cross realm version

Protocol Purpose

The Kerberos protocol is designed to operate across organisational boundaries. A client in one organisation can be authenticated to a server in another. Each organisation wishing to run a Kerberos server establishes its own "realm".

Definition Reference

- http://www.ietf.org/internet-drafts/draft-ietf-krb-wg-kerberos-clarifications-07.txt

Model Authors

- Vishal Sankhla, University of Southern California, August 2004

Alice&Bob style

```
1. C -> ASlocal  : C, TGSlocal, N1
2. ASlocal -> C  : C, Ticket1,
                   {TGSlocal, KC_TGSlocal, Tstart1, Texpire1, N1
                   }_KC_ASlocal
   where Ticket1 : {C, TGSlocal, KC_TGSlocal, Tstart1, Texpire1
                   }_KASlocal_TGSlocal
3. C -> TGSlocal : TGSremote, N2, Ticket1, {C, T1}_KC_TGSlocal
4. TGSlocal -> C : C, Ticket2b,
                   {TGSremote, KC_TGSremote, Tstart2b, Texpire2, N2
                   }_KC_TGSlocal
   where Ticket2b: {C,TGSremote,KC_TGSremote,Tstart2b,Texpire2
                   }_KTGSlocal_TGSremote
5. C -> TGSremote: S,N3,Ticket2b, {C, T2B}_KC_TGSremote
6. TGSremote -> C: C, Ticket3,
                   {Sremote, KC_Sremote, Tstart3,Texpire3}_KC_TGSremote
   where Ticket3 : {C, Sremote, KC_Sremote, Tstart3, Texpire3
                   }_KTGSremote_Sremote
7. C -> Sremote  : Ticket3, {C,T3}_KC_Sremote
8. Sremote -> C  : {T3}_KC_Sremote
```

Problems considered: 8

Attacks Found

None

Further Notes

Agents involved: Client, Local Authentication Server (ASLocal), Local Ticket Granting server (TGSlocal), Remote Ticket Granting server (TGSRemote), Remote Server where the client needs to authenticate (ServerRemote)

