Exercice 60
Assume that the encryption scheme is IND-CPA and that $u, v \in \mathcal{M}_0$ are such that $l(u,\eta) = l(v,\eta)$ and, for any name n not occuring in u, v, $[\![\langle u, n\rangle]\!]_\eta \approx [\![\langle u, k\rangle]\!]_\eta$ and $[\![\langle v, n\rangle]\!]_\eta \approx [\![\langle v, k\rangle]\!]_\eta$. Prove then $[\![\{u\}_k^r]\!]_\eta \approx [\![\{v\}_k^r]\!]_\eta$ for any name r not occurring in u, v, k.

Give an example of such u, v that do contain occurrences of k.

13.6 Proof of soundness of static equivalence: a special case

As we saw above, we need some assumptions on the sequence of terms, ruling out situations such as a key encrypting itself.

Given a sequence of terms $s_1, \ldots, s_n$, we define the relation $k >_{s_1,\ldots,s_n} k'$ between names, as the least transitive relation such that:

If k_1 occurs in u and there is an index i and a subterm $\{u\}_{k_2}^r$ of s_i, then $k_2 >_{s_1,\ldots,s_n} k_1$

A *random seed* is a name that is used as a third argument of an encryption symbol.

Definition 13.6 *A* valid frame *(resp.* valid term sequence*) is a frame (resp. term sequence) $\nu\overline{n}.\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}$ (resp. $s_1, \ldots, s_n$), such that*

1. $s_1, \ldots, s_n \in \mathcal{M}_0$
2. *$\overline{n}$ is the set of names occurring in $s_1, \ldots, s_n$*
3. *$\geq_{s_1,\ldots,s_n}$ is an ordering.*
4. *each random seed is used only once in $s_1, \ldots, s_n$*

Example 13.1 *The following are not valid term sequences*

1. $\{k\}_k^r$
2. $\{\{k_1\}_{k_2}^{r_1}\}_{k_3}^{r_2}, \{\{k_2\}_{k_1}^{r_3}\}_{k_3}^{r_4}$
3. $\{\{k_1\}_{k_2}^{r_1}\}_{k_3}^{r_2}, \{\{k_1\}_{k_3}^{r_3}\}_{k_2}^{r_4}$
4. $\{\{k_1\}_{k_2}^{r_1}\}_{k_2}^{r_2}, \{\{k_2\}_{k_3}^{r_3}\}_{k_3}^{r_4}$

The main restriction imposed by the first condition is the use of names as keys: only atomic keys are considered.

The second condition is not a restriction: we may always bind all names and disclose explicitly the names that are supposed to be available to the attacker.

The third condition is a real (strong) restriction. Some restriction that rules out key cycles is necessary (with the current state of the art). The above condition rules out terms $\{\{u\}_k^{r_1}\}_k^{r_2}$ in the frames. We will see in the section 13.7, that the condition can be slightly relaxed, allowing for such terms in the frames.

The last condition forbids several occurrences of the same ciphertext in the frames. This condition will also be relaxed in the section 13.7.

Theorem 13.1 *Let $\nu\overline{n}.\{x_1 \mapsto s_1, \ldots x_n \mapsto s_n\}$ and $\nu\overline{n'}.\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ be two valid frames.*

$$\nu\overline{n}.\{x_1 \mapsto s_1, \ldots x_n \mapsto s_n\} \sim \nu\overline{n'}.\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \quad \Rightarrow \quad [\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$$

In other words, if the two frames are statically equivalent, then they are computationally indistinguishable.

In what follows, we drop the name binders and the variables, since all names are bound and the variable symbols can be inferred from the context. Furthermore, if P is a predicate symbol and u, v are valid recipes (terms that do not use any names in our case), we write $s_1, \ldots, s_n \models P(u, v)$ insteand of $(u\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}, v\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}) \in P^I$.

Proof : We use an induction on $|s_1| + \cdots + |s_n| + |t_1| + \cdots |t_n|$ where $|s|$ is the number of function symbols and names appearing in s (we do not count the constants). In each case, we leave to the reader the verification that the induction hypothesis is applied to valid sequences indeed.

Base case: $s_1, \ldots, s_n$ and $t_1, \ldots, t_n$ are constants. Since there is no names in the frames, s_i is a valid recipe: $s_1, \ldots, s_n \models EQ(x_i, s_i)$ and therefore $t_1, \ldots, t_n \models EQ(x_i, s_i)$. It follows that, for every i, $s_i = t_i$, hence the indistinguishability of the two sequences.

Induction case: We successively investigate cases. In each case, we assume that the previous cases do not apply.

1. If one of the two sequences contains a pair.
   w.l.o.g., assume $s_1 = \langle s_{11}, s_{12} \rangle$. Then $s_1, \ldots, s_n \models M(\pi_1(x_1))$, hence $t_1, \ldots, t_n \models M(\pi_1(x_1))$. This implies that there are terms $t_{11}, t_{12} \in \mathcal{M}_0$ such that $t_1 = \mathsf{pair} t_{11} t_{12}$.
   Then $s_{11}, s_{12}, s_2, \ldots, s_n \sim t_{11}, t_{12}, t_2, \ldots, t_n$:

$$\begin{array}{rll} s_{11}, s_{12}, s_2, \ldots, s_n \models P(u, v) & \text{iff} & s_1, s_2, \ldots, s_n \models P(u, v)\{x_{11} \mapsto \pi_1(x_1), x_{12} \mapsto \pi_2(x_1)\} \\ & \text{iff} & t_1, t_2, \ldots, t_n \models P(u, v)\{x_{11} \mapsto \pi_1(x_1), x_{12} \mapsto \pi_2(x_1)\} \\ & \text{iff} & t_{11}, t_{12}, t_2, \ldots, t_n \models P(u, v) \end{array}$$

   By induction hypothesis, $[\![s_{11}, s_{12}, s_2, \ldots, s_n]\!]_\eta \approx [\![t_{11}, t_{12}, t_2, \ldots, t_n]\!]$.
   We use then the following exercise:

   Exercice 61
   Let f be a function symbol, whose computational interpretation is a PPT function $[\![f]\!]$. Assume $[\![s_1, \ldots s_p, s_{p+1} \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_p, t_{p+1}, \ldots, t_n]\!]_\eta$ and $r \notin fn(s_1, \ldots, s_n, t_1, \ldots, t_n)$. Prove that $[\![f(s_1, \ldots, s_p \mid r), s_{p+1}, \ldots, s_n]\!]_\eta \approx [\![f(t_1, \ldots, t_p \mid r), t_{p+1}, \ldots, t_n]\!]_\eta$ (r is assumed to be drawn according to a polynomial distribution).

   With the pairing function for f and conclude that $[\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$.

2. If $s_i = s_j$(resp. $t_i = t_j$) for some $i \neq j$ and s_i is not a constant. Then $s_1, \ldots, s_n \models EQ(x_i, x_j)$, hence $t_1, \ldots, t_n \models EQ(x_i, x_j)$, which implies $t_i = t_j$. Then $s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \sim t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ and, by induction hypothesis, $[\![s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n]\!]_\eta$, hence the result.

