AAA Mobile IP

Protocol Purpose

This document specifies a Diameter application that allows a Diameter server to authenticate, authorise and collect accounting information for Mobile IPv4 services rendered to a mobile node.

Definition Reference

- [Per03, CJP03]

Model Authors

- Haykal Tej, Siemens CT IC 3, 2003
- Paul Hankes Drielsma, Information Security Group, ETH Zürich, December 2003
- Sebastian Mödersheim, Information Security Group, ETH Zürich, January 2004
- Luca Compagna, AI-Lab DIST, University of Genova, December 2004

Alice&Bob style

```
1. FA   -> MN:   FA,N_FA
2. MN   -> FA:   N_FA,MN,AAAH,
                 {N_FA,MN,AAAH}_K_MnAAAH
3. FA   -> AAAL: N_FA,MN,AAAH,
                 {N_FA,MN,AAAH}_K_MnAAAH
4. AAAL -> AAAH: N_FA,MN,AAAH,
                 {N_FA,MN,AAAH}_K_MnAAAH
5. AAAH -> HA:   MN,
                 {K_MnHa,K_FaHa}_KAAAHHa,
                 {K_MnFa,K_MnHa}_K_MnAAAH,
                 {MN,
                  {K_MnHa,K_FaHa}_KAAAHHa,
                  {K_MnFa,K_MnHa}_K_MnAAAH
                 }_K_AAAHHa
6. HA   -> AAAH: {K_MnFa,K_MnHa}_K_MnAAAH,
                 {{K_MnFa,K_MnHa}_K_MnAAAH}_K_MnHa,
```

```
                  {{K_MnFa,K_MnHa}_K_MnAAAH,
                   {{K_MnFa,K_MnHa}_K_MnAAAH}_K_MnHa
                  }_K_AAAHHa
 7. AAAH -> AAAL: N_FA,
                  {K_MnFa,K_FaHa}_K_AAAHAAAL,
                  {K_MnFa,K_MnHa}_K_MnAAAH,
                  {{K_MnFa,K_MnHa}_K_MnAAAH}_K_MnHa,
                  {N_FA,
                   {K_MnFa,K_FaHa}_K_AAAHAAAL,
                   {K_MnFa,K_MnHa}_K_MnAAAH,
                   {{K_MnFa,K_MnHa}_K_MnAAAH}_K_MnHa
                  }_K_AAAHAAAL
 8. AAAL -> FA:   N_FA,
                  {K_MnFa,K_FaHa}_K_FaAAAL,
                  {K_MnFa,K_MnHa}_K_MnAAAH,
                  {{K_MnFa,K_MnHa}_K_MnAAAH}_K_MnHa,
                  {N_FA,
                   {K_MnFa,K_FaHa}_K_FaAAAL,
                   {K_MnFa,K_MnHa}_K_MnAAAH,
                   {{K_MnFa,K_MnHa}_K_MnAAAH}_K_MnHa
                  }_K_FaAAAL
 9. FA   -> MN:   {K_MnFa,K_FaHa}_K_FaAAAL,
                  {K_MnFa,K_MnHa}_K_MnAAAH,
                  {{K_MnFa,K_MnHa}_K_MnAAAH}_K_MnHa
```

Problems considered: 7

Attacks Found

```
 i      -> (mn,3): fa,fa
 (mn,3) -> i:      fa,mn,aaah,{fa,mn,aaah}k_mn_aaah
 i      -> (mn,3): {fa,mn,aaah}k_mn_aaah,{{fa,mn,aaah}k_mn_aaah}(mn,aaah)
```

In this type-flaw attack, the intruder replays the message `{fa,mn,aaah}k_mn_aaah` to the mobile node, which expects to receive a message of the form `{fa,NewKey}k_mn_aaah` where NewKey is the new key, which is thus matched with the pair of agent names `mn,aah`. Since the intruder knows these two agent names, he can also produce a message encrypted with this new key as required.

