# Dolev-Yao theory with associative blindpair operators

A Baskar[1], R Ramanujam[2], and S P Suresh[3] †

[1] BITS Pilani K K Birla Goa Campus
[2] Institute of Mathematical Sciences, Chennai
[3] CMI and CNRS UMI 2000 ReLaX

Abstract. In the context of modeling cryptographic tools like blind signatures and homomorphic encryption, the Dolev-Yao model is typically extended with an operator over which encryption is distributive. The intruder deduction problem has a non-elementary upper bound when the extended operator is an abelian group operator. Here we show that the intruder deduction problem is DEXPTIME-complete when we restrict the operator to satisfy only the associative property. We propose an automata-based analysis for the upper bound and use the reachability problem for alternating pushdown systems to show the lower bound.

## 1 Introduction

In the use of logic as a tool for analyzing security of communication protocols, cryptography is abstracted using a term algebra. In these Dolev-Yao style models [11] for cryptographic protocols we use a term algebra containing operations like pairing, encryption, signatures, hash functions, and nonces to build terms that are sent as messages in the protocol. The adversary against a protocol is modeled as a powerful intruder who can control the entire network, and can encrypt and decrypt at will; however, the cryptographic means used are assumed to be perfect. Therefore, while the intruder may not have access to actual private keys possessed by the "honest" participants, he has access to the structural patterns of terms that may be derived from the ones sent by the participants. Since these models are used for algorithmic analysis, the following *intruder deduction problem* is of basic interest: given a finite set of terms $X$ and a term $t$, is there a way for the intruder to derive $t$ from $X$?

In the basic Dolev-Yao model, the main operators are pairing and encryption, but these two do not interact with each other, in the sense that the encryption of a paired term is no different from that of any other term. The Dolev-Yao model abstracts away from the details of the encryption schemes used. However, the scheme used by participants would be known to the intruder, who can well make use of this information. In Dolev-Yao theory, the terms $\{t\}_k$ and $\{t'\}_{k'}$ are assumed to be distinct, unless $t = t'$ and $k = k'$. However, this is in general not true of cryptographic schemes such as the RSA. The algebraic properties of the encryption operator may well dictate the use of an equational theory to which the intruder has access. In such a context, interaction between encryption and other operators may be important. The reader is referred to the excellent survey [10] for studies of this kind.

One way of studying such interaction is by considering an extension of the Dolev-Yao term algebra with additional operators that interact in some specific way with encryption.

†Partially supported by an Infosys Grant

For instance, [12] 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 intruder deduction 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.

In this paper, we study an associative blind pair operator $+$ in which encryption is distributive. This operator satisfies two equations $\{t+t'\}_k = \{t\}_k + \{t'\}_k$ and $(t_1+t_2)+t_3 = t_1+(t_2+t_3)$. We show the intruder deduction problem for the Dolev-Yao term algebra with this extended operator is decidable in exponential time. The standard strategy consists of two steps. The first step is to prove the so-called locality property [13, 8, 5], 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 second step is using the locality property to provide a decision procedure for the intruder deduction problem.

Our system does not have an obvious locality property, so we cannot follow the standard route to decidability. The first contribution of this paper is to show a way of working around this difficulty by proving 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 all terms occurring in a normal derivation of $t$ from $X$ are from $S(X \cup \{t\})$, and the set of terms in $S(X \cup \{t\})$ 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 contribution is to settle the complexity of the intruder deduction problem by proving DEXPTIME-hardness by reduction from the reachability problem for alternating pushdown systems.

In [1], generic decidability results are given for the intruder deduction problem for convergent subterm theories and locally stable equational theories. Later in [9], similar results have been attained for monoidal theories. But our system does not belong to any of these subclasses. In [7], a generic procedure for the intruder deduction problem (deducibility) is given for arbitrary convergent equational theories. This procedure might not terminate but whenever it terminates it gives the correct answer. For the blind signature theory, this procedure terminates and it is implemented in polynomial time. But the modeling of blind signatures using the associative blind pair operator is different and hence the results in this paper. In [2], Dolev-Yao model is extended with an operator which is associative, commutative and idempotent but this operator doesn't interact with the encryption operator.

In earlier work in [4], we proposed similar system described in this paper, but we imposed a restriction on the blind pair operator: one of the components in the blind pair is always of the form $n$ or $\{n\}_k$ where $n$ is an atomic term and the only rule that involves distributing an encryption over a blind pair is the derivation of $[\{t\}_k, n]$ from $[t, \{n\}_{inv(k)}]$ and $k$. This restricted system also satisfies a locality property and using that we get a PTIME algorithm. It turns out that the considered restriction well suffices for the use of blind signatures in applications like voting protocols. In [5], the blind pair operator proposed did not have associativity property and the intruder deduction problem is DEXPTIME-complete but the operator might not satisfy associative property. The strategy is used here is similar to [3].

In Section 2, we present the basic definitions related to the Dolev-Yao system with the blind pair operator which is associative and in which encryption distributes. In Section 3, we

prove a normalization result and a weak subterm property. Section 4 contains details of an automaton-based DEXPTIME decision procedure for the intruder deduction problem. Section 5 contains the DEXPTIME[6] complexity lower bound.

## 2 The Dolev-Yao framework and the intruder deduction problem

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

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

where $m \in \mathcal{B}$, $k \in \mathcal{K}$, and $\{t, t_1, \ldots, t_l\} \subseteq \mathcal{T}$.

Definition 1. *The set of* subterms *of $t$, $st(t)$, is the smallest $Y \subseteq \mathcal{T}$ such that*

- $t \in Y$,
- *if* $(t_1, t_2) \in Y$, *then* $\{t_1, t_2\} \subseteq Y$,
- *if* $t_1 + t_2 + \cdots + t_l \in Y$, *then* $\{t_i + t_{i+1} \ldots + t_j | 1 \leq i \leq j \leq l\} \subseteq Y$, *and*
- *if* $\{t\}_k \in Y$, *then* $\{t, k\} \subseteq Y$.

*The set of subterms of $X$, $st(X)$, is $\bigcup_{t \in X} st(t)$ and its size is at most $(\sum_{t \in X} |t|)^2$.*

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 by "pushing encryptions over blind pairs, all the way inside." Formally, it is defined as follows:

Definition 2. *The* normal form *of a term $t$, denoted by $t\!\downarrow$, is defined inductively as follows.*

- $m\!\downarrow = m$ *for* $m \in \mathcal{B}$
- $(t_1, t_2)\!\downarrow = (t_1\!\downarrow, t_2\!\downarrow)$
- $(t_1 + t_2)\!\downarrow = t_1\!\downarrow + t_2\!\downarrow$
- $\{t\}_k\!\downarrow = \begin{cases} \{t_1\}_k\!\downarrow + \{t_2\}_k\!\downarrow & \text{if } t = t_1 + t_2, \text{ for some } t_1 \text{ and } t_2 \\ \{t\!\downarrow\}_k & \text{otherwise} \end{cases}$

The rules for deriving new terms from existing terms are given in Figure 1. The rules on the left column is referred as *synth*-rules as the conclusion of the rules contain its premises as subterms. The rules on the right column is referred as *analz*-rules as the conclusion of the rules are subterms of the left hand premise.

