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

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

1 **Chennai Mathematical Institute**
**Chennai, India**
{abaskar,spsuresh}@cmi.ac.in

2 **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. The result holds also when blind pairing is associative. The lower bound contrasts with ptime decidability for restricted theories of blind signatures, and the upper bound with non-elementary decidability for abelian group operators with distributive encryption.

## 1 Introduction

Dolev-Yao style models [8] for cryptographic protocols (the so-called "symbolic models") use a term algebra containing operations like pairing, encryption, signatures, hash functions, and nonces to build terms that are sent as messages in the protocol. The adversary against a protocol is modeled as a powerful network, which is only restricted in the way in which messages may be derived from the ones sent by the "honest" principals. Since these models are used for algorithmic analysis, the following *term derivability problem* is of basic interest: given a finite set of terms $X$ and a term $t$, is there a way for the adversary to derive $t$ from $X$?

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

Though the blind pairing constructor finds natural use in the Dolev-Yao modelling of electronic voting protocols [9], 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.

Indeed the result is of a technical nature and relates to the theoretician's toolkit in the study of Dolev-Yao models. The standard strategy to prove the derivability problem decidable is to prove a so-called **locality property** [14, 5], that if $t$ is derivable from $X$, then there is a special kind of derivation (a **normal derivation**) $\pi$ such that every term occurring in $\pi$ 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 backwards reachability problem for alternating pushdown systems. While many lower bound results for the *active* intruder deduction problem exist in the literature, under various settings, this is one of the few lower bound results for the *passive* intruder deduction problem.

The third technical contribution of the paper is the use (in our decision procedure) of the alternating automaton saturation technique in itself (similar to the one in [3]). 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 [13, 11, 10]. Applications of two-way alternating tree automata to security protocol verification has been touched upon in [4]. The saturation technique that we use offers yet another tool that may be of use in other contexts.

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

There is a more restricted way of modelling blind signatures: as seen in [7, 2, 6]. 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 the second component of blind pairs are always of the form $n$ or $\{n\}_k$ where $n$ is an atomic term (or nonce). And the only rule that involves pushing an encryption inside a blind pair is the derivation of $[\{t\}_k, n]$ from $[t, \{n\}_{inv(k)}]$ and $k$. This restricted system also satisfies a locality property.

At the other end of the spectrum, a much more powerful system is considered in [12]. They study an abelian group operator $+$ such that $\{t_1 + \cdots + t_n\}_k = \{t_1\}_k + \cdots + \{t_n\}_k$, i.e. encryption is homomorphic over $+$. They employ a very involved argument and prove the derivability problem in the general case to be decidable with a non-elementary upper bound. They also give a dexptime algorithm in the case when the operator is xor, and a ptime algorithm in the so-called binary case. The blind pair operator we consider has very different characteristics than xor, and the arguments in [12] do not apply here.

## 2 Extension of the Dolev-Yao model with blind pairs

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

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

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

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

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

- $m{\downarrow} = m$ for $m \in \mathcal{N}$
- $(t_1, t_2){\downarrow} = (t_1{\downarrow}, t_2{\downarrow})$
- $[t_1, t_2]{\downarrow} = [t_1{\downarrow}, t_2{\downarrow}]$
- $\{t\}_k{\downarrow} = \begin{cases} [\{t_1\}_k{\downarrow}, \{t_2\}_k{\downarrow}] & \text{if } t = [t_1, t_2] \\ \{t{\downarrow}\}_k & \text{otherwise} \end{cases}$

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

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

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

▶ **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 $X \vdash t$.

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

But there is no obvious locality property for the proof system considered here. For instance, to derive the term $\{a\}_k$ from $[a, b]$, $\{b\}_k$ and $k$, we necessarily need to go via the term $[\{a\}_k, \{b\}_k]$, which is not

a subterm of either the premises or the conclusion. In fact, the structure of terms occurring in a proof of $X \vdash t$ can get very complex. For example, one can code up some kind of a counter – a set $X$ of $O(n)$ terms and another term $t$, each of size $O(1)$ with $X \vdash t$ but such that every proof of $t$ from $X$ has at least $2^n$ terms occurring in it. The idea is to put enough terms in $X$ that can build (using the *encrypt* and *blindsplit* rules) a term $r$ with an encryption sequence consisting of an arbitrary number of blocks of length $n$. When (and only when) successive blocks code up consecutive $n$-bit binary numbers, the structure of $X$ ensures that the encryptions can be peeled off from $r$ inside out. This ensures that there are $2^n$ such blocks and hence an exponential sized proof. The example is given in detail in Appendix A.

## 3 Normal proofs

Even though our proof system lacks an obvious locality property, we can prove a weak locality property, which will help us derive a decision procedure for the derivability problem. This section is devoted to a proof of the weak locality property (or **weak subterm property**).

We first define the notion of a normal proof. These are proofs got by applying the transformations of Figure 2 repeatedly. Any subproof that matches 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.

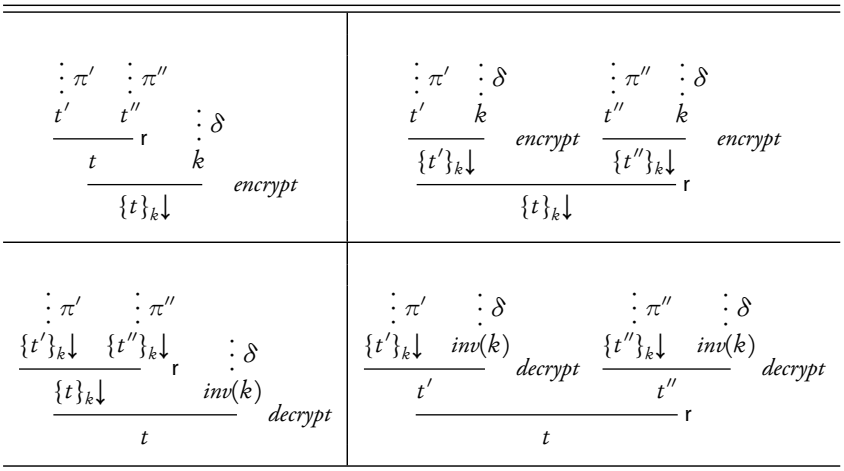



**Figure 2** The normalization rules. Rule r is meant to be either *blindpair* (in which case $t = [t', t'']$), or $blindsplit_0$ (in which case $t' = [t'', t]$), or $blindsplit_1$ (in which case $t' = [t, t'']$).

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

A proof $\pi$ is a **normal proof** if the following two conditions hold:

1. every subproof of $\pi$ is minimal, and
2. the transformations in Figure 2 cannot be applied to $\pi$.

▶ **Lemma 4.** *Whenever $X \vdash t$, there is a normal proof of $t$ from $X$.*

The proof is straightforward, and can be found in Appendix B.

We now state the weak locality property for normal proofs. The standard locality property can be viewed as giving a bound on the "width" and encryption depth of terms occurring in a proof of $X \vdash t$. We prove a weaker property, where only the width of terms is bounded. So the set of terms occurring in any normal proof of $X \vdash t$ is got by encrypting terms (perhaps repeatedly) from a "core" set, using

keys derivable from $X$. The core, it turns out, is $st(X \cup \{t\})$. For every $p \in st(X \cup \{t\})$, define $\mathscr{L}_p$ to be $\{x \in (st(X \cup \{t\}) \cap \mathscr{K})^* \mid X \vdash \{p\}_x\}$. We shall show in the next section that $\mathscr{L}_p$ is regular for each $p$.

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

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

▶ **Lemma 5** (Weak locality property). *Let $\pi$ be a normal proof of $t$ from $X$, and let $\delta$ be a subproof of $\pi$ with root labelled $r$. Then the following hold:*

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

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

## 4 The automaton construction

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

An **alternating pushdown system** is a triple $\mathscr{P} = (P, \Gamma, \hookrightarrow)$, where $P$ is a *finite set of control locations*, $\Gamma$ 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 $i \geq 0$.

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 5 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):

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

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 $\varepsilon$ 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 $\varepsilon$ from $t$ to the set $\{[t,t'],t'\}$. We introduce a final state $f$ and introduce an $\varepsilon$-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 that has an edge labelled $\varepsilon$ 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 3. Some examples that illustrate the saturation procedure are presented in Appendix C.

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 $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ that satisfies the following:
  1. if $t \in Y_0, k \in K_0$ such that $\{t\}_k\!\downarrow \in Y_0$, then $t \overset{k}{\hookrightarrow}_0 \{\{t\}_k\!\downarrow\}$.
  2. if $t, t', t'' \in Y_0$ such that $t$ is the conclusion of an instance of the *blindpair* or *blindsplit*$_i$ rules with premises $t'$ and $t''$, then $t \overset{\varepsilon}{\hookrightarrow}_0 \{t', t''\}$.
- $\hookrightarrow_{i+1}$ is the smallest subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ such that:
  1. if $q \overset{a}{\Rightarrow}_i C$, then $q \overset{a}{\hookrightarrow}_{i+1} C$.
  2. if $\{t\}_k\!\downarrow \in Y_0$ and $t \overset{k}{\Rightarrow}_i C$, then $\{t\}_k\!\downarrow \overset{\varepsilon}{\hookrightarrow}_{i+1} C$.
  3. if $k \in K_0$ and $k \overset{\varepsilon}{\Rightarrow}_i \{f\}$, then $f \overset{k}{\hookrightarrow}_{i+1} \{f\}$.
  4. if $\Gamma \subseteq Y_0$, $t \in Y_0$, and if there is an instance r of one of the rules of Figure 1 (nullary, unary or binary) whose set of premises is (exactly) $\Gamma$ 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 3** The sequence of automata for analysing $X_0 \vdash t_0$, with $Y_0 = \mathit{st}(X_0 \cup \{t_0\})$ 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. Other constructions like two-way automata do not seem immediately applicable to this situation.

