Undecidability of the Secrecy Problem

We show that the secrecy problem for security protocols is in general undecidable, even when only considering well-typed runs. We use a reduction from the reachability problem for two-counter machines.

1 Two-counter machines

A two-counter machine (2CM) is a tuple $M = (Q, F, q_0, \delta)$ where Q is a set of states, $F \subseteq Q$ is the set of final states, and $q_0 \in Q$ is the initial state. The transition relation is given by $\delta \subseteq Q \times \{0, 1\}^2 \times Q \times \{-1, 0, 1\}^2$, where the values denote, in order, the current state, the status of each counter (zero/non-zero), the next state, and the action to be performed on each counter. For any $(q, i_1, i_2, q', j_1, j_2) \in \delta$, we enforce the condition that $j_k = -1$ implies $i_k = 1$ (for $k = 1, 2$). This allows us to decrement only a positive counter.

A *configuration* of such a machine M is a triple (q, n_1, n_2) with $q \in Q$ being the current state and $n_k \in \mathbb{N}$ the two counters. A transition $t = (q, i_1, i_2, q', j_1, j_2)$ is enabled at a configuration $c = (q, n_1, n_2)$ iff $i_k = 0$ iff $n_k = 0$ for $k = 1, 2$. (We will now skip the "$k = 1, 2$" to increase readability, as our results apply equally to both counters.) If t is enabled at c, we have $(q, n_1, n_2) \xrightarrow{t} (q', n_1 + j_1, n_2 + j_2)$.

The initial configuration is $c_0 = (q_0, 0, 0)$. A configuration c is reachable if $c_0 \xrightarrow{*} c$. A configuration (q, n_1, n_2) is final if $q \in F$. The reachability problem for 2CMs is to determine whether, for a given machine M, a final configuration of M is reachable. This problem is known to be undecidable [1, 2]. For a given 2CM $M = (Q, F, q_0, \delta)$, we will define a corresponding protocol **Pr** such that a final configuration is reachable in M iff $\mathbf{Pr}_M$ admits a "leaky run", i.e. a run at the end of which the intruder can derive a designated "secret".

2 Well-typed runs: Undecidability

Assume a set of fresh names $\mathscr{N}$ such that $Q \subseteq \mathscr{N}$. Consider two fixed names z and d. Also consider two fixed agents A and B who share the key $k_{ab} \in \mathscr{K}$. For $u, u' \in \mathscr{N}$ and $q \in Q$, we define:

$$[q, u, u'] \overset{\text{def}}{=} \mathbf{senc}((q, (u, u')), k_{ab}) \quad \text{and} \quad [u, u'] \overset{\text{def}}{=} \mathbf{senc}((u, u'), k_{ab})$$

We now define $\mathbf{Pr}_M$ as the protocol consisting of constants $\mathscr{C}$ and roles $\mathscr{R}$ as follows:

- $\mathscr{C} = Q \cup \{A, B, z, d, s\}$ (with s the designated secret that will be leaked if M admits a run from the initial configuration to a final one)
- $\mathscr{R} = \{\eta_0\} \cup \{\eta_t \mid t \in \delta\} \cup \{\eta_f \mid f \in F\}$ is the set of roles of the protocol, which we will define in detail presently.

Informally, we try to set up the roles such that by following this sequence of roles that mirror the transitions of M, the intruder gets a chain of values of the form $[q, u, u']$, starting from $[q_0, 0, 0]$. (The name z uniquely stands for the counter value zero.) This chain is such that if ever a value of the form $[f, u, v]$ (for some $f \in F$) is received by the intruder, the final role immediately also sends the designated secret s in the open to the intruder, and so we have a breach of secrecy.

The roles are defined as follows. The initial role outputs a coding of the initial configuration to the intruder, along with some other information. (d is a 'dummy' name that we will use to code up the initial configuration. We will shortly illustrate how these names are used.)

$$\eta_0 \overset{\text{def}}{=} A!B : [d, d], [q_0, z, z], [d, d]$$

For every transition $t = (q, i_1, i_2, q', j_1, j_2) \in \delta$, we define $\eta_t \overset{\text{def}}{=} a \cdot a'$ with

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

where $N = \{v'_k \mid k \in \{1, 2\} \text{ and } j_k = 1\}$ is the set of new names generated by A for this send action.

The code for the configuration looks like $[q, w_1, w_2]$, where $w_1, w_2 \in \mathscr{N}$ are names that *represent* the values of the two counters, and q is the current state of the 2CM. We say that a name u represents a natural number n if there is a 'chain' of length n of the form $[z, u_1], [u_1, u_2], \ldots, [u_{n-1}, u]$, where all these terms are in the intruder's database, and all the names are distinct.