3. If some $s_i = \{u_i\}_k^{r_i}$ (or some t_i; this case is symmetric) where u_i is not a constant and k is maximal w.r.t. $\geq_{s_1, \ldots, s_n}$ and $k \notin \{s_1, \ldots, s_n\}$. After possibly renumbering the terms, let $s_1 = \{u_1\}_k^{r_1}, \ldots s_p = \{u_p\}_k^{r_p}$ and $s_{p+1}, \ldots, s_n$ are not encryptions with k.

   k does not occur in $u_1, \ldots, u_p, s_{p+1}, \ldots, s_n$ since every s_i is either a name (different from k), a constant, or a ciphertext (thanks to step 1) and, in the latter case, k does not occur in s_i by maximality of k.

   Let $\sigma = \{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}$ and $\sigma' = \{x_1 \mapsto s'_1, \ldots, x_n \mapsto s'_n\}$ and let ρ be the replacement of $s_1, \ldots, s_p$ with $\{0^{l(u_1, \eta)}\}_k^{r_1}, \ldots, \{0^{l(u_p, \eta)}\}_k^{r_p}$ respectively. We observe first that, for any recipe u, $\rho(u\sigma\downarrow) = u\sigma'\downarrow$ by consistency of use of the random seeds.

If $s_1, \ldots, s_n \models P(u, v)$, $(u\sigma\downarrow, v\sigma\downarrow) \in P^I$. For $P \in \{M, EQ, EK, EL\}$, $(u_1, u_2) \in P^I$ iff $(\rho(u_1), \rho(u_2)) \in P^I$ (for instance, when $P = EL$, this is thanks to the presevation of plaintexts lengths by ρ). Hence

$$\begin{array}{lll} s_1 \ldots, s_n \models P(u,v) & \text{iff} & (\rho(u\sigma\downarrow), \rho(v\sigma\downarrow)) \in P^I \\ & \text{iff} & (u\sigma'\downarrow, v\sigma'\downarrow) \in P^I \\ & \text{iff} & s'_1, \ldots, s'_n \models P(u,v) \end{array}$$

$s_1, \ldots, s_n \sim t_1, \ldots, t_n$ therefore implies $s'_1, \ldots, s'_n \sim t_1, \ldots, t_n$. Since, by assumption, at least one s_i is such that $s_i\{u_i\}_k^{r_i}$ where u_i is not a constant and $s'_i = \{0^{l(u_i,\eta)}\}_k^{r_i}$ has a sctritly smaller size, we may apply the induction hypothesis:

$$[\![s'_1, \ldots, s'_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$$

Then, thanks to our assumptions (validity of the sequences) and lemma 13.2, $[\![s_1, \ldots, s_n]\!]_\eta \approx [\![s'_1, \ldots, s'_n]\!]_\eta$.
We conclude $[\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]$.

4. If there are two indices i, j such that $s_i = \{u_i\}_{s_j}^{r_i}$. Assume w.l.o.g $i = 1$. $s_1, \ldots, s_n \models M(\mathsf{dec}(x_1, x_j))$ implies $t_1, \ldots, t_n \models M(\mathsf{dec}(x_1, x_j))$, hence $t_1 = \{v_1\}_{t_j}^{r'_1}$ for some v_1.
We claim that $u_1, s_2, \ldots, s_n \sim v_1.t_2, \ldots, t_n$:

$$\begin{array}{lll} u_1, s_2 \ldots s_n \models P(w_1, w_2) & \text{iff} & s_1, s_2 \ldots, s_n \models P(w_1\{x_1 \mapsto \mathsf{dec}(x_1, x_j)\}, w_2\{x_1 \mapsto \mathsf{dec}(x_1, x_j)\}) \\ & \text{iff} & t_1, \ldots, t_n \models P(w_1\{x_1 \mapsto \mathsf{dec}(x_1, x_j)\}, w_2\{x_1 \mapsto \mathsf{dec}(x_1, x_j)\}) \\ & \text{iff} & v_1, t_2, \ldots, t_n \models P(w_1, w_2) \end{array}$$

By induction hypothesis, $[\![u_1, s_2, \ldots, s_n]\!]_\eta \approx [\![v_1, t_2, \ldots, t_n]\!]_\eta$. By exercise 61 *and since we assumed that each random seed is used only once*, we conclude $[\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$. (We sill see in the next section how to modify this part, to avoid the assumption on the unique use of random seeds).

5. Now we have only to consider sequences $s_1, \ldots, s_n, t_1, \ldots, t_n$ that consist of encryptions of constants, names that are not used as encryption keys, and constants. If one of the sequence contains at least one ciphertext, assume w.l.o.g. that this is s_1: $s_1 = \{c_1\}_k^{r_1}$. Let $s_1, \ldots s_p$ be all ciphertexts in the sequence $s_1, \ldots, s_n$, whose encryption key is k: $\mathrm{v}s_i = \{c_i\}_k^{r_i}$ for $1 \leq i \leq p$.
For every $1 \leq i, j \leq p$, $s_1, \ldots, s_n \models EK(x_i, x_j)$, hence $t_1, \ldots, t_p \models EK(x_i, x_j)$: for $i = 1, ..., p, t_i = \{c'_i\}_{k'}^{r'_i}$ for some constants c'_i. Furthermore, thanks to EL, for every $i = 1, ..., p$, $l(c_i, \eta) = l(c'_i, \eta)$. It follows that $[\![s_1, \ldots, s_p]\!]_\eta \approx [\![t_1, \ldots, t_p]\!]_\eta$ by lemma 13.2.
The key k does not occur in the sequence $s_{p+1}, ldots, s_n$ and, by symmetry, the key k' does not occur in the sequence $t_{p+1}, \ldots, t_n$. Then, by consistency of use of random seeds,

$$[\![s_1, \ldots, s_n]\!]_\eta = [\![s_1, \ldots, s_p]\!]_\eta \times [\![s_{p+1}, \ldots, s_n]\!]_\eta$$

and

$$[\![t_1, \ldots, t_p]\!]_\eta \times [\![t_{p+1}, \ldots, t_n]\!]_\eta = [\![t_1, \ldots, t_n]\!]_\eta$$

By induction hypothesis, and since $s_{p+1}, \ldots, s_n \sim t_{p+1}, \ldots, t_n$, $[\![s_{p+1}, \ldots, s_n]\!]_\eta \approx [\![t_{p+1}, \ldots, t_n]\!]_\eta$. It follows that

$$[\![s_1, \ldots, s_n]\!]_\eta = [\![s_1, \ldots, s_p]\!]_\eta \times [\![s_{p+1}, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_p]\!]_\eta \times [\![t_{p+1}, \ldots, t_n]\!]_\eta = [\![t_1, \ldots, t_n]\!]_\eta$$

6. We are left to a case where $s_1, \ldots, s_n$, (and $t_1, \ldots t_n$) are sequences of distinct names and constants. If there is at least one name in the sequence, say s_1, then t_1 must also be a name (otherwise $EQ(x_1, t_1)$ would be satisfied in the second sequence and not in the first one). By induction hypothesis, $[\![s_2, \ldots, s_n]\!]_\eta \approx [\![t_2, \ldots, t_n]\!]_\eta$ and then

$$[\![s_1, \ldots, s_n]\!]_\eta = [\![s_1]\!]_\eta \times [\![s_2, \ldots, s_n]\!]_\eta \approx [\![t_1]\!]_\eta \times [\![t_2, \ldots, t_n]\!]_\eta = [\![t_1, \ldots, t_n]\!]_\eta$$

Exercice 62
Give an example showing that the above proof does not work when a random seed may occur twice in a valid frame.

13.7 The proof in the general case

In this section, we prove exactly the same result as in the previous section, however relaxing two assumptions.