We like to emphasize that the subtle difference between the *analz*-rules for the pair operator $(t_0, t_1)$ and blind pair operator $t_0 + t_1$. If we have $(t_0, t_1)$ then we can derive $t_0$ using $split_0$ rule and $t_1$ using $split_1$ rule. But to derive $t_0$ from $t_0 + t_1$ using $blindsplit_1$ rule, we also need $t_1$ (and similarly to derive $t_1$ from $t_0 + t_1$ using $blindsplit_0$ rule, we also need $t_0$).

Definition 3. *A derivation or a proof $\pi$ of a term $t$ from a set of terms $X$ is a tree*

- *whose nodes are labeled by sequents of the form $X \vdash t'$ for some $t' \in \mathcal{T}$ and connected by one of the analz-rules or synth-rules in Figure 1,*
- *whose root is labeled $X \vdash t$, and*
- *whose leaves are labeled by Ax rule in Figure 1.*

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

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

$$\frac{X \vdash t_0 \quad X \vdash t_1}{X \vdash (t_0, t_1)}\ \mathit{pair} \qquad \frac{X \vdash (t_0, t_1)}{X \vdash t_i}\ \mathit{split}_i$$

$$\frac{X \vdash t_0 \quad X \vdash t_1}{X \vdash t_0 + t_1}\ \mathit{blindpair} \qquad \frac{X \vdash t_0 + t_1 \quad X \vdash t_i}{X \vdash t_{1-i}}\ \mathit{blindsplit}_i (i = 0, 1)$$

*synth*-rules *analz*-rules

Fig. 1. Deduction System

*We use $X \vdash t$ to denote that there is a proof of $t$ from $X$. For a set of terms $X$, $cl(X) = \{t \mid X \vdash t\}$ is the closure of $X$.*

*Example 1.* Let $X = \{a + b, \{b\}_k, k, \mathit{inv}(k)\}$ and $t$ to be $a$, then the following derivation shows that $X \vdash t$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{X \vdash a + b}\ \mathit{Ax} \quad \dfrac{}{X \vdash k}\ \mathit{Ax}}{X \vdash \{a\}_k + \{b\}_k}\ \mathit{encrypt} \quad \dfrac{}{X \vdash \{b\}_k}\ \mathit{Ax}}{X \vdash \{a\}_k}\ \mathit{blindsplit} \quad \dfrac{}{X \vdash \mathit{inv}(k)}\ \mathit{Ax}}{X \vdash a}\ \mathit{decrypt}$$

Definition 4. *The intruder deduction problem is the following: given a finite set $X \subseteq \mathcal{T}$ and $t \in \mathcal{T}$, determine whether $X \vdash t$.*

## 3 Weak Locality Property

As we have mentioned earlier, our derivation system lacks the locality property but we prove a weak locality property in this section and use it to solve the intruder deduction problem. Even if there are derivations of $X \vdash t$ with out the weak locality property, there will be one derivation of $X \vdash t$ with the weak locality property. Such a derivation will not have a few patterns (for example split rule will not be applied immediately after a pair rule). If any such pattern occurs, we argue there is a way to get rid of it without changing the final conclusion of the derivation. This is achieved by providing a set of transformation rules which dictate how to replace forbidden derivations by acceptable derivations. We formalize these concepts below.

Definition 5. *A transformation rule is a pair of proofs $(\pi_1, \pi_2)$ such that the roots of $\pi_1$ and $\pi_2$ are the same. Any subproof that matches a pattern of $\pi_1$ is meant to be replaced by the $\pi_2$.*

*A proof $\pi$ is a normal proof if transformation rules in Figure 2 and Figure 3 cannot be applied to $\pi$. The transformation rules in Figure 2 are from [5] and the transformation rules in Figure 3 are included to handle the associative property of the blind pair operator.*

The derivation provided in Example 1 is not a normal proof as we can apply transformation rule in the last row of Figure 2 (for blindsplit rule which is followed by the decrypt rule). Here is the result of applying this transformation rule for the proof in Example 1.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{X \vdash a+b}\,\mathit{Ax} \quad \dfrac{}{X \vdash k}\,\mathit{Ax}}{X \vdash \{a\}_k + \{b\}_k}\,\mathit{encrypt} \quad \dfrac{}{X \vdash \mathit{inv}(k)}\,\mathit{Ax}}{X \vdash a+b}\,\mathit{decrypt} \quad \dfrac{\dfrac{}{X \vdash \{b\}_k}\,\mathit{Ax} \quad \dfrac{}{X \vdash \mathit{inv}(k)}\,\mathit{Ax}}{X \vdash b}\,\mathit{decrypt}}{X \vdash a}\,\mathit{blindsplit}$$

The above derivation is still not a normal proof as the second transformation rule in Figure 2 can be applied. If we apply this transformation rule, we will get the following proof.

$$\dfrac{\dfrac{}{X \vdash a+b}\,\mathit{Ax} \quad \dfrac{\dfrac{}{X \vdash \{b\}_k}\,\mathit{Ax} \quad \dfrac{}{X \vdash \mathit{inv}(k)}\,\mathit{Ax}}{X \vdash b}\,\mathit{decrypt}}{X \vdash a}\,\mathit{blindsplit}$$

The above proof is a normal proof as no transformation rule can be applied.

Lemma 1. *For a given $X \cup \{t\} \subseteq \mathcal{T}$, if $X \vdash t$, then there is a normal proof for $X \vdash t$.*

*Proof.* If a proof for $X \vdash t$ is not a normal proof, then we apply the transform rules in Figure 2 and Figure 3 as long as possible. But it is not clear whether this procedure will terminate and eventually lead to a normal proof. We define a measure for every proof such that application of transformation rule reduces the measure of the proof. This will immediately lead to that the above procedure terminates.

For every proof $\pi$, we define a measure, $d(\pi)$, recursively as follow:

- if the last rule of $\pi$ is an *Ax* rule, $d(\pi) = 1$,
- if $\pi$ has only one immediate subproof $\pi'$ then $d(\pi) = d(\pi') + 1$,and
- if $\pi$ has immediate subproofs $\pi'$ and $\pi''$ and $r$ is the last rule of $\pi$, then

$$d(\pi) = \begin{cases} d(\pi') + d(\pi'') + 2 & \text{if } r = \textit{blindpair} \\ 2^{d(\pi')+d(\pi'')} & \text{if } r = \textit{encrypt} \text{ or } \textit{decrypt} \\ d(\pi') + d(\pi'') + 1 & \text{otherwise} \end{cases}$$

