with PA-ENC-TIMESTAMP pre-authentication method

Protocol Purpose

Mutual authentication

Definition Reference

- http://www.ietf.org/internet-drafts/draft-ietf-krb-wg-preauth-framework-02.txt

Model Authors

- Daniel Plasto for Siemens CT IC 3, 2004
- Vishal Sankhla, University of Southern California, 2004

Alice&Bob style

```
C -> A: U,G,N1,{C,T0}_Kca
A -> C: U,Tcg,{G,Kcg,T1start,T1expire,N1}_Kca

where Tcg := {U,C,G,Kcg,T1start,T1expire}_Kag
      A := Key Distribution Centre

C -> G: S,N2,Tcg,Acg
G -> C: U,Tcs,{S,Kcs,T2start,T2expire,N2}_Kcg

where Acg := {C,T1}_Kcg  (T1 is a timestamp)
      Tcs := {U,C,S,Kcs,T2start,T2expire}_Kgs

C -> S: Tcs,Acs
S -> C: {T2'}_Kcs

where Acs := {C,T2'}_Kcs  (T2 is a timestamp)
```

Problems considered: 7

Attacks Found

None

Further Notes

The AS, TGS and S cache the timestamps they have received in order to prevent replays as specified in RFC 1510.

HLPSL Specification

```
role authenticationServer(
             A,C,G    : agent,
             Kca,Kag   : symmetric_key,
             SND, RCV : channel(dy),
             L        : text set)
played_by A
def=

  local State    : nat,
        N1       : text,
        U        : agent,
        T0       : text,
        Kcg      : symmetric_key,
        T1start  : text,
        T1expire : text

  const sec_a_Kcg : protocol_id

  init  State := 11

  transition
    1. State  = 11 /\ RCV(U'.G.N1'.{C.T0'}_Kca)
```

```
                  /\ not(in(T0',L)) =|>
       State':= 12 /\ Kcg' := new()
                  /\ T1start' := new()
                  /\ T1expire' := new()
                  /\ SND(U'.
                         {U'.C.G.Kcg'.T1start'.T1expire'}_Kag.
                         {G.Kcg'.T1start'.T1expire'.N1'}_Kca)
                  /\ L' := cons(T0',L)
                  /\ witness(A,C,n1,N1')
                  /\ wrequest(A,C,t0,T0')
                  /\ secret(Kcg',sec_a_Kcg,{A,C,G})

end role
```

```
role ticketGrantingServer (
             G,S,C,A      : agent,
             Kag,Kgs      : symmetric_key,
             SND,RCV      : channel(dy),
             L            : text set)
played_by G
def=

  local State    : nat,
        N2       : text,
        U        : agent,
        Kcg      : symmetric_key,
        Kcs      : symmetric_key,
        T1start,T1expire  : text,
        T2start, T2expire : text,
        T1       : text

  const sec_t_Kcg, sec_t_Kcs : protocol_id

  init  State := 21

  transition
    1. State  = 21 /\ RCV(S.N2'.
                         {U'.C.G.Kcg'.T1start'.T1expire'}_Kag.
                         {C.T1'}_Kcg')
```

```
                  /\ not(in(T1',L))
       =|>
       State':= 22 /\ Kcs' := new()
                   /\ T2start' := new()
                   /\ T2expire' := new()
                   /\ SND(U'.
                          {U'.C.S.Kcs'.T2start'.T2expire'}_Kgs.
                          {S.Kcs'.T2start'.T2expire'.N2'}_Kcg')
                   /\ L' := cons(T1',L)
                   /\ wrequest(G,C,t1,T1')
                   /\ witness(G,C,n2,N2')
                   /\ secret(Kcg',sec_t_Kcg,{A,C,G})
                   /\ secret(Kcs',sec_t_Kcs,{G,C,S})

end role
```

```
role server( S,C,G    : agent,
             Kgs      : symmetric_key,
             SND, RCV : channel(dy),
             L        : text set)
played_by S
def=

  local State    : nat,
        U        : agent,
        Kcs      : symmetric_key,
        T2expire : text,
        T2start  : text,
        T2       : text

  const sec_s_Kcs : protocol_id

  init  State := 31

  transition
    1. State  = 31 /\ RCV({U'.C.S.Kcs'.T2start'.T2expire'}_Kgs.
                          {C.T2'}_Kcs')
                   /\ not(in(T2',L)) =|>