First, we relax the condition on keys occurences, in order to allow for instance terms $\{\{u\}_k^r\}_k^{r'}$ in the sequence. We define the relation $\sqsubseteq$ on terms (read $s \sqsubseteq t$ as "s occurs as plaintext in t") as the least symmetric and transitive relation such that:

- If $u \sqsubseteq u_1$ or $u \sqsubseteq u_2$, then $u \sqsubseteq \langle u_1, u_2 \rangle$.
- If $u \sqsubseteq v$ then $u \sqsubseteq \{v\}_k^r$.

Now, $\gg_{s_1 \ldots, s_n}$ is redefined as the (more general) least transitive relation such that, $k_2 \gg_{s_1, \ldots, s_n} l_1$ whenever there is a subterm $\{u\}_{k_2}^r$ of some s_i such that $k_1 \sqsubseteq u$.

Definition 13.7 *A sequence of terms $s_1, \ldots, s_n$ in $\mathcal{M}_0$* has no key cycle *if $\gtrsim_{s_1, \ldots, s_n}$ is an ordering. Dually, if there is a name k such that $k \gg_{s_1, \ldots, s_n} k$, then $s_1, \ldots, s_n$* contains a key-cycle.

Key cycles are defined now according to this new ordering.

Example 13.2 *1. $\{k\}_k^r$ contains a key cycle*

2. $\{\{k_1\}_{k_2}^{r_1}\}_{k_3}^{r_2}, \{\{k_2\}_{k_1}^{r_3}\}_{k_3}^{r_4}$ contains a key-cycle

3. $\{\{k_1\}_{k_2}^{r_1}\}_{k_3}^{r_2}, \{\{k_1\}_{k_3}^{r_3}\}_{k_2}^{r_4}$ does not contains a key-cycle

4. $\{\{k_1\}_{k_2}^{r_1}\}_{k_2}^{r_2}, \{\{k_2\}_{k_3}^{r_3}\}_{k_3}^{r_4}$ does not contains a key-cycle

Second, we relax the conditions on random seeds, allowing several copies of the same ciphertext:

Definition 13.8 *A sequence $s_1, \ldots, s_n$ of terms* uses the random seeds in a consistent way *if*

1. *Any random seed occurring in $s_1, \ldots, s_n$, only occurs in $s_1, \ldots, s_n$ as the third argument of an encryption*
2. *If $\{u_1\}_{k_1}^r$ and $\{u_2\}_{k_2}^r$ are two subterms of $s_1, \ldots, s_n$, then $u_1 = u_2$ and $k_1 = k_2$.*

Definition 13.9 *A sequence of terms (resp. a frame) $s_1, \ldots, s_n$ is* weakly valid *if*

- *it has no key cycle*
- *the sandom seeds are used in a consistent way*

Example 13.3 $\{\{u\}_k^r\}_k^{r'}, \{\{u\}_k^r\}_{k'}^{r''}, \{k'\}_k^{r'''}$ *is a weakly valid sequence, with* $k \gg k'$.

Definition 13.10 *A key* k *is* deducible *from a frame* ϕ, *if there is a recipe* u *such that* $u\sigma_\phi\!\downarrow = k$.

Now we can generalize theorem 13.1 to:

Theorem 13.2 *Let* $\nu\overline{n}.\{x_1 \mapsto s_1, \dots x_n \mapsto s_n\}$ *and* $\nu\overline{n'}.\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ *be two weakly valid frames.*

$$\nu\overline{n}.\{x_1 \mapsto s_1, \dots x_n \mapsto s_n\} \sim \nu\overline{n'}.\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\} \quad \Rightarrow \quad [\![s_1, \dots, s_n]\!]_\eta \approx [\![t_1, \dots, t_n]\!]_\eta$$

The proof is basically the same as the proof of theorem 13.1. The only difference lies in points 3 and 4: at step 3, we need to replace ciphertexts $\{u\}_k^r$ with $\{0^{l(u,\eta)}\}_k^r$ *at every position.* And, at step 4, we have to show that maximal keys for one sequence correspond to maximal keys for the other sequence.

The following two lemmas prepare these steps.

Again, we confuse the frames and the term sequences, as all names are assumed to be bound.

Lemma 13.3 *Let* $v_1, \dots, v_m$ *be a valid sequence of terms, let* $u \in \mathcal{M}_0$, k, r *be names such that* $k \not\sqsubseteq u, v_1, \dots, v_m$, *and* r *occurs in* $u, v_1, \dots, v_m$ *only as a random seed in subterms* $\{u\}_k^r$. *Then*

$$[\![v_1, \dots, v_m]\!]_\eta \approx [\![v_1[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r], \dots, v_m[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r]]\!]_\eta$$

and

$$(\nu\overline{n})v_1, \dots, v_m \sim (\nu\overline{n})v_1[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r], \dots, v_m[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r]$$

Proof : We prove first the first claim of the lemma.

Let τ be a partial assigment of all names, except k and the random seeds s that are used in ciphertexts $\{w\}_k^s$ occurring in $v_1, \dots v_m$. We prove actually

$$[\![v_1, \dots, v_m]\!]_\eta^\tau \approx [\![v_1[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r], \dots, v_m[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r]]\!]_\eta^\tau$$

Assume that $\mathcal{A}$ can distinguish the two above sequences with an advantage ϵ:

$$\begin{aligned}\epsilon = |&\mathbb{P}[k, R, r_1, \dots, r_n : \mathcal{A}([\![v_1, \dots, v_m]\!]_\eta^\tau | R) = 1] \\ &-\mathbb{P}[k, R, r_1, \dots, r_n : \mathcal{A}([\![v_1[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r], \dots, v_m[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r]]\!]_\eta^\tau | R) = 1]|\end{aligned}$$

We construct as follows an adversary $\mathcal{B}$ on IND-CPA: let $w_1, \dots, w_n$ be the set of terms w such that $w \sqsubseteq v_1, \dots, v_m$. $w_1, \dots, w_n$ are ordered in such a way that $i < j$ whenever w_i is a subterm of w_j.

1. $\mathcal{B}$ stores in its memory a table associating the terms w_i with their computational interpretations: For $i = 1$ to n, $\mathcal{B}$
   (a) if w_i is a constant or a name (which is then in he domain of τ), $\mathcal{B}$ stores in its table $[\![w_i]\!]_\eta^\tau$
   (b) if $w_i = \langle w_j, w_k\rangle$, then $\mathcal{B}$ retrieves the values of $[\![w_j]\!]_\eta^\tau$, $[\![w_k]\!]_\eta^\tau$, that are stored in its table, computes teh pair of them and stores the result in the table
   (c) if $w_i = \{w_j\}_{k'}^{r'}$ where $k' \neq k$, then $\mathcal{B}$ retreives $[\![w_j]\!]_\eta^\tau$ and computes $[\![w_i]\!]_\eta^\tau$ and stores the result
   (d) if $w_i = \{w_j\}_k^{r'}$, with $r \neq r'$, then $\mathcal{B}$ retrieves $[\![w_j]\!]_\eta^\tau$ from its table, queries the encryption oracle with $([\![w_j]\!]_\eta^\tau, [\![w_j]\!]_\eta^\tau)$ and stores the result.