Informally, we wish to receive a code for the current configuration from the intruder (along with some bookkeeping information), make the changes necessitated by the transition t in the 2CM, and then send a code for this new configuration to the intruder (potentially along with some new changed information). Essentially, the bookkeeping information is there only to indicate whether some name codes up a non-zero integer or not, and we will now describe via examples how this information is used.

Consider a decrement action first. A decrement of a counter is allowed only if the counter is positive. Therefore, if $j_k = -1$ for some $k \in \{1, 2\}$ in t, there better be some way for us to check that i_k is indeed non-zero in the current configuration, and then decrement the value of counter k. In particular, we can use our notion of a name coding up a counter value in order to do this check. If one of the counters (say counter 1) has a non-zero value (say r, coded up by the name u), we expect the intruder to have a chain starting from $[z, u_1]$ all the way to $[u_{r-1}, u]$. So if A receives a term of the form $[u_{r-1}, u]$, along with the term $[q, u, w_2]$, A is convinced that u is indeed non-zero (since the intruder managed to derive this $[u_{r-1}, u]$ term, presumably from the chain all the way to u that he has access to), and can therefore execute a decrement action on counter 1. Having decremented, the new term will now mention u_{r-1} as the new value for counter 1.

On the other hand, for an increment action, we wish to extend the chain in the intruder's database by the new pair of the form $[u, u']$, where u coded up the old counter value, and u' codes up the new value (which is the old value incremented by 1). So our bookkeeping technique in this case would mandate that A picks a new name u' (which now becomes part of the set N of new names generated for that action), and constructs such a pair $[u, u']$ to be sent as part of the send action a' in the role corresponding to this transition t, in addition to the new configuration which mentions u' as the new value for the incremented counter.

Keeping all this in mind, we show here a few illustrative examples for how the roles for various transitions t look. We also provide a generalization from these cases later. Suppose $t = (q, 0, 1, q', 0, 0)$. This transition is enabled when the first counter is zero and the second is not. There is no change to the counter values. So, as part of the receive action, A receives a term coding up this configuration, and the last part of the chain that certifies that counter 2 is non-zero. A sends out only a term representing the configuration. So the role is $a \cdot a'$, where:

$$a = A?B : [q, z, v_2], [u_2, v_2] \quad \text{and} \quad a' = A!B : [q', z, v_2]$$

Now suppose $t = (q, 0, 1, q', 1, 1)$. The first counter is zero, while the second is not, and both counters are being incremented. The receive action looks the same as for the previous example. However, for the send action, we now need to increment both counters, and extend the respective chains with the new counter values. The role is $a \cdot a'$, where:

$$a = A?B : [q, z, v_2], [u_2, v_2] \quad \text{and} \quad a' = A!B : (\{v_1', v_2'\})[z, v_1'], [q', v_1', v_2'], [v_2, v_2']$$

Now suppose $t = (q, 1, 1, q', 0, -1)$. Both counters are positive, and only the second counter is decremented. As proof of the positivity of the counters, A receives a configuration term and two chain terms. The new counter value for counter 2 comes from the chain received for counter 2, while counter 1's value remains unchanged. No chains need to be extended, so only one term representing the new configuration is sent out. The role is $a \cdot a'$, where:

$$a = A?B : [u_1, v_1], [q, v_1, v_2], [u_2, v_2] \quad \text{and} \quad a' = A!B : [q', v_1, u_2]$$

Having seen all these examples, we are now ready to provide the general description for a role marking a transition in the 2CM. Since we are fitting roles into a general template involving a configuration term and two chain terms, for examples of the previous sort where there is only one term sent out, the chain terms are taken to be $[d, d]$ by fiat. Sending this term to the intruder does not change anything since η_0 has anyway sent this term to the intruder.

For every transition $t = (q, i_1, i_2, q', j_1, j_2) \in \delta$, we define $\eta_t \overset{\text{def}}{=} a \cdot a'$ with

$$a = A?B : [u_1, v_1], [q, w_1, w_2], [u_2, v_2] \quad \text{and} \quad a' = A!B : (\mathrm{N})[u_1', v_1'], [q', w_1', w_2'], [u_2', v_2']$$

where $N = \{v'_k \mid k \in \{1,2\} \text{ and } j_k = 1\}$, and the following conditions hold for each pair $[i,j]$: (we omit the subscript k everywhere, these apply equally for either value of k)

