A decidable subclass of unbounded security protocols

R. Ramanujam and S. P. Suresh

The Institute of Mathematical Sciences
C.I.T. Campus, Chennai 600 113, India.
E-mail: {jam,spsuresh}@imsc.res.in

1 Summary

The verification problem for security protocols can be formulated as follows: given an abstract specification of the protocol as a sequence of communications between agents, is it the case that every run generated by possible multi-sessions between agents, with a hypothetical intruder interleaving arbitrarily many actions, satisfies the given security requirements? There are many requirements but an important (and central) requirement is that of *secrecy*: a secret that is generated by an honest agent should not be leaked to the intruder, who is assumed to have unlimited computational resources and can keep a record of every public system event and utilize it at an arbitrarily later time. However, the intruder cannot generate an honest agent's secret autonomously, nor can it break encryption.

A crucial requirement on runs is that of *freshness*: every time an agent sends out a secret (a nonce), it is a new one — an obvious requirement to avoid the intruder *replaying* old sessions. But this means that when there is no bound on the number of plays of roles by agents, the number of nonces used grows unboundedly as well. [DLMS99] pinpoint to such unbounded generation of nonces as a problem, and use it to show that the secrecy problem for protocols is undecidable, even when the number of roles, the length of each role and message length are bounded. They go on to show that for systems without the freshness constraint, the problem becomes decidable. In fact, we can get decidability as long as the honest agents are finite-state systems, which is equivalent to placing a bound on the number of fresh nonces generated by them.

An alternative to placing bounds on fresh nonces is to look for subclasses of protocols in which, by virtue of the manner in which communication patterns between agents are structured, decidability obtains. The definition of such a subclass is arrived at by a detailed analysis of the undecidability proof; while we cannot hope for an exact characterization, it suffices to come up with a restriction that is strong enough to exclude the "source" of undecidability while yet retaining a large enough class of interesting protocols.

In this paper, we propose a simple syntactic restriction on protocols and show that it achieves this purpose. The condition essentially states that between any two terms that occur in distinct communications, no encrypted subterm

of one can be unified with a subterm of the other. In the absence of such a restriction, the intruder may use such a binding to transfer information from one play to another, and 'pump' this process (using unboundedly many nonces) to generate unboundedly many plays with distinct information content, leading to undecidability. We show how the restriction leads to a bound on the size of (partial) runs that need to be checked for a leak. It is also easily seen that the subclass includes a wide variety of protocols studied in the literature, for instance, most of the protocols presented in the survey ([CJ97]).

It also turns out that without the restriction, the halting problem for two-counter machines may be coded, illustrating the comment above relating to the source of undecidability. On the other hand, for the subclass studied, the decidability result extends to other properties than secrecy as well, those which can be stated in a simple modal logic.

Several approaches have been adapted to obtain decidability of the verification problem for security protocols: We refer to [CS02] for an overview. The approach we follow is close to that of [Low98], but our notion of secrecy differs from the one in that paper. The other source of undecidability, as pointed out in [HT96], is unbounded length of the messages in the runs of the protocol. For a sample of the techniques used to obtain decidability in this case, we refer to [MS01], where the verification problem for *bounded-process protocols* – which essentially come with a bound on the length of their runs – are proved to be decidable. While the focus in this paper is to obtain decidability in the presence of unboundedly many nonces but bounded message length, in a companion paper [RS03], we prove the secrecy problem to be decidable for a subclass of protocols – called *normal protocols* – in the presence of unbounded message length but boundedly many nonces.

2 Security protocols and their semantics

Fix a finite set of *agents* Ag with a special *intruder* $I \in Ag$. $Ag \setminus \{I\}$ is denoted by Ho. The set of *keys* K is $K_{lt} \cup K_{st}$ where K_{lt}, the set of *long-term keys* is the set $\{k_{AB}, pubk_A, privk_A \mid A, B \in Ag, A \neq B\}$, and K_{st} is the set of *short-term keys*. $pubk_A$ is A's *public key* and $privk_A$ is its *private key*. k_{AB} is the long-term key *shared* by A and B. For every $k \in K$ define $\overline{k} \in K$ as follows: for the shared keys and short-term keys $\overline{k} = k$, whereas $\overline{pubk_A} = privk_A$ and $\overline{privk_A} = pubk_A$. $\overline{k}$ is k's *inverse key*. For $A \in Ag$, $K_A \stackrel{\text{def}}{=} \{pubk_B, k_{AB} \mid B \neq A\} \cup \{privk_A\}$ is the set of keys known to A. Also fix an infinite set of *nonces* N. Define the set of *basic terms* T_0 to be $K \cup N \cup Ag$.