(e) if $w_i = \{u\}_k^r$, then $\mathcal{B}$ retrieves $[\![u]\!]_\eta^\tau$ from its table and queries the encryption oracle with $([\![u]\!]_\eta^\tau, 0^{l(u,\eta)})$ and strores the result

At the end of this (PTIME) procedure, $\mathcal{B}$ has stored in its table all the computational interpretations of the subterms of $v_1, \ldots, v_m$, if it was interacting with the left-oracle. Otherwise, the stores contains the computational interpretation of the subterms of $v_1[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r], \ldots, v_m[\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r], u$.

2. $\mathcal{B}$ simulates $\mathcal{A}$ with the inputs corresponding to the values stored at $v_1, \ldots, v_m$ locations. It produces the same output as $\mathcal{A}$.

The advantage of $\mathcal{B}$ is ϵ: if $\mathcal{A}$ distinguishes the two sequences of terms with non negligible probability, then $\mathcal{B}$ breaks IND-CPA.

Now, remains to prove the second claim.

We let ρ be the replacement of $\{u\}_k^r$ with $\{0^{l(u,\eta)}\}_k^r$ and, for every i, $v_i' = \rho(v_i)$. The consistency of the use of random seeds implies that ρ is a bijection, whose inverse is ρ'. We claim that $v_1, \ldots, v_n \sim v_1', \ldots, v_n'$. To see this, let $\sigma = \{x_1 \mapsto v_1, \ldots, x_n \mapsto v_n\}$ and $\sigma' = \{x_1 \mapsto v_1', \ldots, x_n \mapsto v_n'\}$. Note first that there is no recipe w such that $w\sigma \downarrow = k$, because $k \not\sqsubseteq v_i$ for every i (and since, for every rewrite rule $l \to r$, $r\sigma = k$ implies that $r \sqsubseteq l$).

By induction on m, we prove that, for any recipe t, $t\sigma \xrightarrow{m} s$ iff $t\sigma' \xrightarrow{m} \rho(s)$. If $m = 0$ this is because none of the random seeds of $v_1, \ldots, v_n$ is occurring in t, hence $\rho(t\sigma) = t\sigma'$. Otherwise, there is a position p in $t\sigma$, a rewrite rule $l \to r$ and a subsitution θ such that $t\sigma|_p = l\theta$ and $t\sigma[r\theta]_p \xrightarrow{m-1} s$. If $l = \pi_i(\langle x_1, x_2\rangle)$ and $r = x_i$, $\rho(l\theta) = \pi_i(\langle \rho(x_1\sigma), \rho(x_2\sigma)\rangle)$. It follows that $t\sigma'|_p = l\theta'$ and $\rho(t\sigma[r\theta]_p) = t\sigma'[r\theta']_p$ and we may apply the induction hypothesis. If $l = \mathsf{dec}(\{x\}_{k_1}^{r_1}, k_1)$, then $k_1 \neq k$ and $\rho(l\theta) = \mathsf{dec}(\{\rho(x\theta)\}_{k_1}^{r_1}, k_1)$. Again $\rho(t\sigma[x\theta]_p) = t\sigma'[x\theta']_p$ and $t\sigma' \to \rho(u\sigma[r\theta]_p)$: we may apply the induction hypothesis.

Now, $v_1, \ldots, v_n \models P(t_1, t_2)$ iff $(t_1\sigma\downarrow, t_2\sigma\downarrow) \in P^I$. As already observed, for $P \in \{M, EQ, EK, EL\}$, P^I is invariant by ρ (and ρ'): $(t_1\sigma\downarrow, t_2\sigma\downarrow) \in P^I$ iff $(\rho(t_1\sigma\downarrow), \rho(t_2\sigma\downarrow)) \in P^I$. Thanks to what we proved above, $\rho(t_i\sigma\downarrow) = t_i\sigma'\downarrow$, hence $v_1, \ldots, v_n \models P(t_1, t_2)$ iff $(t_1\sigma'\downarrow, t_2\sigma'\downarrow) \in P^I$ iff $v_1', \ldots, v_n' \models P(t_1, t_2)$. This concludes the proof of the second claim.

Lemma 13.4 *Assume that $(s_1, \ldots, s_n), (t_1, \ldots, t_n)$ are weakly valid term sequences such that*

- $(s_1, \ldots, s_n) \sim (t_1, \ldots, t_n)$
- *for every $t = \{u\}_k^r \in \mathcal{M}_0$, if $u \notin \mathcal{W}$ and t occurs in some s_i (resp. some t_i), then there is a recipe v such that $v\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}\downarrow = k$ (resp. $v\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}\downarrow = k$).*

Then, for every $j_1 \neq j_2$,

$s_{j_1} \notin \mathcal{W}$ and $s_{j_1} \sqsubseteq s_{j_2}$ implies $t_{j_1} \notin \mathcal{W}$ and $t_{j_1} \sqsubseteq t_{j_2}$.

Proof : We prove the lemma by induction on $|s_{j_2}|$.

In the base case, $|s_{j_2}| = 1$, then $s_{j_1} = s_{j_2}$. Using EQ, it follows that $t_{j_1} = t_{j_2}$.

Now, for the induction step. If $s_{j_2} = \left\langle s_{j_2}', s_{j_2}'' \right\rangle$, using M, t_{j_2} must also be a pair $t_{j_2} = \left\langle t_{j_2}', t_{j_2}'' \right\rangle$ and

$$(s_1, \ldots, s_{j_2-1}, s_{j_2}', s_{j_2}'', s_{j_2+1}, \ldots, s_n) \sim (t_1, \ldots, t_{j_2-1}, t_{j_2}', t_{j_2}'', t_{j_2+1}, \ldots, t_n)$$

Moreover, either $s_{j_1} \sqsubseteq s_{j_2}'$ or else $s_{j_1} \sqsubseteq s_{j_2}''$. By induction hypothesis, t_{j_1} is a name and $t_{j_1} \sqsubseteq t_{j_2}'$ or $t_{j_1} \sqsubseteq t_{j_2}''$. In any case, $t_{j_1} \sqsubseteq \left\langle t_{j_2}', t_{j_2}'' \right\rangle$.

If $s_{j_2} = \{s'_{j_2}\}_{k_{j_2}}^{r_{j_2}}$, then $s_{j_1} \sqsubseteq s'_{j_2}$ and $s_{j_1} \neq k_{j_2}$ and $s_{j_1} \neq r_{j_2}$ (because the sequences are weakly valid). In particular, $s'_{j_2} \notin \mathcal{W}$, hence, by hypothesis, there is a recipe u such that $u\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}\downarrow = k_{j_2}$. Using M again, $u\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}\downarrow = k'_{j_2}$ is a key and, considering the recipe $\mathsf{dec}(x_{j_2}, u)$, t_{j_2} must be $\{t'_{j_2}\}_{k'_{j_2}}^{r'_{j_2}}$ for some t'_{j_2}. Now, $s_1, \ldots, s_{j_2-1}, s'_{j_2}, s_{j_2+1}, \ldots, s_n) \sim (t_1, \ldots, t_{j_2-1}, t'_{j_2}, t_{j_2+1}, \ldots, t_n)$ and, by induction hypothesis, $s_{j_1} \sqsubseteq s'_{j_2}$ implies $t_{j_1} \sqsubseteq t'_{j_2}$. It follows that $t_{j_1} \sqsubseteq t_{j_2}$.

Proof of the theorem 13.2 : Again, we use an induction on $|s_1| + \cdots + |s_n| + |t_1| + \cdots + |t_n|$ (where constants are not counted in $|s_i|, |t_i|$). The base case and cases 1,2 are exactly the same as in the proof of theorem 13.1.

