Undecidability of the secrecy problem for 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

In this report, we present uniform proofs for two important undecidability results on security protocols, namely the undecidability of the secrecy problem for security protocols both under the restriction of *bounded number of nonces* and under the restriction of *bounded message length*. These results have been proved in ([HT96], [ALV01]) and in ([CDL+99], [DLMS99]) respectively.

We present both the undecidability results by reducing the reachability problem for two-counter machines to the secrecy problem. While our proof of the undecidability of the secrecy problem when restricted to boundedly many nonces does not differ essentially from those in [HT96] or [ALV01], our proof of the undecidability of the secrecy problem when restricted to bounded message length differs substantially from that in [DLMS99] and [CDL+99]. We highlight the important differences below:

- In [DLMS99] the multiset rewriting (MSR) formalism is used to prove the result. The presence of semantically complex features like existential quantification, etc. in the formalism makes it hard to comprehend the essence of the undecidability result. Our proof is an attempt at clearly highlighting the sources of undecidability.

- In [DLMS99], the reduction is essentially from the halting problem for Turing machines. Our proof uses a reduction from the reachability problem for two-counter machines. Technically, this makes our reduction simpler. Essentially, two-counter machines are two-tape Turing machines which operate on a unary alphabet. Hence keeping track of a two-counter machine's configurations is easier, since we can ignore the contents of the cells.

1 Security protocol modelling

In this section, we set up the basic model of security protocols in which our results are proved.

Basic terms

We assume a finite set of *agents* Ag with a special *intruder* $I \in Ag$. The set of *honest agents*, denoted Ho, is defined to be $Ag \setminus \{I\}$. We assume that the set of *keys* K is given by $K_{lt} \cup K_{st}$ 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 $k \in K$ we assume $\overline{k} \in K$ defined 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 $A \in Ag$, $K_A \stackrel{\text{def}}{=} \{pubk_B, k_{AB} \mid B \neq A\} \cup \{privk_A, pubk_A\}$ is the set of keys known to A by default. We also assume an infinite set of *nonces* N. The set of *basic terms* $\mathcal{T}_0$ is defined to be $K \cup N \cup Ag$.

Terms

The set of *information terms* is defined to be

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

where m ranges over $\mathcal{T}_0$ and k ranges over K.

The set of subterms of a given term t, denoted $ST(t)$, is defined to be the least set $T \subseteq \mathcal{T}$ 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 $t \in T$.

For any $T \subseteq \mathcal{T}$, $ST(T)$ is defined to be $\bigcup\limits_{t \in T} ST(t)$.

For a term t and a key k, k *encrypts in* t iff $\exists t' : \{t'\}_k \in ST(t)$. For any two terms t and t', t' is *referred to* in t iff either t' is a subterm of t or t' encrypts in t. For a set of terms T and a term t', t' is *referred to* in T if t' is referred to in t for some $t \in T$. $EST(t)$, the set of *encrypted subterms* of $t \in \mathcal{T}$ is the set $\{t' \in ST(t) \mid \exists t'', k : t' = \{t''\}_k\}$. For $t \in \mathcal{T}$, $|t|$, the *size* of t is defined inductively as follows: $|m| \stackrel{\text{def}}{=} 1$ for $m \in \mathcal{T}_0$; $|(t, t')| \stackrel{\text{def}}{=} |t| + |t'|$; and $|\{t\}_k| \stackrel{\text{def}}{=} |t| + |k|$.

In our modelling of protocols we assume that long-term keys are never passed from one agent to another as part of messages. Hence we want to formally characterise when a term can be part of a message used in a protocol. We say that a term t is *well-formed* if and only if $ST(t) \cap K_{lt} = \emptyset$.

Actions

An *action* is either a *send action* of the form $A!B{:}(M)t$ or a *receive action* of the form $A?B{:}t$ where:

- $A \in Ho, B \in Ag$ and $A \neq B$,
- $t \in \mathcal{T}$ is well-formed, and

- for all $m \in M$, $m \in N \cup K_{st}$ and m is referred to in t.

For simplicity of notation, we write $A!B{:}t$ instead of $A!B{:}(\emptyset)\ t$. The set of all actions is denoted by Ac, the set of all send actions is denoted by $Send$, and the set of all receive actions is denoted by Rec.

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

For any action a, $|a|$ is defined to be $|term(a)|$. For any send action $A!B{:}(M)t$ with $B \in Ho$, we say that $B?A{:}t$ is said to be its *matching receive.*

For a of the form $A!B{:}(M)t$, we define $term(a)$ to be t and $NT(a)$ to be M. Similarly for a of the form $A?B{:}t$, we define $term(a)$ to be t and $NT(a)$ to be $\emptyset$. For $\eta = a_1 \cdots a_\ell \in Ac^*$, $terms(\eta)$ is defined to be $\{term(a_i) \mid 1 \leq i \leq \ell\}$ and $NT(\eta)$ is defined to be $NT(a_1) \cup \cdots \cup NT(a_\ell)$. For a sequence $\eta = a_1 \cdots a_\ell \in Ac^*$, $Actions(\eta)$ is the set $\{a_1, \ldots, a_\ell\}$. For a subset R of Ac^*, $terms(\mathsf{R})$ is defined to be $\bigcup\limits_{\eta \in \mathsf{R}} terms(\eta)$ and $NT(\mathsf{R})$ is defined to be $\bigcup\limits_{\eta \in \mathsf{R}} NT(\eta)$.

For notational ease, we overload notation and define $ST(a)$ and $EST(a)$ for any action a to be $ST(term(a))$ and $EST(term(a))$, respectively.

Ac_A, the set of A-*actions* is given by $\{C!D{:}(M)t, C?D{:}t \in Ac \mid C = A\}$. For any $\eta = a_1 \cdots a_\ell \in Ac^*$ and $A \in Ag$, $\eta \upharpoonright A$, A's view of η, is defined to be the subsequence $a_{i_1} \cdots a_{i_r}$ of η, where $\{i_1, \ldots, i_r\} = \{i \leq \ell \mid a_i \in Ac_A\}$.

Example 1.1 For example, let η be the following sequence:

$$
\begin{array}{llllll}
1. & A & ! & B & : & (x) & \{A, x\}_{pubk_B} \\
2. & B & ? & A & : & & \{A, x\}_{pubk_B} \\
3. & B & ! & A & : & (y) & \{x, y\}_{pubk_A} \\
4. & A & ? & B & : & & \{x, y\}_{pubk_A} \\
5. & A & ! & B & : & & \{y\}_{pubk_B} \\
6. & B & ? & A & : & & \{y\}_{pubk_B}
\end{array}
$$

Then $\eta \upharpoonright A$ is the sequence η_1 given below:

$$
\begin{array}{lllllll}
1. & A & ! & B & : & (x) & \{A, x\}_{pubk_B} \\
2. & A & ? & B & : & & \{x, y\}_{pubk_A} \\
3. & A & ! & B & : & & \{y\}_{pubk_B}
\end{array}
$$

Similarly $\eta \upharpoonright B$ is the sequence η_2 given below:

$$
\begin{array}{lllllll}
1. & B & ? & A & : & & \{A, x\}_{pubk_B} \\
2. & B & ! & A & : & (y) & \{x, y\}_{pubk_A} \\
3. & B & ? & A & : & & \{y\}_{pubk_B}
\end{array}
$$

■

Protocol specifications

Definition 1.2 *A* **protocol** *is a pair* $\mathsf{Pr} = (\mathsf{C}, \mathsf{R})$ *where*

- C*, the set of* constants *of* Pr*, denoted* $CT(\mathsf{Pr})$*, is a subset of* $\mathcal{T}_0$*,*

- R*, the set of* roles *of* Pr*, denoted* $Roles(\mathsf{Pr})$*, is a finite nonempty subset of* Ac^+ *such that for all* $\eta \in \mathsf{R}$*, there is some* $A \in Ho$ *such that* $\eta \in Ac_A^+$*, and*

- $\mathsf{C} \cap NT(\mathsf{R}) = \emptyset$.

Example 1.3 A version of the Needham-Schroeder protocol ([NS78]) is presented in this example. The protocol $\mathsf{Pr}_{\mathsf{NS}}$ is given by (C, R) where $\mathsf{C} = \emptyset$, and $\mathsf{R} = \{\eta_1, \eta_2\}$ where η_1 and η_2 are as given in Example 1.1.