Define the set of information terms to be

$$\mathcal{T} \quad ::= \quad m \mid (t, t') \mid \{t\}_k$$

where m ranges over $T_0 \setminus K_{lt}$ and k ranges over K. We define the set of subterms of a term t, $ST(t)$, to be the least set T such that: $t \in T$; if $(t, t') \in T$ then $t \in T$ and $t' \in T$; and if $\{t\}_k \in T$ then $t \in T$. $ST(T) = \bigcup_{t \in T} ST(t)$ for any

$$\frac{}{T \cup \{t\} \vdash t}\,\mathsf{Ax}_a$$

$$\frac{T \vdash (t_1, t_2)}{T \vdash t_i}\,\mathsf{unpair}_i (i = 1, 2)$$

$$\frac{T \vdash \{t\}_k \qquad T \vdash \overline{k}}{T \vdash t}\,\mathsf{decrypt}$$

$$\frac{T \vdash \{\{t\}_k\}_{\overline{k}}}{T \vdash t}\,\mathsf{reduce}$$

analz-rules

$$\frac{}{T \cup \{t\} \vdash t}\,\mathsf{Ax}_s$$

$$\frac{T \vdash t_1 \qquad T \vdash t_2}{T \vdash (t_1, t_2)}\,\mathsf{pair}$$

$$\frac{T \vdash t \qquad T \vdash k}{T \vdash \{t\}_k}\,\mathsf{encrypt}$$

synth-rules

Fig. 1. analz and synth rules.

$T \subseteq \mathcal{T}$. For a set of terms T and a key k we say that k is *referred to* in T if $k \in T$ or $\exists t : \{t\}_k \in T$. $EST(t)$, the set of *encrypted subterms* of any $t \in \mathcal{T}$ is the set $\{t' \in ST(t) \mid \exists t'', k : t' = \{t''\}_k\}$. $|t|$, the *size* of t is defined inductively as follows: $|m| = 0; |(t, t')| = |t| + |t'| + 1; |\{t\}_k| = |t| + 1$.

$\Sigma = \{A!B{:}(M)t, A?B{:}t \mid A, B \in Ag, A \neq B, t \in \mathcal{T}, M \subseteq ST(t) \cap T_0\}$ is the set of *actions*. For $a = A!B{:}(M)t$, $term(a) = t$ and $NT(a) = M$. Similarly for $a = A?B{:}t$, $term(a) = t$ and $NT(a) = \emptyset$. For any action a, $|a|$ is defined to be $|term(a)|$. For any send action $A!B{:}(M)t$, $B?A{:}t$ is said to be its *matching receive*. $terms(a_1 \cdots a_\ell) = \{term(a_i) \mid 1 \leq i \leq \ell\}$ and $NT(a_1 \cdots a_\ell) = NT(a_1) \cup \cdots \cup NT(a_\ell)$. For any $\eta \in \Sigma^*$, $CT(\eta) \overset{\text{def}}{=} (T_0 \cap ST(terms(\eta))) \setminus NT(\eta)$ is the set of *constants* of η. An *event* is a pair (η, i) where $\eta \in \Sigma^+$ and $1 \leq i \leq |\eta|$. The set of all events is called *Events*. For $e = (a_1 \cdots a_\ell, i) \in Events$, $act(e) = a_i$.

Note that B is (merely) the intended receiver in $A!B{:}(M)t$ and the purported sender in $A?B{:}t$. As we will see later, every send action is an instantaneous receive by the intruder, and similarly, every receive action is an instantaneous send by the intruder.

Σ_A, the set of A-*actions* is given by $\{C!D{:}(M)t, C?D{:}t \in \Sigma \mid C = A\}$. For any $\eta = a_1 \cdots a_\ell \in \Sigma^*$ and any $A \in Ag$, $\eta \restriction A$ is given by $a_{i_1} \cdots a_{i_r}$ where $\{i_1, \ldots, i_r\} = \{i \leq \ell \mid a_i \in \Sigma_A\}$.

Definition 2.1 *A* sequent *is of the form $T \vdash t$ where $T \subseteq \mathcal{T}$ and $t \in \mathcal{T}$. An* analz*-proof (*synth*-proof) π of $T \vdash t$ is an inverted tree whose nodes are labelled by sequents and connected by one of the* analz*-rules (*synth*-rules) in Figure 1, whose root is labelled $T \vdash t$, and whose leaves are labelled by instances of the* Ax_a *rule (*Ax_s *rule). For a set of terms T,* analz(T) *(*synth(T)*) is the set of terms t such that there is an* analz*-proof (a* synth*-proof) of $T \vdash t$. For ease of notation,* synth(analz(T)) *is denoted by $\overline{T}$.*

The definitions of analz and synth are due to [Pau98]. We will assume a number of basic properties of synth and analz proved in [Pau98].

Definition 2.2 *An* information state s *is a tuple* $(s_A)_{A \in Ag}$ *where for each agent* A, $s_A \subseteq \mathcal{T}$. $\mathcal{S}$ *denotes the set of all information states. The notions of an action enabled at a state and update of a state on an action are given as follows:*

- $A!B{:}(M)t$ *is* enabled *at* s *iff* $t \in \overline{s_A \cup M}$, *and if none of the terms in* M *occurs in* s.
- $A?B{:}t$ *is* enabled *at* s *iff* $t \in \overline{s_I}$.
- $update(s, A!B{:}(M)t) = s'$ *where* $s'_A = s_A \cup M$, $s'_I = s_I \cup \{t\}$, *and* $s'_C = s_C$ *for all the other* $C \in Ag$.
- $update(s, A?B{:}t) = s'$ *where* $s'_A = s_A \cup \{t\}$ *and* $s'_C = s_C$ *for all other* $C \in Ag$.