The above definition might look cryptic at first: for instance why the encrypt/decrypt rule increases the measure exponentially. We are using the subproof $\delta$ twice on the right hand sides of the last four transformations. So additive increase will not help our objective: the measure should decrease after applying the transformation rules. But fortunately the encrypt/decrypt

| | |
|---|---|
| $\dfrac{\dfrac{\dfrac{\vdots\,\pi_0 \quad \vdots\,\pi_1}{X \vdash t_0 \quad X \vdash t_1}\,\textit{pair}}{X \vdash (t_0, t_1)}\,\textit{split}_i}{X \vdash t_i}$ | $\dfrac{\vdots\,\pi_i}{X \vdash t_i}$ |
| $\dfrac{\dfrac{\vdots\,\pi_0 \quad \vdots\,\pi_1}{X \vdash t \quad X \vdash k}\,\textit{encrypt} \qquad \vdots\,\delta}{X \vdash \{t\}_k \qquad X \vdash \textit{inv}(k)}\,\textit{decrypt}}{X \vdash t}$ | $\dfrac{\vdots\,\pi_0}{X \vdash t}$ |
| $\dfrac{\dfrac{\vdots\,\pi_0 \quad \vdots\,\pi_1}{X \vdash t_0 \quad X \vdash t_1}\,\textit{blindpair} \qquad \vdots\,\delta}{X \vdash t_0 + t_1 \qquad X \vdash t_i}\,\textit{blindsplit}}{X \vdash t_{1-i}}$ | $\dfrac{\vdots\,\pi_i}{X \vdash t_{1-i}}$ |
| $\dfrac{\dfrac{\vdots\,\pi' \quad \vdots\,\pi''}{X \vdash t' \quad X \vdash t''}\,\textit{blindpair} \qquad \vdots\,\delta}{X \vdash t' + t'' \qquad X \vdash k}\,\textit{encrypt}}{X \vdash \{t'\}_k\downarrow + \{t''\}_k\downarrow}$ | $\dfrac{\dfrac{\vdots\,\pi' \quad \vdots\,\delta}{X \vdash t' \quad X \vdash k}\,\textit{encrypt} \qquad \dfrac{\vdots\,\pi'' \quad \vdots\,\delta}{X \vdash t'' \quad X \vdash k}\,\textit{encrypt}}{X \vdash \{t'\}_k\downarrow + \{t''\}_k\downarrow}\,\textit{blindpair}$ |
| $\dfrac{\dfrac{\vdots\,\pi' \quad \vdots\,\pi''}{X \vdash \{t'\}_k \quad X \vdash \{t''\}_k}\,\textit{blindpair} \qquad \vdots\,\delta}{X \vdash \{t'\}_k + \{t''\}_k \qquad X \vdash \textit{inv}(k)}\,\textit{decrypt}}{X \vdash t' + t''}$ | $\dfrac{\dfrac{\vdots\,\pi' \quad \vdots\,\delta}{X \vdash \{t'\}_k \quad X \vdash \textit{inv}(k)}\,\textit{decrypt} \qquad \dfrac{\vdots\,\pi'' \quad \vdots\,\delta}{X \vdash \{t''\}_k \quad X \vdash \textit{inv}(k)}\,\textit{decrypt}}{X \vdash t' \qquad X \vdash t''}\,\textit{blindpair}}{X \vdash t' + t''}$ |
| $\dfrac{\dfrac{\vdots\,\pi' \quad \vdots\,\pi''}{X \vdash \{t'\}_k + \{t''\}_k \quad X \vdash \{t'\}_k}\,\textit{blindsplit}_0 \qquad \vdots\,\delta}{X \vdash \{t''\}_k \qquad X \vdash \textit{inv}(k)}\,\textit{decrypt}}{X \vdash t''}$ | $\dfrac{\dfrac{\dfrac{\vdots\,\pi' \quad \vdots\,\delta}{X \vdash \{t'\}_k + \{t''\}_k \quad X \vdash \textit{inv}(k)}\,\textit{decrypt} \qquad \dfrac{\vdots\,\pi'' \quad \vdots\,\delta}{X \vdash \{t'\}_k \quad X \vdash \textit{inv}(k)}\,\textit{decrypt}}{X \vdash t' + t'' \qquad X \vdash t'}\,\textit{blindsplit}_0}{X \vdash t''}$ |
| $\dfrac{\dfrac{\vdots\,\pi \quad \vdots\,\pi'}{X \vdash \{t'\}_k + \{t''\}_k \quad X \vdash \{t''\}_k}\,\textit{blindsplit}_1 \qquad \vdots\,\delta}{X \vdash \{t'\}_k \qquad X \vdash \textit{inv}(k)}\,\textit{decrypt}}{X \vdash t'}$ | $\dfrac{\dfrac{\dfrac{\vdots\,\pi \quad \vdots\,\delta}{X \vdash \{t'\}_k + \{t''\}_k \quad X \vdash \textit{inv}(k)}\,\textit{decrypt} \qquad \dfrac{\vdots\,\pi' \quad \vdots\,\delta}{X \vdash \{t''\}_k \quad X \vdash \textit{inv}(k)}\,\textit{decrypt}}{X \vdash t' + t'' \qquad X \vdash t''}\,\textit{blindsplit}_1}{X \vdash t'}$ |

Fig. 2. Transformation rules

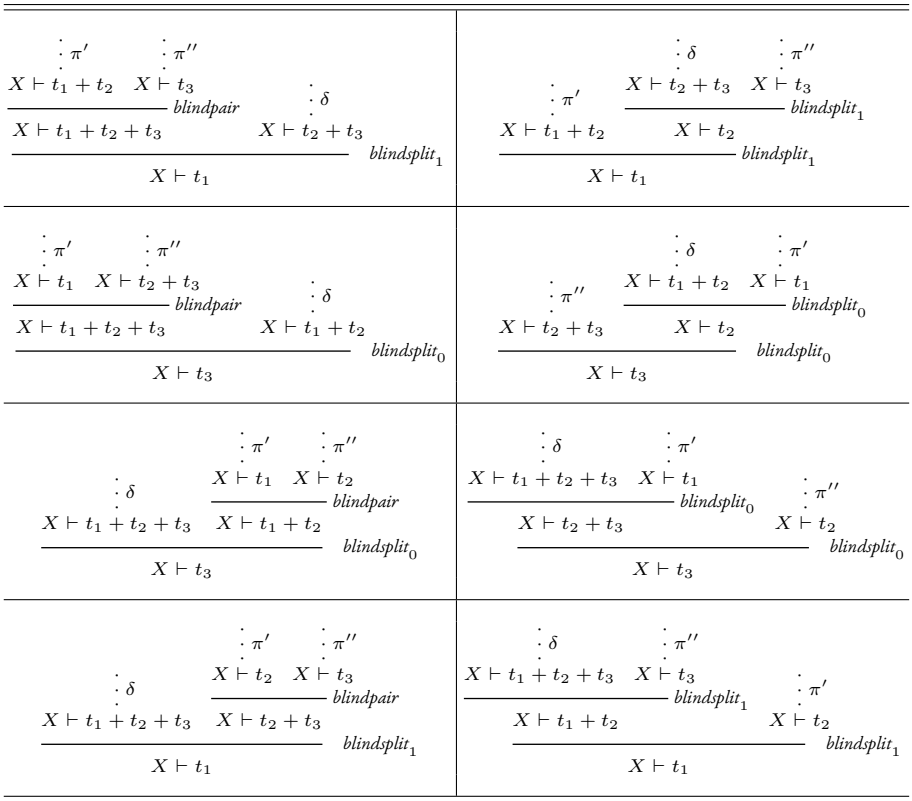

Fig. 3. Transformation rules for the associative case

rule on the left hand side builds on a bigger subproof whereas the encrypt/decrypt rule on the right hand side builds on smaller subproofs. By using this observation, we have defined the measure so that it would reduce even though $\delta$ is appearing more than once on the right hand side of the transformation rules.