The protocol has two roles: we call η_1 the initiator role and η_2 the responder role. A sends the new nonce x to B as a challenge to prove his (B's) identity. She then receives a response to it as also a challenge from B in the form of a nonce y. She finally responds to B's challenge by sending back y. Since only B can decrypt the contents of the first message, A is at least convinced that B is alive. Similarly, B first receives a challenge from A and responds to it while issuing his own challenge. He finally receives the response to his challenge. Since only A could have decrypted the contents of the message sent by B, B is at least convinced that she is alive. ∎

Example 1.4 Here is another example of a protocol, We call this $\mathsf{Pr}_{\mathsf{ut}}$ (ut stands for "untitled"). It is given by (C, R) where:

- $\mathsf{C} = \emptyset$,

- $\mathsf{R} = \{\zeta_1, \zeta_2\}$ where ζ_1 is the following sequence:

$$\begin{array}{llllll} 1. & A & ! & B & : & (x) \quad \{A, \{x\}_{pubk_B}\}_{pubk_B} \\ 2. & A & ? & B & : & \qquad\ \{x\}_{pubk_A} \end{array}$$

and ζ_2 is the following sequence:

$$\begin{array}{llllll} 1. & B & ? & A & : & \{A, \{x\}_{pubk_B}\}_{pubk_B} \\ 2. & B & ! & A & : & \{x\}_{pubk_A} \end{array}$$

Here again we can call the role ζ_1 the initiator role and the role ζ_2 the responder role. The initiator issues a challenge, response to which will ensure her at least of the responder's being alive in the network. The responder plays the passive role of just responding to challenges. ∎

Events

An *event* is a pair (η, i) where $\eta \in Ac^+$ and $1 \leq i \leq |\eta|$. The set of all events is named *Events*. For $e = (a_1 \cdots a_\ell, i) \in \mathit{Events}$, $act(e) \stackrel{\text{def}}{=} a_i$. For $\eta \in Ac^+$, $\mathit{Events}(\eta) \stackrel{\text{def}}{=} \{(\eta, i) \mid 1 \leq i \leq |\eta|\}$. For any $\xi = e_1 \cdots e_\ell \in \mathit{Events}^*$, $\mathit{Events}(\xi) \stackrel{\text{def}}{=} \{e_1, \ldots, e_\ell\}$. For $E \subseteq \mathit{Events}$, $\mathit{Actions}(E) \stackrel{\text{def}}{=} \{act(e) \mid e \in E\}$ and for $\xi \in \mathit{Events}^*$, $\mathit{Actions}(\xi) \stackrel{\text{def}}{=} \mathit{Actions}(\mathit{Events}(\xi))$.

For notational ease, we overload notation and define $NT(e)$, $term(e)$, $ST(e)$, and $EST(e)$ for any event e to be $NT(act(e))$, $term(act(e))$, $ST(act(e))$, and $EST(act(e))$, respectively.

Definition 1.5 *$\rightarrow_\ell \subseteq \mathit{Events} \times \mathit{Events}$, the* local dependency relation *between events, is defined as follows: $e \rightarrow_\ell e'$ iff there exists $\eta \in Ac^+$ and $i < |\eta|$ such that $e = (\eta, i)$ and $e' = (\eta, i+1)$.*

For any $e \in \mathit{Events}$, $LP(e)$, the local past *of e is defined to be the set $\{e' \in \mathit{Events} \mid e' \stackrel{+}{\rightarrow}_\ell e\}$. e is a* local maximum *if there is no $e' \in \mathit{Events}$ such that $e \rightarrow_\ell e'$.*

Substitutions, plays and events of a protocol

A *substitution* σ is a map from $\mathcal{T}_0$ to $\mathcal{T}$ such that:

- $\sigma(Ag) \subseteq Ag$, $\sigma(I) = I$, and for all $A, B \in Ho$, if $A \neq B$ then $\sigma(A) \neq \sigma(B)$,
- $\sigma(N) \subseteq \mathcal{T}$,
- $\sigma(K_{st}) \subseteq K_{st}$, and
- $\sigma(k_{AB}) = k_{\sigma(A)\sigma(B)}$, $\sigma(pubk_A) = pubk_{\sigma(A)}$, and $\sigma(privk_A) = privk_{\sigma(A)}$.

An important point to note about substitutions is that nonces can be substituted with arbitrary terms. Thus our formal model allows the possibility of some kinds of type-flaw attacks to be carried out by the intruder. But this also adds another source of unboundedness in the model (apart from the infinite set of nonces), since when a protocol is run, there is no a priori bound on the size of the messages occurring in the run. Such a substitution with arbitrary terms can be done only by the intruder, since the honest agents are assumed to follow the protocol. This is captured formally in our model by the definitions of a substitution being suitable for an action, a sequence of actions, for a protocol, etc.

σ is extended to terms, actions, sequences of actions and events as follows:

- $\sigma((t, t')) = (\sigma(t), \sigma(t'))$.
- $\sigma(\{t\}_k) = \{\sigma(t)\}_{\sigma(k)}$.
- $\sigma(A!B{:}(M)t) = \sigma(A)!\sigma(B){:}(\sigma(M))\sigma(t)$.
- $\sigma(A?B{:}t) = \sigma(A)?\sigma(B){:}\sigma(t)$.
- $\sigma(a_1 \cdots a_\ell) = \sigma(a_1) \cdots \sigma(a_\ell)$.
- $\sigma((\eta, i)) = (\sigma(\eta), i)$.

A substitution σ is said to be *suitable* for a iff:

- if $a \in Ac_A$ then $\sigma(a) \in Ho$,
- for all $n \in NT(a) \cap N$, $\sigma(n) \in N$,
- for $m \in NT(a)$ and $n \in ST(term(a)) \setminus NT(a)$, $\sigma(m) \neq \sigma(n)$ and
- for all $m \neq n \in NT(a)$, $\sigma(m) \neq \sigma(n)$.

σ is suitable for $a_1 \cdots a_\ell$ iff for all $i \leq \ell$, σ is suitable for a_i. σ is said to be *suitable for* a protocol Pr if $\sigma(t) = t$ for all constants $t \in CT(\mathsf{Pr})$.

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

A substitution σ is said to be *well-typed* iff for $n \in N$, $\sigma(n) \in N$. A play η' of Pr is said to be a *well-typed play* iff $\eta' = \sigma(\eta)$ for some well-typed substitution σ and $\eta \in \mathit{Roles}(\mathsf{Pr})$. Similarly an event $e \in \mathit{Events}(\mathsf{Pr})$ is a *well-typed event* iff $e = (\eta, i)$ where η is a well-typed play of Pr.

Example 1.6

- Here are two substitutions suitable for $\mathsf{Pr}_{\mathsf{NS}}$, which was presented in Example 1.3:
  - σ_1 given by: $\sigma_1(x) = m$, $\sigma_1(y) = n$, $\sigma_1(B) = I$ and σ_1 is identity on all other terms in $\mathcal{T}_0 \setminus K_{lt}$.
  - σ_2 given by: $\sigma_2(x) = m$, $\sigma_2(y) = n$ and σ_2 is identity on all other terms in $\mathcal{T}_0 \setminus K_{lt}$.

  Examples of plays of $\mathsf{Pr}_{\mathsf{NS}}$ include $\eta'_1 \stackrel{\text{def}}{=} \sigma_1(\eta_1)$ and $\eta'_2 \stackrel{\text{def}}{=} \sigma_2(\eta_2)$.
- Here are three substitutions suitable for the protocol presented in Example 1.4.
  - ς_1 given by: $\varsigma_1(x) = m$ and ς_1 is identity on all other terms in $\mathcal{T}_0 \setminus K_{lt}$.
  - ς'_2 given by: $\varsigma'_2(x) = (A, \{m\}_{pubk_B})$, $\varsigma'_2(A) = I$, and ς'_2 is identity on all other terms in $\mathcal{T}_0 \setminus K_{lt}$.
  - ς''_2 given by: $\varsigma''_2(x) = m$, $\varsigma''_2(A) = I$, and ς''_2 is identity on all other terms in $\mathcal{T}_0 \setminus K_{lt}$.

  Examples of plays of $\mathsf{Pr}_{\mathsf{ut}}$ include $\zeta'_1 \stackrel{\text{def}}{=} \varsigma_1(\zeta_1)$, $\zeta'_2 \stackrel{\text{def}}{=} \varsigma'_2(\zeta_2)$ and $\zeta''_2 \stackrel{\text{def}}{=} \varsigma''_2(\zeta_2)$.

■

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

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

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

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

analz-rules $\qquad\qquad$ synth-rules

Figure 1: analz and synth rules.

$$\frac{\dfrac{\dfrac{\dfrac{}{T \vdash t}\,\mathsf{Ax}_a}{T \vdash t_1}\,\mathsf{split}_1 \qquad \dfrac{\dfrac{\dfrac{}{T \vdash t}\,\mathsf{Ax}_a}{T \vdash t_2}\,\mathsf{split}_2}{T \vdash \overline{k'}}\,\mathsf{split}_2}{T \vdash t_3}\,\mathsf{decrypt} \qquad \dfrac{\dfrac{\dfrac{}{T \vdash t}\,\mathsf{Ax}_a}{T \vdash t_2}\,\mathsf{split}_2}{T \vdash \overline{k}}\,\mathsf{split}_1}{\dfrac{T \vdash t_4}{T \vdash m}\,\mathsf{split}_1}\,\mathsf{decrypt}$$

Figure 2: An example analz-proof.

Message generation rules

Definition 1.7 *A* sequent *is of the form $T \vdash t$ where $T \subseteq \mathcal{T}$ and $t \in \mathcal{T}$.*

An analz*-proof π of $T \vdash t$ is an inverted tree whose nodes are labelled by sequents and connected by one of the* analz*-rules in Figure 1, whose root is labelled $T \vdash t$, and whose leaves are labelled by instances of the Ax_a rule. For a set of terms T,* analz(T) *is the set of terms t such that there is an* analz*-proof of $T \vdash t$.*

A synth*-proof π of $T \vdash t$ is an inverted tree whose nodes are labelled by sequents and connected by one of the* synth*-rules in Figure 1, whose root is labelled $T \vdash t$, and whose leaves are labelled by instances of the Ax_s rule. For a set of terms T,* synth(T) *is the set of terms t such that there is 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].