We extend the notion of update to sequences of actions as follows: $update(s, \varepsilon) = s$, $update(s, \eta \cdot a) = update(update(s, \eta), a)$.

Definition 2.3 *A* **protocol** Pr *is a sequence* $a_1 b_1 \cdots a_\ell b_\ell \in \Sigma^+$ *such that:*

- *for all* $i : 1 \leq i \leq \ell$, b_i *is* a_i*'s matching receive,*
- *for all* $k \in K_{st}$ *referred to in* $ST(terms(\mathsf{Pr}))$, $k \in NT(\mathsf{Pr})$, *and*
- *for* $s_0 = (K_A \cup CT(\mathsf{Pr}))_{A \in Ag}$, *for all* $i : 1 \leq i \leq \ell$, a_i *is enabled at* $update(s_0, a_1 b_1 \cdots a_{i-1} b_{i-1})$.

One of the standard presentations of protocols is as a sequence of *communications* of the form $A \rightarrow B : (M)t$. For technical convenience, we split each communication of the above form into a pair of actions, $A!B{:}(M)t$ and $B?A{:}t$. We also require that all the short-term keys used in the protocol are freshly generated. This is a standard requirement and explains precisely why these keys are called "short-term".

Given a protocol Pr, $Roles(\mathsf{Pr}) \stackrel{\text{def}}{=} \{\mathsf{Pr} \upharpoonright A \mid A \in Ag \text{ and } \mathsf{Pr} \upharpoonright A \neq \varepsilon\}$.

A *substitution* σ is a map from T_0 to T_0 such that: $\sigma(Ag) \subseteq Ag$, if $A \neq B$ then $\sigma(A) \neq \sigma(B)$, $\sigma(N) \subseteq N$, $\sigma(K_{st}) \subseteq K_{st}$, $\sigma(k_{AB}) = k_{\sigma(A)\sigma(B)}$, $\sigma(pubk_A) = pubk_{\sigma(A)}$, and $\sigma(privk_A) = privk_{\sigma(A)}$. Substitutions are extended to terms and actions pointwise. σ is *suitable* for a iff for $m \neq n \in NT(a)$, $\sigma(m) \neq \sigma(n)$. For $\eta = a_1 \cdots a_\ell \in \Sigma^*$, σ is suitable for η iff it is suitable for a_i for all $i \leq \ell$, and $\sigma(\eta) = \sigma(a_1) \cdots \sigma(a_\ell)$. A substitution σ is said to be *suitable for* a protocol Pr if for all $t \in CT(\mathsf{Pr}), \sigma(t) = t$.

Definition 2.4 *A protocol* $\mathsf{Pr} = a_1 b_1 \cdots a_\ell b_\ell$ *is* structured *iff for all substitutions* σ, σ' *suitable for* η, $\sigma(EST(t_i)) \cap \sigma'(EST(t_j)) = \emptyset$ *for* $i \neq j \leq \ell$, $t_i = term(a_i)$ *and* $t_j = term(a_j)$.

The above definition constrains unifiability of encrypted subterms of *different messages*. One could make the definition stronger by constraining the unifiability of different encrypted subterms in the same message. Lemma 3.4 shows that the definition as stated above is adequate for achieving decidability. The stronger definition might lead to better bounds, though. The syntactic condition that we have proposed is of intrinsic interest independent of its impact on decidability issues. It is part of the prudent engineering practices for cryptographic protocols advocated in [AN96].

Given a protocol Pr, $\eta' \in \Sigma^*$ is a *play* of Pr if $\eta' = \sigma(\eta)$ where $\eta \in \mathit{Roles}(\mathsf{Pr})$ and σ is a substitution suitable for Pr and η. $\mathit{Plays}(\mathsf{Pr})$ is the set of all plays of Pr. $\mathit{Events}(\mathsf{Pr}) = \{(\eta, i) \in \mathit{Events} \mid \eta \in \mathit{Plays}(\mathsf{Pr})\}$.

Define a function *infstate* from $\mathcal{S} \times \mathit{Events}(\mathsf{Pr})^*$ to $\mathcal{S}$ by induction as follows:

- $\mathit{infstate}(s_0, \varepsilon) = s_0$.
- If $\mathit{infstate}(s_0, \xi) = s$ and $\xi' = \xi \cdot e$, then $\mathit{infstate}(s_0, \xi') = \mathit{update}(s, \mathit{act}(e))$.

If $\mathit{infstate}(s_0, \xi) = s$, for any $A \in Ag$, $\mathit{infstate}_A(s_0, \xi) = s_A$.