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

The following lemma essentially shows that the saturation procedure terminates in exponential time. The proof is in Appendix D.

▶ **Lemma 6.** 1. *For all $i \geq 0$ and all $a \in \Sigma \cup \{\varepsilon\}$, the relation $\stackrel{a}{\Rightarrow}_i$ 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 $\stackrel{a}{\hookrightarrow}_{i+1}$ 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 $C \subseteq Q$, $q \stackrel{a}{\hookrightarrow}_i C$ if and only if $q \stackrel{a}{\hookrightarrow}_{d'} C$.*

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

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

▶ **Theorem 7** (Soundness). *For any $i$, any $t \in Y_0$, and any keyword $x$, if $t \stackrel{x}{\Rightarrow}_i \{f\}$, then $X_0 \vdash \{t\}_x\downarrow$.*

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

▶ **Lemma 8.** *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) $t \stackrel{x}{\Rightarrow}_{i,d} C$, 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, formulatng Lemma 4 precisely turned out to be the trickiest part of the upper bound proof. Due to lack of space, we present the proof in Appendix D.

▶ **Theorem 9** (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 $t \stackrel{x}{\Rightarrow}_i \{f\}$.*

The proof is by induction on derivations, and is presented in full detail in Appendix D.

▶ **Theorem 10.** *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 6, there is $d'$ such that for all $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, and $C \subseteq Q$, and any $i \geq 0$,

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

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

By the soundness theorem (Theorem 7), for all $i$, any $t \in Y_0$ and any keyword $x$, if $t \stackrel{x}{\Rightarrow}_i \{f\}$, then $X_0 \vdash \{t\}_x\downarrow$. In particular, this holds for $i = d'$. On the other hand, by the completeness theorem (Theorem 9), whenever $X_0 \vdash \{t\}_x\downarrow$ for $t \in Y_0$ and keyword $x$, there is an $i$ such that $t \stackrel{x}{\Rightarrow}_i \{f\}$, and hence $t \stackrel{x}{\Rightarrow}_{d'} \{f\}$. Thus to check whether $X_0 \vdash t_0$, it suffices to check if $t_0 \stackrel{\varepsilon}{\Rightarrow}_{d'} \{f\}$. But by construction, if $t_0 \stackrel{\varepsilon}{\Rightarrow}_{d'} \{f\}$, then $t_0 \stackrel{\varepsilon}{\hookrightarrow}_{d'+1} \{f\}$, but this means that $t_0 \stackrel{\varepsilon}{\hookrightarrow}_{d'} \{f\}$.

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

## 5 A DEXPTIME lower bound for the derivability problem

We recall the following fact about alternating pushdown systems.

▶ **Theorem 11** ([15]). *The* **backwards-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 $t\in\mathscr{T}$.

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

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

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

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

This gets translated to the following term $\mathsf{r}_m$:

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

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

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

We claim that $(s,x_s)\Rightarrow_{\mathscr{P}}(f,x_f)$ iff $X\vdash\{s\}_{x_se}$. A detailed proof for both directions is presented in Appendix E. Here we just present a high-level sketch of the proof.

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

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

The harder direction of the proof is given by the following lemma.

▶ **Lemma 12.** *For any configuration $(a,x)$, if there is a normal proof of $X\vdash\{a\}_{xe}$, then*

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

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

▶ **Lemma 13.** *If there is a normal proof $\pi$ of $X\vdash\{a\}_{xe}$, then either $(a,x)=(f,x_f)$ or there is a rule of $\mathscr{P}$, $(a,y)\hookrightarrow\{(b_1,y_1),\ldots,(b_n,y_n)\}$, and $z\in\Gamma^*$ such that $x=yz$, and for each $j\le n$, a* subproof *$\pi_j$ of $\pi$ with conclusion $X\vdash\{b_j\}_{y_jze}$.*

**Proof.** The observation that drives the proof of this lemma is the following. Its proof is given in Appendix E.

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

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

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

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

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

And the following theorem is the end result.

▶ **Theorem 14.** *The passive intruder deduction problem is* dexptime*-hard.*

## 6 Discussion

We can think of a number of extensions of our system by considering more algebraic properties of the blind pair operator, like associativity, commutativity, unitariness, etc. It then becomes more convenient to treat an extension of the Dolev-Yao model with a polyadic $+$ operator, over which encryption distributes. In this framework, a very powerful system is studied in [12], where $+$ is treated as an abelian group operator.

The decidability results in [12] are driven by a set of normalization rules whose effect is drastically different from ours. Our rules ensure that the "width" of terms occurring in a normal proof of $X \vdash t$ is bounded by $X \cup \{t\}$. But their normalization rules ensure that the encryption depth of terms occurring in a normal proof of $X \vdash t$ is bounded by $X \cup \{t\}$. On the other hand, the width of terms, represented by coefficients in the $+$-terms, can grow unboundedly. The rest of their decidability proof is an involved argument using algebraic methods.

The techniques of our paper do not seem to extend to the system with an abelian group operator, nor for slightly weaker systems where $+$ is associative and commutative, or when $+$ is a (not necessarily commutative) group operator and the term syntax allows terms of the form $-t$. But the techniques for our upper bound proofs extends to the case when $+$ is just an associative operator (not necessarily commutative, or has inverses). These results will be presented in a companion paper. 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.

We have concentrated on the passive intruder in this paper. It is interesting to consider the active intruder deduction problem for these systems, and more generally, 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. That is left 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 Vincent Bernat and Hubert Comon-Lundh. Normal proofs in intruder theories. In *ASIAN*, pages 151–166, 2006.

3 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.

4 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`.

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

6 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.

7 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.

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

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

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

11 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.

12 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.

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

14 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.

15 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 Lower bound example

The following example constructs a set of terms $X$ (containing $10 \cdot n + 12$ terms each of constant size) and a term $t$ (also of constant size) such that $X \vdash t$, but every proof of $X \vdash t$ contains a term of size at least $2^n$.

We will assume four keys $k, k', k_0, k_1$ which are all non-symmetric and not inverses of one another. For notational simplicity, we use $\{t\}_i$ instead of $\{t\}_{k_i}$. We let $\underline{m}$ be reverse of the $n$-bit binary representation of $m \in \{0, \ldots, 2^n - 1\}$. We shall show how to start with a set of terms of encryption depth at most 1, and derive another term of encryption depth at most 1, while necessarily seeing a term of the form $\{e\}_{k\underline{2^n-1}k\cdots k\underline{i+1}k\underline{i}k\cdots\underline{0}k'}$. In this example, all lowercase letters other than $k$ and its variants above stand for atomic terms.

Here are the terms that allow us to build up a term of the form $\{e\}_{k\underline{i_r}k\underline{i_{r-1}}k\cdots k\underline{i_1}k'}$. We will introduce more terms later that help us force $i_{j+1} = i_j + 1$ in the above.

$$\{s\}_{k'}, [\{a_1\}_0, s], [\{a_2\}_0, a_1], \ldots, [\{a_n\}_0, a_{n-1}], [\{b\}_k, a_n]$$

$$[\{c_1\}_0, b], [\{c_1\}_1, b], [\{c_2\}_0, c_1], [\{c_2\}_1, c_1], \ldots, [\{c_n\}_0, c_{n-1}], [\{c_n\}_1, c_{n-1}]$$

$$[\{b\}_k, c_n]$$

$$[\{d_1\}_1, b], [\{d_2\}_1, d_1], \ldots, [\{d_n\}_1, d_{n-1}], [\{e\}_k, d_n]$$

We will now try to go down from $\{e\}_{k\underline{i_r}k\underline{i_{r-1}}k\cdots k\underline{i_1}k'}$ to $\{f\}_{k'}$ while checking that consecutive blocks code up consecutive numbers. The logic is simple. If the term encrypted is still $e$, it means that every "bit" to the right of the "current bit position" is a 0. If this is the case, and the current bit is $i$, the corresponding bit in the previous block should be $1 - i$. Once a 1 has been seen to the right of the current bit position, and the current bit is $i$, the corresponding bit in the previous block should be $i$. Once a 1 has been seen, the term that is encrypted is changed from $e$ to $f$. We also let $g$ and $h$ code up the fact that we are looking to verify that the corresponding bit in the previous block is a 0 and 1, respectively.

$$[\{e\}_k, e], [[\{e\}_0, \{h_n\}_0], e], [[\{e\}_1, \{g_n\}_1], f]$$

$$[\{f\}_k, e], [[\{f\}_0, \{g_n\}_0], f], [[\{f\}_1, \{h_n\}_1], f]$$

$$[\{g_n\}_0, g_{n-1}], [\{g_n\}_1, g_{n-1}], [\{g_{n-1}\}_0, g_{n-2}], [\{g_{n-1}\}_1, g_{n-2}], \ldots, [\{g_1\}_0, p]$$

$$[\{g_n\}_k, g_n], [\{g_{n-1}\}_k, g_{n-1}], \ldots, [\{g_1\}_k, g_1]$$

$$[\{h_n\}_0, h_{n-1}], [\{h_n\}_1, h_{n-1}], [\{h_{n-1}\}_0, h_{n-2}], [\{h_{n-1}\}_1, h_{n-2}], \ldots, [\{h_1\}_1, p]$$

$$[\{h_n\}_k, h_n], [\{h_{n-1}\}_k, h_{n-1}], \ldots, [\{h_1\}_k, h_1]$$

$$[\{p\}_k, p], [\{p\}_0, p], [\{p\}_1, p], [\{p\}_{k'}, \{s\}_{k'}]$$

Take $X$ be the set of all the above terms. We claim that the term $\{f\}_{k'}$ is derived from $X$, but that the term $\{e\}_{k\underline{2^n-1}k\cdots k\underline{i+1}k\underline{i}k\cdots\underline{0}k'}$ will necessarily have to occur in any proof of $\{f\}_{k'}$ from $X$.