HLPSL Specification

```
role client(C,
            ASlocal,
            TGSlocal,
            TGSremote,
            Sremote    : agent,
            KC_ASlocal : symmetric_key,
            SND, RCV   : channel(dy))
played_by C def=

  local State      : nat,
        T1,T2B,T3  : text,
        KC_TGSlocal,
        KC_TGSremote,
        KC_Sremote : symmetric_key,
        Ticket1,
        Ticket2b,
        Ticket3    : {agent.agent.symmetric_key.text.text}_symmetric_key,
        Tstart1,
        Texpire1,
        Tstart2b,
```

```
      Texpire2,
      Tstart3,
      Texpire3   : text,
      N1,N2,N3   : text

const sec_c_KC_TGSlocal,
      sec_c_KC_TGSremote,
      sec_c_KC_Sremote,
      sec_c_T3   : protocol_id

init  State := 0

transition

step1.
  State = 0  /\ RCV(start)
  =|>
  State':= 1 /\ N1' := new()
             /\ SND(C.TGSlocal.N1')

step2.
  State = 1 /\ RCV(C.Ticket1'.
                  {TGSlocal.KC_TGSlocal'.Tstart1'.Texpire1'.N1}_KC_ASlocal)
  =|>
  State':= 2 /\ N2' := new()
             /\ T1' := new()
             /\ SND(TGSremote.N2'.Ticket1'.{C.T1'}_KC_TGSlocal')
             /\ witness(C,TGSlocal,t1,T1')
             /\ request(C,ASlocal,n1,N1)
             /\ secret(KC_TGSlocal',sec_c_KC_TGSlocal,{ASlocal,C,TGSlocal})

step3.
  State = 2  /\ RCV(C.Ticket2b'.
             {TGSremote.KC_TGSremote'.Tstart2b'.Texpire2'.N2}_KC_TGSlocal)
  =|>
  State':= 3 /\ N3' := new()
             /\ T2B' := new()
             /\ SND( Sremote.N3'.Ticket2b'.{C.T2B'}_KC_TGSremote')
             /\ witness(C,TGSremote,t1r,T2B')
             /\ request(C,TGSlocal,n1r,N2)
             /\ secret(KC_TGSremote',sec_c_KC_TGSremote,{TGSlocal,C,TGSremote})
```

```
  step4.
    State = 3  /\ RCV(C.Ticket3'.
                    {Sremote.KC_Sremote'.Tstart3'.Texpire3'.N3}_KC_TGSremote )
    =|>
    State':= 4 /\ T3' := new()
               /\ SND (Ticket3'.{C.T3'}_KC_Sremote')
               /\ witness(C,Sremote,t2b,T3')
               /\ request(C,TGSremote,n2,N3)
               /\ secret(KC_Sremote',sec_c_KC_Sremote,{TGSremote,C,Sremote})
               /\ secret(T3',sec_c_T3,{C,Sremote})

  step5.
    State = 4  /\ RCV( {T3}_KC_Sremote ) =|>
    State':= 5 /\ request(C,Sremote,t2a,T3)

end role
```

```
role aSlocalRole(C,
                 ASlocal,
                 TGSlocal          : agent,
                 KC_ASlocal,
                 KASlocal_TGSlocal : symmetric_key,
                 SND ,RCV          : channel(dy))
played_by ASlocal def=

  local State            : nat,
        N1               : text,
        Tstart1,Texpire1 : text,
        KC_TGSlocal      : symmetric_key

  const sec_a_KC_TGSlocal : protocol_id

  init  State := 6

  transition

  step1.
    State = 6  /\ RCV( C.TGSlocal.N1') =|>
```

```
    State':= 7 /\ Tstart1' := new()
               /\ Texpire1' := new()
               /\ KC_TGSlocal' := new()
               /\ SND(C.
              {C.TGSlocal.KC_TGSlocal'.Tstart1'.Texpire1'}_KASlocal_TGSlocal.
              {TGSlocal.KC_TGSlocal'.Tstart1'.Texpire1'.N1'}_KC_ASlocal)
               /\ witness(ASlocal,C,n1,N1')
               /\ secret(KC_TGSlocal',sec_a_KC_TGSlocal,{ASlocal,C,TGSlocal})

end role
```

```
role tGSlocalRole(C,
                  ASlocal,
                  TGSlocal,TGSremote  : agent,
                  KASlocal_TGSlocal,
                  KTGSlocal_TGSremote : symmetric_key,
                  SND ,RCV            : channel(dy),
                  L                   : text set)
played_by TGSlocal def=

  local State              : nat,
        N2                 : text,
        Tstart1,  Texpire1 : text,
        Tstart2b, Texpire2 : text,
        KC_TGSlocal        : symmetric_key,
        KC_TGSremote       : symmetric_key,
        T1                 : text

  const sec_tl_KC_TGSlocal,
        sec_tl_KC_TGSremote : protocol_id

  init  State := 8

  transition

  step1.
    State = 8  /\ RCV(TGSremote.N2'.
              {C.TGSlocal.KC_TGSlocal'.Tstart1'.Texpire1'}_KASlocal_TGSlocal.
              {C.T1'}_KC_TGSlocal')
```

```
              /\ not(in(T1',L)) =|>
    State':= 9 /\ Tstart2b' := new()
              /\ Texpire2' := new()
              /\ KC_TGSremote' := new()
              /\ SND(C.
            {C.TGSremote.KC_TGSremote'.Tstart2b'.Texpire2'}_KTGSlocal_TGSremote.
            {TGSremote.KC_TGSremote'.Tstart2b'.Texpire2'.N2'}_KC_TGSlocal')
              /\ L' = cons(T1',L)
              /\ wrequest(TGSlocal,C,t1,T1')
              /\ witness(TGSlocal,C,n1r,N2')
              /\ secret(KC_TGSlocal',sec_tl_KC_TGSlocal, {ASlocal,C,TGSlocal})
              /\ secret(KC_TGSremote',sec_tl_KC_TGSremote, {TGSlocal,C,TGSremote})

end role
```

```
role tGSremoteRole(C,
                   TGSlocal,
                   TGSremote,
                   Sremote             : agent,
                   KTGSlocal_TGSremote,
                   KTGSremote_Sremote  : symmetric_key,