Example 1.8 Let $T = \{t\}$ where $t = (\{\{(m, n)\}_k\}_{k'}, (\overline{k}, \overline{k'}))$. The analz-proof given in Figure 2 shows that $m \in$ analz(T). To save on space, we denote $\{\{(m, n)\}_k\}_{k'}$ by t_1, $(\overline{k}, \overline{k'})$ by t_2, $\{(m, n)\}_k$ by t_3 and (m, n) by t_4. ■

$$\dfrac{\dfrac{\dfrac{\dfrac{}{T\vdash m}\,\mathsf{Ax}_s \quad \dfrac{}{T\vdash n}\,\mathsf{Ax}_s}{T\vdash t_2}\,\mathsf{pair} \quad \dfrac{}{T\vdash k}\,\mathsf{Ax}_s}{T\vdash t_1}\,\mathsf{encrypt} \quad \dfrac{}{T\vdash k'}\,\mathsf{Ax}_s}{T\vdash t}\,\mathsf{encrypt}$$

Figure 3: An example synth-proof.

Example 1.9 Let $T = \{m, n, k, k'\}$ and $t = \{\{(m, n)\}_k\}_{k'}$. The synth-proof given in Figure 3 shows that $t \in \mathsf{synth}(T)$. To save on space, we denote $\{(m, n)\}_k$ by t_1 and (m, n) by t_2. ■

Information states and updates

Definition 1.10 *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 for all* $m \in M$ *and* $A \in Ag$, $m \notin ST(s_A)$.
- $A?B{:}t$ *is* enabled *at* s *iff* $t \in \overline{s_I}$.
- $update(s, A!B{:}(M)t) \stackrel{\text{def}}{=} s'$ *where* $s'_A = s_A \cup M$, $s'_I = s_I \cup \{t\}$, *and* $s'_C = s_C$ *for all the other* $C \in Ag$.
- $update(s, A?B{:}t) \stackrel{\text{def}}{=} s'$ *where* $s'_A = s_A \cup \{t\}$ *and* $s'_C = s_C$ *for all other* $C \in Ag$.

For any information state s and a finite set of actions Ac', we define $update(s, Ac')$ to be a state s' such that:

- for all $A \in Ho$, $s'_A = s_A \cup \{t \mid$ there exists $a \in Ac' \cap Ac_A$ such that $term(a) = t$ or $t \in NT(a)\}$.
- $s'_I = s_I \cup \{t \mid$ there exists $a \in Send \cap Ac'$ with $term(a) = t\}$.

We define $update(s, \eta)$ to be $update(s, Actions(\eta))$,for an information state s and a sequence of actions η.

A sequence of actions $\eta = a_1 \cdots a_\ell$ is said to be *send-admissible* with respect to a state s_0 if for all $i \leq \ell$, if $a_i \in Send$ then a_i is enabled at $update(s_0, a_1 \cdots a_{i-1})$.

Given a protocol Pr, $s_0 \in \mathcal{S}$ is said to be an *initial information state* of Pr if for all $A \in Ho$, $(s_0)_A = K_A \cup CT(\mathsf{Pr})$ 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 1.11 *$infstate(s, E)$ is defined to be $update(s, Actions(E))$ for any $E \subseteq Events$ and information state s. For any $\xi \in Events^*$ and information state s, $infstate(s, \xi)$ is defined to be $infstate(s, Events(\xi))$. We use the notations $infstate_A(s, E)$ and $infstate_A(s, \xi)$ for $(infstate(s, E))_A$ and $(infstate(s, \xi))_A$, respectively. For $e \in Events$, $E \subseteq Events$, and an information state s, e is* enabled at (s, E) *iff $LP(e) \subseteq E$ and $act(e)$ is enabled at $infstate(s, E)$.*

Runs of a protocol

We first isolate the important notion of admissible runs (also called runs) without reference to any protocol.

Definition 1.12 *The set of* admissible runs *(also called* runs*), denoted by $\mathcal{R}$, is defined to be the least subset of $\mathcal{S} \times Events^*$ such that:*

- $(s_0, \varepsilon) \in \mathcal{R}$ *for every* $s_0 \in \mathcal{S}$.
- *Suppose* $(s_0, \xi) \in \mathcal{R}$ *and suppose* $e \in Events$ *is enabled at* $(s_0, Events(\xi))$. *Then* $(s_0,\ \xi \cdot e) \in \mathcal{R}$.

Now we define the notion of a run specific to a protocol.

Definition 1.13 *Given a protocol* Pr*, we say that* (s_0, ξ) *is a* run of Pr *iff* $s_0 \in \mathsf{Init}(\mathsf{Pr})$ *and* $Events(\xi) \subseteq Events(\mathsf{Pr})$*. The set of all runs of* Pr *is denoted by* $\mathcal{R}(\mathsf{Pr})$.

For a fixed set $T \subseteq \mathcal{T}_0$, a run (s_0, ξ) is said to be a T-*run* of Pr if for all $e \in Events(\xi)$, $ST(e) \cap \mathcal{T}_0 \subseteq T$. The set of all T-runs of Pr is denoted by $\mathcal{R}_T(\mathsf{Pr})$.

A run (s_0, ξ) is a *well-typed run* of Pr iff for all $e \in Events(\xi)$, e is a well-typed event. The set of all well-typed runs of Pr is denoted by $\mathcal{R}_{\mathsf{wt}}(\mathsf{Pr})$.

Example 1.14

- An example run of $\mathsf{Pr}_{\mathsf{NS}}$ (which turns out to be well-typed) is given by (s_0, ξ_1) where
  - $s_0 = (K_A)_{A \in Ag}$ and
  - $\xi_1 \stackrel{\text{def}}{=} (\eta_1', 1) \cdot (\eta_2', 1) \cdot (\eta_2', 2) \cdot (\eta_1', 2) \cdot (\eta_1', 3) \cdot (\eta_2', 3)$. It is presented explicitly below:

$$\begin{array}{lllllll}
(\eta_1', 1) & A & ! & I & : & (m) & \{A, m\}_{pubk_I} \\
(\eta_2', 1) & B & ? & A & : & & \{A, m\}_{pubk_B} \\
(\eta_2', 2) & B & ! & A & : & (n) & \{m, n\}_{pubk_A} \\
(\eta_1', 2) & A & ? & I & : & & \{m, n\}_{pubk_A} \\
(\eta_1', 3) & A & ! & I & : & & \{n\}_{pubk_I} \\
(\eta_2', 3) & B & ? & A & : & & \{n\}_{pubk_B}
\end{array}$$

- An example run of $\mathsf{Pr}_{\mathsf{ut}}$ is given by (s_0, ξ_2) where
  - $s_0 = (K_A)_{A \in Ag}$ and
  - $\xi_2 \stackrel{\text{def}}{=} (\zeta_1', 1) \cdot (\zeta_2', 1) \cdot (\zeta_2', 2) \cdot (\zeta_3', 1) \cdot (\zeta_3', 2) \cdot (\zeta_1', 2)$. It is presented explicitly below:

$$\begin{array}{lllllll}
(\zeta_1', 1) & A & ! & B & : & (m) & \{A, \{m\}_{pubk_B}\}_{pubk_B} \\
(\zeta_2', 1) & B & ? & I & : & & \{I, \{A, \{m\}_{pubk_B}\}_{pubk_B}\}_{pubk_B} \\
(\zeta_2', 2) & B & ! & I & : & & \{A, \{m\}_{pubk_B}\}_{pubk_I} \\
(\zeta_2'', 1) & B & ? & I & : & & \{I, \{m\}_{pubk_B}\}_{pubk_B} \\
(\zeta_2'', 2) & B & ! & I & : & & \{m\}_{pubk_I} \\
(\zeta_1', 2) & A & ? & B & : & & \{m\}_{pubk_A}
\end{array}$$

■

The secrecy problem

Definition 1.15 *Given a protocol* Pr *and* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr})$, *secrets*$(s_0, \xi)$ *is defined to be the set of basic terms* m *such that for some prefix* ξ' *of* ξ *and some* $A \in Ho$, *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* $\mathit{secrets}(s_0, \xi) \cap \mathsf{analz}(\mathit{infstate}_I(s_0, \xi)) \neq \emptyset$.

Example 1.16 The runs (s_0, ξ_1) and (s_0, ξ_2) of Example 1.14 are both leaky. For instance, letting $\xi_1' = (\eta_1', 1) \cdot (\eta_2', 1) \cdot (\eta_2', 2)$, it is easy to check that $n \in \mathsf{analz}(\mathit{infstate}_B(s_0, \xi_1'))$ but $n \notin \mathsf{analz}(\mathit{infstate}_I(s_0, \xi_1'))$. Thus by definition $n \in \mathit{secrets}(s_0, \xi_1)$. It is also easy to check that $n \in \mathsf{analz}(\mathit{infstate}_I(s_0, \xi_1))$. Thus ξ_1 is leaky.