## B Normalization proofs

▶ **Lemma 15.** *Whenever $X \vdash t$, there is a normal proof of $t$ from $X$.*

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

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

We can view normal proofs as the result of repeatedly applying the reduction steps in Figure 2 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 $\pi$ to $\pi'$, $d(\pi') < d(\pi)$. This immediately proves that the normalization procedure terminates.

The non-trivial cases are the reductions in Figure 2. 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. ◀

▶ **Lemma 16.** *Let $\pi$ be a normal proof of $t$ from $X$, and let $\delta$ be a subproof of $\pi$ with root labelled $r$. Then the following hold:*

1. *If $\delta$ is not a purely synthetic proof, for every $u$ occurring in $\delta$ there is $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.*
2. *If $\delta$ is a purely synthetic proof, for every $u$ occurring in $\delta$, either $u \in st(r)$ or there is $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.*
3. *If the last rule of $\delta$ 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 $\delta'$ of $\delta$, and prove it for $\delta$ itself.

- Suppose $\delta$ is of the following form:

$$\frac{}{r}\ Ax$$

  Then $r \in X \subseteq st(X)$, and we are done.
- Suppose $\delta$ is the following form (and $r = (r', r'')$):

$$\frac{\begin{array}{cc}\vdots\,\delta' & \vdots\,\delta'' \\ r' & r''\end{array}}{r}\ pair$$

  In this case, $\delta$ is a purely synthetic proof, and we aim to prove that for every $u$ occurring in $\delta$, 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 $\delta'$ or $\delta''$ 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 $\delta$ is of the following form:

$$\frac{\begin{array}{c}\vdots\,\delta' \\ (r, r')\end{array}}{r}\ split$$

We have to consider the following cases:

1. Suppose $\delta'$ is not a purely synthetic proof and for every $u$ occurring in $\delta'$ 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 $\delta$, 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 $\delta'$ is a purely synthetic proof. But then $\delta'$ has to end in an application of the *pair* rule, and therefore one of the premises of the last rule of $\delta'$ has to be $r$, and this contradicts minimality of $\delta$. So this case is not possible.

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

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

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

- Suppose $\delta$ is of the following form:

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

We have to consider the following cases:

1. Suppose $\delta'$ is not a purely synthetic proof and for every $u$ occurring in $\delta'$ 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 $\delta''$, 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 $\delta$, there is a $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.
2. Suppose $\delta'$ is a purely synthetic proof. But then $\delta'$ 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 $\delta$, as we can see by reasoning similar to the case when $\delta$ ends with a *split*. So this case is not possible.

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

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

We have to consider the following cases:

1. Suppose $r$ is not a blind pair, and hence $\delta$ is a purely synthetic proof. Then we aim to prove that for every $u$ occurring in $\delta$, 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 $\delta'$ or occurs in $\delta''$ 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 $\delta''$. 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 $\delta$ is not a purely synthetic proof. We aim to prove that for every $u$ occurring in $\delta$, there is $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$. We consider the following subcases:

   a. Suppose $\delta'$ is not a purely synthetic proof and for every $u$ occurring in $\delta'$ 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 $\delta''$, then since $k$ is atomic, $\delta''$ 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 $\delta$, there is a $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.

   b. Suppose $\delta'$ is a purely synthetic proof. We note that $r'$ is a blind pair, and hence the last rule of $\delta'$ is not *encrypt* (since $\delta'$ is purely synthetic). The only other possibility is that the last rule of $\delta'$ is *blindpair*, but that would violate the normality of $\delta$, as one of the transformations specified by the first row of Figure 2 would apply to $\delta$. So this case is not possible.

- Suppose $\delta$ is of the following form:

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

We first note that $inv(k)$ is an atomic key and hence $\delta''$ should end with the *analz* rule. Hence for every $u$ occurring in $\delta''$, there exists $p \in st(X)$ and a keyword $x$ such that $u = \{p\}_x\downarrow$.

We now consider $\delta'$. It cannot end in a *blindpair* rule, since one of the rules specified by the second row of Figure 2 would apply to $\delta$, thereby contradicting normality of $\delta$. Nor can $\delta'$ end in an *encrypt* rule, since then the major premise of the last rule of $\delta'$ would be $r$, and this contradicts the minimality of $\delta$. The only possibilities therefore are that $\delta'$ 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 $\delta'$ is in $st(X)$. Hence $\{r\}_k$, as well as $r$, are in $st(X)$ as well.

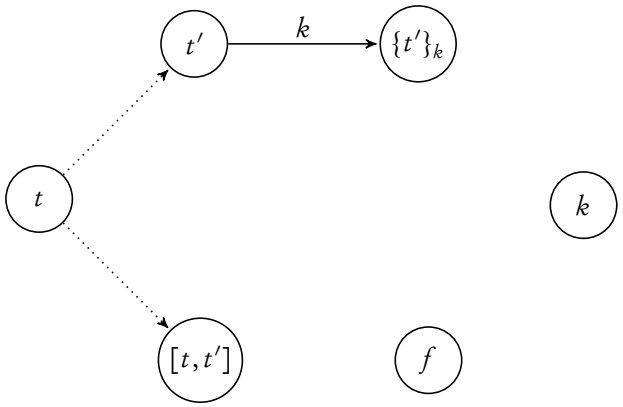
We now consider the case when the last rule of $\delta'$ is $blindsplit_1$. Let $r_1$ be the major premise of this rule, and $r_2$ the minor premise. Now it cannot be the case that $r_1$ is of the form $[\{r\}_k, \{r'\}_k]$. For, in that case $r_2$ would have been $\{r'\}_k$, and one of the normalization rules specified by the second row of Figure 2 would have applied to $\delta$, and this contradicts its normality.

We also know from the induction hypothesis (applied to $\delta'$) 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.

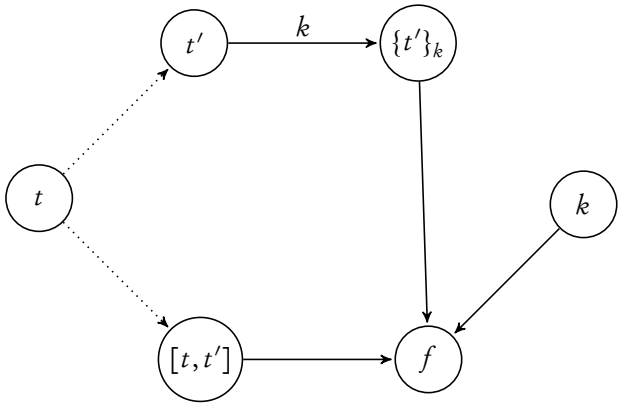
◀

## C Examples of the automaton construction

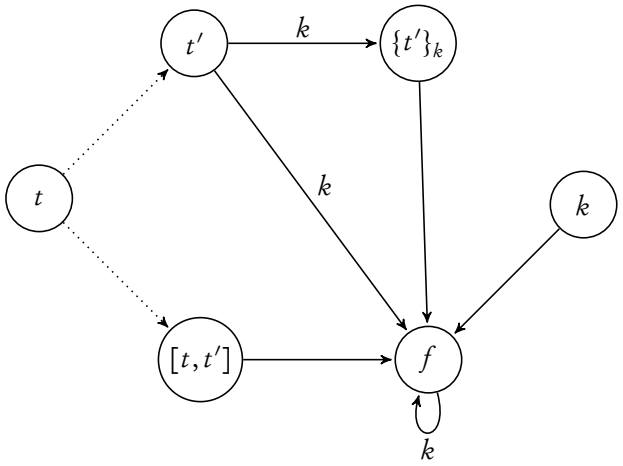
The first example we look at is a derivation of $\{t\}_k$ from $X = \{[t, t'], \{t'\}_k, k\}$. We will show *parts* of the successive stages of the automaton construction corresponding to this derivation. In this example and the next, we have only displayed enough states and edges that help us verify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



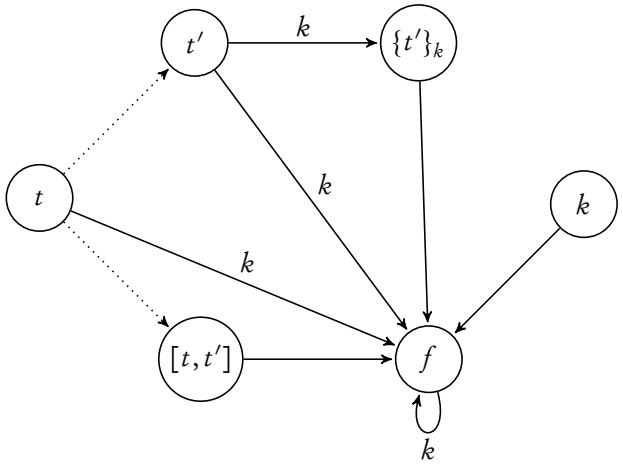

Stage 0. Notice that the dotted edges are part of the same and-edge.