Now we prove the application of the transformation rules reduces the measure of the proof. It is easy to see that the measure goes down for the first three rules in Figure 2. So we consider transformations in the fourth row and fifth row. We observe that the measure of the proof on the left is $2^{d(\pi')+d(\pi'')+d(\delta)+2}$, while the measure of the proof on the right is $2^{d(\pi')+d(\delta)} + 2^{d(\pi'')+d(\delta)} + 2$. Let $d(\pi') = m$, $d(\pi'') = n$, and $d(\delta) = p$, and assume without loss of generality that $m \geq n$. Then (since $m, n, p > 0$) $2^{m+n+p+2} > 2^{m+p+1} + 2 \geq 2^{m+p} + 2^{n+p} + 2$. The argument for the last two transformations in Figure 2 is similar.

Now we consider the transformations in Figure 3. The measure of the proof on the left is $d(\pi') + d(\pi'') + d(\delta) + 3$, while the measure of the proof on the right is $d(\pi') + d(\pi'') + d(\delta) + 2$.

We introduce a bit of notation first to conveniently state the weak locality lemma. We say that a proof $\pi$ of $X \vdash t$ is purely synthetic if either it ends in an application of the *blindpair* or *pair* rules, or it ends in an application of the *encrypt* rule and $t\downarrow$ is not a blind pair. A keyword

is an element of $\mathcal{K}^*$. Given a term $t$ and a keyword $x = k_1 \cdots k_n$, we use $\{t\}_x$ to denote $\{\cdots\{t\}_{k_1}\cdots\}_{k_n}$.

Lemma 2. *Let $\pi$ be a normal proof of $t$ from $X$, and let $\delta$ be a subproof of $\pi$ with root labeled $r$. Then for every $u$ occurring in $\delta$, the following hold:*

1. *Either $u \in st(r)$, or there are $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x$,*
2. *if $\delta$ is not a purely synthetic proof, then there exist $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x$, and*
3. *If the last rule of $\delta$ is the decrypt or split rule with the left side premise $X \vdash r_1$, then $r_1 \in st(X)$.*

*Proof.* We assume the claim for every proper subproof of $\delta$ and prove it for $\delta$ itself. Moreover, the second part of the claim is stronger than the first part. So we prove only the second part if $\delta$ is not a purely synthetic proof.

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

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

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

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

  In this case, $\delta$ is a purely synthetic proof. We aim to prove that for every $u$ occurring in $\delta$, either $u \in st(r)$ or there are $p \in st(X)$ 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 exist $p \in st(X)$ 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, when $u = r$.
- Suppose $\delta$ is of the following form:

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

  We have to consider the following cases:
  1. Suppose $\delta'$ is not a purely synthetic proof. By induction hypothesis, for every $u$ occurring in $\delta'$ there are $p' \in st(X)$ and keyword $x'$ such that $u = \{p'\}_{x'}\downarrow$. In particular, there are $p \in st(X)$ 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 are $p \in st(X)$ 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 the last rule of $\delta'$ the *pair* rule, and therefore one of the premises of the last rule of $\delta'$ has to be $r$ but this would violate the normality of $\delta$, as the transformation rule specified in the first row of Figure 2 can be applied to $\delta$. So this case is not possible.

– Suppose $\delta$ is of the following form (and $r = r' + r''$):

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

We argue exactly as in the case when the last rule of $\delta$ is a *pair*.

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

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

We have to consider the following cases:

1. Suppose $\delta'$ is not a purely synthetic proof. By induction hypothesis, for every $u$ occurring in $\delta'$, there are $p' \in st(X)$ and keyword $x'$ such that $u = \{p'\}_{x'}\downarrow$. We turn our attention to $u$ occurring in $\delta''$. By induction hypothesis, either $u \in st(s)$ or there are $v \in st(X)$ and keyword $y$ such that $u = \{v\}_y\downarrow$. But note that $s \in st(r + s)$, and there are $p \in st(X)$ and keyword $x$ such that $r + s = \{p\}_x$. Therefore, if $u \in st(s)$, clearly there are $v' \in st(X)$ and a keyword $x'$ 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 are $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.
2. Suppose $\delta'$ is a purely synthetic proof. But then the last rule of $\delta'$ is not the *encrypt* rule, and hence the last rule of $\delta'$ is an instance of the *blindpair* rule but this would violate the normality of $\delta$, as the transformation rule specified in the third row of Figure 2 can be applied to $\delta$. So this case is not possible.

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

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

We have to consider the following cases:

1. Suppose $r$ is not a blind pair, and hence $\delta$ is a purely synthetic proof. Then we aim to prove that for every $u$ occurring in $\delta$, either $u \in st(r)$ or there are $p \in st(X)$ 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 exist $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$. But since $r' \in st(\{r'\}_k)$, 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, when $u = r$.

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 are $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$. We consider the following sub-cases:
   (a) Suppose $\delta'$ is not a purely synthetic proof, and for every $u$ occurring in $\delta'$, there are $p' \in st(X)$ and keyword $x'$ such that $u = \{p'\}_{x'}\downarrow$. In particular, there are $p \in st(X)$ and keyword $x$ such that $r' = \{p\}_x\downarrow$. But this means that $r = \{p\}_{xk}\downarrow$. Suppose $u$ occurs in $\delta''$. Since $k$ is atomic, the last rule of $\delta''$ is an *analz* rule. So there are $q \in st(X)$ and keyword $y$ such that $u = \{q\}_y\downarrow$. Thus we have proved that for every $u$ occurring in $\delta$, there are $p \in st(X)$ 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 the transformation rule specified by the fourth row of Figure 2 can be applied to $\delta$. So this case is not possible.

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

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

We first note that $inv(k)$ is an atomic key. Hence the last rule of $\delta''$ should be an *analz* rule. Hence for every $u$ occurring in $\delta''$, there exist $p \in st(X)$ and keyword $x$ such that $u = \{p\}_x\downarrow$.
We now consider $\delta'$. The last rule of $\delta'$ cannot be a *blindpair* rule, since the transformation rule specified by the fifth row of Figure 2 can be applied to $\delta$, thereby contradicting the normality of $\delta$. Nor can the last rule of $\delta'$ be an *encrypt* rule; otherwise, the transformation rule specified by the second row of Figure 2 can be applied to $\delta$ and this would contradict the normality of $\delta$.
The remaining possibility for the last rule of $\delta'$ is one 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 $blindsplit_0$ (the proof is similar when the last rule of $\delta'$ is $blindsplit_1$). Let $r_1$ be the major premise and $r_2$ be the minor premise of this rule. Now it cannot be the case that $r_1$ is of the form $\{r\}_k + \{r'\}_k$. For, in that case $r_2$ would have been $\{r'\}_k$, and the transformation rule specified by the sixth row of Figure 2 can be applied to $\delta$, and this would contradict the normality of $\delta$.
We also know from the induction hypothesis (applied to $\delta'$), there are $p \in st(X)$ and keyword $x$ such that $r_1 = \{p\}_x$. But since $r_1$ is of the form $\{r\}_k + r_2$, where $r_2$ is not of the form $\{r'\}_k$ for any $r'$, we conclude that $x = \varepsilon$ and $r_1 = p \in st(X)$. It follows that $r \in st(X)$ as well. □

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

Fix a finite set of terms $X_0$ and a term $t_0$. Let $Y_0$ denotes $st(X_0 \cup \{t_0\})$ and $K_0 = Y_0 \cap \mathcal{K}$. In this section, we address the question of whether there exists a normal proof of $t_0$ from $X_0$.