Similarly, it is easy to see that $m \in \mathsf{analz}(\mathit{infstate}_A(s_0, \xi_2'))$ but $m \notin \mathsf{analz}(\mathit{infstate}_I(s_0, \xi_2'))$ (where xi_2' is the sequence $((\zeta_1', 1))$. Thus by definition $m \in \mathit{secrets}(s_0, \xi_2)$. It is also easy to check that $m \in \mathsf{analz}(\mathit{infstate}_I(s_0, \xi_2))$. Thus ξ_2 is leaky. ■

The following are some versions of the secrecy problem that we are interested in.

Definition 1.17 The secrecy problem: *The problem of determining for a given protocol* Pr *whether some run of* Pr *is leaky.*

The secrecy problem for T-runs (for fixed T): *The problem of determining for a given protocol* Pr *whether some* T*-run of* Pr *is leaky.*

The secrecy problem for well-typed runs: *The problem of determining for a given protocol* Pr *whether some well-typed run of* Pr *is leaky.*

2 Undecidability results

In this section we prove that the secrecy problem for security protocols is in general undecidable. In fact we prove that the secrecy problem for well-typed runs as well as the secrecy problem for T-runs (even for a fixed finite T) are undecidable. These statements do not necessarily imply that the general secrecy problem is undecidable, but we will show that in fact our proofs yield that result as well.

The undecidability result for well-typed runs was first proved by [CDL+99] (see also [DLMS99]) in the setting of multi-set rewriting. We use a different reduction from that used in [CDL+99]. Our reduction is much simpler than the ones currently found in the literature. To our knowledge, ours is also the first detailed proof of this result, a fact which can be attributed to the simplicity of our reduction.

The undecidability result for T-runs has been proved in various places, including for instance, [HT96] and [ALV01]. Our proof is a variant of the one in [ALV01].

Two-counter machines

Our undecidability results use a reduction from the reachability problem for two-counter machines. We recall the relevant definitions below:

A two-counter machine is a tuple $M = (Q, F, q_0, \delta)$ where:

- Q is a finite set of *states*,
- $F \subseteq Q$ is the set of *final states*,
- $q_0 \in Q$ is the *initial state*,
- $\delta \subseteq Q \times \{0,1\}^2 \times Q \times \{-1,0,1\}^2$ is the *transition relation* with the condition that whenever $(q, i_1, i_2, q', j_1, j_2) \in \delta$ then $j_k = -1$ implies $i_k = 1$, for $k = 1,2$ (we can decrement a positive counter only).

The other standard notions relating to two-counter machines are defined below:

- A *configuration* of a two-counter machine $M = (Q, F, q_0, \delta)$ is a triple (q, n_1, n_2) with $q \in Q, n_k \in \mathbb{N}$ (the n_k's are counters).
- For a configuration (q, n_1, n_2) of M and a transition $t = (q, i_1, i_2, q', j_1, j_2) \in \delta$, t is *enabled* at (q, n_1, n_2) iff for $k = 1, 2$, $i_k = 0$ iff $n_k = 0$ and whenever t is enabled at (q, n_1, n_2) we have the *reduction* $(q, n_1, n_2) \xrightarrow{t} (q', n_1 + j_1, n_2 + j_2)$.
- A configuration (q, n_1, n_2) is *reachable* if $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$.
- A configuration (q, n_1, n_2) is *final* if $q \in F$.
- The *reachability problem for two-counter machines* is the problem of determining for a given two-counter machine $M = (Q, F, q_0, \delta)$ whether a final configuration of M is reachable.

We assume the well-known fact that the reachability problem for two-counter machines is undecidable.

2.1 The secrecy problem for well-typed runs

Let $M = (Q, F, q_0, \delta)$ be an arbitrary two-counter machine. We will define a protocol $\mathsf{Pr}_M = (\mathsf{C}, \mathsf{R})$ such that a final configuration of M is reachable iff there is some $(s_0, \xi) \in \mathcal{R}_{\mathsf{wt}}(\mathsf{Pr}_M)$ which is leaky. Before defining the actual reduction, we set up some basic notation: For simplicity, assume $Q \subseteq N$. Let z be a fixed nonce from N. We fix honest agents A, B and a long-term shared key k_{AB}. Then we define the following terms:

$$\text{for any } u, u' \in N, \text{ and } q \in Q, \quad [q, u, u'] \stackrel{\text{def}}{=} \{(q, (u, u'))\}_{k_{AB}}.$$
$$\text{for any } u, u' \in N, \quad [u, u'] \stackrel{\text{def}}{=} \{(u, u')\}_{k_{AB}}.$$

The protocol Pr_M is defined as follows:

Definition 2.1 $\mathsf{Pr}_M \stackrel{\text{def}}{=} (\mathsf{C}, \mathsf{R})$ *where:*

- $\mathsf{C} = Q \cup \{A, B, z\}$ *and*
- $\mathsf{R} = \{\eta_0\} \cup \{\eta_t \mid t \in \delta\} \cup \{\eta_f \mid f \in F\}$ *where:*

- $\eta_0 \stackrel{\text{def}}{=} A!B{:}\,[z,z],[q_0,z,z],[z,z]$.
- *for each transition* $t=(q,i_1,i_2,q',j_1,j_2)\in\delta$, $\eta_t \stackrel{\text{def}}{=} a\cdot a'$ *with:*

$$\begin{aligned}a &= A?B{:}\,[u_1,v_1],[q,w_1,w_2],[u_2,v_2];\\ a' &= A!B{:}\,(M)\ [u'_1,v'_1],[q',w'_1,w'_2],[u'_2,v'_2]\end{aligned}$$

where, for $k\in\{1,2\}$, *the following conditions hold:*

$$\begin{aligned}&i_k=0 \textit{ iff } w_k=z\\ &j_k=0 \textit{ iff } w_k=w'_k \textit{ and } u_k=v_k=u'_k=v'_k=z\\ &j_k=1 \textit{ iff } w_k\neq w'_k,\ u'_k=w_k,\ v'_k=w'_k \textit{ and } u_k=v_k=z\\ &j_k=-1 \textit{ iff } w_k\neq w'_k,\ v_k=w_k,\ u_k=w'_k \textit{ and } u'_k=v'_k=z\\ &\textit{if } j_k=1 \textit{ then } v'_k\notin \mathsf{C}\cup ST(a)\cup\{v'_{2-k+1}\}\\ &M=\{v'_k \mid k\in\{1,2\} \textit{ and } j_k=1\}.\end{aligned}$$

For any η_t *as given above, and* $k\in\{1,2\}$, *the notation* $\mathsf{inctr}_k(\eta_t)$ *is used to denote* w_k *and the notation* $\mathsf{outctr}_k(\eta_t)$ *is used to denote* w'_k.

- *For each* $f\in F$, $\eta_f \stackrel{\text{def}}{=} a\cdot a'\cdot a''$ *where:*

$$\begin{aligned}a &= A?B{:}\,[f,w_1,w_2];\\ a' &= A!B{:}\,(\{x\})\ \{x\}_{k_{AB}};\\ a'' &= A!B{:}\,x\end{aligned}$$

where $x\notin \mathsf{C}\cup ST(a)$.

The role corresponding to the transition $t=(q,0,1,q',1,-1)$ is presented by way of example:

$$\begin{aligned}&A?B{:}\,[z,z],[q,z,w_2],[w'_2,w_2];\\ &A!B{:}\,(\{w'_1\})\ [z,w'_1],[q',w'_1,w'_2],[z,z]\end{aligned}$$

The role η_0 starts off the simulation of the two-counter machine. The role η_f checks if a final configuration with state f is reached and if so signals it by contriving to "leak" a freshly minted secret. The role η_t simulates the transition $t\in\delta$.

Lemma 2.2 *For all* $(s_0,\xi)\in\mathcal{R}(\mathsf{Pr}_M)$, $k_{AB}\notin ST(\mathit{infstate}_I(s_0,\xi))$.

Proof: The proof is an easy induction on $|\xi|$. For $\xi=\varepsilon$, it is clear that k_{AB} does not belong to $ST(\mathit{infstate}_I(s_0,\xi))$ (which is the same as $ST((s_0)_I)$).