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

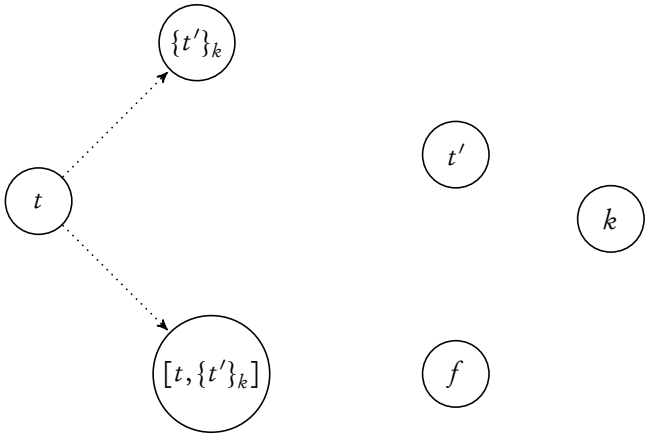



Stage 2. The $k$-labelled edge from $f$ to $f$ is added because of the edge from $k$ to $f$. The $k$-labelled edge from $t'$ to $f$ is added because there was a $k$-labelled path from $t'$ to $f$ in the previous stage.

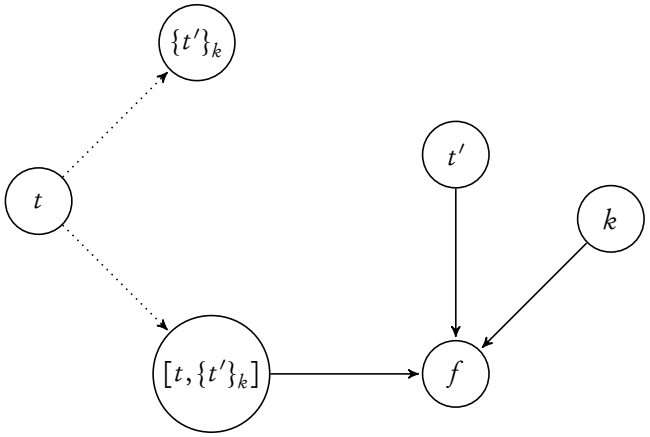



Stage 3. The $k$-labelled edge from $t$ to $f$ is added because there are $k$-labelled paths both from $[t,t']$ and $t'$ in the previous stage. This edge verifies that $X \vdash \{t\}_k$.

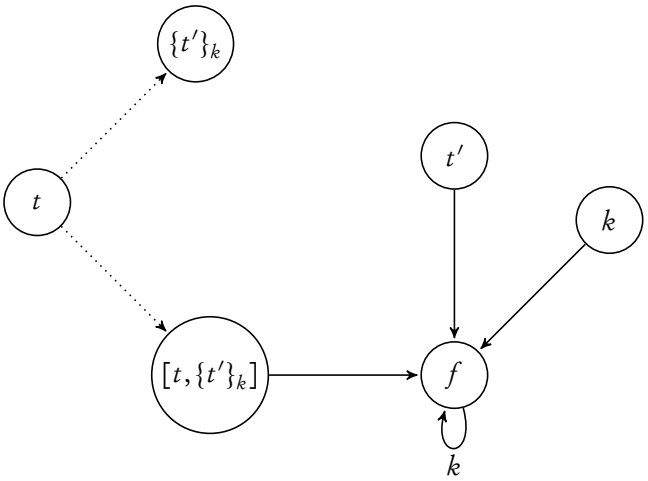
The second example is a derivation of $t$ from the set $X = \{[t,\{t'\}_k], t', k\}$.



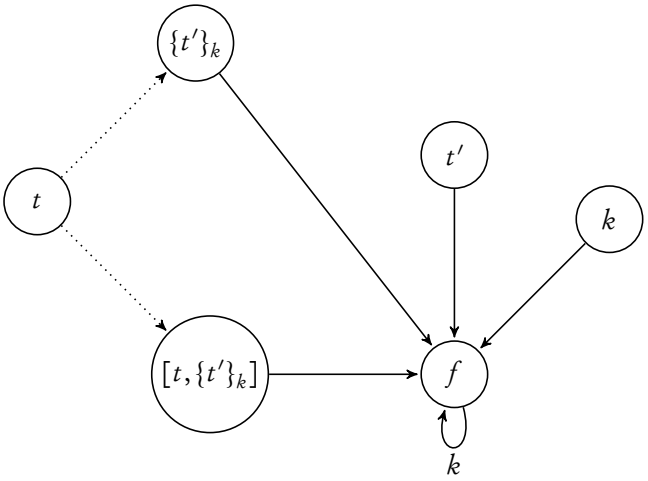

Stage 0. Notice that the dotted edges are part of the same and-edge.



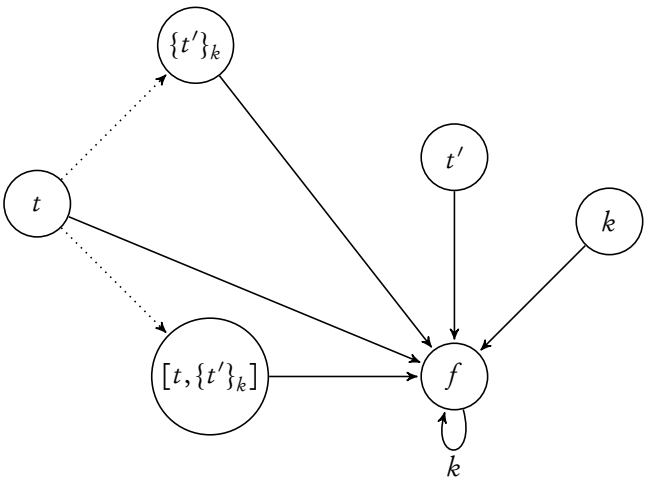

Stage 1. At this stage we add edges to $f$ from all terms derivable using the *Ax* rule.




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 it is derivable in one step from $t'$ and $k$ and there are edges from both $t'$ and $k$ to $f$.




Stage 4. The $\varepsilon$-labelled edge from $t$ to $f$ is added because there are $\varepsilon$-labelled edges both from $\{t'\}_k$ and $[t,\{t'\}_k]$ to $f$ in the previous stage. This edge verifies that $X \vdash t$.

## D Proofs for the automaton construction

▶ **Lemma 17.** 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 is $C' \subseteq Q$ such that $q \overset{\varepsilon}{\hookrightarrow}_i C'$ and $C' \overset{a}{\Rightarrow}_{i,j} C$.

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.

◀

▶ **Lemma 18.** *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 19.** *(**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 $\pi$ be a normal proof of $\{t\}_x\downarrow$ from $X$. The following cases need to be considered:

- Suppose the last rule r of $\pi$ 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 18, 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 $\pi$ ends in an application of the *Ax*, *pair*, *split*, and *decrypt* rules.
- Suppose $\pi$ is the following proof:

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

  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 $\pi$ ends in a $\textit{blindsplit}_i$ rule or a *blindpair* rule. The reasoning in all three cases is similar. We consider the case when $\pi$ has the following form:

$$\dfrac{\begin{array}{cc}\vdots\,\pi' & \vdots\,\pi'' \\ X \vdash [\{t\}_x\downarrow, t'] & X \vdash t'\end{array}}{X \vdash \{t\}_x\downarrow}\ \textit{blindsplit}_1$$

  By Lemma 5, 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 18, $t \overset{x}{\Rightarrow}_j \{f\}$ for some $j$.

◀

▶ **Lemma 20.** *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 $X_0 \cup \{D\}_y \vdash \{t\}_{xy}$.*

**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 $t \overset{x}{\Rightarrow}_{0,1} C$, 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 $\textit{blindsplit}_0$ 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 $C \not\subseteq Y_0$). 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 $C' \not\subseteq Y_0$, $X_0 \vdash x'y$. So $X_0 \vdash \{t\}_{x'y}$.