The weak locality property (Lemma 2) provides a key to the solution – every term occurring in such a proof is of the form $\{p\}_x$ for $p \in Y_0$ and $x \in K_0^*$.

For every $p \in Y_0$, define $\mathcal{L}_p = \{x \in K_0^* \mid X_0 \vdash \{p\}_x\}$. It is easy to see that $\mathcal{L}_p$ satisfies the following equations:

$$\text{if } x \in \mathcal{L}_p \text{ and } x \in \mathcal{L}_{p'} \text{ then } x \in \mathcal{L}_{p+p'},$$
$$\text{if } x \in \mathcal{L}_p \text{ and } x \in \mathcal{L}_{p+p'}, \text{ then } x \in \mathcal{L}_{p'},$$
$$\text{if } x \in \mathcal{L}_{p'} \text{ and } x \in \mathcal{L}_{p+p'}, \text{ then } x \in \mathcal{L}_p$$
$$kx \in \mathcal{L}_p \text{ iff } x \in \mathcal{L}_{\{p\}_k},$$
$$\text{if } x \in \mathcal{L}_p \text{ and } \varepsilon \in \mathcal{L}_k, \text{ then } xk \in \mathcal{L}_p, \text{ and}$$
$$\text{if the empty string } \varepsilon \in \mathcal{L}_{\{p\}_k} \text{ and } \varepsilon \in \mathcal{L}_{inv(k)}, \text{ then } \varepsilon \in \mathcal{L}_p.$$

If $p, p', p+p'$ are considered as states and $x$ is accepted from $p$ as well as $p'$, then we want $x$ is to be accepted from $p + p'$. To capture this we need an and edge (labeled with $\epsilon$) from $p$ and $p'$ to $p + p'$. This suggests the construction of an alternating automaton $\mathcal{A}$ such that checking $X \vdash \{t\}_x$ is equivalent to checking whether there is an accepting path of $x$ from $t$ in $\mathcal{A}$. First we recall the definition of alternating automaton and other related notions.

Definition 6. *An* alternating automaton *is $\mathcal{A} = (Q, \Sigma, \hookrightarrow, F)$, where $Q$ is a finite set of states, $\Sigma$ is a finite alphabet, $\hookrightarrow \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ is the transition relation, and $F \subseteq Q$ is the set of final states.*

*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 $(q, a, C) \in \hookrightarrow$. For ease of notation, we 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 $q \overset{x}{\Rightarrow}_{\mathcal{A},i} C$ iff*

- *$C = \{q\}, x = \varepsilon$, and $i = 0$, or*
- *there is a transition $q \overset{a}{\hookrightarrow} \{q_1, \ldots, q_n\}$ of $\mathcal{A}$, $y \in \Sigma^*$, and $i_1, \ldots, i_n \geq 0$ such that $i = i_1 + \cdots + i_n + 1$ and $x = ay$ and for all $j \in \{1, \ldots, n\}$, $q_j \overset{y}{\Rightarrow}_{\mathcal{A},i_j} C_j$ such that $C = C_1 \cup \cdots \cup C_n$.*

*For $C = \{q_1, \ldots, q_m\}$ and $C' \subseteq Q$, we use the notation $C \overset{x}{\Rightarrow}_{\mathcal{A},i} C'$ to mean that for all $j \leq m$, there exist $i_j$ such that $q_j \overset{x}{\Rightarrow}_{\mathcal{A},i_j} C_j$, and $i = i_1 + \cdots + i_m, C' = C_1 \cup \cdots \cup C_m$. We also say $q \overset{x}{\Rightarrow}_{\mathcal{A}} C$ and $C \overset{x}{\Rightarrow}_{\mathcal{A}} C'$ to mean that there is some $i$ such that $q \overset{x}{\Rightarrow}_{\mathcal{A},i} C$ and $C \overset{x}{\Rightarrow}_{\mathcal{A},i} C'$, respectively.*

*We say a word $x$ has an accepting run from $q$ iff $q \overset{x}{\Rightarrow}_{\mathcal{A}} C$ such that $C \subseteq F$. For a given $q$, is the set of words accepted by $\mathcal{A}$ with $q$ as initial state.*

$$\mathcal{L}(\mathcal{A}, q) = \{x \in \Sigma^* \mid q \overset{x}{\Rightarrow}_{\mathcal{A}} C \text{ such that } C \subseteq F\}$$

We typically drop the subscript $\mathcal{A}$ if it is clear from the context which alternating automaton is referred to.

Now we construct an alternating automaton $\mathcal{A}$ such that $\mathcal{L}_p = \mathcal{L}(\mathcal{A}, p)$ for each $p \in Y_0$. The states of the automaton are terms from $Y_0$, and the transition relation is a direct transcription of the equations in 1. For instance there is an edge labeled $k$ from $t$ to $\{t\}_k$, and there is an edge labeled $\varepsilon$ from $t$ to the set $\{t + t', t'\}$. We introduce a final state $f$ and introduce an $\varepsilon$-labeled edge from $t$ to $f$ whenever $\varepsilon \in \mathcal{L}_t$.

Definition 7. *Let $\mathcal{A}_0$ be given by $(Q, \Sigma, \hookrightarrow_0, F)$ where $Q = Y_0 \cup \{f\}$ ($f \notin Y_0$), $\Sigma = K_0$, $F = \{f\}$, and $\hookrightarrow_0$ be the smallest subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ that satisfies the following:*

- *if $t \in Y_0, k \in K_0$ such that $\{t\}_k\downarrow \in Y_0$, then $t \overset{k}{\hookrightarrow}_0 \{t\}_k\downarrow$.*
- *if $t, t', t'' \in Y_0$ such that $t$ is the conclusion of a blindpair or blindsplit$_i$ rule with premises $t'$ and $t''$, then $t \overset{\varepsilon}{\hookrightarrow}_0 \{t', t''\}$.*
- *if $t \in X_0$, then $t \overset{\varepsilon}{\hookrightarrow}_0 \{f\}$.*
- *if $k \in X_0 \cap K_0$, then $f \overset{k}{\hookrightarrow}_0 \{f\}$.*

There is one issue in this automaton $\mathcal{A}_0$: if $kx \in \mathcal{L}_t$ then $x \in \mathcal{L}_{\{t\}_k}$. These cannot be represented directly by a transition in the automaton. Thus we define a revised automaton that has an edge labeled $\varepsilon$ from $\{t\}_k$ to $q$ whenever the original automaton has an edge labeled $k$ 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 $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_i, \ldots$, each of which adds transitions to the previous one, as given by the below definition.