1. $[0,0]$: $w = w' = z$ and $u = v = u' = v' = d$
2. $[0,1]$: $u' = w = z$ and $u = v = d$ and $v' = w' \notin \mathscr{C}$
3. $[1,0]$: $w' = w = v$ and $u' = v' = d$ and $u \neq v \notin \mathscr{C}$
4. $[1,1]$: $w = v = u'$ and $w' = v'$ and $u \neq v \neq v' \notin \mathscr{C}$
5. $[1,-1]$: $w = v$ and $w' = u$ and $u' = v' = d$ and $u \neq v \notin \mathscr{C}$

For each final state $f \in F$, $\eta_f \overset{\text{def}}{=} a \cdot a'$ where

$$a = A?B : [f, w_1, w_2] \quad \text{and} \quad a' = A!B : s \quad \text{and} \quad w_1 \neq w_2 \notin \mathscr{C}.$$

Having set all this machinery up, we now need to prove that this protocol does indeed admit a breach of secrecy if and only if a final configuration is reachable in M. This requires a few lemmas. Firstly, observe that the intruder never gets to know the key shared between A and B.

Obs 1. *Suppose ρ is a run of $\mathbf{Pr}_M$ and $ks = (X_A)_{A\in\mathscr{A}}$ is the knowledge state at the end of ρ. Then, $X_I \nvdash k_{ab}$.*

We now make the idea of coding up state-specific – a name u represents a number n in a state ks if there is a chain of length n of the form $[z,u_1],[u_1,u_2],\ldots,[u_{n-1},u]$ such that each of these terms belong to X_I in ks. We can lift this notion to representing in a run ρ if the ks considered is the global knowledge state at the end of ρ.

Using these, we now get to the crucial lemmas that show that the role η_t faithfully simulates the transition t in M. We will use the following notation: $t = (q, i_1, i_2, q', j_1, j_2)$ and $\eta_t = a \cdot a'$, where $a = A?B : [u_1,v_1],[q,w_1,w_2],[u_2,v_2]$ and $a' = A!B : (N)[u'_1,v'_1],[q',w'_1,w'_2],[u'_2,v'_2]$.

Suppose ρ is a run of $\mathbf{Pr}_M$ and ks the global knowledge state at the end of ρ. Also suppose that t is a transition of M, $\eta_t = a \cdot a'$, and (q,n_1,n_2) a configuration of M represented in ks.

Lemma 2. *If t is enabled at (q,n_1,n_2), then there is a well-typed substitution σ suitable for η_t in $\mathbf{Pr}_M$ s.t:*

- *$\sigma(w_k)$ represents n_k in ks*
- *$\sigma(a)$ is enabled at ks and $\sigma(a')$ at the state ks' obtained by the transition $(ks, \sigma(a))$*
- *$\sigma(w'_k)$ represents $n_k + j_k$ in ks'', the state obtained by the transition $(ks, \sigma(a \cdot a'))$.*

Proof.

Suppose t is enabled at (q,n_1,n_2). Then $i_k = 0$ iff $n_k = 0$. Also suppose that the name r_k represents n_k in ks. We now define a σ suitable for $\mathbf{Pr}_M$ and η_t as follows.

- $\sigma(w_k) = r_k$, and σ is identity on $\mathscr{C}$
- For each fresh $m \in \mathrm{N}$ generated for use in the action involved in t, $\sigma(m)$ is a distinct name not appearing in ks.
- If $i_k = 1$ then $\sigma(u_k) = r'_k$, where r'_k represents $n_k - 1$ in ks. (Since $i_k = 1$ and t is enabled at the current configuration, $n_k = 1$. Combined with the fact that r_k represents n_k in ks, there is at least one such r'_k that we can pick, from the intruder's chain.)

It is evident that σ is well-typed and suitable for η_t. We merely need to show that it satisfies the three properties required by the lemma. The first follows immediately by the definition, since r_k represents n_k at ks.

We now show that $\sigma(a)$ is enabled at ks. Note that (q, n_1, n_2) is represented in ks by $[q, r_1, r_2]$, and therefore $X_I \vdash [q, r_1, r_2]$. Now, we just need to show that $X_I \vdash \sigma([u_k, v_k])$. Two cases arise.

- $i_k = 0$: In this case, $u_k = v_k = d$. Now, any run in $\mathbf{Pr}_M$ starts with the role η_0, and this role only has one send action, which sends out $[d, d]$ to the network. Thus, $X_I \vdash [d, d]$.
- $i_k = 1$: In this case, $v_k = w_k$. Since $\sigma(u_k) = r'_k$ (by the definition of σ), we have that $\sigma([u_k, v_k]) = [r'_k, r_k]$, and since $\sigma(w_k) = r_k$ (and w_k is non-zero), $X_I \vdash [r'_k, r_k]$ and we are done.