◀

## E The lower bound proofs

▶ **Lemma 21.** *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 $\mathscr{P}$ (numbered $m$, say), $(a,y) \hookrightarrow \{(b_1,y_1),\ldots,(b_n,y_n)\}$, $z \in \Gamma^*$, 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 that $i = i_1 + \cdots + i_n + 1$. By induction we know that $X \vdash \{b_j\}_{y_j ze}$ for all $j$. We observe that we can encrypt $\mathsf{r}_j$ using the sequence of keys $ze$, and then by a series of applications of the *blindsplit* rule with all the $\{b_i\}_{y_i ze}$, get $[\{\mathsf{c}_m\}_{dze}, \{a\}_{yze}]$. We can now encrypt $\{\mathsf{c}_m\}_d$ using the sequence of keys $ze$, and then apply the *blindsplit* rule to get $\{a\}_{yze} = \{a\}_{xe}$, as desired. ◀

▶ **Lemma 22.** 1. *No term $t \in X$ is a subterm of another term $t' \in X$.*
2. *The only term $t$ in $st(X)$ such that $e \in st(t)$ is $\{f\}_{x_f e}$.*
3. *If $p \in st(X)$ is a blind pair, then $e \notin st(p)$.*
4. *If $p \in st(X)$ and $\mathsf{c}_m \in st(p)$ then $p \in st(\mathsf{r}_m)$.*
5. *If $p \in st(X)$ and $p$ is a blind pair, then $\{\mathsf{c}_m\}_d \in st(p)$ for a unique $m \leq \ell$ and $p$ is not of the form $\{q\}_k$ for any $q$ and $k$.*
6. *If $\{p'\}_{w'}$ is the major premise of a blindsplit rule whose conclusion $\{p\}_w$ contains $\{\mathsf{c}_m\}_d$ as a subterm (for some $m \leq \ell$), then $w = w'$.*

▶ **Lemma 23.** *Let $\pi$ be a normal proof of $X \vdash \{a\}_{xe}$, where $(a,x)$ is a configuration of $\mathscr{P}$. Then*
1. *every term occurring $r$ in $\pi$ is of the form $\{p\}_w$ where $p \in st(X)$ and $w \in \Gamma^* \cup \Gamma^* e$.*
2. *if $r$ is a blind pair, there is a unique $p$ and $w$ such that $r = \{p\}_w$.*

**Proof.** 1. The subterm property for normal proofs guarantees that every term occurring in $\pi$ is of the form $\{p\}_w$, where $p \in st(X \cup \{a\})$ and $w \in (\Gamma \cup \{e\})^*$. Let us first observe that $a \in st(X)$ and hence $p \in st(X)$. Suppose a term of the form $\{q\}_{ye y'}$ occurs in $\pi$, where $y' \neq \varepsilon$. Since the conclusion of $\pi$ is $\{a\}_{xe}$ where $x \in \Gamma^*$, there has to be an occurrence of a rule in $\pi$ with one of its premises of the form $\{r\}_{zez'}$ with $r \in st(X)$ and $z' \neq \varepsilon$, and conclusion $\{t\}_w$ (with $t \in st(X)$, $e \notin st(t)$, and $w \in \Gamma^* \cup \Gamma^* e$).

But this rule cannot be an *encrypt*. Nor can it be a *blindpair*, since then $t$ has to have an occurrence of $e$, but the only term in $st(X)$ with an $e$ is $\{f\}_{x_f e}$ and that cannot be derived using an instance of *blindpair*. Suppose $\{t\}_w$ is derived from $\{r\}_{zez'}$ and $\{t'\}_{w'}$ using *blindsplit*. If $\{r\}_{zez'}$ is the major premise of this *blindsplit* rule, then it is easy to see that $\{t\}_w$ is of the form $\{u\}_{zez'}$. But since $\{t\}_w$ is

not of that form, it is clear that $\{t'\}_{w'}$ is the major premise of the *blindsplit* rule. But now we observe that if $\{t'\}_{w'}$ is of the form $\{u'\}_{ez'}$ then $\{t\}_w$ would also be of the form $\{v\}_{ez'}$. Since it is not of that form, and since $\{r\}_{zez'} \in st(\{t'\}_{w'})$, it has to be the case that $e \in st(t')$. But that is not possible since $t'$ is a blind pair and in $st(X)$, and $e$ does not occur in such terms. The conclusion thus follows.

2. Suppose $r$ is a blind pair. Then so is $p$, and by Lemma 22(5), there is some $m \leq \ell$ such that $\{\mathsf{c}_m\}_d \in st(p)$, and also $p$ is not of the form $\{q\}_k$ for any $q$ and $k$. From this it follows that there is a unique $p$ and $w$ such that $r = \{p\}_w$.

◀

▶ **Lemma 24.** *Let $\pi$ be a normal proof of $X \vdash \{a\}_{xe}$. Let $\delta$ be a subproof of $\pi$ with conclusion $u$ and immediate subproofs $\delta'$ and $\delta''$ with conclusions $u'$ and $u''$, respectively. Then:*

1. *if the last rule of $\delta$ is an application of encrypt, there are $p \in X$, $w \in \Gamma^*$, and $k \in \Gamma \cup \{e\}$ such that $u' = \{p\}_w$, $u'' = k$, and $u = \{p\}_{wk}$.*
2. *if the last rule of $\delta$ is an application of blindpair, there are $b \in M$, $p, p' \in st(X)$, and $w, w'', w'' \in \Gamma^*$ such that $u' = \{p'\}_{w'e}$, $u'' = \{b\}_{w''e}$, and $u = \{p\}_{we}$.*
3. *if the last rule of $\delta$ is an application of blindsplit, there are $p, p', p'' \in st(X)$, and $w, w'', w'' \in \Gamma^*$ such that $u' = \{p'\}_{w'e}$, $u'' = \{p''\}_{w''e}$, and $u = \{p\}_{we}$.*

**Proof.** 1. Clearly $u = \{p\}_{wk}$, where $p \in st(X)$ and $wk \in \Gamma^* \cup \Gamma^* e$. It follows that $w \in \Gamma^*$ and $k \in \Gamma \cup \{e\}$. To see that $p \in X$, we just need to observe that $\pi$ (and hence $\delta$) is a normal proof, and our normalization rules do not allow an *encrypt* rule whose major premise is the conclusion of a *blindpair* or a *blindsplit* rule. Thus $\delta'$ ends in an application of *encrypt* and by induction, $p \in X$. Or $\delta'$ ends in an application of *Ax*, and clearly we can take $p \in X$ and $w = \varepsilon$.

2. Suppose one of $\delta'$ and $\delta''$ end with an application of either the *blindpair* rule or the *blindsplit* rule. Then by induction hypothesis, either $u'$ or $u''$ is encrypted by $e$ at the end. It follows that $u$ also is encrypted by $e$ at the end. But $u$ is a blind pair and $e$ is a subterm of $u$, so $u \notin st(X)$. Thus $u = \{p\}_{we}$ for some $p \in st(X)$ and $w \in \Gamma^*$. If on the other hand both $\delta'$ and $\delta''$ end with an application of the *encrypt* rule, then $u' = \{p'\}_{w'}$ and $u'' = \{p''\}_{w''}$ for $p', p'' \in X$. But $u$ is a blind pair and of the form $\{p\}_w$ for some $p \in st(X)$. By inspection of $X$, $\{\mathsf{c}_m\}_d \in st(p)$ for some $m \leq \ell$, and so $\{\mathsf{c}_m\}_d \in st(p') \cup st(p'')$. But the only term of $X$ with $\{\mathsf{c}_m\}_d$ as a subterm is $\mathsf{r}_m$ and it, in turn, is not a subterm of any other term of $X$. Thus it is not possible that both $\delta'$ and $\delta''$ end in *encrypt*. We have thus established that $u = \{p\}_{we}$, $u' = \{p'\}_{w'e}$, and $u'' = \{p''\}_{w''e}$, where $p, p', p'' \in st(X)$. Inspection of $X$ reveals that if $p$ is a blind pair and a subterm of $X$, then one of its components is not a blind pair and does not contain $\{\mathsf{c}_m\}_d$ as a subterm, for any $m$. Thus we are justified in claiming that $p'' \in M$.

3. Suppose one of $\delta'$ and $\delta''$ end with an application of either the *blindpair* rule or the *blindsplit* rules. Then by induction hypothesis, either $u'$ or $u''$ is encrypted by $e$ at the end. It follows that $u$ also is encrypted by $e$ at the end. Thus $u = \{p\}_{we}$ for some $p \in st(X)$ and $w \in \Gamma^*$. If on the other hand both $\delta'$ and $\delta''$ end with an application of the *encrypt* rule, then $u' = \{p'\}_{w'}$ and $u'' = \{p''\}_{w''}$ for $p', p'' \in X$. But one of them, say $u'$ is a blind pair and $\{p''\}_{w''}$ is one of the components of $\{p'\}_{w'}$. But this cannot happen when both $p'$ and $p''$ are in $X$, as can be seen easily by inspecting $X$. Thus it is not possible that both $\delta'$ and $\delta''$ end in *encrypt*. We have thus established that $u = \{p\}_{we}$, $u' = \{p'\}_{w'e}$, and $u'' = \{p''\}_{w''e}$, where $p, p', p'' \in st(X)$.

◀

▶ **Lemma 25.** *Let $\pi$ be a normal proof of $X \vdash \{a\}_{xe}$. Let $\delta$ be a subproof of $\pi$ with conclusion $\{p\}_{we}$, and let $m \leq \ell$. Then:*

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

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

**Proof.** 1. Let $\delta'$ and $\delta''$ be immediate subproofs of $\delta$, with conclusions $\{p'\}_{w'e}$ and $\{b\}_{w''e}$, respectively, with $b \in M$.