Definition 8. *For each $i > 0$, $\mathcal{A}_i$ is given by $(Q, \Sigma, \hookrightarrow_i, F)$ where $\hookrightarrow_i$ is the smallest subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$ such that:*

1. *if $q \overset{a}{\Rightarrow}_{i-1} C$, then $q \overset{a}{\hookrightarrow}_i C$.*
2. *if $\{t\}_k\downarrow \in Y_0$ and $t \overset{k}{\Rightarrow}_{i-1} C$, then $\{t\}_k\downarrow \overset{\varepsilon}{\hookrightarrow}_i C$.*
3. *if $k \in K_0$ and $k \overset{\varepsilon}{\Rightarrow}_{i-1} \{f\}$, then $f \overset{k}{\hookrightarrow}_i \{f\}$.*
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 (unary or binary) whose set of premises is (exactly) $\Gamma$ and conclusion is $t$, then the following holds:*

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

*We use $\hookrightarrow_i$ for $\hookrightarrow_{\mathcal{A}_i}$ and $\Rightarrow_i$ for $\Rightarrow_{\mathcal{A}_i}$*

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

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

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

The the number of subterms is $O(n^2)$ if $X_0, t_0$ is of size $O(n)$. So we have to iterate the saturation procedure at most $2^{n^2}$ (the number of subsets of states) times.

Theorem 3. *Given a finite $X_0 \subseteq \mathcal{T}$ and $t_0 \in \mathcal{T}$, checking whether $X_0 \vdash t_0$ is solvable in time $O(2^{n^2})$ where $n = \sum_{t \in X_0} |t| + |t_0|$.*

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

In this section, we reduce the reachability problem of alternating pushdown systems to the intruder deduction problem. The reduction is similar to the reduction in [5] with a few modifications.

Definition 9. *An alternating pushdown system (APDS) is a triple* $\mathcal{P} = (P, \Gamma, \Delta)$, *where*

- $P$ *is a finite set of* control locations,
- $\Gamma$ *is a finite* stack alphabet, *and*
- $\Delta \subseteq (P \times \Gamma^*) \times 2^{(P \times \Gamma^*)}$ *is a 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* $(a, x) \overset{i}{\Rightarrow}_{\mathcal{P}} C$ *iff:*

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

*We use* $(a, x) \Rightarrow_{\mathcal{P}} C$ *to denote* $(a, x) \overset{i}{\Rightarrow}_{\mathcal{P}} C$ *for some* $i$.

Theorem 4 ([14]). *The* reachability problem for alternating pushdown systems, *which asks, given an APDS* $\mathcal{P}$ *and configurations* $(s, x_s)$ *and* $(f, x_f)$, *whether* $(s, x_s) \Rightarrow_{\mathcal{P}} (f, x_f)$, *is* DEXPTIME*-complete.*

We reduce this problem to the problem of checking whether $X \vdash t$ in our proof system, given $X \subseteq \mathcal{T}$ and $t \in \mathcal{T}$. We use $\{c\}_x \wedge \{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \overset{Ass}{\Longrightarrow} \{b\}_y$, called associative rewrite terms, to denote the following term

$$\{b_1\}_{y_1} + \{c\}_x + \{b_2\}_{y_2} + \{c\}_x + \cdots + \{c\}_x + \{b_n\}_{y_n} + \{c\}_x + \{b\}_y + \{c\}_x + \{b_1\}_{y_1} + \{c\}_x + \{b_2\}_{y_2} + \{c\}_x + \cdots + \{c\}_x + \{b_n\}_{y_n}$$

where $c, b_1, \ldots, b_n, b$ be set of basic terms and let $x, y_1, \ldots, y_n, y$ be keywords.

Definition 10. *Suppose* $\mathcal{P} = (P, \Gamma, \hookrightarrow)$ *is an APDS, and* $(s, x_s)$ *and* $(f, x_f)$ *are two configurations of* $\mathcal{P}$. *The rules in* $\hookrightarrow$ *are numbered* $1$ *to* $l$.

*We define a set of terms* $X$ *such that* $(s, x_s) \Rightarrow_{\mathcal{P}} (f, x_f)$ *iff* $X \vdash \{s\}_{x_s e}$.

- $P \cup C$ *is taken to be a set of basic terms, where* $C = \{c_1, \ldots c_l\}$,
- $\Gamma \cup \{e, d\}$ *is taken to be a set of keys, such that* $e, d \notin \Gamma$, *and none of the keys in* $\Gamma \cup \{e\}$ *is an inverse of another,*
- $X_1 = \{\{f\}_{x_f e}\} \cup \{\{c\}_d \mid c \in C\}$.
- $X_2 = \{\{c_i\}_d \wedge \{b_1\}_{x_1} \wedge \cdots \wedge \{b_n\}_{x_n} \overset{Ass}{\Longrightarrow} \{a\}_x \mid (a, x) \hookrightarrow_{\mathcal{P}} \{(b_1, x_1), \ldots, (b_n, x_n)\}$ *is the* $i$ *th rule of* $\hookrightarrow\}$, *and*

*In the rest of the section, we assume* $X = X_1 \cup X_2 \cup \Gamma \cup \{e\}$.

Lemma 4. *If* $\{c\}_d \wedge \{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \stackrel{Ass}{\Longrightarrow} \{b\}_y$ *is an associative rewrite term in* $X_2$ *and* $z \in \Gamma^*$ *such that for all* $i \leq n : X \vdash \{b_i\}_{y_i z e}$*, then* $X \vdash \{b\}_{yze}$.

We can encrypt $\{c\}_d$ using the keys in $ze$ to derive $X \vdash \{c\}_{dze}$. Using blindsplit rule on associative rewrite term, we can derive $X \vdash \{b\}_{yze}$.

Using the above lemma we can prove if $(a, x) \Rightarrow_i \{(f, x_f)\}$, then $X \vdash \{a\}_{xe}$.

Lemma 5. *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 $\mathcal{P}$, $(a, y) \hookrightarrow \{(b_1, y_1), \ldots, (b_n, y_n)\}$, $z \in \Gamma^*$, and $i_1, \ldots, i_n \geq 0$ such that $x = yz$ and $(b_j, y_j z) \Rightarrow_{i_j} \{(f, x_f)\}$ for all $j \in \{1, \ldots, n\}$, and such that $i = i_1 + \cdots + i_n + 1$. By induction we know that $X \vdash \{b_j\}_{y_j ze}$ for all $j$. It immediately follows from the definition of $X_2$ and Lemma 4 that $X \vdash \{a\}_{yze}$. Since $x = yz$, $X \vdash \{a\}_{xe}$.

To prove the converse of Lemma 5, we have to prove some properties of the normal proof of $X \vdash \{a\}_{xe}$. First, we make some observations about the normal proof $\pi$ of $X \vdash \{a\}_{xe}$. There are no *pair*, *split*, *decrypt* rules in $\pi$. This is easy to see from the set $X$ and the conclusion. Most importantly, there are no *blindpair* rules in $\pi$. Since the conclusion is not a blindpair term, the transformation rules in Figure 1 eliminate the *blindpair* rules.

Lemma 6. *Let* $\pi$ *be a normal proof of* $X \vdash \{a\}_{xe}$*, for* $a \in P$ *and* $x \in \Gamma^*$*. Then any term* $u$ *occurring in* $\pi$ *is of the form* $\{p\}_w$*, for* $p \in st(X)$ *and* $w \in \Gamma^* \cup \Gamma^* e$.

*Proof.* 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 $\{q\}_{yey'}$ occurs in $\pi$, such that $y' \neq \varepsilon$. Since the conclusion of $\pi$ is $\{a\}_{xe}$ with $x \in \Gamma^*$, there has to be an occurrence of a rule $R$ with a premise $\{r\}_{zez'}$ and a conclusion $\{t\}_w$ such that $r, t \in st(X)$, $z' \neq \varepsilon$, $e \notin st(t)$, and $w \in \Gamma^* \cup \Gamma^* e$. Clearly $R$ cannot be an $\mathcal{A}x$ rule.