Given a protocol Pr, $s_0 \in \mathcal{S}$ is said to be an *initial information state* of Pr if for all $A \in \mathit{Ho}$, $(s_0)_A = K_A \cup CT(\mathsf{Pr})$ and there exists a subset T of $N \cup K_{st}$ such that $(s_0)_I = K_I \cup CT(\mathsf{Pr}) \cup T$. The set of all initial information states of Pr is denoted by $\mathsf{Init}(\mathsf{Pr})$.

Definition 2.5 *Given a protocol* Pr*, the set of* runs *of* Pr*,* $\mathcal{R}(\mathsf{Pr})$ *is inductively defined as follows:*

- $(s_0, \varepsilon) \in \mathcal{R}(\mathsf{Pr})$ *for every* $s_0 \in \mathsf{Init}(\mathsf{Pr})$.
- *Suppose* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr})$ *and* $\mathit{infstate}(s_0, \xi) = s$. *Suppose there is* (η, i) *such that for all* $1 \leq j < i$, (η, j) *occurs in* ξ, (η, i) *does not occur in* ξ, *and* $\mathit{act}(\eta, i)$ *is enabled at* s. *Then* $(s_0,\ \xi \cdot (\eta, i)) \in \mathcal{R}(\mathsf{Pr})$.

Note that the set of runs of a protocol is typically infinite. Thus we are in the domain of infinite state systems and typically the reachability problem for such systems is undecidable.

Definition 2.6 *Given a protocol* Pr *and* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr})$, $\mathit{secrets}(s_0, \xi)$ *is defined to be the set of basic terms* m *such that for some prefix* ξ' *of* ξ *and some* $A \in \mathit{Ho}$, *letting* $\mathit{infstate}(s_0, \xi') = s$, m *belongs to* $\mathsf{analz}(s_A) \setminus \mathsf{analz}(s_I)$. (s_0, ξ) *is* leaky *iff* $\mathit{secrets}(s_0, \xi) \cap \mathsf{analz}(\mathit{infstate}_I(s_0, \xi)) \neq \emptyset$. Pr preserves secrecy *iff for all runs* (s_0, ξ) *of* Pr*,* (s_0, ξ) *is non-leaky.*

Note that the following property is true: For all $T \subseteq \mathcal{T}$, and $t, t' \in \mathcal{T}$, $\overline{\overline{T \cup \{t\}} \ \cup \ \{t'\}} = \overline{T \cup \{t, t'\}}$. This immediately implies that for any state s and actions a, a', $\mathit{update}(\mathit{update}(s, a), a') = \mathit{update}(\mathit{update}(s, a'), a)$. Thus for any finite set of actions $\Sigma' = \{a_1, \ldots, a_\ell\}$ and a state s, it makes sense to define $\mathit{update}(s, \Sigma')$ to be $\mathit{update}(s, a_1 \cdots a_\ell)$.

Let $e \in \mathit{Events}, E \subseteq \mathit{Events}$ and $s \in \mathcal{S}$. We say that e is enabled at (s, E) iff:

- for some $\eta \in \Sigma^*$ and $i \leq |\eta|$, $e = (\eta, i) \notin E$, for all $j < i$, $(\eta, j) \in E$, and
- letting $\Sigma' = \{\mathit{act}(e') \mid e' \in E\}$ and $a = \mathit{act}(e)$, a is enabled at $\mathit{update}(s, \Sigma')$.

Definition 2.7 *Suppose* $(s, \xi) \in \mathcal{S} \times \mathit{Events}^*$ *with* $\xi = e_1 \cdots e_\ell$. *We say that* $G = (E, \rightarrow)$ *is* a minimal causal graph (MCG) of (s, ξ) *iff:*

- $E = \{e_1, \ldots, e_\ell\}$,
- $\rightarrow \subseteq E \times E$ *such that for all* $i, j \leq \ell$, *if* $e_i \rightarrow e_j$ *then* $i < j$, *and*
- *for all* $e \in E$, ${}^\bullet e = \{e' \in E \mid e' \rightarrow e\}$ *is* a minimal set *such that* e *is enabled at* $(s, {}^\bullet e)$.

Note that the notion of MCG is similar to that of *bundles* in the strand space context ([FHG99]). There are outward differences due to the fact that some of the steps in the construction of a message by the intruder is explicitly represented in a bundle.

Proposition 2.8 *Suppose* Pr *is a protocol and* $(s, \xi) \in \mathcal{R}(\mathsf{Pr})$. *Then there is at least one MCG of* (s, ξ).

Definition 2.9 *Suppose* $\mathsf{Pr} = a_1 b_1 \cdots a_\ell b_\ell$ *is a protocol,* $(s, \xi) \in \mathcal{R}(\mathsf{Pr})$ *and* $(E, \rightarrow)$ *is an MCG of* (s, ξ). *We say that an edge* (e, e') *is* in order *iff: either* $(\exists i \leq \ell$, $act(e)$ *is an instance of* a_i *and* $act(e')$ *is an instance of* $b_i)$, *or* $(\exists i < j \leq \ell$, $act(e)$ *is an instance of* a_i *or* b_i *and* $act(e')$ *is an instance of* a_j *or* $b_j)$. *An edge is said to be* out of order *if it is not in order.*

Proposition 2.10 *Suppose* $\mathsf{Pr} = a_1 b_1 \cdots a_\ell b_\ell$ *is a protocol,* $(s, \xi) \in \mathcal{R}(\mathsf{Pr})$ *and* $G = (E, \rightarrow)$ *is an MCG of* (s, ξ).