Now $\{p\}_{we}$ is a blind pair, and by inspection of $X$, $\{\mathsf{c}_m\}_d \in st(p)$ for some $m \leq \ell$. Now either $\{p'\}_{w'e} \in comps(\{t\}_{we})$ in which case we are done, or $\{p'\}_{w'e}$ is itself a blind pair. In this case, we notice that $\delta'$ cannot end with an application of *blindsplit*, because one of the premises of that rule would be $\{p\}_{we}$, contradicting the fact that $\pi$ is normal (and hence minimal). It cannot also end in an application of *encrypt*, since it will follow that $p' = \mathsf{r}_m$, and it cannot be a subterm of $p$. Thus $\delta'$ ends in an application of *blindpair*, and by induction hypothesis, all components of $\{p'\}_{w'e}$ occur in $\delta'$. Combined with $\{b\}_{w''e}$, we can conclude that all components of $\{p\}_{we}$ occur in $\delta$.

2. Let $\delta'$ and $\delta''$ be immediate subproofs of $\delta$, and $\{p'\}_{w'e}$ and $\{p''\}_{w''e}$, respectively, be their conclusions.

Suppose, without loss of generality, that $\{p'\}_{w'e}$ is the major premise and suppose $\{\mathsf{c}_m\}_d \in st(p)$. Then by Lemma 22(6), $w = w'$, and it is also seen that $residues(p) = residues(p') \cup \{\{b\}_y\}$, where $\{b\}_{ywe} = \{p''\}_{w''e}$. It is clear that $\delta'$ does not end in a *blindpair* rule. If it ends in an application of *encrypt*, then (since $\{\mathsf{c}_m\}_d \in st(p')$ and $p' \in X$) $p' = \mathsf{r}_m$ and thus $p'$ has no residues. If $\delta'$ ends in an application of *blindsplit*, then by induction hypothesis, for all residues of $r'$ of $p'$, $\{r'\}_{w'e} = \{r'\}_{we}$ occurs in $\delta'$. Either way, since $\{p''\}_{w''e}$ also occurs in $\delta$, we can conclude that for all residues $r$ of $p$, $\{r\}_{we}$ occurs in $\delta$.

◀

▶ **Lemma 26.** *If there is a normal proof $\pi$ of $X \vdash \{a\}_{xe}$, then either $(a,x) = (f, x_f)$ or there is a rule (numbered $m$, say) of $\mathscr{P}$, $(a,y) \hookrightarrow \{(b_1,y_1),\ldots,(b_n,y_n)\}$, and $z \in \Gamma^*$ such that $x = yz$, and for each $j \leq n$, a* subproof *$\pi_j$ of $\pi$ with conclusion $X \vdash \{b_j\}_{y_j ze}$.*

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

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

Suppose $p \notin st(\mathsf{r}'_m)$. Then, it is clear that $\mathsf{r}'_m \in st(p')$, where $p'$ is the unique term in $st(X)$ such that $t' = \{p'\}_{ze}$, for a unique $z$. Now clearly $p' \notin X$ and hence $t'$ is not the result of an application of *encrypt*. It cannot also be the result of an application of *blindsplit*, since then one of the premises has to be $\{a\}_{xe}$, contradicting minimality. Thus $t'$ is the result of an application *blindpair*, but the previous lemma tells us that $\{r\}_{ze}$ for all $\{r\}_{ze} \in comps(\{p'\}_{ze})$. But notice that $\mathsf{r}'_m \in st(p')$, and hence we can conclude that among $comps(p')$ will be found $\{b_j\}_{y_j}$ for every $(b_j, y_j)$ on the right hand side of the rule numbered $m$. So by induction hypothesis, we can conclude that for each $j$, there is a subproof $\pi_j$ of $\pi$ whose conclusion is $X \vdash \{b_j\}_{y_j ze}$, and we are done. ◀

## F Blind pair as an associative operator: upper bound

We introduce a new syntax of terms, where we have a polyadic $+$ operator instead of the blind pairing operator. We will still continue to call the proof rules *blindpair* and *blindsplit*, for the sake of simplicity.

$$\mathscr{T} ::= m \mid (t_1, t_2) \mid \{t\}_k \mid t_1 + t_2 \cdots + t_l$$

where $m \in \mathscr{N}$, $k \in \mathscr{K}$, and $\{t, t_1, \ldots, t_l\} \subseteq \mathscr{T}$. We shall once again consider only normal forms, which are obtained by pushing encryptions inside $+$ terms recursively.

The set of **subterms** of $t$, $st(t)$, is the smallest $X \subseteq \mathscr{T}$ such that 1) $t \in X$, 2) if $(t, t') \in X$, then $\{t, t'\} \subseteq X$, 3) if $t_1 + t_2 + \ldots t_l \in X$ and none of $t_i$'s are headed with $+$ then $\{t_i + t_{i+1} \ldots t_j | 1 \leq i \leq j \leq l\} \subseteq X$ and $st(t_i) \subseteq X$ for every $i$ and 4) if $\{t\}_k \in X$ then $\{t, k\} \subseteq X$. $st(X)$ is defined to be $\bigcup_{t \in X} st(t)$.

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

**Figure 4** Proof system for normal terms (with assumptions from $X \subseteq \mathscr{T}$).

We next consider the weak locality lemma for this system. Here we need to add some normalization rules to ensure that an application of the *blindpair* rule does not occur above the major premise of an application of the *blindsplit* rule. For example, the system allows the following derivation.

$$\dfrac{\dfrac{t_1 + t_2 \quad t_3}{t_1 + t_2 + t_3}\ blindpair \qquad t_2 + t_3}{t_1}\ blindsplit_1$$

But we can replace it with the derivation given below, which is well-behaved.

$$\dfrac{t_1 + t_2 \qquad \dfrac{t_2 + t_3 \quad t_3}{t_2}\ blindsplit_1}{t_1}\ blindsplit_1$$

We need to extend the normalization rules in Figure 2 to make this work. We will also ensure that there is no occurrence of *blindpair* above the minor premise of *blindsplit*.

The proofs of the following two lemmas is pretty much on the lines of the corresponding statements for the binary operator case, so we skip the proofs.

▶ **Lemma 27.** *If $X \vdash t$ then there is a normal proof of $t$ from $X$.*

▶ **Lemma 28.** *Let $\pi$ be a normal proof of $t$ from $X$, and let $\delta$ be a subproof of $\pi$ with root labeled $r$. Then the following hold:*

1. *If $\delta$ is not a purely synthetic proof, for every $u$ occurring in $\delta$ there is $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.*

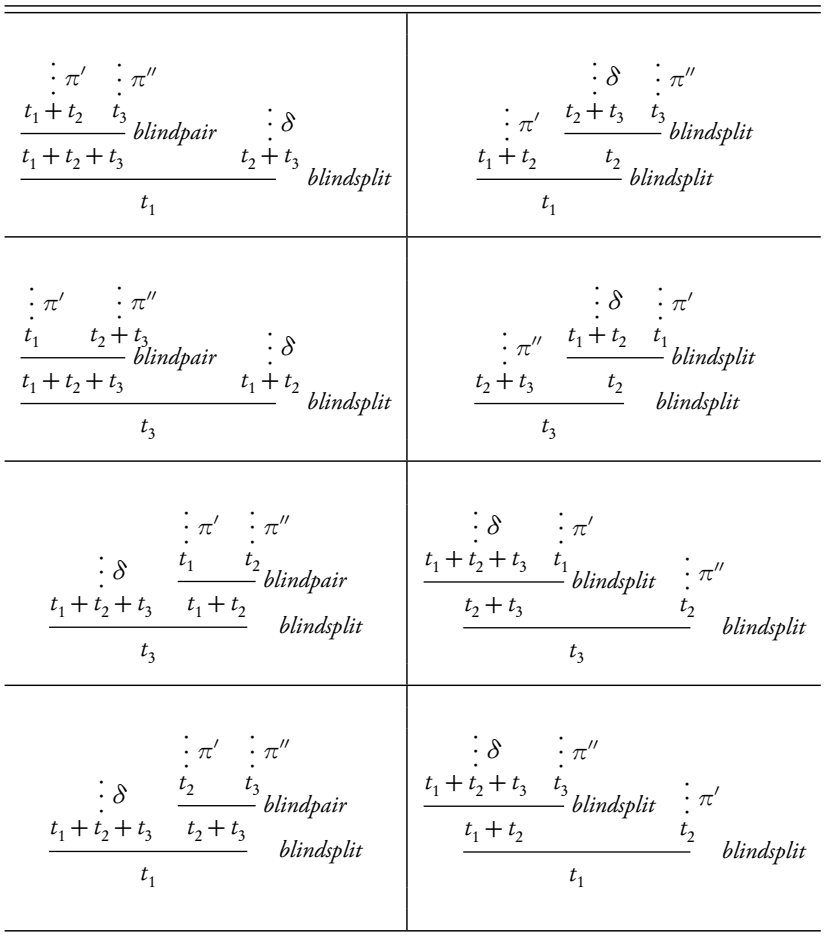



**Figure 5** The normalization rules for the associative case

2. *If $\delta$ is a purely synthetic proof, for every $u$ occurring in $\delta$, either $u \in st(r)$ or there is $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.*
3. *If the last rule of $\delta$ is decrypt or split with major premise $r_1$, then $r_1 \in st(X)$.*

Once we have the weak locality property, the automaton construction proceeds exactly as before. The only change is that the definition of $st(X)$ is potentially $O(n^2)$ where $n$ is the size of $X$.