HLPSL Specification

```
role aaa_MIP_MN (MN, AAAH, FA : agent,
                 Snd, Rcv     : channel(dy),
                 K_MnAAAH     : symmetric_key)
played_by MN
def=

  local  State          : nat,
         K_MnFa,K_MnHa : symmetric_key

  init   State := 0

  transition

   1. State = 0
      /\ Rcv(FA.FA)
      =|>
      State' := 1
      /\  Snd(FA.MN.AAAH.{FA.MN.AAAH}_K_MnAAAH)

   2. State = 1
      /\ Rcv( {K_MnFa'.K_MnHa'}_K_MnAAAH.
             {{K_MnFa'.K_MnHa'}_K_MnAAAH}_K_MnHa')
      =|>
      State' := 2
      /\ wrequest(MN,AAAH,k_mnha2,K_MnHa')
      /\ wrequest(MN,AAAH,k_mnfa2,K_MnFa')

end role
```

```
role aaa_MIP_FA (FA,AAAL,AAAH,MN: agent,
                 Snd, Rcv: channel(dy),
```

```
                 K_FaAAAL: symmetric_key)
played_by FA
def=

  local
    State              : nat,
    K_MnFa, K_FaHa     : symmetric_key,
    SignedRegReq       : {agent.(agent.agent)}_symmetric_key,
    KeyMnHaKeyMnFa     : {symmetric_key.symmetric_key}_symmetric_key,
    SignKeyMnHaKeyMnFa :
         {{symmetric_key.symmetric_key}_symmetric_key}_symmetric_key

  init State := 0

  transition

   1. State = 0
      /\ Rcv(start)
      =|>
      State' := 1
      /\ Snd(FA.FA)

   2. State = 1
      /\ Rcv(FA.MN.AAAH.SignedRegReq')
      =|>
      State' := 2
      /\ Snd(FA.MN.AAAH.SignedRegReq')

   3. State = 2
      /\ Rcv( FA.{K_MnFa'.K_FaHa'}_K_FaAAAL.
              KeyMnHaKeyMnFa'.SignKeyMnHaKeyMnFa'.
              {FA.{K_MnFa'.K_FaHa'}_K_FaAAAL.
               KeyMnHaKeyMnFa'.SignKeyMnHaKeyMnFa'}_K_FaAAAL)
      =|>
      State' := 3
      /\ Snd(KeyMnHaKeyMnFa'.SignKeyMnHaKeyMnFa')
      /\ wrequest(FA,AAAH,k_faha1,K_FaHa')
      /\ wrequest(FA,AAAH,k_mnfa1,K_MnFa')

end role
```

```
role aaa_MIP_AAAL (AAAL,AAAH,FA,MN: agent,
                   Snd, Rcv: channel(dy),
                   K_FaAAAL,K_AAAHAAAL: symmetric_key)
played_by AAAL
def=

  local
    State                : nat,
    K_MnFa,K_FaHa        : symmetric_key,
    SignedRegReq         : {agent.(agent.agent)}_symmetric_key,
    KeyMnFaKeyMnHa       : {symmetric_key.symmetric_key}_symmetric_key,
    SignedKeyMnFaKeyMnHa :
           {{symmetric_key.symmetric_key}_symmetric_key}_symmetric_key

  init State := 0

  transition

   1. State = 0
      /\ Rcv(FA.MN.AAAH.SignedRegReq')
      =|>
      State' := 1
      /\ Snd(FA.MN.AAAH. SignedRegReq')

   2. State = 1
      /\ Rcv( FA.{K_MnFa'.K_FaHa'}_K_AAAHAAAL.
              KeyMnFaKeyMnHa'.SignedKeyMnFaKeyMnHa'.
              {FA.{K_MnFa'.K_FaHa'}_K_AAAHAAAL.
               KeyMnFaKeyMnHa'.SignedKeyMnFaKeyMnHa'}_K_AAAHAAAL)
      =|>
      State' := 2
      /\ Snd( FA.{K_MnFa'.K_FaHa'}_K_FaAAAL.
                  KeyMnFaKeyMnHa'.SignedKeyMnFaKeyMnHa'.
             {FA.{K_MnFa'.K_FaHa'}_K_FaAAAL.
                  KeyMnFaKeyMnHa'.SignedKeyMnFaKeyMnHa'}_K_FaAAAL)

end role
```

```
role aaa_MIP_AAAH (AAAH,AAAL,HA,FA,MN : agent,
           Snd, Rcv : channel(dy),
           K_MnAAAH,
           K_AAAHAAAL,
           KAAAHHa : symmetric_key)
played_by AAAH
def=

  local  State                : nat,
         K_FaHa,K_MnHa,K_MnFa : symmetric_key

  const  secFAHA, secFAMN, secMNHA   : protocol_id

  init   State := 0

  transition

   1. State   = 0
      /\ Rcv(FA.MN.AAAH.{FA.MN.AAAH}_K_MnAAAH)
      =|>
      State' := 1
      /\ K_MnHa' := new()
      /\ K_MnFa' := new()
      /\ K_FaHa' := new()
      /\ Snd( MN.{K_MnHa'.K_FaHa'}_KAAAHHa.
              {K_MnFa'.K_MnHa'}_K_MnAAAH.
              {MN.{K_MnHa'.K_FaHa'}_KAAAHHa.
                  {K_MnFa'.K_MnHa'}_K_MnAAAH}_KAAAHHa)