- Suppose $R$ is an *encrypt* rule. The conclusion $\{r\}_{zez'k}$ is not equal to $\{t\}_w$ for any $t \in st(X)$, and $w \in \Gamma^* \cup \Gamma^* e$ such that $e \notin st(t)$, which is a contradiction.
- Suppose $R$ is a *blindsplit* rule with $\{r\}_{zez'}$ as the left side premise. Then it is clear that $\{t\}_w$ is also of the form $\{q\}_{zez'}$ which is a contradiction.
- Suppose $R$ is a *blindsplit* rule with $\{r\}_{zez'}$ as the right side premise and $t'$ as the left side premise of $R$. It is easy to see that $t' = \{t\}_w + \{r\}_{zez'}$. Since $w \in \Gamma^* \cup \Gamma^* e$, it cannot have common suffix with $zez'$. Hence, $t' \in st(X)$. Since $e$ is a proper subterm of $t'$, it should be equal to $\{f\}_{x_f e}$ which contradicts that $t'$ is a blind pair term. □

The following lemma constrains the structure of rules that occur in any normal proof of $X \vdash \{a\}_{xe}$. This lemma is weaker than its counterpart in [5] as the right side premise of blindsplit may be a blindpair term.

Lemma 7. *Let* $\pi$ *be a normal proof of* $X \vdash \{a\}_{xe}$*, for* $a \in P$ *and* $x \in \Gamma^*$*. Let* $\delta$ *be a subproof of* $\pi$ *with root labeled* $r$.

1. *If the last rule of $\delta$ is an encrypt rule, then $r = \{p\}_w$ for some $p \in X$ and keyword $w \in \Gamma^* \cup \Gamma^* e$.*
2. *If the last rule of $\delta$ is a blindsplit rule, then $r = \{p\}_{we}$, where $p \in st(X)$ and $w \in \Gamma^*$.*

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

1. Suppose the last rule of $\delta$ is an *encrypt* rule, and has the following structure:

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

   If the last rule of $\delta'$ is an *Ax* rule, then $r' \in X$. Hence, $r$ is of the form $\{r'\}_k$ with $r' \in X$. If the last rule of $\delta'$ is an *encrypt* rule, then $r' = \{p\}_w$ for some $p \in X$. In that case we are done, since $r = \{p\}_{wk}$. The other option is that the last rule of $\delta'$ is a *blindsplit* rule, in which case $r'$ is of the form $\{p\}_{we}$ (by part 2 of this lemma applied to $\delta'$). But then $r = \{p\}_{wek}$, and that violates Lemma 6, so this case cannot arise.
2. Suppose the last rule of $\delta$ is a *blindsplit* rule and has the following form:

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

   - Suppose the last rule of $\delta'$ is an *Ax* rule. Then $r' \in X_2$. Now let us look at the last rule of $\delta''$. If it is either *Ax* rule or *encrypt* rule, $r''$ is of the form, $\{p''\}_{z''}$, for some $p'' \in X$. Suppose $p'' \in X_2$ then there is a $c \in C$ which is a subterm of $r'$ and $p''$ which contradicts that no $c \in C$ is a subterm of two terms in $X_2$. Hence $p'' \notin X_2$. Moreover, $p''$ cannot be in $X_3$ as $r'$ is neither starting or ending with $\{c\}_w$ for any $c \in C$ and $w$. If the last rule of $\delta''$ is a *blindsplit* rule, then by hypothesis, $r''$ is of the form $\{p''\}_{z''e}$. But this contradicts that $r''$ is a subterm of $r' \in X_2$ (No term in $X_2$ contains $e$). Hence the last rule of $\delta'$ cannot be an *Ax* rule.
   - If the last rule of $\delta'$ is *blindsplit* rule, then by induction hypothesis, $r' = \{p\}_{w'e}$ for $p \in st(X)$ and $w' \in \Gamma^*$. Hence, $r = \{p'\}_{z''e}$ for some $p' \in st(X)$ and $z'' \in \Gamma^*$.
   - Suppose the last rule of $\delta'$ is *encrypt*, then $r' = \{p'\}_z$ for $p' \in X$. If $z \in k^* e$, then $r$ is of the required form. Suppose $z$ is not ending with $e$.
     Clearly, $p'$ is a blind pair term, so $p' \in X_2$. Now let us look at the last rule of $\delta''$.
     - If it is *blindsplit* rule, then $r''$ is of the form $\{p''\}_{z''e}$, using induction hypothesis. Then $r'$ is also of the form $\{p'\}_{z'e}$, as no blindpair term in $X$ contains $e$. Hence $r$ is also of this form.
     - Suppose the last rule of $\delta''$ is *Ax* rule. Clearly, $r''$ cannot be a blind pair term as it should contain a different $\{c_m\}_d$. It is easy to see that $r'' \notin X_3$ as $p'$ is neither starting nor ending with a term in $X_3$. If $r'' = \{a\}_{xe}$, then $r'$ is of the form $\{p'\}_{z'e}$ as no blind pair term in $X$ contains $e$. Hence $r$ is of the required form.

- Suppose the last rule of $\delta''$ is *encrypt*. Then $r''$ is of the form $\{p''\}_{z''}$ for some $p'' \in X$. If $p'' \in X_1$, then $r'$ is of the required form, and hence $r$. Moreover, $p''$ cannot be in $X_3$ as $p'$ is neither starting nor ending with a term in $X_3$. If $p'' \in X_2$, then we cannot use it to blind split $r'$. Hence $r$ is of the form $\{p\}_{we}$ for some $p \in st(X)$. □

We now state an important property of normal proofs from rewrite systems – namely that whenever the "conclusion" of a rewrite term is provable, all the "premises" are provable too. The proof of the lemma is given in appendix.

Lemma 8. *Let $\pi$ be a normal proof of $X \vdash \{a\}_{xe}$, for $a \in P$ and $x \in \Gamma^*$. Then either $\{a\}_{xe} \in X_1$ or there is a rewrite term $\{c_m\}_d \wedge \{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \stackrel{Ass}{\Longrightarrow} \{a\}_y$ in $X_2$, and $z \in \Gamma^*$ such that $x = yz$, for all $i \leq n$, $\{b_i\}_{y_i ze}$ occurs in $\pi$.*

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

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

- If $r \notin X_3, t = t_1 + t_2 + \cdots + t_i + r + t_{i+1} + \ldots t_n$, and none of the $t_i\downarrow$'s are headed with $+$, then *residues*$(t, r) = \bigcup_{i=1}^{n} \{t_i\} \setminus \{\{c_m\}_d\}$.
- *residues*$(t, r) = \emptyset$, otherwise.