Let us assume now that the two sequences only consists of ciphertexts, constants and names.

- If in one of the two sequences, there is a subterm $\{u\}_k^r$ such that u is not a constant and k is not deducible.

  Assume for instance that such a term occurs as a subterm in the sequence $s_1, \ldots, s_n$. Consider a key k, which is maximal w.r.t. $\gg_{s_1,\ldots,s_n}$ among the keys that are not deducible from $s_1, \ldots, s_n$ and that encrypt at least one non-constant term.

  We claim that $k \not\sqsubseteq s_1, \ldots, s_n$. If it was the case, say $k \sqsubseteq s_1$, we prove, by induction on s_1, that either k is deducible from the sequence, or else there is a non-deducible key k' such that $k' \gg k$. In the base case, $s_1 = k$ is deducible. If $s_1 = \langle s_{11}, s_{12}\rangle$, then either $k \sqsubseteq s_{11}$ or else $k \sqsubseteq s_{12}$ and, by induction hypothesis, k is deducible from $s_{11}, s_{12}, s_2, \ldots, s_n$ or else there is a non deducible key k' such that $k' \gg_{s_{11},s_{12},s_2,\ldots,s_n} k$. In the first case, k is deducible from $s_1, \ldots, s_n$ (replacing x_{11} with $\pi_1(x_1)$ and x_{12} with $\pi_2(x_1)$ in the recipe) and, in the second case, $k' \gg_{s_1,\ldots,s_n} k$. Now, if $s_1 = \{s_{11}\}_{k'}^r$, then, by weak validity of the sequence, $k \neq k'$. Then $k' \gg_{s_1,\ldots,s_n} k$. Either k' is not deducible, and we are done, or else, there is a recipe u such that $u\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}\downarrow = k'$. Then $\mathsf{dec}(x_1, u)$ is a recipe yielding s_{11}. Since, in addition, $k \sqsubseteq s_{11}$, by induction hypothesis, either k is deducible from $s_{11}, s_2, \ldots, s_n$, hence from $s_1, \ldots, s_n$ (replacing x_1 with $\mathsf{dec}(x_1, u)$ in the recipe) or there is a key k', that is not deducible from $s_{11}, \ldots, s_n$, such that $k' \gg_{s_{11},s_2,\ldots,s_n} k$. In the latter case, k' is neither deducible from $s_1, \ldots, s_n$ and $k' \gg_{s_1,\ldots,s_n} k$, which concludes the proof of our claim.

  Then, we may use the lemma 13.3: let, for every i, $s'_i = s_i\{\{u\}_k^r \mapsto \{0^{l(u,\eta)}\}_k^r\}$ where $\{u\}_k^r$ is any subterm of the sequence $s_1, \ldots, s_n$ such that u is not a constant. (There is such a term by assumption). By lemma 13.3,

  $$[\![s_1, \ldots, s_n]\!]_\eta \approx [\![s'_1, \ldots, s'_n]\!]_\eta$$

  and $s_1, \ldots, s_n \sim s'_1, \ldots, s'_n$. It follows that $s'_1, \ldots, s'_n \sim t_1, \ldots, t_n$, hence, by induction hypothesis, $[\![s'_1, \ldots, s'_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$ and, since $[\![s'_1, \ldots, s'_n]\!]_\eta \approx [\![s_1, \ldots, s_n]\!]_\eta$, we get the desired result: $[\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$.

- We assume now that all terms of both sequences are either names, constants or ciphertexts and that every ciphertext $\{u\}_k^r$ occurring in the sequences is such that either k is deducible or else u is a constant. Furthermore, none of the sequences contain two identical terms.

  If one of the sequences contains a ciphertext $\{u\}_k^r$ such that k is deducible: $s_i = \{u_i\}_k^{r_i}$ and there is a recipe v such that $v\sigma\downarrow = k$ where $\sigma = \{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}$.

  After a possible renumbering, let $s_1 = \{u_1\}_k^{r_1}$ be a term in the sequence, whose encryption key k is deducible and which is maximal w.r.t. $\sqsubseteq$. This implies that s_1 has no other occurrence in the sequence (because the plaintext of encryptions with non-deducible keys are constant).

Since $s_1, \ldots, s_n \models M(\mathsf{dec}(x_1, v))$, we must have $t_1, \ldots, t_n \models M(\mathsf{dec}(x_1, v))$: $t_1 = \{v_1\}_{k'}^{r'_1}$ and k' is a deducible key (using the recipe v). By lemma 13.4, t_1 is also maximal w.r.t. $\sqsubseteq$, and, for the same reason as above, has no other occurrence in the sequence.

As before, $u_1, s_2, \ldots, s_n \sim v_1, t_2, \ldots, t_n$. By induction hypothesis, we therefore have $[\![u_1, s_2, \ldots, s_n]\!]_\eta \approx [\![v_1, t_2, \ldots, t_n]\!]_\eta$. Then, by maximality w.r.t. $\sqsubseteq$ of s_1, t_1 and by the consistency of use of random numbers, r has no other occurrence in the sequence $s_1, \ldots, s_n$ and r' has no other occurrence in the sequence $t_1, \ldots, t_n$. From $[\![u_1, s_2, s_2, s_3, \ldots, s_n]\!]_\eta \approx [\![v_1, t_2, t_2, t_3, \ldots, t_n]\!]_\eta$ and exercice 61, it follows $[\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$.

Now, we can proceed with the last cases of the proof, as in the proof of theorem 13.1 and conclude.

Exercice 63

We say that frame $(\nu\overline{n})\ s_1, \ldots, s_n$ is *transparent* if:

- for every ciphertext $\{u\}_k^r$ that occurs in $s_1, \ldots, s_n$, either u is constant or else k is deducible.
- The random seeds are used in a consistent way

Given a transparent frame $\phi = (\nu\overline{n}).s_1, \ldots, s_n$, we define the *flatened* frame $F(\phi)$ by induction as follows:

- If there is an index i such that $s_i = \langle s_{i1}, s_{i2}\rangle$, then $F(\phi) = F((\nu\overline{n}).s_1, \ldots, s_{i-1}, s_{i1}, s_{i2}, s_{i+1}, \ldots, s_n)$.
- If there is are indices i, j such that $s_i = \{s_{i1}\}_{s_j}^{r_i}$, then $F(\phi) = F((\nu\overline{n}).s_1, \ldots, s_{i-1}, s_{i1}, s_{i+1}, \ldots, s_n)$
- Otherwise, $F(\phi) = \phi$.

1. Show that, if ϕ is a transparent frame, then $F(\phi)$ contains only names, constants and encryptions of constants with non-deducible keys.
2. Show that, if ϕ, ϕ' are two transparent frames (that may contain key-cycles), then $F(\phi) \sim F(\phi')$ implies $[\![F(\phi)]\!]_\eta \approx [\![F(\phi')]\!]_\eta$
3. Given two transparent frames ϕ, ϕ', show that $\phi \sim \phi'$ implies $F(\phi) \sim F(\phi')$.
4. Given two transparent frames ϕ, ϕ' such that $\phi \sim \phi'$, show that $[\![\phi]\!]_\eta \approx [\![\phi']\!]_\eta$ iff $[\![F(\phi)]\!]_\eta \approx [\![F(\phi')]\!]_\eta$.
5. Prove an extension of the theorem 13.2, in which the frames may contain key-cycles on deducible keys.