       State':= 32 /\ SND({T2'}_Kcs')
```

```
                    /\ L' = cons(T2',L)
                    /\ request(S,C,t2a,T2')
                    /\ witness(S,C,t2b,T2')
                    /\ secret(Kcs',sec_s_Kcs,{G,C,S})

end role
```

```
role client( C,G,S,A      : agent,
             U            : agent,
             Kca          : symmetric_key,
             SND,RCV      : channel(dy))
played_by C
def=

  local State    : nat,
        Kcs      : symmetric_key,
        T1expire : text,
        T2expire : text,
        T1start  : text,
        T2start  : text,
        Kcg      : symmetric_key,
        Tcg,Tcs  : {agent.agent.agent.symmetric_key.text.text}_symmetric_key,
        T0,T1,T2 : text,
        N1,N2    : text

  const sec_c_Kcg, sec_c_Kcs : protocol_id

  init  State := 1

  transition
    1. State  = 1 /\ RCV( start )  =|>
       State':= 2 /\ N1' := new()
                  /\ T0' := new()
                  /\ SND(U.G.N1'.{C.T0'}_Kca)
                  /\ witness(C,A,t0,T0')

    2. State  = 2 /\ RCV(U.Tcg'.{G.Kcg'.T1start'.T1expire'.N1}_Kca) =|>
       State':= 3 /\ N2' := new()
                  /\ T1' := new()
```

```
                    /\ SND(S.N2'.Tcg'.{C.T1'}_Kcg')
                    /\ witness(C,G,t1,T1')
                    /\ request(C,A,n1,N1)
                    /\ secret(Kcg',sec_c_Kcg,{A,C,G})

     3. State  = 3 /\ RCV(U.Tcs'.{S.Kcs'.T2start'.T2expire'.N2}_Kcg) =|>
        State':= 4 /\ T2' := new()
                    /\ SND(Tcs'.{C.T2'}_Kcs')
                    /\ witness(C,S,t2a,T2')
                    /\ request(C,G,n2,N2)
                    /\ secret(Kcs',sec_c_Kcs,{G,C,S})

     4. State  = 4 /\ RCV({T2}_Kcs) =|>
        State':= 5
                    /\ request(C,S,t2b,T2)

  end role
```

```
role session(A,G,C,S                    : agent,
             U                          : agent,
             Kca,Kgs,Kag                : symmetric_key,
             LS,LG,LA                   : text set)
def=

        local
             SendC,ReceiveC             : channel (dy),
             SendS,ReceiveS             : channel (dy),
             SendG,ReceiveG             : channel (dy),
             SendA,ReceiveA             : channel (dy)
  composition
           client(C,G,S,A,U,Kca,SendC,ReceiveC)
      /\   server(S,C,G,Kgs,SendS,ReceiveS,LS)
      /\   ticketGrantingServer(G,S,C,A,Kag,Kgs,SendG,ReceiveG,LG)
      /\   authenticationServer(A,C,G,Kca,Kag,SendA,ReceiveA,LA)

end role
```

```
role environment() def=

  local LS, LG, LA : text set

  const a,g,c,s                 : agent,
        kgi,
        kca,kgs,kag             : symmetric_key,
        kia                     : symmetric_key,
        u3,
        u1,u2                   : agent,
        t0,t1,t2a,t2b,n1,n2     : protocol_id

  init LS = {} /\ LG = {} /\ LA = {}

  intruder_knowledge = {a,g,c,s,u1,u2,kia
                        }

  composition

        session(a,g,c,s,u1,kca,kgs,kag,LS,LG,LA) % normal session
 /\     session(a,g,i,s,u2,kia,kgs,kag,LS,LG,LA) % i is Client

end role
```

```
goal

  %secrecy_of Kcg,Kcs
  secrecy_of sec_a_Kcg,
             sec_t_Kcg, sec_t_Kcs,
             sec_s_Kcs,
             sec_c_Kcg, sec_c_Kcs

  %Client authenticates AuthenticationServer on n1
  authentication_on n1
  %Client authenticates TicketGrantingServer on n2
  authentication_on n2
  %Client authenticates Server on t2b
  authentication_on t2b
  %Server authenticates Client on t2a
```

```
  authentication_on t2a
  %TicketGrantingServer weakly authenticates Client on t1
  authentication_on t1
  %AuthenticationServer weakly authenticates Client on t0
  authentication_on t0

end goal
```

```
environment()
```

References