Showing that $\sigma(a')$ is enabled at ks' is easier. Note that the term sent in a' is derivable using N, $k_{k_{ab}}$ and the term received in action a, for all u, v and w values – all of which belong to ks'. Thus, $X'_A \vdash [u'_1, v'_1], [q', w'_1, w'_2], [u'_2, v'_2]$, which is the only criterion for $\sigma(a')$ being enabled at ks'.

We now show that $\sigma(w'_k)$ represents $n_k + j_k$ in ks''. Three cases arise.

- $j_k = 0$: Simplest case, $w_k = w'_k$ and done.
- $j_k = -1$: By definition, $\sigma(u_k)$ represents $n_k - 1$, and $w'_k = u_k$ according to our role description. Since the intruder database grows monotonically, $\sigma(u_k) = \sigma(w'_k)$ continues to represent $n_k - 1$ in ks''.
- $j_k = 1$: According to our role description, $w_k = u'_k$ and $w'_k = v'_k$. Also note that by the update resulting from executing a' at ks', in the resultant state ks'', the intruder has $X''_I \vdash [\sigma(u'_k), \sigma(v'_k)]$. Since $\sigma(w_k) = \sigma(u'_k)$ continues to represent n_k in ks'' (by monotonicity), and $[\sigma(u'_k), \sigma(v'_k)]$ extends the intruder's 'chain' by one, we have that $\sigma(w'_k) = \sigma(v'_k)$ represents $n_k + 1$ in ks''. ■

Lemma 3. *If there is a substitution σ suitable for η_t in $\mathbf{Pr}_M$ such that $\sigma(w_k)$ represents n_k in ks and $\sigma(a)$ is enabled at ks, then t is enabled at (q, n_1, n_2).*

Proof. Suppose σ is such a suitable substitution. In order to show that t is enabled at (q, n_1, n_2), we need to show that $i_k = 0$ iff $n_k = 0$. Two cases arise, one where $i_k = 0$ and one where $i_k = 1$.

- $i_k = 0$: By the role description, $w_k = z$, and hence $\sigma(w_k) = z$. Since z uniquely represents only 0 in any state, and we know that $\sigma(w_k)$ represents n_k, $n_k = 0$.
- $i_k = 1$: By the role description, $w_k = v_k$ and $u_k \neq v_k$.

  L3.1 From a being enabled at ks, we know that $X_I \vdash [q, \sigma(w_1), \sigma(w_2)]$. Now, for any term of the form $[q, t, t']$ derivable by the intruder, $t, t' \neq d$ (from our role description).

  L3.2 The only term of the form $[t, t]$ derivable by the intruder is $[d, d]$ (since such a term only gets into the intruder's database at the initial role – every other role extends the chain by a different name).

  L3.3 For any term of the form $[t, t']$ derivable by the intruder where $t \neq t'$, we have $t' \neq z$ (a chain cannot be extended with zero, since all names along a chain are distinct, and every chain begins with zero).

  From L3.1, we know that $\sigma(w_k) = \sigma(v_k) \neq d$. Since $X_I \vdash [\sigma(u_k), \sigma(v_k)]$ (by the enabledness of a at ks), this fact combined with L3.2 gives us $\sigma(u_k) \neq \sigma(v_k)$. Finally, by applying L3.3 to this fact, we have $\sigma(v_k) \neq z$. However, since $\sigma(w_k) = \sigma_{v_k}$, we have that $\sigma(w_k) \neq z$, and since z uniquely represents 0, we have that $n_k \neq 0$, and this concludes the proof. ∎

Theorem 4. $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$ *iff* (q, n_1, n_2) *is represented in some run of* $\mathbf{Pr}_M$ *iff it is represented in some well-typed run of* $\mathbf{Pr}_M$.

Proof. We prove this in two parts.

T4.1 First we prove that if $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$, then there is a well-typed run of $\mathbf{Pr}_M$ which represents (q, n_1, n_2). This proof is by induction on the length of the derivation $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$. The base case is when this length is 0, in which case $q = q_0$ and $n_k = 0$. Then the run $(\eta_0, \sigma, 1)$ gives us the result, for any well-typed substitution σ which is identity on $\mathscr{C}$.