Exercice 64

If we allow arbitrary ground terms (not containing decryption or pairing symbols) as keys, show that the soundness of static equivalence does not hold.

More precisely, construct (from any IND-CPA encryption scheme) another IND-CPA encryption scheme, for which, for two well-chosen terms t_1, t_2, $\{u\}_{t_1}^r \sim \{u\}_{t_2}^r$, none of the names occurring in t_1, t_2 occurs in u, and $[\![\{u\}_{t_1}^r]\!]_\eta \not\approx [\![\{u\}_{t_2}^r]\!]_\eta$.

Exercice 65

Assume here that the encryption scheme is not only IND-CPA, but also *which-key concealing*, which is defined as follows: for any PPT machine $\mathcal{A}$ that has access to two oracles, and security parameter η, let

$$\epsilon_1(\mathcal{A}, \eta) = \begin{array}{l} |\mathbb{P}\left[k, k' \leftarrow \mathcal{K}(\eta), R \leftarrow U : \mathcal{A}^{\mathcal{O}^1_k, \mathcal{O}^1_{k'}}(0^\eta \mid R) = 1\right] \\ \qquad -\mathbb{P}\left[k \leftarrow \mathcal{K}(\eta), R \leftarrow U : \mathcal{A}^{\mathcal{O}^2_k \mathcal{O}^2_k}(0^\eta \mid R) = 1\right]| \end{array}$$

Where $\mathcal{O}_k^1(x,y) = \mathcal{E}(x,k,r)$ and $\mathcal{O}_k^2(x,y) = \mathcal{E}(y,k,r)$ if $|x| = |y|$ (and 0 otherwise).

The encryption scheme is which-key concealing if, for every PPT machine $\mathcal{A}$, $\epsilon_1(\mathcal{A},\eta)$ is negligible.

We consider a new definition of static equivalence $\sim_1$, in which we do not have the predicate EK but, instead, a unary predicate Cipher, which is true exactly on terms that are in $\mathcal{M}_0$ and whose top symbol is an encryption.

1. Show that $\nu k, k'k'', r, r'.\ \{k''\}_k^r, \{k''\}_{k'}^{r'} \sim_1 \nu k, k', k'', r, r'.\ \{k''\}_k^r, \{k''\}_k^{r'}$
2. Let $u_1, \ldots, u_m \in \mathcal{M}_0$ be such that all names occurring in $u_1, \ldots, u_m$ are in the domain of τ and $k, k_1, \ldots, k_m, n_1, \ldots, n_m \notin \mathsf{Dom}(\tau)$. Assume the encryption scheme is which-key concealing. Prove
$$[\![\{u_1\}_{k_1}^{n_1}, \ldots, \{u_m\}_{k_m}^{n_m}]\!]_\eta^\tau \approx [\![\{0^{l(u_1,\eta)}\}_k^{n_1}, \ldots, \{0^{l(u_m,\eta)}\}_k^{n_m}]\!]_\eta^\tau$$
3. Assume that $(s_1, \ldots, s_n)$ and $(t_1, \ldots, t_n)$ are two valid sequences of terms and that the encryption scheme is which-key concealing, then prove
$$(\nu\overline{n})s_1, \ldots, s_n \sim_1 (\nu\overline{m})t_1, \ldots, t_n \Rightarrow [\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$$

13.8 Completeness

The completeness problem is the converse of theorem 13.2: given two sequences of terms that are computationally indistinguishable, are they statically equivalent ? In other words, completeness ensures that we did not give too much power to the symbolic attacker.

The main issue is to be sure that the distinguishing capabilities of the symbolic attacker, the predicates, can be implemented. That is what we consider first.

13.8.1 Predicate implementation

We assume a set of function symbols, all of which have a computational interpretation as a PPT algorithm. Then, if $\mathcal{M}$ is the set of ground terms constructed using this set of function symbols and a set of names, for every mapping τ from names to bitstrings, $[\![\_]\!]_\eta^\tau$ is the unique extension of τ as a homomorphism from $\mathcal{M}$ to $\{0,1\}^*$. As before, we consider name distributions that are parametrized by a security parameter $\eta \in \mathbb{N}$ and, when τ is a partial interpretation only, $[\![\_]\!]_\eta^\tau$ is the corresponding distribution.

Definition 13.11 *A predicate $P \in \mathcal{P}$ of arity k is* implementable *if there is a PPT algorithm $[\![P]\!]$ such that:*

$$\forall s_1, \ldots, s_k \in \mathcal{M}, \exists Q \in \mathrm{POL}_1, \exists N \in \mathbb{N}, \forall \eta > N.$$
$$\mathbb{P}\left[(x_1, \ldots, x_k) \leftarrow [\![s_1, \ldots, s_k]\!]_\eta : [\![P]\!]^{\mathcal{A}}(x_1, \ldots, x_k) = P^I(s_1, \ldots, s_k)\right] > \frac{1}{2} + \frac{1}{Q(\eta)}$$

We will see later examples of predicate symbols that are (not) implementable.

Definition 13.12 *Given a convergent rewrite system $\mathcal{S}$ on $\mathcal{M}$, and an interpretation of the predicate symbols, a* $\mathcal{S}, \mathcal{P}^I$-computational structure *is a computational interpretation of the function symbols and the predicate symbols such that*

$$\forall s, t \in \mathcal{M}, \forall \eta \in \mathbb{N}, \forall \tau.\ (s =_{\mathcal{S}} t \quad\Rightarrow\quad [\![s]\!]_\eta^\tau = [\![t]\!]_\eta^\tau)$$

and, for every $P \in \mathcal{P}$, P is implementable.

Note here that we require not only indistinguishability, but true equality: τ is universally quantified over the assignments of appropriate lengths.

13.8.2 Completeness

Theorem 13.3 (completeness) *For any computational structure,*

$$[\![s_1,\ldots,s_n]\!]_\eta \approx [\![t_1,\ldots,t_n]\!]_\eta \Rightarrow (\nu\overline{n})\ s_1,\ldots,s_n \sim (\nu\overline{m})\ t_1,\ldots,t_n.$$

Proof : By definition, if $s_1,\ldots,s_n \not\sim t_1,\ldots,t_n$, then there are recipes $u_1,\ldots,u_k$ and a predicate $P \in \mathcal{P}$ such that $P^I(u_1\sigma\downarrow,\ldots,u_k\sigma\downarrow) \not\Leftrightarrow P^I(u_1\sigma'\downarrow,\ldots,u_k\sigma'\downarrow)$. where $\sigma = \{x_1 \mapsto s_1,\ldots,x_n \mapsto s_n\}$ and $\sigma' = \{x_1 \mapsto t_1;\ldots,x_n \mapsto t_n\}$. For instance, assume w.l.o.g. that $P^I(u_1\sigma\downarrow,\ldots,u_k\sigma\downarrow) = 1$ and $P^I(u_1\sigma'\downarrow,\ldots,u_k\sigma'\downarrow) = 0$.