                   SND ,RCV            : channel(dy),
                   L                   : text set )
played_by TGSremote def=

  local State              : nat,
        N3                 : text,
        Tstart2b, Texpire2 : text,
        Tstart3,  Texpire3 : text,
        KC_TGSremote,
        KC_Sremote         : symmetric_key,
        T2B                : text

  const sec_tr_KC_Sremote,
        sec_tr_KC_TGSremote : protocol_id

  init  State := 10

  transition
```

```
  step1.
    State = 10  /\ RCV(Sremote.N3'.
            {C.TGSremote.KC_TGSremote'.Tstart2b'.Texpire2'}_KTGSlocal_TGSremote.
            {C.T2B'}_KC_TGSremote')
      /\ not(in(T2B',L)) =|>
    State':= 11 /\ Tstart3' := new()
                /\ Texpire3' := new()
                /\ SND(C.
             {C.Sremote.KC_Sremote'.Tstart3'.Texpire3'}_KTGSremote_Sremote.
             {Sremote.KC_Sremote'.Tstart3'.Texpire3'.N3'}_KC_TGSremote')
                /\ L' := cons(T2B',L)
                /\ wrequest(TGSremote,C,t1r,T2B')
                /\ witness(TGSremote,C,n2,N3')
                /\ secret(KC_Sremote',sec_tr_KC_Sremote,{TGSremote,C,Sremote})
                /\ secret(KC_TGSremote',sec_tr_KC_TGSremote,{TGSlocal,C,TGSremote})

end role
```

```
role sremoteRole(C,
                 TGSremote,
                 Sremote            : agent,
                 KTGSremote_Sremote : symmetric_key,
                 SND ,RCV           : channel(dy),
                 L                  : text set )
played_by Sremote def=

  local State             : nat,
        Tstart3, Texpire3 : text,
        KC_Sremote        : symmetric_key,
        T3                : text

  const sec_s_KC_Sremote,
        sec_s_T3          : protocol_id

  init  State := 12

  transition
```

```
  step1.
    State = 12  /\
        RCV({C.Sremote.KC_Sremote'.Tstart3'.Texpire3'}_KTGSremote_Sremote.
             {C.T3'}_KC_Sremote')
                /\ not(in(T3',L)) =|>
    State':= 13 /\ SND({T3'}_KC_Sremote')
                /\ L' := cons(T3',L)
                /\ witness(Sremote,C,t2a,T3')
                /\ request(Sremote,C,t2b,T3')
                /\ secret(KC_Sremote',sec_s_KC_Sremote,{TGSremote,C,Sremote})
                /\ secret(T3',sec_s_T3,{C,Sremote})

end role
```

```
role session(C,ASlocal,TGSlocal,TGSremote,Sremote   : agent,
             KC_ASlocal,KASlocal_TGSlocal           : symmetric_key,
             KTGSlocal_TGSremote,KTGSremote_Sremote : symmetric_key,
             LTGSlocal, LTGSremote, LSremote        : text set )
def=

  local Send1, Send2, Send3, Send4, Send5,
        Receive1, Receive2, Receive3, Receive4, Receive5: channel (dy)

  composition
    client(C,ASlocal,TGSlocal,TGSremote,Sremote,KC_ASlocal,Send1,Receive1)
    /\  aSlocalRole(C,ASlocal,TGSlocal,
                    KC_ASlocal, KASlocal_TGSlocal,Send2,Receive2)
    /\  tGSlocalRole(C,ASlocal,TGSlocal,TGSremote,
                     KASlocal_TGSlocal, KTGSlocal_TGSremote,
                     Send3,Receive3,LTGSlocal)
    /\  tGSremoteRole(C,TGSlocal,TGSremote,Sremote,
                      KTGSlocal_TGSremote,KTGSremote_Sremote,
                      Send4,Receive4,LTGSremote)
    /\  sremoteRole(C,TGSremote,Sremote,KTGSremote_Sremote,
                    Send5,Receive5,LSremote)

end role
```

```
role environment() def=

  local LTGSL, LTGSR, LS : text set

  const c, asl, tgsl, tgsr, s : agent,
        ki_aslocal,
        kc_aslocal,
        kaslocal_tgslocal,
        ktgslocal_tgsremote,
        ktgsremote_sremote    : symmetric_key,

        t1,t1r,t2a,t2b,n1,n1r,n2: protocol_id

  init LTGSL = {} /\ LTGSR = {} /\ LS = {}

  intruder_knowledge = {c,asl,tgsl,tgsr,s,ki_aslocal
                        }

  composition

        session(c,asl,tgsl,tgsr,s,
                kc_aslocal,kaslocal_tgslocal,ktgslocal_tgsremote,
                ktgsremote_sremote,LTGSL,LTGSR,LS)
 /\     session(i,asl,tgsl,tgsr,s,
                ki_aslocal,kaslocal_tgslocal,ktgslocal_tgsremote,
                ktgsremote_sremote,LTGSL,LTGSR,LS)

end role
```

```
goal

  %secrecy_of KC_TGSlocal, KC_TGSremote, KC_Sremote, T3
  secrecy_of sec_c_KC_TGSlocal,sec_c_KC_TGSremote,sec_c_KC_Sremote,sec_c_T3,
             sec_a_KC_TGSlocal,
             sec_tl_KC_TGSlocal,sec_tl_KC_TGSremote,
             sec_tr_KC_Sremote,sec_tr_KC_TGSremote,
             sec_s_KC_Sremote,sec_s_T3
```

```
  %Client authenticates ASlocalRole   on n1
  authentication_on n1
  %Client authenticates TGSlocalRole  on n1r
  authentication_on n1r
  %Client authenticates TGSremoteRole on n2
  authentication_on n2
  %Client authenticates SremoteRole   on t2a
  authentication_on t2a
  %SremoteRole   authenticates Client   on t2b
  authentication_on t2b
  %TGSlocalRole  weakly authenticates Client  on t1
  weak_authentication_on t1
  %TGSremoteRole weakly authenticates Client on t1r
  weak_authentication_on t1r

end goal
```

```
environment()
```

References