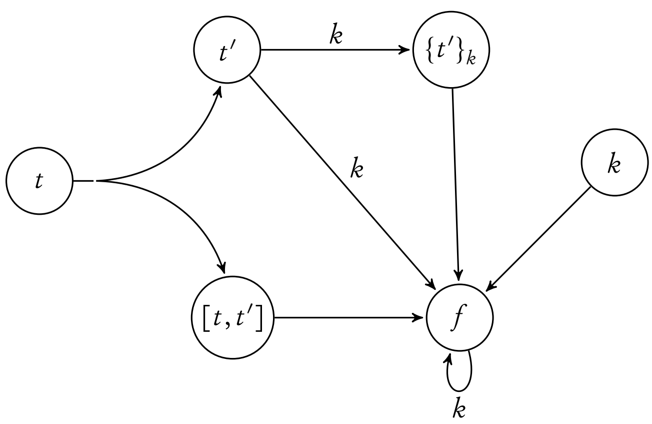
The first example we look at is a derivation of $\{t\}_k$ from $X = \{[t, t'], \{t'\}_k, k\}$. We will show *parts* of the successive stages of the automaton construction corresponding to this derivation. In this example and the next, we have only displayed enough states and edges that help us verify the existence of the appropriate derivation.



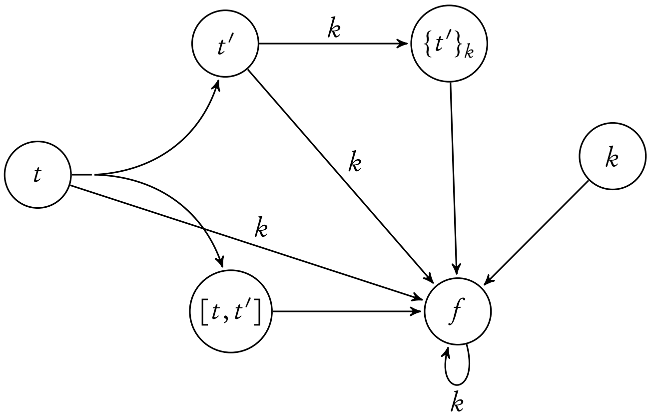
Stage 0. Notice AND-edge from t.



Stage 1. At this stage we add edges to f from all terms derivable using the *Ax* rule.

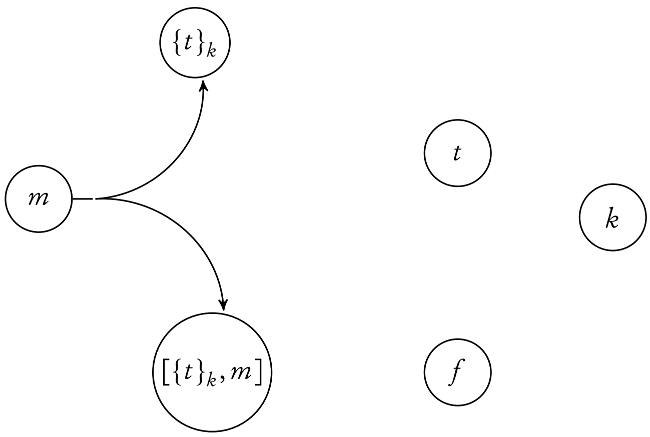


Stage 2. The k-labelled edge from f to f is added because of the edge from k to f. The k-labelled edge from t' to f is added because there was a k-labelled path from t' to f in the previous stage.

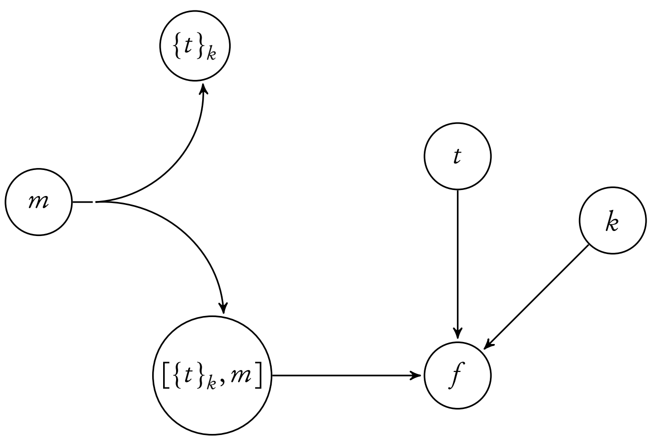


Stage 3. The k-labelled edge from t to f is added because there are k-labelled paths both from $[t,t']$ and t' in the previous stage. This edge verifies that $X \vdash \{t\}_k$.