Moreover, $u_1,\ldots,u_k$ do not contain any name, by a simple induction on the length of the rewrite sequence, there is a deterministic polynomial time Turing machine $\mathcal{B}$, which, on input $[\![v_1,\ldots,v_n]\!]^\tau_\eta$, computes $[\![P(u_1\theta\downarrow,\ldots,u_k\theta\downarrow)]\!]^\tau_\eta$, where $\theta = \{x_1 \mapsto v_1,\ldots,x_n \mapsto v_n\}$. By definition, for every $\eta \in \mathbb{N}$, for every assignment τ of keys and random numbers occcurring in $v_1,\ldots,v_n$,

$$\mathcal{B}([\![v_1,\ldots v_n]\!]^\tau_\eta) = [\![P]\!]([\![u_1\theta\downarrow]\!]^\tau_\eta,\ldots [\![u_k\theta\downarrow]\!]^\tau_\eta)$$

So,

$$\begin{array}{l}\mathbb{P}\left[(x_1,\ldots,x_n) \leftarrow [\![v_1,\ldots,v_n]\!]_\eta : \mathcal{B}(x_1,\ldots,x_n) = 1\right] = \\ \qquad \mathbb{P}\left[(y_1,\ldots,y_k) \leftarrow [\![u_1\theta\downarrow,\ldots,u_k\theta\downarrow]\!]_\eta : [\![P]\!](y_1,\ldots,y_k) = 1\right]\end{array}$$

On the other hand, according to the definition, there are Q_1 and N_1 such that, for all $\eta > N_1$,

$$\mathbb{P}\left[(y_1,\ldots,y_k) \leftarrow [\![u_1\sigma\downarrow,\ldots,u_k\sigma\downarrow]\!]_\eta : [\![P]\!](y_1,\ldots,y_k) = 1\right] > \frac{1}{2} + \frac{1}{Q_1(\eta)}$$

and there are Q_2 and N_2 such that, for all $\eta > N_2$,

$$\mathbb{P}\left[(y_1,\ldots,y_k) \leftarrow [\![u_1\sigma'\downarrow,\ldots,u_k\sigma'\downarrow]\!]_\eta : [\![P]\!](y_1,\ldots,y_k) = 0\right] > \frac{1}{2} + \frac{1}{Q_2(\eta)}$$

Altogether, if we let

$$\begin{array}{l}\epsilon(\eta) = \\ \qquad \mathbb{P}\left[(x_1,\ldots,x_n) \leftarrow [\![s_1,\ldots,s_n]\!]_\eta : \mathcal{B}(x_1,\ldots,x_n) = 1\right] - \\ \qquad\qquad \mathbb{P}\left[(x_1,\ldots,x_n) \leftarrow [\![t_1,\ldots,t_n]\!]_\eta : \mathcal{B}(x_1,\ldots,x_n) = 1\right]\end{array},$$

For $\eta > \max(N_1,N_2)$,

$$\epsilon(\eta) > (\frac{1}{2} + \frac{1}{Q_1(\eta)}) - (\frac{1}{2} - \frac{1}{Q_2(\eta)}) = \frac{1}{Q_1(\eta)} + \frac{1}{Q_2(\eta)}$$

It follows that $([\![s_1,\ldots,s_n]\!]_\eta)_{\eta\in\mathbb{N}} \not\approx ([\![t_1,\ldots,t_n]\!]_\eta)_{\eta\in\mathbb{N}}$.

13.8.3 Examples of computational structures

M, the predicate defining "valid messages" can be implemented if there is an algorithm, which can recognize when a message is a pair (which we always assume), an algorithm which can recognize when a message is a key, and also an algorithm, which decides when the decryption algorithm is used with a valid input.

More precisely, we assume a particular bitstring $\perp$ (an error message), which is returned when π_i is applied on a bitstring, which is not a pair, or when there is an attempt to encrypt a message, which is not a key or when trying to decrypt something, which is not a ciphertext. Following a strict interpretation, we also assume that any function applied to the error message $\perp$ returns $\perp$. This corresponds actually to a typing predicate, which we assume to be implemented deterministically, for instance using tags (it would also work with a probabilistic implementation of such predicates). Finally, we assume *confusion freeness* as defined below (a definition taken from [203], who also show that such a condition is necessary for completeness, and is not ensured by IND-CPA):

Definition 13.13 *An encryption scheme is* confusion free *if, for every bitstring x,*

$$\mathbb{P}\left[k_1, k_2 \leftarrow \mathcal{K}(\eta), r \leftarrow U : \mathcal{D}(\mathcal{E}(x, k_1 \mid r), k_2) \neq \perp\right] = \nu(\eta)$$

is negligible.

In other words: it is very likely to get an error message when trying to decrypt with a wrong key.

Proposition 13.1 *If the encryption scheme is confusion-free, then the predicates M, EQ are implementable.*

Proof : Let us start with M. Let $[\![M]\!](x) = 1$ iff $x \neq \perp$.

Let $s \in \mathcal{M}$. We have to compute

$$\epsilon(s, \eta) \stackrel{\text{def}}{=} \mathbb{P}\left[x \leftarrow [\![s]\!]_\eta : M^I(s) \neq [\![M]\!](x)\right]$$

If $s \in \mathcal{M}_0$, then $M^I(s) = 1$ and $[\![M]\!](x) = 1$ and therefore $\epsilon(s, \eta) = 0$. So, we only have to consider the case $M^I(s) = 0$. We proceed by induction on s. If s is a constant, then the only possibility for not being accepted by M^I is $s = \perp$, in which case $[\![M]\!](x) = 0$.

Otherwise, if s is a pair or an encryption with a valid key, then, since M^I has a strict interpretation, there must be a direct subterm of s which does not belong to $\mathcal{M}_0$. Then, we use the induction hypothesis and the strictness of $[\![M]\!]$: if s_1, s_2 are the direct subterms of s,

$$\begin{array}{rcl}
\epsilon(s, \eta) & = & \mathbb{P}\left[(x_1, x_2) \leftarrow [\![s_1, s_2]\!]_\eta : [\![M]\!](x_1) = 1 \wedge [\![M]\!](x_2) = 1\right] \\
& \leq & \min(\mathbb{P}\left[(x_1, x_2) \leftarrow [\![s_1, s_2]\!]_\eta : [\![M]\!](x_1) = 1\right], \mathbb{P}\left[(x_1, x_2) \leftarrow [\![s_1, s_2]\!]_\eta : [\![M]\!](x_2) = 1\right]) \\
& = & \min(\mathbb{P}\left[x_1 \leftarrow [\![s_1]\!]_\eta : [\![M]\!](x_1) = 1\right], \mathbb{P}\left[x_2 \leftarrow [\![s_2]\!]_\eta : [\![M]\!](x_2) = 1\right]) \\
& \leq & \min(\epsilon(s_1, \eta), \epsilon(s_2, \eta))
\end{array}$$

If $s = \{u\}_v^r$ and v is not a valid key or if s is a projection of a term which is not a pair, or if s is a decryptoin of a term which is not a ciphertext, then $[\![s]\!]_\eta$ is always $\perp$, by definition, hence $[\![M]\!](x) = 0$ (for all $x \in [\![s]\!]_\eta$).

Finally, if $s = \mathcal{D}(\{t\}_{k_1}^r, k_2)$, if $\{t\}_{k_1}^r$ is not in M^I, we are back to the previous computation. Assume now that $\{t\}_{k_1}^r \in M^I$ and therefore that $k_2 \neq k_1$. Then, by definition of confusion freeness, $\epsilon(s, \eta) = \nu(\eta)$

Next, $[\![EQ]\!](x, y)$ is defined as $[\![M]\!](x) \wedge [\![M]\!](y) \wedge x = y$, which is implementable, as $M^{\mathcal{A}}$ is implementable.

Now there are several symmetric encryption schemes that are proved to ensure confusion-freeness. Typically, the encryption schemes that satisfy, additionally to IND-CPA, some integrity properties. A discussion and constructions on such authenticated encryption schemes can be found in [54].