Suppose $(s_0,\xi)\in\mathcal{R}(\mathsf{Pr}_M)$ where $\xi=\xi'\cdot e$. Let T' and T be used to denote $\mathit{infstate}_I(s_0,\xi')$ and $\mathit{infstate}_I(s_0,\xi)$, respectively. By induction hypothesis $k_{AB}\notin ST(T')$. This means that for all $e'\in \mathit{Events}(\xi')$, $\mathit{term}(\mathit{act}(e'))$ does not contain k_{AB} as a subterm. The conditions imposed on substitutions being suitable for a protocol and role ensure that if $\mathit{act}(e)$ is a send action, then $\mathit{term}(\mathit{act}(e))$ does not contain k_{AB} as a subterm unless $\mathit{term}(\mathit{act}(e'))$ contains it as a subterm, for some $e'\in\mathit{Events}(\xi')$. Therefore if $\mathit{act}(e)$ is a send then k_{AB} does not occur as a subterm in $\mathit{term}(\mathit{act}(e))$ and thus $k_{AB}\notin ST(T)$. On the other hand if $\mathit{act}(e)$ is a receive action then $T=T'$ and hence the fact that $k_{AB}\notin ST(T)$ follows from the induction hypothesis. ■

Lemma 2.3 *For all* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$*, letting* $infstate_I(s_0, \xi) = T$*, if* $[u, u'] \in \overline{T}$ *then* $[u, u'] \in \mathsf{analz}(T)$*. Similarly for terms of the form* $[q, u, u']$*.*

Proof: It suffices to note that by lemma 2.2, $k_{AB} \notin ST(T)$ and hence $k_{AB} \notin \overline{T}$ as well. The statement of the lemma follows immediately from this. ∎

Lemma 2.4 $\mathcal{R}(\mathsf{Pr}_M) = \mathcal{R}_{\mathsf{wt}}(\mathsf{Pr}_M)$*.*

Proof: We have to prove that for all $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, (s_0, ξ) is well-typed. We proceed by induction on $|\xi|$. The base case, when $|\xi| = 0$, is trivial.

Suppose that $(s_0, \xi \cdot e) \in \mathcal{R}(\mathsf{Pr}_M)$ and that (s_0, ξ) is well-typed. The conditions imposed on substitutions being suitable for a protocol and a role ensure that if $act(e)$ is a send action, then $(s_0, \xi \cdot e)$ is well-typed as well. On the other hand, if $act(e)$ is a receive action, then $term(act(e))$ is always a tuple of terms of the form $[q, u, u']$ or $[u, u']$ which belong to $\overline{infstate_I(s_0, \xi)}$. But by lemma 2.3, these in fact belong to $\mathsf{analz}(infstate_I(s_0, \xi))$, and therefore each of them occurs as subterms of $term(act(e'))$ for some $e' \in Events(\xi)$. Since (s_0, ξ) is well-typed, the u's and u''s are guaranteed to be *nonces*. This ensures that $(s_0, \xi \cdot e)$ is well-typed as well. ∎

Definition 2.5

1. *We say that a number n is represented in an information state s by a nonce u if there exist* distinct *nonces $u_0, \ldots, u_n$ such that $u_0 = z$, $u_n = u$, and for all $i < n$, $[u_i, u_{i+1}] \in \overline{s_I}$.*

2. *We say that a configuration (q, n, n') is represented in an information state s by the term $[q, u, u']$ if u represents n in s, u' represents n' in s, and $[q, u, u'] \in \overline{s_I}$.*

3. *For some $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, we say that a number n is represented in (s_0, ξ) by a nonce u if n is represented in $infstate(s_0, \xi)$ by u.*

4. *For some $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, we say that a configuration (q, n_1, n_2) is represented in (s_0, ξ) by a term $[q, u_1, u_2]$ if (q, n_1, n_2) is represented in $infstate(s_0, \xi)$ by $[q, u_1, u_2]$.*

From the definition it follows that in all states s, z represents 0. Note that there can be many representatives for the same number or configuration in a given state. While this does not affect the proofs which follow, the other possibility of many numbers being represented by the same nonce is undesirable. Fortunately the following lemma says that we do not have to worry about this eventuality.

Definition 2.6 *An information state s is a* good state *if for all nonces u, there is at most one $u' \neq u$ such that $[u', u] \in \overline{s_I}$.*

It is trivial to observe that in a good state, every nonce represents at most one number and every term represents at most one configuration. It is also an immediate observation that z represents only 0 in any state and that only z represents 0 in any state.

Lemma 2.7 *For all* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$*,* $\mathit{infstate}(s_0, \xi)$ *is a good state.*

Proof: We prove the statement by induction on $|\xi|$. Throughout we will let $\mathit{infstate}(s_0, \xi)$ be denoted by s. The base case is when $|\xi| = 0$ in which case $s = s_0$. Suppose $[u', u] \in \overline{s_I}$. Then by lemma 2.3 $[u', u] \in \mathsf{analz}(s_I)$ which contradicts the fact that $\mathsf{analz}((s_0)_I) \subseteq \mathcal{T}_0$. Thus no term of the form $[u', u]$ belongs to s_I. Hence s is a good state.

Now for the induction step. Suppose $\xi = \xi' \cdot e$ and let $\mathit{infstate}(s_0, \xi')$ be denoted by s'. By induction hypothesis s' is a good state. We have to prove that s is also a good state. Suppose $\mathit{act}(e)$ is a receive action. Then $s'_I = s_I$ and thus s is a good state as well. Suppose on the other hand that $\mathit{act}(e)$ is a send action. Now consider any nonce u. If for all u', $[u', u] \notin ST(\mathit{term}(\mathit{act}(e)))$ then for any u', $[u', u] \in \overline{s_I}$ iff $[u', u] \in \overline{s'_I}$ and thus there is at most one $u' \neq u$ such that $[u', u] \in \overline{s_I}$ (by the induction hypothesis). Otherwise, suppose $[u', u] \in ST(\mathit{term}(\mathit{act}(e)))$. Now from the definition of Pr_M it is clear that either $u \notin ST(s'_A)$ for all $A \in Ag$ (i.e. u is a new nonce) or $u' = u = z$. In the first case, it is seen from the definition of Pr_M that there is at most one u' such that $[u', u] \in ST(\mathit{term}(\mathit{act}(e)))$ and thus at most one u' such that $[u', u] \in \overline{s_I}$. In the second case, clearly if there is at most one $u' \neq z$ such that $[u', z] \in \overline{s'_I}$, the same continues to hold with respect to $\overline{s_I}$ as well. Thus s is a good state even in the case when $\mathit{act}(e)$ is a send action. This completes the induction step and the proof. ■

The following lemma states that the role η_t faithfully simulates the transition t.

Lemma 2.8 *Suppose* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$*,* $s = \mathit{infstate}(s_0, \xi)$*,* $t = (q, i_1, i_2, q', j_1, j_2) \in \delta$*,* $\eta_t = a \cdot a'$ *and* (q, n_1, n_2) *is a configuration of* M *represented in* s*.*

1. *If* t *is enabled at* (q, n_1, n_2) *then there is a substitution* σ *suitable for* Pr_M *and* η_t *such that:*
   - $\sigma(\mathsf{inctr}_k(\eta_t))$ *represents* n_k *in* s *(for* $k = 1, 2$*),*
   - $\sigma(a)$ *is enabled in* s *and* $\sigma(a')$ *is enabled at* $\mathit{update}(s, \sigma(a))$*, and*
   - $\sigma(\mathsf{outctr}_k(\eta_t))$ *represents* $n_k + j_k$ *in* $\mathit{update}(s, \sigma(\eta_t))$ *(for* $k = 1, 2$*).*
2. *If there is a substitution* σ *suitable for* Pr_M *and* η_t *such that* $\sigma(\mathsf{inctr}_k(\eta_t))$ *represents* n_k *(for* $k = 1, 2$*) then* t *is enabled at* (q, n_1, n_2)*.*

Proof: Suppose $t = (q, i_1, i_2, q', j_1, j_2)$ and suppose

$a = A?B{:}\,[u_1, v_1], [q, w_1, w_2], [u_2, v_2];$
$a' = A!B{:}\,(M)\ [u'_1, v'_1], [q', w'_1, w'_2], [u'_2, v'_2]$

1. Suppose t is enabled at (q, n_1, n_2). This means that for $k = 1, 2$, $i_k = 0$ iff $n_k = 0$. Let r_k represent n_k in s. We choose a substitution σ suitable for Pr_M and η_t which is compatible with the following conditions:
   - for $k = 1, 2$, $\sigma(w_k) = r_k$,
   - for $m \in M$ and $A \in Ag$, $\sigma(m) \notin ST(s_A)$, and

- for $k = 1, 2$, if $j_k = -1$ then $\sigma(u_k) = r'_k$ where r'_k represents $n_k - 1$ in s and $[r'_k, r_k] \in \overline{s_I}$ (since s is a good state by Lemma 2.7, there exists a unique r'_k with this property).

Clearly such a σ can be chosen. Let $s' = \mathit{update}(s, \sigma(a))$ and $s'' = \mathit{update}(s', \sigma(a'))$.