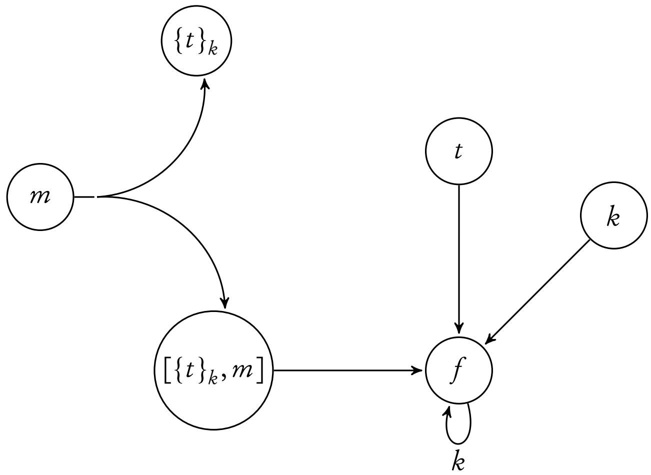
The second example is a derivation of m from the set $X = \{[\{t\}_k, m], t, k\}$.



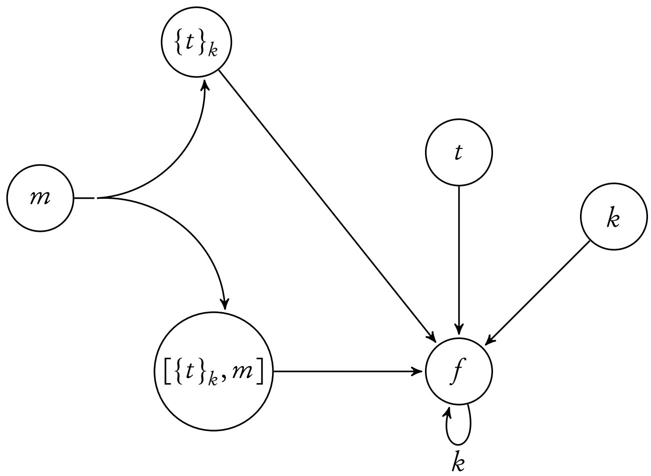
Stage 0. Notice the AND-edge from m.



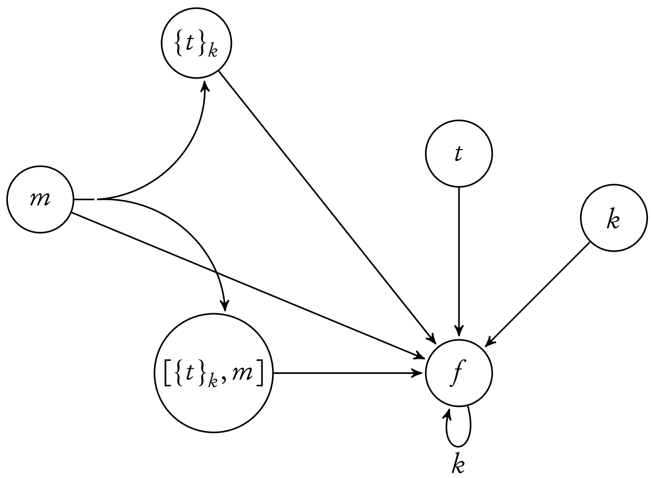
Stage 1. At this stage we add edges to f from all terms derivable using the *Ax* rule.



Stage 2. The k-labelled edge from f to f is added because of the edge from k to f.



Stage 3. The edge from $\{t\}_k$ to f is added because there is a path labelled k from t to f.



Stage 4. The ε-labelled edge from m to f is added because there are ε-labelled edges both from $\{t\}_k$ and $[\{t\}_k, m]$ to f in the previous stage. This edge verifies that $X \vdash m$.