In the case where this length is non-zero, suppose $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2) \xrightarrow{t} (q', n'_1, n'_2)$. By IH, there is a run ρ of $\mathbf{Pr}_M$ which represents (q, n_1, n_2). Let ks be the final state of ρ. There is a well-typed substitution σ suitable for η_t in $\mathbf{Pr}_M$ which obeys the conditions in Lemma 2. In particular, $\sigma(w'_k)$ represents $n_k + j_k = n'_k$ in the state obtained by the transition $(ks, \sigma(\eta_t))$. It is not hard to see that we can extend ρ with $\sigma(a) \cdot \sigma(a')$ to get a well-typed run of $\mathbf{Pr}_M$. Since $\sigma(w'_k)$ represents n'_k at the end of this run, this new well-typed run represents (q, n'_1, n'_2).

T4.2 We show that if there is a run ρ of $\mathbf{Pr}_M$ in which (q, n_1, n_2) is represented, then $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$. This proof is by induction on the length of ρ. For the base case, with a run of length 0, the statement is vacuously true.

In the case where the run has positive length, suppose (q', n'_1, n'_2) is represented by $[q', w'_1, w'_2]$ in a run $\rho = \rho' \cdot e$ (where e is a final action) of $\mathbf{Pr}_M$. Let ks and ks' be the final states at the ends of ρ and ρ' respectively. If (q', n'_1, n'_2) is already represented in ρ', then by IH, we get $(q_0, 0, 0) \xrightarrow{*} (q', n'_1, n'_2)$. Otherwise, $X'_I \vdash [q', w'_1, w'_2]$ in ks, but $X''_I \nvdash [q', w'_1, w'_2]$ in ks'.

Note that since I does not have access to k_{ab}, and a term of the form $[q', w'_1, w'_2]$ cannot be obtained by applying elimination rules to any term I already had access to in ks', it must be the case that this term was added to database due to the update caused by the event e. Therefore, in particular, e is a send event (since receive events do not cause changes to the intruder's database), and $[q', w'_1, w'_2]$ is a term obtained by applying some elimination rule(s) to the terms occurring in e. Hence, e can be of two kinds:

- $e = (\eta_{I0}, \sigma, 1)$: The only such term sent out by e is $[q_0, z, z]$, so $(q', n'_1, n'_2) = (q_0, 0, 0)$.
- $e = (\eta_{It}, \sigma, 2)$ for some $t \in \delta$: Note that in this case, the last event of ρ' must be $(\eta_{It}, \sigma, 1)$. It is clear that the knowledge state in ρ' just before the event $(\eta_{It}, \sigma, 1)$ represents the configuration (q, n_1, n_2) (where the n' values in (q', n'_1, n'_2) are such that $n'_k = n_k + j_k$). Since this is a proper prefix of ρ' and also a valid run of $\mathbf{Pr}_M$, by IH, we know that $(q_0, 0, 0) \xrightarrow{*} (q, n_1, n_2)$. By Lemma 3, we know that t is enabled at (q, n_1, n_2), and thus we can extend the path in the 2CM to get $(q_0, 0, 0) \xrightarrow{*} (q', n'_1, n'_2)$. ■

Theorem 5. *A final configuration is reachable in M iff there is a leaky run of $\mathbf{Pr}_M$ iff there is a well-typed leaky run of $\mathbf{Pr}_M$.*

Proof. We first prove that if a final configuration is reachable in M then there is a well-typed leaky run of $\mathbf{Pr}_M$. Suppose $(q_0, 0, 0) \xrightarrow{*} (f, n_1, n_2)$ for $f \in F$. Then, by Lemma 4, there is a well-typed run ρ of $\mathbf{Pr}_M$ representing (f, n_1, n_2). If ks is the knowledge state at the end of ρ, we have that $X_I \vdash [f, r_1, r_2]$ for names r_1, r_2. Hence the sequence of events $e_1 \cdot e_2$ is enabled at ks, where $e_i = (\eta_{If}, \sigma, i)$ for $i = 1, 2$, and some well-typed σ such that $\sigma(x) \neq \sigma(y)$ for all $y \neq x$. It then follows that $\rho \cdot e_1 \cdot e_2$ is also a well-typed but leaky run of $\mathbf{Pr}_M$.

We now prove that if there is a leaky run of $\mathbf{Pr}_M$ then a final configuration is reachable in M. Suppose there is such a leaky run ρ. Such a run would include an instance of η_{If} for some $f \in F$. However, this requires some configuration of the form (f, n_1, n_2) to be represented by a term in ρ, and by Lemma 4, we have that (f, n_1, n_2) is reachable in M. ■

Thus, we get that the secrecy problem is undecidable.

References

[1] Wikipedia. Two counter machines are Turing-equivalent. `https://tinyurl.com/und2cm`

[2] Hopcroft, J. E. and Ullman, J. D. Introduction to Automata Theory, Languages, and Computation. 1979. ISBN 81-85015-96-1. (Section 7.8)