Suppose $\{r\}_{ze}$ occurs as the root of a subproof $\delta$ of $\pi$, for any $r \in st(X_2)$. We prove that either $\{r\}_{ze} \in X_1$, or there is $t \in X_2$ such that $\{b\}_{yze}$ occurs in a proper subproof of $\delta$ for every $\{b\}_y \in$ *residues*$(t, r)$. The statement of the lemma follows immediately. We distinguish the following three cases:

- Suppose the last rule of $\delta$ is *Ax* rule. Then $\{r\}_{ze} \in X$. Since the only terms from $X_1$ contain $e$ as a proper subterm, $\{r\}_{ze} \in X_1$.
- Suppose the last rule of $\delta$ is *encrypt*. Then $\{r\}_{ze} = \{p\}_{we}$ for some $p \in X$. But then it has to be the case that $r \in X_2 \cup X_3$. It is easy to see that *residues*$(t, t') = \emptyset$ for every $t \in X_2$ and for every $t' \in X_2 \cup X_3$. Hence, our statement is vacuously true.
- Suppose the last rule of $\delta$ is a *blindsplit* rule. Let $\{u\}_{ze}$ be the left side premise and $\{u'\}_{z'e}$ be the right side premise such that $u, u' \in st(X_2)$. By induction hypothesis, there is $t \in X_2$ such that $\{b\}_{yze}$ occurs in $\delta$ for every $\{b\}_y \in$ *residues*$(t, u)$. We distinguish two cases now.
  - Suppose, $\{u'\}_{z'e} = \{c_m\}_{dz'e}$. Then *residues*$(t, r) =$ *residues*$(t, r + \{c_m\}_d) =$ *residues*$(t, \{c_m\}_d + r)$. Hence the claim.
  - Suppose, $\{u'\}_{z'e} = \{b_j\}_{y_j ze}$. Then *residues*$(t, r) =$ *residues*$(t, u) \cup \{\{b_j\}_{y_j})\}$. It is easy to see that $\{b_j\}_{y_j we}$ occurs in a proper subproof of $\delta$. Moreover, for every $\{b\}_y \in$ *residues*$(t, u)$, $\{b\}_{yze}$ occurs in a proper subproof $\delta$ ( by induction hypothesis). Hence the claim.
  - Suppose $\{u'\}_{z'e}$ is headed with $+$, then it is easy to see that *residues*$(r, t) =$ *residues*$(u, t) \cup$ *residues*$(u', t)$. We use the fact that $u$ and $u'$ are subterms of a same term from $X_2$. Now using induction hypothesis, we can claim that for every $\{b\}_y \in$ *residues*$(t, r)$, and hence $\{b\}_{yze}$ occurs in a proper subproof of $\delta$. Hence the claim. □

Lemma 9. *For any configuration $(a, x)$, if there is a normal proof of $X \vdash \{a\}_{xe}$, then $(a, x) \Rightarrow_{\mathcal{P}} (f, x_f)$.*

*Proof.* By Lemma 8, $X \vdash \{a\}_{xe}$ means that either $\{a\}_{xe} \in X_1$ or there is an associative rewrite term $\{c\}_d \wedge \{b_1\}_{y_1} \wedge \cdots \wedge \{b_n\}_{y_n} \stackrel{Ass}{\Longrightarrow} \{a\}_y$ in $X_2$, and $z \in \Gamma^*$ such that $x = yz$ and for all $i \leq n$, $\{b_i\}_{y_i ze}$ occurs in $\pi$.

In the first case ($\{a\}_{xe} \in X_1$), $a = f$ and $x = x_f$, and it follows that $(a, x) \Rightarrow_{\mathcal{P}} (f, x_f)$. In the second case, by induction hypothesis, $(b_i, y_i z) \Rightarrow_{\mathcal{P}} (f, x_f)$, for all $i \leq n$. Combined with $(a, y) \hookrightarrow \{(b_1, y_1), \ldots, (b_n, y_n)\}$, it follows that $(a, x) = (a, yz) \Rightarrow_{\mathcal{P}} (f, x_f)$.

For a given APDS $\mathcal{P} = (P, \Gamma, \Delta)$ and configurations $(s, x_s)$ and $(f, x_f)$, we have constructed a set of terms $X$ such that $X \vdash \{s\}_{x_s e}$ iff $(s, x_s) \Rightarrow_{\mathcal{P}} (f, x_f)$. This reduction and the fact that reachability problem for alternating pushdown systems is DEXPTIME-hard lead to the following main result.

Theorem 5. *Given a finite $X \subseteq \mathcal{T}$ and a term $t \in \mathcal{T}$, checking whether $X \vdash t$ is* DEXPTIME*-hard.*

## 6 Discussion

The techniques of our paper do not seem to extend to the system with abelian group operators, 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 $-t$. 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 $X \cup \{t\}$. But 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. But the relationships between the two techniques need to be studied in more depth and might be useful to solve weaker systems and the system with an abelian group operators. We leave this for future work.

## References


1. Abadi, M., Cortier, V.: Deciding knowledge in security protocols under equivalence thoeries. Theoretical Computer Science 367(1–2), 2–32 (2006)
2. Avanesov, T., Chevalier, Y., Rusinowitch, M., Turuani, M.: Satisfiability of general intruder constraints with and without a set constructor. Journal of Symbolic Computation 80, 27–61 (2017)
3. Baskar, A.: Decidability Results For Extended Dolev-Yao Theories. Ph.D. thesis, Chennai Mathematical Institute (2011)
4. Baskar, A., Ramanujam, R., Suresh, S.: Knowledge-based modelling of voting protocols. In: Samet, D. (ed.) Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge. pp. 62–71 (2007)

5. Baskar, A., Ramanujam, R., Suresh, S.: A DEXPTIME-complete Dolev-Yao theory with distributive e encryption. In: Proceedings of the 35th International Symbosium on Mathematical Foundations of Computer Science. Lecture Notes in Computer Science, vol. 6281, pp. 102–113 (2010)
6. Baskar, A., Ramanujam, R., Suresh, S.: Dolev-yao theory with associative blindpair operators (2019), technical report available at `http://www.cmi.ac.in/ spsuresh/pdfs/ciaa19-tr.pdf`
7. Ciobâca, S., Delaune, S., Kremer, S.: Computing knowledge in security protocols under convergent equational theories. Journal of Automated Reasoning 48(2), 219–262 (2012)
8. Comon-Lundh, H., Shmatikov, V.: Intruder Deductions, Constraint Solving and Insecurity Decisions in Presence of Exclusive or. In: Proceedings of the 18th IEEE Synposium on Logic in Computer Science (LICS). pp. 271–280 (June 2003)
9. Cortier, V., Delaune, S.: Decidability and combination results for two notions of knowledge in security protocols. Journal of Automated Reasoning 48(4), 441–487 (2012)
10. Cortier, V., Delaune, S., Lafourcade, P.: A survey of algebraic properties used in cryptographic protocols. Journal of Computer Security 14(1), 1–43 (2006)
11. Dolev, D., Yao, A.: On the Security of public-key protocols. IEEE Transactions on Information Theory 29, 198–208 (1983)
12. Lafourcade, P., Lugiez, D., Treinen, R.: Intruder deduction for the equational theory of abelian groups with distributive encryption. Information and Computation 205(4), 581–623 (April 2007)
13. Rusinowitch, M., Turuani, M.: Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete. Theoretical Computer Science 299, 451–475 (2003)
14. Suwimonteerabuth, D., Schwoon, S., Esparza, J.: Efficient algorithms for alternating pushdown systems with an application to the computation of certificate chains. In: Proceedings of the Automated Technology for Verification (ATVA2006). LNCS, vol. 4218, pp. 141–153 (2006)