1. *If* (e, e') *is out of order for some* $e, e' \in E$, *then* $act(e)$ *is a send and* $act(e')$ *is a receive.*
2. *If for some* $e = (\eta, i)$ *every edge* (e, e') *is out of order, then for all* $e' \in E$, e' *is not of the form* (η, j) *with* $j > i$.
3. *Suppose* $e_1, \ldots, e_k$ *is a sequence of events from* E *such that for all* $i < k$, $e_i \rightarrow e_{i+1}$ *and* (e_i, e_{i+1}) *is in order. Then* $k \leq 2 \cdot \ell$.

Definition 2.11 *Suppose* Pr *is a protocol,* (s_0, ξ) *is a run of* Pr *and* $G = (E, \rightarrow)$ *is an MCG of* (s_0, ξ). *For any* $E' \subseteq E$, $max(E')$ *denotes the set* $\{e \in E' \mid \text{for all } e' \in E' : \neg(e \rightarrow e')\}$ *and* $terms(E')$ *denotes the set* $\{term(a) \mid a \in act(E')\}$.

Proposition 2.12 1. *Whenever there is a* synth*-proof of* $T \vdash t$ *and* $T \subseteq T'$, *there is also a* synth*-proof of* $T' \vdash t$. *Similarly for* analz*-proofs.*

2. *If* π *is a* synth*-proof of* $T \vdash t$, *then for any sequent* $T \vdash t'$ *labelling a node of* π, *either* $t' \in ST(t)$ *or* t' *is a key which encrypts a subterm of* t.
3. *In any* synth*-proof of* $T \vdash t$ *where* $|t| \leq B$, *there are at most* B *occurrences of axioms.*
4. *If* $T \vdash t$ *labels the root of an* analz*-proof whose leftmost axiom is labelled by* $T \vdash t'$, *then* $t \in ST(t')$.

3 Decidability

In this section we prove the main result of the paper.

Theorem 3.1 *The secrecy problem for the class of structured protocols is decidable.*

Proof. Suppose $\mathsf{Pr} = c_1 d_1 \cdots c_\ell d_\ell$ is a given structured protocol. If Pr does not preserve secrecy then it has a leaky run. Let $B = max_{i \leq \ell}\{|term(c_i)|\}$. It follows from Lemma 3.2 that Pr has a leaky run of length at most $B' = (B+1)^{2 \cdot \ell}$. Thus it suffices to check for the existence of a leaky run of length at most B'. The set of runs of length at most B' is a finite set which can be effectively constructed. Of course, it is also effectively checkable whether a given run is leaky or not. Thus the decidability of secrecy problem for structured protocols is proved, assuming Lemma 3.2.

For the rest of the discussion we fix a structured protocol $\mathsf{Pr} = c_1 d_1 \cdots c_\ell d_\ell$ which *does not preserve secrecy.* Let $B = max_{i \leq \ell}\{|term(c_i)|\}$. We fix a leaky run (s_0, ξ) of Pr of minimum length. We also suppose that $\xi = e_1 \cdots e_k$. We fix an MCG $(E, \rightarrow)$ of (s_0, ξ). We introduce the following notation for some of the substrings of ξ. For any $j : 1 \leq j \leq k$, ξ_j denotes $e_1 \cdots e_j$, s_j denotes $infstate(s_0, \xi_j)$ and T_j denotes $(s_j)_I$. For $i, j : 1 \leq i < j \leq k$, ξ_j^{-i} denotes $e_1 \cdots e_{i-1} e_{i+1} \cdots e_j$, s_j^{-i} denotes $infstate(s_0, \xi_j^{-i})$ and T_j^{-i} denotes $(s_j^{-i})_I$. We also define T_0 to be $(s_0)_I$. Further we let X denote $secrets(s_0, \xi)$ and X' denote $secrets(s_0, \xi_{k-1})$. It is clear that $X \cap \mathsf{analz}(T_k) \neq \emptyset$ while $X' \cap \mathsf{analz}(T_{k-1}) = \emptyset$, since (s_0, ξ) is a minimal leaky run. For all $i \leq k$, we let $a_i = act(e_i)$ and $t_i = term(a_i)$.

Lemma 3.2 $|\xi| \leq (B+1)^{2 \cdot \ell}$.

