```
--[ AnswerRequest($IB, adec(request, ~ltkIB)) ]->
[ Out( h(adec(request, ~ltkIB)) ) ]
rule Secrecy_claim:
   [ Secret(A, B, wd) ] --[ Secret(A, B, wd ]-> []
rule Cu_1_send:
let m = <Cu, ~POI_c>
in
[ Fr(~POI_C)
, !Ltk(Cu, ltkCu)
, !Pk(IB, pkIB)
]
--[ Send(Cu, wd)
]->
[ St_Cu_1(Cu, ltkCu, pkIB, IB, POI_c)
, Out(<wd,sign(wd,ltkCu)>)
]
restriction Equality:
"All x y #i. Eq(x,y) @i ==> x = y"
lemma executable:
exists-trace
"Ex Cu WD wd #i #j. Send(Cu,wd)@i & Recv(WD,wd) @j"


lemma auth_injective:
"
( All IB Kcib #i. SessKeyC(IB, Kcib) @ #i
==>
( (Ex #a. AnswerRequest(IB, Kcib) @ a
& (All #j. SessKeyC(IB, Kcib) @ #j ==> #i = #j)
)
| (Ex #r. LtkReveal(IB) @ r & r < i)
)
)
```