      /\ witness(AAAH,FA,k_faha1,K_FaHa')
      /\ witness(AAAH,HA,k_faha2,K_FaHa')
      /\ witness(AAAH,FA,k_mnfa1,K_MnFa')
      /\ witness(AAAH,MN,k_mnfa2,K_MnFa')
      /\ witness(AAAH,MN,k_mnha2,K_MnHa')
      /\ witness(AAAH,HA,k_mnha1,K_MnHa')

   2. State = 1
      /\ Rcv( {K_MnFa.K_MnHa}_K_MnAAAH.
             {{K_MnFa.K_MnHa}_K_MnAAAH}_K_MnHa.
              {{K_MnFa.K_MnHa}_K_MnAAAH.
               {{K_MnFa.K_MnHa}_K_MnAAAH}_K_MnHa}_KAAAHHa)
```

```
      =|>
      State' := 2
      /\ Snd( FA.{K_MnFa.K_FaHa}_K_AAAHAAAL.{K_MnFa.K_MnHa}_K_MnAAAH.
                {{K_MnFa.K_MnHa}_K_MnAAAH}_K_MnHa.
              {FA.{K_MnFa.K_FaHa}_K_AAAHAAAL.{K_MnFa.K_MnHa}_K_MnAAAH.
                {{K_MnFa.K_MnHa}_K_MnAAAH}_K_MnHa}_K_AAAHAAAL)
      /\ secret(K_FaHa,secFAHA,{FA,HA})
      /\ secret(K_MnFa,secFAMN,{FA,MN})
      /\ secret(K_MnHa,secMNHA,{MN,HA})

end role
```

```
role aaa_MIP_HA (HA,AAAH,MN: agent,
                 Snd,Rcv: channel(dy),
                 K_AAAHHa: symmetric_key)
played_by HA
def=

  local
    State                 : nat,
    K_MnFa,K_FaHa, K_MnHa : symmetric_key,
    KeyMnFaKeyMnHa        : {symmetric_key.symmetric_key}_symmetric_key

  init State := 0

  transition

   1. State = 0
      /\ Rcv( MN.{K_MnHa'.K_FaHa'}_K_AAAHHa.KeyMnFaKeyMnHa'.
              {MN.{K_MnHa'.K_FaHa'}_K_AAAHHa.KeyMnFaKeyMnHa'}_K_AAAHHa)
      =|>
      State' := 1
      /\ Snd( KeyMnFaKeyMnHa'.{KeyMnFaKeyMnHa'}_K_MnHa'.
              {KeyMnFaKeyMnHa'.{KeyMnFaKeyMnHa'}_K_MnHa'}_K_AAAHHa)
      /\ wrequest(HA,AAAH,k_faha2,K_FaHa')
      /\ wrequest(HA,AAAH,k_mnha1,K_MnHa')
end role
```

```
role session(MN,FA,AAAL,AAAH,HA: agent,
             Kmn3ah,Kfa3al,K3ah3al,Kha3ah: symmetric_key) def=

   local     MNs,MNr,
             FAs,FAr,
             Ls, Lr,
             Hs, Hr,
             HAs, HAr: channel(dy)

   composition

           aaa_MIP_MN(MN,AAAH,FA,MNs,MNr,Kmn3ah)

        /\ aaa_MIP_FA(FA,AAAL,AAAH,MN,FAs,FAr,Kfa3al)

        /\ aaa_MIP_AAAL(AAAL,AAAH,FA,MN,Ls,Lr,Kfa3al,K3ah3al)

        /\ aaa_MIP_AAAH(AAAH,AAAL,HA,FA,MN,Hs,Hr,Kmn3ah,K3ah3al,Kha3ah)

        /\ aaa_MIP_HA(HA,AAAH,MN,HAs,HAr,Kha3ah)
end role
```

```
role environment() def=

  const k_mnha1, k_mnfa1, k_faha1                   : protocol_id,
        k_mnha2, k_mnfa2, k_faha2                   : protocol_id,
        mn, fa, aaal, aaah, ha                      : agent,
        k_mn_aaah, k_fa_aaal, k_aaah_aaal, k_ha_aaah : symmetric_key

  intruder_knowledge = {mn,fa,aaal,aaah,ha}

  composition

        session(mn,fa,aaal,aaah,ha,
                k_mn_aaah,k_fa_aaal,k_aaah_aaal,k_ha_aaah)

end role
```

```
goal

  %secrecy_of K_MnFa, K_FaHa, K_MnFa
  secrecy_of secFAHA, secFAMN, secMNHA
  %AAA_MIP_FA weakly authenticates AAA_MIP_AAAH on k_faha1
  weak_authentication_on k_faha1
  %AAA_MIP_FA weakly authenticates AAA_MIP_AAAH on k_mnfa1
  weak_authentication_on k_mnfa1
  %AAA_MIP_HA weakly authenticates AAA_MIP_AAAH on k_faha2
  weak_authentication_on k_faha2
  %AAA_MIP_HA weakly authenticates AAA_MIP_AAAH on k_mnha1
  weak_authentication_on k_mnha1
  %AAA_MIP_MN weakly authenticates AAA_MIP_AAAH on k_mnha2
  weak_authentication_on k_mnha2
  %AAA_MIP_MN weakly authenticates AAA_MIP_AAAH on k_mnfa2
  weak_authentication_on k_mnfa2

end goal

environment()
```

References

[CJP03] Pat Calhoun, Tony Johansson, and Charles Perkins. Diameter Mobile IPv4 Application, October 2003. Work in Progress.

[Per03] Charles Perkins. Mobile IPv4 Challenge/Response Extensions (revised), October 2003. Work in Progress.