Proof. Suppose that $|\xi| > (B+1)^{2 \cdot \ell}$. Now consider the set E' of all $e \in E$ such that there is a path (of length ≥ 0) from e to e_k using only edges which are in order. We will show below (Lemma 3.3) that there are at most $B+1$ edges into any event of E. From this and Proposition 2.10 it follows that $|E'| \leq (B+1)^{2 \cdot \ell}$. Therefore there is some $e \in E \setminus E'$. Let e_r be a maximal such element. It is easy to see that $e_r \neq e_k$ and all edges out of e_r are out of order (this includes the case when there are no edges out of e_r). Let E'' be the set $\{e \in E \mid e_r \rightarrow e\}$. Let $X = T_0 \cap (\mathsf{analz}(T_r) \setminus \mathsf{analz}(T_{r-1}))$ and let s_0' be a state such that $(s_0')_A = (s_0)_A$ for all $A \in Ho$ and $(s_0')_I = (s_0)_I \cup X$. For every $e_u \in E''$, it follows from Lemma 3.4 that $t_u \in \overline{T_{u-1}^{-r} \cup X}$. Further, letting $e_r = (\eta, i)$, it follows from Proposition 2.10 that there is no event in E of the form (η, j) with $j > i$. Also for all $m \in X$, m is generated by a_r and hence the enabledness of e_q at (s_0', ξ_{q-1}) is not affected, for $q < r$.

From this it follows that (s_0', ξ_k^{-r}) is also a run of Pr. By mimicking the argument given in the beginning of Lemma 3.4, it is easy to see that $\mathsf{analz}(T_k) \cap T_0 \subseteq \mathsf{analz}(T_k^{-r} \cup X)$. Further since $X \cap secrets(s_0, \xi) = \emptyset$, $secrets(s_0, \xi) = secrets(s_0', \xi_k^{-r})$. Thus (s_0', ξ_k^{-r}) is leaky as well, contradicting the fact that (s_0, ξ) is a leaky run of Pr of minimum length. This shows that our assumption that $|\xi| > (B+1)^{2 \cdot \ell}$ is wrong. Thus the lemma is proved, assuming Lemma 3.3 and Lemma 3.4.

Lemma 3.3 *For all* $e \in E$, $|max({}^{\bullet}e)| \leq B + 1$.

Proof. Fix $e \in E$. Let $m = max\{i \leq k \mid e_i \in {}^{\bullet}e\}$. By Proposition 2.12 any synth-proof of a term of size at most B has at most B axiom occurrences. Clearly every one of these axioms are of the form $\mathsf{analz}(T_m) \vdash t$. We prove below that whenever $t \in \mathsf{analz}(T_m)$ then it is also the case that $t \in \mathsf{analz}(T_0 \cup terms(E'))$ where

$E' \subseteq \{e_1, \ldots, e_m\}$ with $|max(E')| = 1$. Then it is clear that $|max(^{\bullet}e)| \leq B+1$ (the bound is $B+1$ rather than B because if $e = (\eta, i)$ with $i > 0$ then $(\eta, i-1)$ might also be in $max(^{\bullet}e)$).

We introduce a bit of notation for what follows: Say that for $E_1, E_2 \subseteq E$, $E_2 \prec E_1$ if one of the following two conditions hold:

1. $E_2 \subsetneq E_1$
2. $E_2 \not\subseteq E_1$ and for all $e_i \in max(E_2) \setminus max(E_1)$, there is an $e_j \in max(E_1) \setminus max(E_2)$ such that $i < j$.

It is easy to see that $\prec$ is a strict partial order and that whenever $E_3 \prec E_2$ and $E_2 \prec E_1$, it is also the case that $E_3 \cup E_2 \prec E_1$.

We now prove that for any $t \in \mathsf{analz}(T_m)$ there is some $E' \subseteq \{e_1, \ldots, e_m\}$ with $|max(E')| = 1$ such that $t \in \mathsf{analz}(T_0 \cup terms(E'))$. It suffices to prove that whenever $t \in \mathsf{analz}(T_0 \cup terms(E_1))$ for $E_1 \subseteq \{e_1, \ldots, e_m\}$ with $|max(E_1)| > 1$, then it is also the case that $t \in \mathsf{analz}(T_0 \cup terms(E_2))$ where $E_2 \prec E_1$.

Suppose $e_r, e_s \in max(E_1)$. Let π be an analz-proof of $T_0 \cup terms(E_1) \vdash t$. Clearly, at most one of $T_0 \cup terms(E_1) \vdash t_s$ and $T_0 \cup terms(E_1) \vdash t_r$ can be the leftmost axiom of π. Suppose $T_0 \cup terms(E_1) \vdash t_r$ is not the leftmost axiom of π.

Assume that for some subproof π' of π with root labelled $T_0 \cup terms(E_1) \vdash t'$, for all proper subproofs π'' of π' with root labelled $T_0 \cup terms(E_1) \vdash t''$, if $T_0 \cup terms(E_1) \vdash t_r$ is not the leftmost axiom of π'', then there is a proof of $T_0 \cup terms(E_2) \vdash t''$ for some $E_2 \prec E_1$. The only nontrivial case in the induction step is when π' is of the following form:

$$\dfrac{\begin{matrix}(\pi'_1) \\ \vdots \\ T_0 \cup terms(E_1) \vdash \{t'\}_k\end{matrix} \qquad \begin{matrix}(\pi'_2) \\ \vdots \\ T_0 \cup terms(E_1) \vdash \overline{k}\end{matrix}}{T_0 \cup terms(E_1) \vdash t'}\ \mathsf{decrypt}$$

