es-secret-server[11,40]

es-secret-server{$table:ut2, $encrypt:ut2, $decrypt:ut2}
(*es*; *T*; *L*; *i*)
$\equiv_{\text{def}}$ let *ds*= "$table" : secret-table(*T*) in
(↑discrete(*i*;"$table"))
& (∀*l*∈*L*.
destination(*l*) = *i*
& state *ds*
rcv(*l*,"$encrypt")::(ℕ + Atom1)
× data(*T*) sends ["$encrypt", *e*.next("$table" when *e*)::ℕ
× Atom1] on lnk-inv(*l*)
& state *ds*
rcv(*l*,"$decrypt")::(ℕ + Atom1)
× Atom1 sends ["$decrypt", *e*.decrypt("$table" when *e*;val(*e*)):data(*T*)] on lnk-inv(*l*))
& (@*i* only map(λ*l*.rcv(*l*,"$encrypt");*L*) affect "$table":secret-table(*T*)
c∧ ((∀*l*∈*L*.
@*i*
events of kind rcv(*l*,"$encrypt") change "$table" to
λ*s*,*v*. encrypt(*s*."$table";*v*) State(*ds*) (val::(ℕ + Atom1) × data(*T*)))
& ∀*e*@*i*.
∃*e*′:E
(*e* leaks "$table" to *e*′
⇒ (∃*l*∈*L*.(kind(*e*) = rcv(*l*,"$encrypt")
& kind(*e*′) = rcv(lnk-inv(*l*),"$encrypt"))
∨ (kind(*e*) = rcv(*l*,"$decrypt")
& kind(*e*′) = rcv(lnk-inv(*l*),"$decrypt"))))
& ∀*e*@*i*. ¬*e* copies "$table"
& ∀*e*@*i*.
(↑first(*e*))
⇒ (atoms-distinct("$table" when *e*)
& ptr("$table" when *e*) = 0
& (∀*n*:ℕ, *j*:Id.
(*n* < ‖"$table" when *e*‖)
⇒ *j* ‖ st-atom("$table" when *e*;*n*)
⇒ (*j* = *i*)))))

clarification:

es-secret-server{$table:ut2, $encrypt:ut2, $decrypt:ut2}
(*es*; *T*; *L*; *i*)
$\equiv_{\text{def}}$ let *ds*= "$table" : secret-table(*T*) in
(↑es-isconst(*es*;*i*;"$table"))
& l_all(*L*;IdLnk;*l*.destination(*l*) = *i* ∈ Id

& es-kind-sends-iff(es;rcv(l,"$encrypt");:($\mathbb{N}$ + Atom1)
× data(T);lnk-inv(l);"$encrypt";:$\mathbb{N}$ × Atom1;ds;e.next(es-when(es; "$table"; e)))
& es-kind-sends-iff(es;rcv(l,"$decrypt");:($\mathbb{N}$ + Atom1)
× Atom1;lnk-inv(l);"$decrypt";data($T$);$ds$;$e$.decrypt(es-when
(es; "$table"; e);es-val(es; e))))
& (es-frame(es;i;map(λl.rcv(l,"$encrypt");$L$);"$table";secret-table(T))
c∧ (l_all(L;IdLnk;l.effect-p(es;i;ds;rcv(l,"$encrypt");:($\mathbb{N}$ + Atom1)
× data(T);"$table";$\lambda s,v$. encrypt($s$."$table";v)))
& alle-at(es;i;e.$\exists e'$:es-E(es)
(es-leaks(es;e;"$table";$e'$)
⇒ l_exists(L;IdLnk;l.(es-kind(es; e)
=
rcv(l,"$encrypt")
∈ Knd
& es-kind(es; e') = rcv(lnk-inv(l),"$encrypt") ∈ Knd)
∨ (es-kind(es; e) = rcv(l,"$decrypt") ∈ Knd
& es-kind(es; e')
=
rcv(lnk-inv(l),"$decrypt")
∈ Knd))))
& alle-at(es;i;e.¬es-copies(es;e;"$table"))
& alle-at(es;i;e.(↑es-first(es; e))
⇒ (atoms-distinct(es-when(es; "$table"; e))
& ptr(es-when(es; "$table"; e)) = 0 ∈ $\mathbb{N}$
& (∀n:$\mathbb{N}$, j:Id.
(n < ‖es-when(es; "$table"; e)‖)
⇒ es-atom(es;j;st-atom(es-when(es; "$table"; e);n))
⇒ (j = i ∈ Id))))))

---

http://www.nuprl.org/FDLcontent/p0_963683_/p58_241630_{es-secret-server}.html