▶ **Theorem 29.** *For the system with an associative blind pair, the problem of checking whether $X \vdash t$ is solvable in time $O(2^{n^2})$, where $n$ is the sum of the sizes of terms in $X$ and $t$.*

## G Blind pair as an associative operator: lower bound

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

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

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

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

This gets translated to the following term $\mathsf{r}_m$:

$$\{b_1\}_{x_1} + \cdots + \{b_n\}_{x_n} + \{\mathsf{c}_m\}_d + \{a\}_x + \{\mathsf{c}_m\}_d + \{b_n\}_{x_n} + \cdots + \{b_1\}_{x_1}$$

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

We again claim that $(s, x_s) \Rightarrow_{\mathscr{P}} (f, x_f)$ iff $X \vdash \{s\}_{x_s e}$. In this section, we just present an outline of the proof, highlighting the significant differences from the earlier lower bound proof.

The most important point to note is that our normalization rules prohibit the conclusion of a *blindpair* rule appearing as a major premise or a minor premise of a *blindsplit* rule. So in any normal proof $\pi$ of $X \vdash \{a\}_{xe}$ for a configuration $(a, x)$, all branches consist of a sequence of *blindsplit* rules (near the root) followed by a sequence of *encrypt* rules (near the leaves).

For any term $t$ whose normal form is $t_1 + \ldots + t_n]$, we define *comps*$(t)$ to be the set $\{t_1, \ldots, t_n\}$. If $t \in st(X)$ such that $\{\mathsf{c}_m\}_d \in st(t)$, then *residues*$(t)$ is defined by the following:

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

▶ **Lemma 30.** *Let $\pi$ be a normal proof of $X \vdash \{a\}_{xe}$. Let $\delta$ be a subproof of $\pi$ with conclusion $\{p\}_{we}$ for $p \in st(X)$, and let $m \leq \ell$. Then:*

1. *if $\{p\}_{we}$ is headed with a $+$ and $\{\mathsf{c}_m\}_d \notin st(p)$, then $X \vdash \{r\}_{we}$ is the conclusion of some subproof of $\delta$, for every $\{r\}_{we} \in comps(\{p\}_{we})$.*
2. *if $\{p\}_{we}$ is headed with a $+$ and $\{\mathsf{c}_m\}_d \in st(p)$, then $X \vdash \{r\}_{we}$ is the conclusion of some subproof of $\delta$, for every $r \in residues(p)$.*

**Proof.** 1. Suppose $\{p\}_{we}$ is headed with a $+$ and $\{\mathsf{c}_m\}_d \notin st(p)$. Then $\delta$ has to end with a *blindsplit* rule. Let $\{p'\}_{w'e}$ and $\{p''\}_{w''e}$ be the major and minor premises, respectively. If $\{\mathsf{c}_m\}_d \notin st(\{p'\}_{w'e})$ then by induction hypothesis, all the components of $\{p'\}_{w'e}$, and hence all components of $\{p\}_{we}$ occur earlier in the proof.

   If on the other hand $\{\mathsf{c}_m\}_d \in st(\{p'\}_{w'e})$, then $\{\mathsf{c}_m\}_d \in st(\{p''\}_{w''e}$ as well, and by induction hypothesis $\{r\}_{w''e}$ occurs earlier in the proof for every residue $r$ of $p''$. But it can be argued that $w = w''$ and that $comps(p) \subseteq residues(p'')$. Thus all components of $\{p\}_{we}$ occur earlier in the proof.

2. Suppose $\{p\}_{we}$ is headed with a $+$ and $\{\mathsf{c}_m\}_d \in st(\{p\}_{we})$. If $\delta$ ends with an *encrypt* rule, then it can be seen that $p \in X$ and that *residues*$(p) = \emptyset$. So the statement of the lemma is vacuously true.

   Otherwise $\delta$ ends with a *blindsplit* rule. Let $\{p'\}_{w'e}$ and $\{p''\}_{w''e}$ be the major and minor premises, respectively. Then $\{\mathsf{c}_m\}_d \in st(\{p'\}_{w'e})$ and so by induction hypothesis, $\{r\}_{w'e}$ occurs earlier in the proof for every $r \in residues(p')$. It can be seen that $w = w'$, so the statement of the lemma holds for some of the residues of $p$.

   Now if $\{\mathsf{c}_m\}_d \in st(\{p''\}_{w''e}$, then by inspecting the structure of $X$, one can prove that all residues of $p$ are also residues of $p''$ and that $w'' = w$. So the desired conclusion follows by induction hypothesis. Otherwise $\{\mathsf{c}_m\}_d$ is not in $st(\{p''\}_{w''e})$ and thus by induction hypothesis, all components of $\{p''\}_{w''e}$ occur earlier in the proof. It can again be shown that $w'' = w$. Now observe that *residues*$(p) \subseteq residues(p') \cup comps(p'')$ and the desired conclusion follows.

◀

▶ **Lemma 31.** *If there is a normal proof $\pi$ of $X \vdash \{a\}_{xe}$, then either $(a, x) = (f, x_f)$ or there is a rule of $\mathscr{P}$, $(a, y) \hookrightarrow \{(b_1, y_1), \ldots, (b_n, y_n)\}$, and $z \in \Gamma^*$ such that $x = yz$, and for each $j \leq n$, a* subproof *$\pi_j$ of $\pi$ with conclusion $X \vdash \{b_j\}_{y_j ze}$.*

**Proof.** Let $\pi$ be a normal proof of $X \vdash \{a\}_{xe}$ and suppose that $(a, x) \neq (f, x_f)$. Then it is clear that for all prefixes $y$ of $xe$, $\{a\}_y \notin X$. It is clear that the last rule is an application of *blindsplit*, with major premise

$t$ and minor premise $t'$. Now $t$ is a headed with a $+$, and $t = \{p\}_{ze}$ for $p \in st(X)$. If $\{\mathsf{c}_m\}_d \notin st(t)$, then by the previous lemma, all components of $t$, including $\{a\}_{xe}$ occur earlier in the proof, contradicting minimality of $\pi$. Hence $\{\mathsf{c}_m\}_d \in st(t)$ and it can also be seen that $x = z$. Hence $\{r\}_{xe}$ occurs earlier in the proof for all residues $r$ of $t$.

Now it can be seen that it is not the case that $t'$ is headed with a $+$. For in that case, either $\{\mathsf{c}_m\}_d \in st(t')$ and we can see that $t' = \{p'\}_{xe}$ for some $p' \in st(X)$, and that the $a \in residues(p')$. By the previous lemma, $\{a\}_{xe}$ occurs earlier in the proof, contradicting minimality. Or $\{\mathsf{c}_m\}_d \notin st(t')$ and we can see that $\{a\}_{xe} \in comps(t')$, and from the previous lemma it would follow that $\{a\}_{xe}$ occurs earlier in the proof, again contradicting minimality.

It follows that $t$ is either of the form $\{a\}_{xe} + \{\mathsf{c}_m\}_{dze}$ or of the form $\{\mathsf{c}_m\}_{dze} + \{a\}_{xe}$, and that $t' = \{\mathsf{c}_m\}_{dze}$, for some $z$. But now notice that an appropriate encryption of every term in the RHS of the $m^{\text{th}}$ rule is a residue of $t$ and hence occurs earlier in the proof. So it follows that to see that $(a, y)$ is the LHS of the $m^{\text{th}}$ rule, where $x = yz$. And the previous lemma ensures that for each $j \leq n$, there is a *subproof* $\pi_j$ of $\pi$ with conclusion $X \vdash \{b_j\}_{y_j ze}$, where the $m^{\text{th}}$ rule is $(a, y) \hookrightarrow \{(b_1, y_1), \ldots, (b_n, y_n)\}$. ◀

## H Extension to constructed keys

We now consider an extension where we consider constructed keys. We allow terms of the form $\{t\}_u$ now, where $u$ is any term. We also define $inv(t)$ to be $t$ itself for $t \notin \mathcal{K}$.

The strategy for the upper bound is as before. We apply the same normalization rules as before. We then prove the weak locality property for normal proofs. With this achieved, the only change in the automaton construction is to treat all of $st(X_0 \cup \{t_0\})$ as the possible set of keys $K_0$. The rest of the proofs go through almost verbatim. But the most nontrivial work is in proving weak locality. The proof is elaborated in the rest of this section.

The set of keys of a term $t$, $keys(t)$, is defined recursively as follows:

- $keys(m) = \emptyset$ if $m \in \mathcal{N}$
- $keys((t, t')) = keys(t) \cup keys(t')$
- $keys([t, t']) = keys(t) \cup keys(t')$, if $[t, t']$ is not of the form $\{r\}_u\downarrow$
- $keys(\{r\}_u) = keys(r) \cup keys(u) \cup \{u\}$.

$kst(t) = st(keys(t))$.

Let us make the following observations about *keys* and *kst*.

▶ **Lemma 32.** 1. *If $t \in st(t')$, then $kst(t) \subseteq kst(t')$.*
2. *If $[r, s]$ is of the form $\{p\}_x$ for some $p \in st(X)$ and some sequence of terms $x$, then $kst([r, s]) \subseteq (kst(r) \cap kst(s)) \cup st(X)$.*

▶ **Lemma 33.** *Let $\pi$ be a normal proof of $t$ from $X$, and let $\delta$ be a subproof of $\pi$ with root labelled $r$. Then the following hold:*

1. *If $\delta$ is not a purely synthetic proof, for every $u$ occurring in $\delta$, either $u \in kst(r)$ or there is $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $u = \{p\}_x\downarrow$.*
2. *If $\delta$ is a purely synthetic proof, for every $u$ occurring in $\delta$, either $u \in st(r)$ or there is $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $u = \{p\}_x\downarrow$.*
3. *If the last rule of $\delta$ 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 $\delta'$ of $\delta$, and prove it for $\delta$ itself.