Suppose the leftmost axiom of π' is not $T_0 \cup terms(E_1) \vdash t_r$. Then the same holds for π'_1 as well and hence there is an analz-proof of $T_0 \cup terms(E_2) \vdash \{t'\}_k$ for some $E_2 \prec E_1$. Suppose now that the leftmost axiom of π'_2 is $T \vdash t_r$. Then $\overline{k} \in ST(t_r)$. Assume that $T \vdash t_u$ is the leftmost axiom in π'_1. Then k is used in t_u, and since $\neg(e_r \rightarrow e_u)$, it is not the case that $\overline{k}$ is generated by a_r. It must be the case that it is generated by a_v for some $v < r$. Since $\overline{k} \in \mathsf{analz}(T_m)$ it must be the case that $\overline{k} \notin secrets(s_0, \xi_m)$. Which means that $\overline{k} \in \mathsf{analz}(T_v)$, which means that there is a proof π'' of $T_v \vdash \overline{k}$. In other words π'' can be viewed as a proof of $T_0 \cup terms(E_3) \vdash \overline{k}$ for $E_3 = \{e_1, \ldots, e_v\}$. Replacing π'_2 by π'' in π' leads us to the fact that $T_0 \cup terms(E_2) \cup terms(E_3) \vdash t'$. Now, if E_2 contains an event $e_{v'}$ with $v' > v$ then $E_3 \prec E_2$ and hence $E_2 \cup E_3 \prec E_1$. Otherwise the maximum index of an event occurring in E_2 is $v' \leq v$, but then it is clear that $E_2 \cup E_3 \prec E_1$ since $r > v$ and e_r occurs in E_1. This completes the induction step and the proof.

The following lemma crucially uses the fact the the given protocol Pr is structured.

Lemma 3.4 *Suppose* $e_i, e_j \in E$ *such that* $e_j \rightarrow e_i$ *and* (e_j, e_i) *is out of order. Let* $X = T_0 \cap (\mathsf{analz}(T_j) \setminus \mathsf{analz}(T_{j-1}))$. *Then* $t_i \in \overline{T_{i-1}^{-j} \cup X}$.

Proof. We first claim that $\mathsf{analz}(T_{i-1}) \cap T_0 \subseteq \mathsf{analz}(T_{i-1}^{-j} \cup X)$. Suppose $r \in \mathsf{analz}(T_{i-1}) \cap T_0$. If $r \in \mathsf{analz}(T_{j-1})$ we are through. If $r \in ST(t_j) \setminus \mathsf{analz}(T_{j-1})$ then since ξ_j is nonleaky and $r \in \mathsf{analz}(T_{i-1})$, $r \in \mathsf{analz}(T_j)$ and hence $r \in X$. If $r \notin ST(t_j)$ then observe that for any $t \in \mathsf{analz}(T_{i-1}) \setminus ST(t_j)$, a simple induction on analz-proofs using the above facts shows that $t \in \mathsf{analz}(T_{i-1}^{-j} \cup X)$, and hence $r \in \mathsf{analz}(T_{i-1}^{-j} \cup X)$.

We now proceed to prove that $t \in \overline{T_{i-1}^{-j} \cup X}$. Since (e_j, e_i) is out of order, and since a_j is a send and a_i is a receive, it is the case that $\exists \ell_1 < \ell_2 \leq \ell$ such that a_j is an instance of c_{ℓ_2} and a_i is an instance of d_{ℓ_1}. Now since Pr is structured, $EST(t_i) \cap EST(t_j) = \emptyset$.

We now prove that for all $t \in ST(t_i)$: if $t \in \overline{T_{i-1}}$ then $t \in \overline{T_{i-1}^{-j} \cup X}$. Let π be a synth-proof of $\mathsf{analz}(T_{i-1}) \vdash t$. We prove by induction that for all subproofs π' of π whose root is labelled with $\mathsf{analz}(T_{i-1}) \vdash t'$, $t' \in \overline{T_{i-1}^{-j} \cup X}$.

Assume that for some subproof π' of π with root labelled $\mathsf{analz}(T_{i-1}) \vdash t'$, for all proper subproofs of π'' of π' with root $\mathsf{analz}(T_{i-1}) \vdash t''$, $t'' \in \overline{T_{i-1}^{-j} \cup X}$. The nontrivial case is when π' is a one-node proof of the following form:

$$\frac{}{\mathsf{analz}(T_{i-1}) \vdash t'}\ \mathsf{Ax}_s$$

Then it is clear that $t' \in \mathsf{analz}(T_{i-1}) = \mathsf{analz}(T_{i-1}^{-j} \cup \{t_j\})$. There are three subcases now:

- $t' \in ST(t_j) \cap ST(t)$: Since $EST(t_i) \cap EST(t_j) = \emptyset$, t' does not contain any encrypted subterm. Therefore $t' \in \mathsf{synth}(ST(t') \cap T_0)$. Also $ST(t') \subseteq \mathsf{analz}(T_{i-1})$. Therefore $t' \in \mathsf{synth}(\mathsf{analz}(T_{i-1}) \cap T_0) \subseteq \mathsf{synth}(\mathsf{analz}(T_{i-1}^{-j}) \cup X) \subseteq \overline{T_{i-1}^{-j} \cup X}$.
- $t' \notin ST(t_j)$: Now it follows that there is $t'' \in T_{i-1}^{-j}$ such that $t' \in \mathsf{analz}(\{t''\} \cup (\mathsf{analz}(T_{i-1}) \cap T_0))$. But this means that $t' \in \mathsf{analz}(\{t''\} \cup (\mathsf{analz}(T_{i-1}^{-j} \cup X)) \subseteq \mathsf{analz}(T_{i-1}^{-j} \cup X) \subseteq \overline{T_{i-1}^{-j} \cup X}$.
- $t' \notin ST(t)$: Since $t' \notin ST(t)$ but still occurs in a minimal synth-proof of t, t' is a key which encrypts some subterm of t. Thus $t' \in \mathsf{analz}(T_{i-1}) \cap T_0 \subseteq \mathsf{analz}(T_{i-1}^{-j} \cup X) \subseteq \overline{T_{i-1}^{-j} \cup X}$.

This suffices to prove the lemma as the other cases in the induction step are trivial.

4 Discussion

As mentioned earlier, relaxing the syntactic restriction on protocols allows us to code the halting problem for two-counter machines as a secrecy problem. The idea in the coding is to represent transitions of two-counter machines as roles of the protocol. The terms used in the protocol represent configurations of the two-counter machine, which are of the form (q, m, n) for some natural numbers m and n. The roles of the protocol look like the following:
$A?B\colon \{q, y, x\}_{k_{AB}}, \{x', x\}_{k_{AB}};\quad A!B\colon (y')\ \{q', y', x'\}_{k_{AB}}, \{y, y'\}_{k_{AB}}.$

Note that the syntax restriction is not respected by this protocol as distinct communications have encrypted subterms – $\{q, y, x\}_{k_{AB}}$ and $\{q', y', x'\}_{k_{AB}}$, for instance – which are unifiable. The ability to generate new nonces allows us to code the natural numbers, and the unifiability of encrypted terms allows us to code the behavior of the machines which use the output configuration of one transition as the input configuration of another. This is the key to undecidability.

Unlike the above protocol, which is designed to code up a machine, most standard protocols in the literature – for instance many of the protocols presented in [CJ97] – which aim to communicate secrets in a well-designed way, can be transformed easily to respect the proposed syntax restriction by simply introducing message numbers in all the encrypted components. The exception to this are protocols like the Yahalom protocol as given in [CJ97], where some agents forward message components which cannot be decrypted by them. While these protocols cannot be made to conform to the proposed syntactic restrictions by simple transformations as given above, there are nevertheless more sophisticated transformations which can handle such protocols. See [Low98] for a discussion.

Secrecy is studied in this paper only as a representative problem in the verification of security protocols. In fact, we can extend the decidability result in this paper to the verification problem of a simple modal logic in which one can state other versions of secrecy and authentication as well.

References

[AN96] Abadi, M. and Needham, R. "Prudent engineering practices for cryptographic protocols", *IEEE Transactions on Software Engineering*, vol 22, 6-15, 1996.

[CJ97] Clark, J. and Jacob, J., "A survey of authentication protocol literature", Web draft version 1.0, `http://www.cs.york.ac.uk./~jac`, 1997.

[CS02] Comon, H. and Shmatikov, V., "Is it possible to decide whether a cryptographic protocol is secure or not?", *Journal of Telecommunications and Information Technology*, 2002.

[DLMS99] Durgin, N.A., Lincoln, P.D., Mitchell, J.C. and Scedrov, A., "The undecidability of bounded security protocols", *Workshop on Formal Methods and Security Protocols (FMSP'99).* Electronic proceedings available at: `http://www.cs.bell-labs.com/who/nch/fmsp99/program.html`.

[FHG99] Fábrega, J., Herzog, J. and Guttman, J., "Strand Spaces: Proving Security Protocols Correct", *Journal of Computer Security*, vol. 7, 191-230, 1999.

[HT96] Heintze, N. and Tygar, J. D., "A model for secure protocols and their composition", *IEEE Trans. on Soft. Engg*, vol 22, 16-30, 1996.

[Low98] Lowe, G., "Towards a completeness result for model checking of security protocols", Technical Report 1998/6, Department of Mathematics and Computer Science, University of Leicester, 1998.

[MS01] Millen, J. and Shmatikov, V., "Constraint Solving for Bounded-Process Cryptographic Protocol Analysis", *CCS'01*, 2001, 166-175.

[Pau98] Paulson, L., "The inductive approach to verifying cryptographic protocols", *Journal of computer security*, vol 6, 1998, 85-128.

[RS03] Ramanujam, R. and Suresh, S.P., "An equivalence on terms for security protocols", To be published in *Proceedings of AVIS'03*, April 2003.