- From the definition it is immediate that $\sigma(\mathsf{inctr}_k(\eta_t))$, which is the same as $\sigma(w_k)$ represents n_k at s, for $k = 1, 2$.
- We now prove that $\sigma(a)$ is enabled at s and $\sigma(a')$ is enabled at s'. Since $[q, r_1, r_2]$ represents (q, n_1, n_2) in s, $[q, r_1, r_2] \in \overline{s_I}$. For $k = 1, 2$, if $j_k \neq -1$ then $u_k = v_k = z$ and so $[\sigma(u_k), \sigma(v_k)] = [\sigma(z), \sigma(z)] = [z, z]$. Now from the definition of Pr_M it follows that if any number is represented in any run (s_0, ξ) of Pr_M, then $[z, z] \in \overline{\mathit{infstate}_I(s_0, \xi)}$. Therefore $[\sigma(u_k, \sigma(v_k)] = [z, z] \in \overline{s_I}$. Otherwise, $j_k = -1$ and now $v_k = w_k$ and therefore $[\sigma(u_k), \sigma(v_k)] = [\sigma(u_k), \sigma(w_k)] = [r'_k, r_k] \in \overline{s_I}$ (by choice of σ). From this it follows that a is enabled at s. By assumption, for all $m \in M$ and for all $A \in Ag$, $\sigma(m) \notin ST(s_A)$. Besides by definition of Pr_M, $M \cap ST(a) = \emptyset$, and thus for all $A \in Ag$, $\sigma(m) \notin ST(s'_A)$ as well. Also it is quite easy to verify that $\mathit{term}(a') \in \overline{s'_A \cup M}$. Thus a' is enabled in s'.
- Now we prove that $\sigma(\mathsf{outctr}_k(\eta_t)) = \sigma(w'_k)$ represents $n_k + j_k$ in s'' (for $k = 1, 2$). If $j_k = 0$ then $w_k = w'_k$, for $k = 1, 2$ (by definition of Pr_M). Hence it follows that $\sigma(w'_k)$ represents $n_k + j_k$ in s'. If $j_k = -1$ then by definition of σ, $\sigma(u_k)$ represents $n_k - 1 = n_k + j_k$ in s. By definition of Pr_M, $w'_k = u_k$ and thus it follows that $\sigma(w'_k)$ represents $n_k + j_k$ in s and hence in s'' as well. If $j_k = 1$ then since $[\sigma(u'_k), \sigma(v'_k)] \in \overline{s''_I}$, $w'_k = v'_k$, $w_k = u'_k$, and $\sigma(u'_k) = \sigma(w_k) = r_k$ represents n_k in s and hence in s'' as well, $\sigma(w'_k) = \sigma(v'_k)$ represents $n_k + j_k = n_k + 1$ in s''.

2. Suppose σ is a substitution suitable for Pr_M and η_t such that for $k = 1, 2$, $\sigma(\mathsf{inctr}_k(\eta_t)) = \sigma(w_k)$ represents n_k at s. We need to show that $i_k = 0$ iff $n_k = 0$.

   Suppose $i_k = 0$. Then by definition of Pr_M, $w_k = z$, and hence $\sigma(w_k) = z$. Since z represents only 0 in any state and we are given $\sigma(w_k)$ represents n_k at s, $n_k = 0$.

   Suppose $n_k = 0$. Since $\sigma(w_k)$ represents $n_k = 0$ at s and since only z represents 0 in any state, $\sigma(w_k) = z$. But according to definition of Pr_M, this happens only when $i_k = 0$.

■

Theorem 2.9 *1. $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$ iff there is a run (s_0, ξ) of Pr_M in which (q, n_1, n_2) is represented.*

2. A final configuration is reachable in M iff there is a leaky run of Pr_M.

Proof:

1. There are two directions to the proof.

   $(\Rightarrow)$: Let m be the length of the derivation $(q_0,0,0)\xrightarrow{*}(q,n_1,n_2)$. We prove the result by induction on m. The base case is when $m=0$ in which case $q=q_0$ and $n_1=n_2=0$. Then the run $(s_0,(\eta_0,1))$ satisfies the statement of the theorem, where $s_0 \in \mathsf{Init}(\mathsf{Pr}_M)$.

   Suppose $(q_0,0,0)\xrightarrow{*}(q,n_1,n_2)\xrightarrow{t}(q',n_1',n_2')$. By the induction hypothesis there is a run $(s_0,\xi) \in \mathcal{R}(\mathsf{Pr}_M)$ in which (q,n_1,n_2) is represented. Let $s = \mathit{infstate}(s_0,\xi)$. Let $t=(q,i_1,i_2,q',j_1,j_2)$ and $\eta_t = a\cdot a'$. By lemma 2.8, there is a substitution σ suitable for Pr_M and η_t such that $\sigma(a)$ is enabled at s, $\sigma(a')$ is enabled at $\mathit{update}(s,\sigma(a))$, and $\sigma(\mathsf{outctr}_k(\eta_t))$ represents $n_k+j_k=n_k'$ in $\mathit{update}(s,\sigma(\eta_t))$. Letting $e=(\sigma(\eta_t),1)$ and $e'=(\sigma(\eta_t),2)$ it is easy to see that $(s_0,\xi\cdot e\cdot e') \in \mathcal{R}(\mathsf{Pr}_M)$. Further, since $[q',\sigma(w_1'),\sigma(w_2')] \in \mathit{infstate}_I(s_0,\xi\cdot e\cdot e')$, it is clear that (q',n_1',n_2') is represented in $(s_0,\xi\cdot e\cdot e')$.

   $(\Leftarrow)$: We prove by induction on $|\xi|$ that for all $(s_0,\xi) \in \mathcal{R}(\mathsf{Pr}_M)$, if (q,n_1,n_2) is represented in (s_0,ξ) then $(q_0,0,0)\xrightarrow{*}(q,n_1,n_2)$.

   The base case is when $|\xi|=0$ and then the statement is vacuously true since no configuration is represented in (s_0,ξ).

   Suppose (q',n_1',n_2') is represented in $(s_0,\xi) \in \mathcal{R}(\mathsf{Pr}_M)$, where $\xi = \xi''\cdot e'$. If (q',n_1',n_2') is already represented in (s_0,ξ'') then by induction hypothesis it follows that $(q_0,0,0)\xrightarrow{*}(q',n_1',n_2')$. Otherwise there are two cases to consider:

   $e'=(\eta_0,1)$**:** Then it is clear that $(q',n_1',n_2')=(q_0,0,0)$ and hence the (q',n_1',n_2') is vacuously reachable from $(q_0,0,0)$.

   $e'=(\sigma(\eta_t),2)$ **for some** $t\in\delta$**:** Let $t=(q,i_1,i_2,q',j_1,j_2)$ and let $\eta_t=a\cdot a'$. It is clear that $\sigma(\mathsf{outctr}_k(\eta_t))$ represents n_k' for $k=1,2$. Further for $k=1,2$, $n_k'=n_k+j_k$ where $\sigma(\mathsf{inctr}_k(\eta_t))$ represents n_k in $\mathit{infstate}(s_0,\xi'')$. Further there is a proper prefix ξ of ξ'' such that (q,n_1,n_2) is represented in $\mathit{infstate}(s_0,\xi)$, and $\sigma(\eta_t)$ is enabled at $\mathit{infstate}(s_0,\xi)$. By induction hypothesis we have that (q,n_1,n_2) is a reachable configuration and by lemma 2.8, we know that t is enabled at (q,n_1,n_2). Therefore $(q,n_1,n_2)\xrightarrow{t}(q',n_1+j_1,n_2+j_2)=(q',n_1',n_2')$. Thus (q',n_1',n_2') is also a reachable configuration.

2. There are two directions to the proof.

   $(\Rightarrow)$: Suppose a final configuration (f,n_1,n_2) is reachable in M. Then there is a run $(s_0,\xi) \in \mathcal{R}(\mathsf{Pr}_M)$ representing (f,n_1,n_2). Thus $[f,r_1,r_2] \in \mathit{infstate}_I(s_0,\xi)$ for some nonces r_1 and r_2. and hence some instance η of η_f is enabled at (s_0,ξ). Letting $e_i=(\eta,i)$ for $i=1,2,3$, it is easy to see that $(s_0,\xi\cdot e_1\cdot e_2\cdot e_3) \in \mathcal{R}(\mathsf{Pr}_M)$, and by definition of Pr_M this run is patently leaky.

   $(\Leftarrow)$: Suppose there is a leaky run $(s_0,\xi) \in \mathcal{R}(\mathsf{Pr}_M)$. According to the definition of Pr_M, this means that some instance of η_f for $f\in F$ has been played out as part of ξ. But this means that some configuration of the form (f,n_1,n_2) is represented in (s_0,ξ) which implies that a final configuration is reachable in M.

■

The main conclusion of this section is stated below.

Theorem 2.10 *1. The general secrecy problem is undecidable.*

2. The secrecy problem for well-typed runs is undecidable.

Proof:

1. This immediately follows from item 2 of Theorem 2.9 and the fact that the reachability problem for two-counter machines is undecidable.

2. This follows immediately from item 1 of this theorem and from Lemma 2.4, where it is shown that all runs of Pr_M are well-typed.