The converse of theorem 13.2 follows then from the theorem 13.3 and the implementability of EK, EL, which is satisfied by some (but not all) IND-CPA encryption schemes.

Exercice 66
Given an IND-CPA encryption scheme, construct another IND-CPA encryption scheme for which the predicates EK and EL are implementable.

Exercice 67
We extend here the definition of the computational interpretation of terms to terms that may contain decryption symbols. $[\![\mathsf{dec}(s, t)]\!]_\eta^\tau$ is defined by:

1. draw all names occurring in s, t and not in $\mathsf{Dom}(\tau)$,

2. The previous step yielda, together with τ, an assignment τ'; Apply the function $\mathcal{D}$ to $[\![s]\!]_\eta^{\tau'}$ and $[\![t]\!]_\eta^{\tau'}$. This function returns a special bitstring $\perp$ in case it is not defined on the input.

An encryption scheme is *decryption-confusing* if:

1. for every name k, $[\![\{0^\eta\}_k^r]\!]_\eta \approx [\![k]\!]_\eta$
2. for every $x \in \mathcal{M}_0$ and every two distinct names k_1, k_2 not occurring in x,

$$[\![\{\{x\}_{k_1}^{r_1}\}_{k_2}^{r_2}, k_2]\!]_\eta \approx [\![\{x\}_{k_1}^{r_1}, k_2]\!]_\eta$$

1. Show that, if the encryption scheme is decryption confusing, then
   (a) for every distinct names k_1, k_2,

   $$[\![\mathsf{dec}(k_1, k_2), k_2]\!]_\eta \approx [\![k_1, k_2]\!]_\eta \approx [\![\{k_1\}_{k_2}^r, k_2]\!]$$

   (b) For any term x not containing projection symbols, and for any names k_1, k_2 such that $k_1 \neq k_2$, and k_1, k_2 do not occur in x,

   $$[\![\mathsf{dec}(\{x\}_{k_1}^r, k_2), k_2]\!]_\eta \approx [\![\{x\}_{k_1}^r, k_2]\!]_\eta \approx [\![\{\{x\}_{k_1}^{r_1}\}_{k_2}^{r_2}, k_2]\!]_\eta$$

2. In what follows, we assume the encryption scheme IND-CPA and *which-key concealing*, as defined in exercise 65.

   The definition of static equivalence is modified, using a different set of predicate symbols. We use three predicates El and Eq, m which are interpreted in a slightly different way as before since, now, decryption can not fail. m is interpreted as the set of terms, which do not contain projection symbols. El is interpreted as the set of pairs of terms whose lengths are identical, where the length L is defined by:

   $$\begin{array}{rcll} L(k) & = & S & \text{For } k \in \mathcal{N} \\ L(c) & = & S & \text{for any constant c} \end{array} \qquad \begin{array}{rcl} L(D(x,k)) & = & L(x) \\ L(\{x\}_k^r) & = & L(x) \\ L(\langle x, y\rangle) & = & L(x) + L(y) \end{array}$$

   S is an integer constant, used only for the purpose of this definition. Also, we assume $L(c) = S$ for every constant, for simplicity, which means that $\mathcal{W}$ is now restricted to words whose length is a multiple of S. We can think of S as the length of a block in a block cipher encryption scheme.

   Eq is interpreted as equality on the terms that are in m. The rewrite system is unchanged. This defines a staticic equivalence $\sim_c$.

   Show that the following sequences are (statically as well as computationally) distinguishable:

   (a) $(\{\langle \mathbf{0}, k_0\rangle\}_{k_1}^r, k_2, k_1)$ and $(\{\langle \mathbf{1}, k_0\rangle\}_{k_1'}^{r'}, k_1', k_2')$

   (b) $(\{\{\langle \mathbf{0}, k_0\rangle\}_{k_2}^{r_1}\}_{k_1}^{r_2}, k_2, k_1)$ and $(\{\{\langle \mathbf{0}, k_0\rangle\}_{k_2'}^{r_1'}\}_{k_1'}^{r_2'}, k_1', k_2')$

   (c) $(\{k_0\}_{k_1}^{r_1}, k_1, \{\langle k_2, k_3\rangle\}_{k_1}^{r_2}, k_2)$ and $(\{k_0'\}_{k_1'}^{r_1'}, k_2', \{\langle k_2', k_3'\rangle\}_{k_1'}^{r_2'}, k_1')$

   Where $\mathbf{0}$ and $\mathbf{1}$ are sequences of 0's and 1's respectively, of the same appropriate length.

3. Show that the following are statically equivalent:

   (a) $(\{\{\langle \mathbf{0}, k_0\rangle\}_{k_3}^{r_1}\}_{k_1}^{r_2}, k_2, k_1)$ and $(\{\{\langle \mathbf{1}, k_0\rangle\}_{k_3'}^{r_1'}\}_{k_1'}^{r_2'}, k_1', k_2')$

(b) $(\{k_0\}_{k_1}^{r_1}, k_1, k_2, \{\{k_3\}_{k_2}^{r_2}\}_{k_1}^{r_3})$ and $(\{k'_0\}_{k'_1}^{r'_1}, k'_2, k'_1, \{\{k'_3\}_{k'_2}^{r'_2}\}_{k'_1}^{r'_3})$

4. *Valid* sequences are defined as before: sequences of messages in $\mathcal{M}_0$ (in particular not containing decryption symbols) with a consistent use of random numbers and no key-cycle.

   We say that a term sequence is *transparent* if, for every ciphertext $\{u\}_k^r$ occurring in $s_1, \ldots, s_n$, either k is deducible from $s_1, \ldots, s_n$ or else $u \in \mathcal{W}$.

   Let $(s_1, \ldots, s_n)$ be a transparent sequence such that

   (a) $s_1, \ldots, s_n$ are names or ciphertexts
   (b) the only occurrences of $w \in \mathcal{W}$ in the sequence are in expressions $\{w\}_k^r$ where k is not deducible.
   (c) $s_1, \ldots, s_n$ do not contain any pairing
   (d) for every $i \neq j$, $s_i \not\sqsubseteq_{(s_1,\ldots,s_n)} s_j$

   Prove then
$$[\![s_1, \ldots, s_n]\!]_\eta \approx [\![\{0^{l(s_1,\eta)}\}_{k_1}^{r_1}, \ldots, \{0^{l(s_n,\eta)}\}_{k_n}^{r_n}]\!]_\eta$$
   where $k_1, \ldots, k_n \in \mathcal{N}$ are distinct and $r_1, \ldots, r_n$ are distinct.

5. Prove that, if $(s_1, \ldots, s_n)$ and $(t_1, \ldots, t_n)$ are valid transparent sequences of terms and $(s_1, \ldots, s_n) \sim_c (t_1, \ldots, t_n)$, then $s_i \sqsubseteq_{(s_1,\ldots,s_n)} s_j$ implies $t_i \sqsubseteq_{(t_1,\ldots,t_n)} t_j$.

6. Assuming the encryption scheme is decryption confusing, if $s_1, \ldots, s_n$ and $t_1, \ldots, t_n$ are valid sequences, prove
$$(s_1, \ldots, s_n) \sim_c (t_1, \ldots, t_n) \Rightarrow [\![s_1, \ldots, s_n]\!]_\eta \approx [\![t_1, \ldots, t_n]\!]_\eta$$