- Suppose $\delta$ is of the following form:

$$\frac{}{r}\ Ax$$

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

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

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

  In this case, $\delta$ is a purely synthetic proof, and we aim to prove that for every $u$ occurring in $\delta$, either $u \in st(r)$ or there is $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $u = \{p\}_x\downarrow$. But any such $u$ either occurs in $\delta'$ or $\delta''$ 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 $x \in (st(X) \cup kst(r'))^*$ such that $u = \{p\}_x\downarrow$. But since $r' \in st(r)$, the desired conclusion follows. We argue similarly in the second case. Finally $r \in st(r)$, and so we are done in the third case as well.
- Suppose $\delta$ is of the following form:

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

  We have to consider the following cases:
  1. Suppose $\delta'$ is not a purely synthetic proof and for every $u$ occurring in $\delta'$, either $u \in kst((r, r'))$ or there is a $p' \in st(X)$ and $x' \in (st(X) \cup kst((r, r')))^*$ such that $u = \{p'\}_{x'}\downarrow$. In particular, there is a $p \in st(X)$ and $x \in (st(X))^*$ such that $(r, r') = \{p\}_x\downarrow$ (since $(r, r') \notin kst((r, r'))$). 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 $\delta$, there is a $p \in st(X)$ and $x \in (st(X))^*$ such that $u = \{p\}_x\downarrow$. We have also proved that the major premise of the last rule is in $st(X)$.
  2. Suppose $\delta'$ is a purely synthetic proof. But then $\delta'$ has to end in an application of the *pair* rule, and therefore one of the premises of the last rule of $\delta'$ has to be $r$, and this contradicts minimality of $\delta$. So this case is not possible.
- Suppose $\delta$ is the following form (and $r = [r', r'']$):

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

  In this case, $\delta$ is a purely synthetic proof, and we aim to prove that for every $u$ occurring in $\delta$, either $u \in st(r)$ or there is $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $u = \{p\}_x\downarrow$. But any such $u$ either occurs in $\delta'$ or $\delta''$ 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 $x \in (st(X) \cup kst(r'))^*$ such that $u = \{p\}_x\downarrow$. But since $r' \in st(r)$, the desired conclusion follows. We argue similarly in the second case. Finally $r \in st(r)$, and so we are done in the third case as well.
- Suppose $\delta$ is of the following form:

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

  We have to consider the following cases:
  1. Suppose $\delta'$ is not a purely synthetic proof and for every $u$ occurring in $\delta'$, either $u \in kst([r, s])$ or there is a $p' \in st(X)$ and $x' \in (st(X) \cup kst([r, s]))^*$ such that $u = \{p'\}_{x'}\downarrow$. Since $kst([r, s]) \subseteq kst(r) \cup st(X)$, the desired conclusion follows for $u$ occurring in $\delta'$. In particular, there is a

$p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $[r,s] = \{p\}_x\downarrow$ ($\because$ $[r,s] \notin st(kst([r,s]))$). The same holds for $r$ and $s$ as well.

Turning our attention to $u$ occurring in $\delta''$, either $u \in st(s)$ or there is $v \in st(X)$ and $y \in (st(X) \cup kst(s))^* \subseteq (st(X) \cup kst([t,s]))^* \subseteq (st(X) \cup kst(r))^*$ such that $u = \{v\}_y\downarrow$. But observe that $kst(s) \subseteq kst([r,s])st(X) \cup kst(r)$. Therefore for all $u$ occurring in $\delta''$, there is $v' \in st(X)$ and $z \in (st(X) \cup kst(r))^*$ such that $u = \{v'\}_z$.

Thus we have proved that for every $u$ occurring in $\delta$, there is a $p \in st(X)$ and $x$ such that $u = \{p\}_x\downarrow$.

2. Suppose $\delta'$ is a purely synthetic proof. But then $\delta'$ 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 $\delta$, as we can see by reasoning similar to the case when $\delta$ ends with a *split*. So this case is not possible.

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

$$\dfrac{\begin{array}{cc}\vdots\,\delta' & \vdots\,\delta'' \\ r' & h\end{array}}{r}\ encrypt$$

We have to consider the following cases:

1. Suppose $r$ is not a blind pair, and hence $\delta$ is a purely synthetic proof. Then we aim to prove that for every $u$ occurring in $\delta$, either $u \in st(r)$ or there is $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $u = \{p\}_x\downarrow$. But any such $u$ either occurs in $\delta'$ or occurs in $\delta''$ 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 $x \in (st(X) \cup kst(r'))^*$ 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 $\delta''$ (using the fact that $h \in st(r)$). 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 $\delta$ is not a purely synthetic proof. We aim to prove that for every $u$ occurring in $\delta$, either $u \in kst(r)$ or there is $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $u = \{p\}_x\downarrow$. We consider the following subcases:
   a. Suppose $\delta'$ is not a purely synthetic proof and for every $u$ occurring in $\delta'$ either $u \in kst(r') \subseteq kst(r)$ or there is a $p' \in st(X)$ and $x' \in (st(X) \cup kst(r'))^* \subseteq (st(X) \cup kst(r))^*$ such that $u = \{p'\}_{x'}\downarrow$. In particular, there is a $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $r' = \{p\}_x\downarrow$ (since $r' \notin kst(r')$). But this means that $r = \{p\}_{xh}\downarrow$, and hence of the desired form. If $u$ occurs in $\delta''$, then either $u \in st(h) \subseteq kst(r)$ or there is a $q \in st(X)$ and $y \in (st(X) \cup kst(h))^* \subseteq (st(X) \cup kst(r))^*$ such that $u = \{q\}_y\downarrow$ Thus we have proved that for every $u$ occurring in $\delta$, either $u \in kst(r)$ or there is a $p \in st(X)$ and $x \in (st(X) \cup kst(r))^*$ such that $u = \{p\}_x\downarrow$.
   b. Suppose $\delta'$ is a purely synthetic proof. We note that $r'$ is a blind pair, and hence the last rule of $\delta'$ is not *encrypt* (since $\delta'$ is purely synthetic). The only other possibility is that the last rule of $\delta'$ is *blindpair*, but that would violate the normality of $\delta$, as one of the transformations specified by the first row of Figure 2 would apply to $\delta$. So this case is not possible.

- Suppose $\delta$ is of the following form:

$$\dfrac{\begin{array}{cc}\vdots\,\delta' & \vdots\,\delta'' \\ \{r\}_h & inv(h)\end{array}}{r}\ decrypt$$

We first note that either $inv(h)$ is an atomic key and hence $\delta''$ ends with the *analz* rule. In this case, for every $u$ occurring in $\delta''$, there exists $p \in st(X)$ and $x \in (st(X))^*$ such that $u = \{p\}_x\downarrow$ (since $kst(h) = \emptyset$). On the other hand, $inv(h) = h$ is a constructed key, and for every $u$ occurring in $\delta''$,

either $u \in st(h) \subseteq kst(\{r\}_h)$ or there exists $p \in st(X)$ and $x \in (st(X) \cup kst(h))^* \subseteq (st(X) \cup kst(\{r\}_h))^*$ such that $u = \{p\}_x\downarrow$.
We now consider $\delta'$. It cannot end in a *blindpair* rule, since one of the rules specified by the second row of Figure 2 would apply to $\delta$, thereby contradicting normality of $\delta$. Nor can $\delta'$ end in an *encrypt* rule, since then the major premise of the last rule of $\delta'$ would be $r$, and this contradicts the minimality of $\delta$. The only possibilities therefore are that $\delta'$ 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 $\delta'$ is in $st(X)$. Hence $\{r\}_h$, as well as $r$, are in $st(X)$ as well. In the third case, for every $u$ occurring in $\delta'$, either $u \in kst(\{r\}_h)$ or there exists $p \in st(X)$ and $x \in (st(X) \cup kst(\{r\}_h))^*$ such that $u = \{p\}_x\downarrow$.
We now consider the third case again, when the last rule of $\delta'$ 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 $[\{r\}_h, \{r'\}_h]$. For, in that case $r_2$ would have been $\{r'\}_h$, and one of the normalization rules specified by the second row of Figure 2 would have applied to $\delta$, and this contradicts its normality.
We also know from the induction hypothesis (applied to $\delta'$) that there is a $p \in st(X)$ and $x \in (st(X) \cup kst(\{r\}_h)^*$ such that $r_1 = \{p\}_x$ (since $r_1 \notin kst(\{r\}_h)$). But since $r_1$ is of the form $[\{r\}_h, r_2]$, where $r_2$ is not of the form $\{r'\}_h$ for any $r'$, we conclude that $x = \varepsilon$ and $r_1 = p \in st(X)$. It follows that $r \in st(X)$ as well.

◀

▶ **Theorem 34.** *For the system with constructed keys, the problem of checking whether $X \vdash t$ is solvable in time $O(2^n)$, where $n$ is the sum of the sizes of terms in $X$ and $t$.*