■

2.2 Secrecy problem for T-runs

Fix a finite set of nonces N_0 and a finite set of short-term keys K_0. This fixes a finite set $T \subseteq \mathcal{T}_0$ and with respect to this T, we can define the secrecy problem for T-runs. In this section we show that this problem is undecidable. The proof is again a reduction from the reachability problem for two-counter machines. For the purposes of coding up arbitrary two-counter machines, we make the following assumptions about N_0. We assume that x, z, u_1 and u_2 are *fixed, distinct* nonces which belong to N_0

Let $M = (Q, F, q_0, \delta)$ be a two-counter machine. For simplicity we assume that $Q \subseteq \mathbb{N}$. We will define a protocol $\mathsf{Pr}_M = (\mathsf{C}, \mathsf{R})$ such that a final configuration of M is reachable iff there is a leaky T-run of Pr_M. Before defining the actual reduction, we set up some basic notation: We fix honest agents A, B and the long-term shared key k_{AB}. Then we define the following terms:

$\underline{0} = z.$
$\underline{i+1} = (\underline{i}, z).$
for any terms t_1, t_2, t_3, $\quad [t_1, t_2, t_3] \stackrel{\text{def}}{=} \{(t_1, (t_2, t_3))\}_{k_{AB}}.$

The protocol Pr_M is defined as follows:

Definition 2.11 $\mathsf{Pr}_M \stackrel{\text{def}}{=} (\mathsf{C}, \mathsf{R})$ *where:*

- $\mathsf{C} = \{A, B, z\}$ *and*
- $\mathsf{R} = \{\eta_0\} \cup \{\eta_t \mid t \in \delta\} \cup \{\eta_f \mid f \in F\}$ *where:*
  - $\eta_0 \stackrel{\text{def}}{=} A!B{:}\,[\underline{q_0}, z, z],$

– *for each transition* $t = (q, i_1, i_2, q', j_1, j_2) \in \delta$, $\eta_t = a \cdot a'$ *with:*

$A?B$: $[\underline{q}, w_1, w_2]$;
$A!B$: $[\underline{q'}, w_1', w_2']$

where, for $k \in \{1,2\}$, *the following conditions hold:*

$i_k = 0$ *and* $j_k = 0$ *iff* $w_k = w_k' = z$
$i_k = 0$ *and* $j_k = 1$ *iff* $w_k = z$ *and* $w_k' = (z, z)$
$i_k = 1$ *and* $j_k = 0$ *iff* $w_k = w_k' = u_k$
$i_k = 1$ *and* $j_k = 1$ *iff* $w_k = u_k$ *and* $w_k' = (u_k, z)$
$i_k = 1$ *and* $j_k = -1$ *iff* $w_k' = u_k$ *and* $w_k = (u_k, z)$

For any η_t *as given above, and* $k \in \{1,2\}$, *the notation* $\mathsf{inctr}_k(\eta_t)$ *is used to denote the term* w_k *and the notation* $\mathsf{outctr}_k(\eta_t)$ *is used to denote the term* w_k'.

– *For each* $f \in F$, $\eta_f \stackrel{\text{def}}{=} a \cdot a' \cdot a''$ *with:*

$a = A?B$: $[\underline{f}, u_1, u_2]$;
$a' = A!B$: $(\overline{\{x\}})\ \{x\}_{k_{AB}}$;
$a'' = A!B$: x

where $x \notin \mathsf{C} \cup \{z, u_1, u_2\}$.

The role corresponding to the transition $t = (q, 0, 1, q', 1, -1)$ is presented by way of example:

$A?B$: $[\underline{q}, z, (u_2, z)]$;
$A!B$: $[\underline{q'}, (z, z), u_2]$

The role η_0 starts off the simulation of the two-counter machine. The role η_f checks if a final configuration with state f is reached and if so signals it by contriving to "leak"a freshly minted secret. The role η_t simulates the transition $t \in \delta$.

Lemma 2.12 *For all* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, $k_{AB} \notin ST(\mathit{infstate}_I(s_0, \xi))$.

Proof: The proof is an easy induction on $|\xi|$. For $\xi = \varepsilon$, it is clear that k_{AB} does not belong to $ST(\mathit{infstate}_I(s_0, \xi))$ (which is the same as $ST((s_0)_I)$).

Suppose $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$ where $\xi = \xi' \cdot e$. Let T' and T denote $\mathit{infstate}_I(s_0, \xi')$ and $T = \mathit{infstate}_I(s_0, \xi)$ respectively. By induction hypothesis k_{AB} does not belong to $ST(T')$. This means that for all $e' \in \mathit{Events}(\xi')$, $\mathit{term}(\mathit{act}(e'))$ does not contain k_{AB} as a subterm. The conditions imposed on substitutions being suitable for a protocol and role ensure that if $\mathit{act}(e)$ is a send action, then $\mathit{term}(\mathit{act}(e))$ does not contain k_{AB} as a subterm unless $\mathit{term}(\mathit{act}(e'))$ contains it as a subterm, for some $e' \in \mathit{Events}(\xi')$. Therefore if $\mathit{act}(e)$ is a send then k_{AB} does not occur as a subterm in $\mathit{term}(\mathit{act}(e))$ and thus $k_{AB} \notin ST(T)$. On the other hand if $\mathit{act}(e)$ is a receive action then $T = T'$ and hence the fact that $k_{AB} \notin ST(T)$ follows frm the induction hypothesis. ■

Lemma 2.13 *For all* $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, *letting* $\mathit{infstate}_I(s_0, \xi) = T$, *if* $[t, t', t''] \in \overline{T}$ *then* $[t, t', t''] \in \mathsf{analz}(T)$.

Proof: It suffices to note that by lemma 2.12, $k_{AB} \notin ST(T)$ and hence $k_{AB} \notin \overline{T}$ as well. The statement of the lemma follows immediately from this. ■

Lemma 2.14 *For all $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, if for all σ and $f \in F$, $(\sigma(\eta_f), 1) \notin Events(\xi)$, then (s_0, ξ) is a T-run of Pr_M.*

Proof: We have to prove that for all $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, (s_0, ξ) is a T-run. We proceed by induction on $|\xi|$. The base case, when $|\xi| = 0$, is trivial.

Suppose that $(s_0, \xi \cdot e) \in \mathcal{R}(\mathsf{Pr}_M)$ and that (s_0, ξ) is a T-run. Again the conditions imposed on substitutions being suitable for a protocol and a role ensure that if $act(e)$ is a send action, since no new nonce is generated in event e (because e is not an instance of (η_f, i) for any $f \in F$) and since the protocol specification mentions only terms built using T, $(s_0, \xi \cdot e)$ is a T-run as well, since the protocol specification mentions only terms built using T. On the other hand, if $act(e)$ is a receive action, then $term(act(e))$ is always some term of the form $[t, t', t'']$ which belong to $\overline{infstate_I(s_0, \xi)}$. But since $k_{AB} \notin \overline{infstate_I(s_0, \xi)}$, $[t, t', t'']$ must have occurred in a previous send event. The induction hypothesis ensures that in this case $(s_0, \xi \cdot e)$ is a T-run as well. ■

Definition 2.15 1. *We say that a configuration (q, n, n') is represented in an information state s if the term $[\underline{q}, \underline{n}, \underline{n'}] \in \overline{s_I}$.*

2. *For some $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, we say that a configuration (q, n, n') is represented in (s_0, ξ) if (q, n, n') is represented in $infstate(s_0, \xi)$.*

The following lemma states that the role η_t faithfully simulates the transition t.

Lemma 2.16 *Suppose $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, $s = infstate(s_0, \xi)$, $t = (q, i_1, i_2, q', j_1, j_2) \in \delta$, $\eta_t = a \cdot a'$ and (q, n_1, n_2) is a configuration of M represented in s.*

1. *If t is enabled at (q, n_1, n_2) then there is a substitution σ suitable for Pr_M and η_t such that:*
   - *$\sigma(\mathsf{inctr}_k(\eta_t))$ represents n_k in s (for $k = 1, 2$),*
   - *$\sigma(a)$ is enabled in s and $\sigma(a')$ is enabled at $update(s, \sigma(a))$, and*
   - *$\sigma(\mathsf{outctr}_k(\eta_t))$ represents $n_k + j_k$ in $update(s, \sigma(\eta_t))$ (for $k = 1, 2$).*

2. *If there is a substitution σ suitable for Pr_M and η_t such that $\sigma(\mathsf{inctr}_k(\eta_t))$ represents n_k (for $k = 1, 2$) then t is enabled at (q, n_1, n_2).*

Proof: Suppose $t = (q, i_1, i_2, q', j_1, j_2)$ and suppose

$a = A?B{:}\,[\underline{q}, w_1, w_2]$;
$a' = A!B{:}\,[\underline{q'}, w_1', w_2']$

1. Suppose t is enabled at (q, n_1, n_2). This means that for $k = 1, 2$, $i_k = 0$ iff $n_k = 0$. We define a substitution σ as follows:

   - for all $m \in \mathcal{T}_0 \setminus \{u_1, u_2\}$, $\sigma(m) = m$,
   - for $k = 1, 2$,
     - if $i_k = 0$ then $\sigma(u_k) = u_k$
     - if $i_k = 1$ and $j_k \neq 1$ then $\sigma(u_k) = \underline{n_k}$
     - if $j_k = -1$ then $\sigma(u_k) = \underline{n_k - 1}$

   It is easily seen that σ is suitable for Pr_M and η_t. Let $s' = \mathit{update}(s, \sigma(a))$ and $s'' = \mathit{update}(s', \sigma(a'))$.

   - If $i_k = 0$ then $w_k = z$, and since in this case $n_k = 0$ as well it is immediate that $\sigma(\mathsf{inctr}_k(\eta_t))$ represents n_k in s. If $i_k = 1$ and $j_k \neq -1$ then $w_k = u_k$, and since $\sigma(u_k) = \underline{n_k}$ it is clear that $\sigma(\mathsf{inctr}_k(\eta_t))$ represents n_k in s. If $j_k = -1$ then $w_k = (u_k, z)$, and since $\sigma(u_k) = \underline{n_k - 1}$ it is clear that $\sigma(\mathsf{inctr}_k(\eta_t))$ represents n_k in s.
   - We are given that (q, n_1, n_2) is represented in s, i.e., $[\underline{q}, \underline{n_1}, \underline{n_2}] \in \overline{s_I}$. But since $\sigma(w_k) = \underline{n_k}$ for $k = 1, 2$, it is easy to see that $\sigma(a)$ is enabled at s. Since $\sigma(\mathit{term}(a')) \in \overline{\{z, k_{AB}\}}$, it is immediate that $\sigma(a')$ is enabled at $\mathit{update}(s, \sigma(a))$.
   - If $j_k = 0$ then $\mathsf{outctr}_k(\eta_t) = \mathsf{inctr}_k(\eta_t)$ and thus $\sigma(\mathsf{outctr}_k(\eta_t))$ represents $n_k = n_k + j_k$ in s''. If $j_k = 1$ then $\mathsf{outctr}_k(\eta_t) = (\mathsf{inctr}_k(\eta_t), z)$ and thus $\sigma(\mathsf{outctr}_k(\eta_t))$ represents $n_k + 1 = n_k + j_k$ in s''. If $j_k = -1$ then $\mathsf{inctr}_k(\eta_t) = (\mathsf{outctr}_k(\eta_t), z)$ and thus $\sigma(\mathsf{outctr}_k(\eta_t))$ represents $n_k - 1 = n_k + j_k$ in s''.

2. Suppose σ is a substitution suitable for Pr_M and η_t such that for $k = 1, 2$, $\sigma(\mathsf{inctr}_k(\eta_t)) = \sigma(w_k)$ represents n_k at s. We need to show that $i_k = 0$ iff $n_k = 0$.

   Suppose $i_k = 0$. Then by definition of Pr_M, $w_k = z$, and hence $\sigma(w_k) = z$. Since z represents only 0 in any state and we are given $\sigma(w_k)$ represents n_k at s, $n_k = 0$.

   Suppose $n_k = 0$. Since $\sigma(w_k)$ represents $n_k = 0$ at s and since only z represents 0 in any state, $\sigma(w_k) = z$. But according to definition of Pr_M, this happens only when $i_k = 0$.

■

Theorem 2.17 *1.* $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$ *iff there is a T-run (s_0, ξ) of Pr_M in which (q, n_1, n_2) is represented.*

2. A final configuration is reachable in M iff there is a leaky T-run of Pr_M.

Proof:

1. There are two directions to the proof.

($\Rightarrow$) : Let m be the length of the derivation $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$. We prove the result by induction on m. The base case is when $m = 0$ in which case $q = q_0$ and $n_1 = n_2 = 0$. Then the run $(s_0, (\eta_0, 1))$ satisfies the statementof the theorem, where $s_0 \in \mathsf{Init}(\mathsf{Pr}_M)$.

Suppose $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2) \xrightarrow{t} (q', n_1', n_2')$. By the induction hypothesis there is a T-run $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$ in which (q, n_1, n_2) is represented. Let $\mathit{infstate}(s_0, \xi)$ be denoted by s. Let $t = (q, i_1, i_2, q', j_1, j_2)$ and $\eta_t = a \cdot a'$. By lemma 2.16, there is a substitution σ suitable for Pr_M and η_t such that $\sigma(a)$ is enabled at s, $\sigma(a')$ is enabled at $\mathit{update}(s, \sigma(a))$, and $\sigma(\mathsf{outctr}_k(\eta_t))$ represents $n_k + j_k$ in $\mathit{update}(s, \sigma(\eta_t))$. Letting $e = (\sigma(\eta_t), 1)$ and $e' = (\sigma(\eta_t), 2)$ it is easy to see that $(s_0, \xi \cdot e \cdot e') \in \mathcal{R}(\mathsf{Pr}_M)$. Further, since $[q', \sigma(w_1'), \sigma(w_2')] \in \mathit{infstate}_I(s_0, \xi \cdot e \cdot e')$, it is clear that (q', n_1', n_2') is represented in $(s_0, \xi \cdot e \cdot e')$. Further $(s_0, \xi \cdot e \cdot e')$ is a T-run as well, from the induction hypothesis and using the fact that e and e' do not generate any new nonces and appealing to Lemma 2.14.

($\Leftarrow$) : We prove by induction on $|\xi|$ that for all $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, if (q, n_1, n_2) is represented in (s_0, ξ) then $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$.

The base case is when $|\xi| = 0$ and then the statement is vacuously true since no configuration is represented in (s_0, ξ).

Suppose (q', n_1', n_2') is represented in $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$, where $\xi = \xi'' \cdot e'$. If (q', n_1', n_2') is already represented in (s_0, ξ'') then by induction hypothesis it follows that $(q_0, 0, 0) \xrightarrow{*} (q', n_1', n_2')$. Otherwise there are two cases to consider:

$e' = (\eta_0, 1)$**:** Then it is clear that $(q', n_1', n_2') = (q_0, 0, 0)$ and hence the (q', n_1', n_2') is vacuously reachable from $(q_0, 0, 0)$.

$e' = (\sigma(\eta_t), 2)$ **for some** $t \in \delta$**:** Let $t = (q, i_1, i_2, q', j_1, j_2)$ and let $\eta_t = a \cdot a'$. It is clear that $\sigma(\mathsf{outctr}_k(\eta_t))$ represents n_k' for $k = 1, 2$. Further for $k = 1, 2$, $n_k' = n_k + j_k$ where $\sigma(\mathsf{inctr}_k(\eta_t))$ represents n_k in $\mathit{infstate}(s_0, \xi'')$. Further there is a proper prefix ξ of ξ'' such that (q, n_1, n_2) is represented in $\mathit{infstate}(s_0, \xi)$, and $\sigma(\eta_t)$ is enabled at $\mathit{infstate}(s_0, \xi)$. By induction hypothesis we have that (q, n_1, n_2) is a reachable configuration and by lemma 2.16, we know that t is enabled at (q, n_1, n_2). Therefore $(q, n_1, n_2) \xrightarrow{t} (q', n_1 + j_1, n_2 + j_2) = (q', n_1', n_2')$. Thus (q', n_1', n_2') is also a reachable configuration.

2. There are two directions to the proof.

($\Rightarrow$) : Suppose a final configuration (f, n_1, n_2) is reachable in M. Then there is a T-run $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$ representing (f, n_1, n_2). Thus $[\underline{f}, \underline{n_1}, \underline{n_2}] \in \mathit{infstate}_I(s_0, \xi)$, and hence $\sigma(\eta_f)$ is enabled at (s_0, ξ), where σ satisfies the property that it is identity on both x and z, and further $\sigma(w_k) = \underline{n_k}$ for $k = 1, 2$. For each $f \in F$ and $1 \leq i \leq 3$, let $e_i^f = (\sigma(\eta_f), i)$. If $e_1^f \cdot e_2^f$ is not enabled at (s_0, ξ) for the chosen f, it must be because some $e_2^{f'}$ occurs in $\mathit{Events}(\xi)$ for some $f' \in F$, generating the same x as a new nonce. But then $(s_0, \xi \cdot e_3^{f'}) \in \mathcal{R}(\mathsf{Pr}_M)$ and is patently leaky. If $e_1^f \cdot e_2^f$ is enabled at (s_0, ξ) then clearly $(s_0, \xi \cdot e_1^f \cdot e_2^f \cdot e_3^f) \in \mathcal{R}(\mathsf{Pr}_M)$, and this is also leaky. This run is also a T-run by definition.

($\Leftarrow$) : Suppose there is a leaky run $(s_0, \xi) \in \mathcal{R}(\mathsf{Pr}_M)$. According to the definition of Pr_M, this means that some instance of η_f for $f \in F$ has been played out as part of ξ. But this means that some configuration of the form (f, n_1, n_2) is represented in (s_0, ξ) which implies that a final configuration is reachable in M.

■

The main conclusion of this section is stated below.

Theorem 2.18

The secrecy problem for T-runs is undecidable.

Proof: This immediately follows from item 2 of Theorem 2.17 and the fact that the reachability problem for two-counter machines is undecidable. ■

References

[ALV01] Roberto M. Amadio, Denis Lugiez, and Vincent Vanackère. On the symbolic reduction of processes with cryptographic functions. Technical Report 4147, INRIA, March 2001.

[CDL+99] Iliano Cervesato, Nancy A. Durgin, Patrick D. Lincoln, John C. Mitchell, and Andre Scedrov. A Meta-notation for Protocol Analysis. In P. Syverson, editor, *Proceedings of the 12th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, 1999.

[DLMS99] Nancy A. Durgin, Patrick D. Lincoln, John C. Mitchell, and Andre Scedrov. The undecidability of bounded security protocols. In *Proceedings of the Workshop on Formal Methods and Security Protocols (FMSP'99)*, 1999.

[HT96] Nevin Heintze and Doug Tygar. A model for secure protocols and their composition. *IEEE Transactions on Software Engineering*, 22:16–30, 1996.

[NS78] Roger M. Needham and Michael D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Communications of the ACM*, 21(12):993–999, 1978.

[Pau98] Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of computer security*, 6:85–128, 1998.