IEEE P802.11
Wireless LANs

|  |
| --- |
| GCM is Missing From The Pseudocode |
| Date: 2013-09-16 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | Aruba Networks | 1322 Crossman ave, Sunnyvale, CA  | +1 408 227 4500 | dharkins at aruba networks dot com |
|  |  |  |  |  |

Abstract

The representative of the Swiss National Body in ISO commented that 11ad added GCM to 802.11 but that GCM is not in the RSNA frame pseudocode.

### 11.8.2.2 Per-MSDU/Per-A-MSDU Tx pseudo-code

// Note that it is assumed that no entry in the key

// mapping table is of an unsupported cipher type

Set the Key ID subfield of the IV field to 0.

if cipher type of entry is AES-CCM then

 Transmit the MSDU or A-MSDU, to be protected after fragmentation using AES-CCM

else if cipher type of entry is AES-GCM then

 Transmit the MSDU or A-MSDU, to be protected after fragmentation using AES-GCM

else if cipher type of entry is TKIP then

 Compute MIC using michael algorithm and entry’s Tx MIC key.

Append MIC to MSDU

Transmit the MSDU, to be protected with TKIP

else if cipher type of entry is WEP then

 Transmit the MSDU, to be protected with WEP

endif

…

if MPDU has an individual RA and cipher type of entry is not TKIP then

discard the entire MSDU or A-MSDU and generate one or more MAUNITDATA-

STATUS.indication primitives to notify the LLC that the

MSDUs were undeliverable due to a null key

 else if cipher type of entry is AES-CCM then

Transmit the MSDU or A-MSDU, to be protected after fragmentation using AES-CCM

else if cipher type of entry is AES-GCM then

 Transmit the MSDU or A-MSDU, to be protected after fragmentation using AES-GCM

else if cipher type of entry is TKIP then

Compute MIC using michael algorithm and entry’s Tx MIC key.

Append MIC to MSDU

Transmit the MSDU, to be protected with TKIP

else if cipher type of entry is WEP then

Transmit the MSDU, to be protected with WEP

endif

### 11.8.2.4 Per-MPDU Tx pseudo-code

if dot11RSNAActivated = TRUE then

if MPDU is member of an MSDU that is to be transmitted without protections

transmit the MPDU without protections

 else if MSDU or A-MSDU that MPDU is a member of is to be protected using AES-CCM

Protect the MPDU using entry’s key and AES-CCM

Transmit the MPDU

 else if MSDU or A-MSDU that MPDU is a member of is to be protected using AES-GCM

 Protect the MPDU using entry’s key and AES-GCM

 Transmit the MPDU

else if MSDU that MPDU is a member of is to be protected using TKIP

Protect the MPDU using TKIP encryption

Transmit the MPDU

 else if MSDU that MPDU is a member of is to be protected using WEP

Encrypt the MPDU using entry’s key and WEP

Transmit the MPDU

 else

// should not arrive here

endif

endif

### 11.8.2.6 Per-MPDU Rx pseudo-code

if dot11RSNAActivated = TRUE then

if the Protected Frame subfield of the Frame Control field is 0 then

if Protection for TA is off for Rx then

Receive the unencrypted MPDU without protections

else

discard the frame body without indication to LLC and increment dot11WEPExcludedCount

endif

else if Protection is true for TA then

if ((MPDU has individual RA and Pairwise key exists for the MPDU’s TA) or

 (MPDU has a group addressed RA and network type is IBSS and IBSS GTK exists

 for MPDU’s RA)) then

if MPDU has individual RA then

 lookup pairwise key using Key ID from MPDU

else

lookup group key using Key ID from MPDU

 endif

 if key is null then

discard the frame body and increment dot11WEPUndecryptableCount

 else if entry has an AES-CCM key then

decrypt frame using AES-CCM key

discard the frame if the integrity check fails and increment

dot11RSNAStatsCCMPDecryptErrors

 else if entry has an AES-GCM key then

 decrypt frame using AES-GCM key

 discard the frame if the integrity check fails

 else if entry has a TKIP key then

prepare a temporal key from the TA, TKIP key and PN

decrypt the frame using ARC4

discard the frame if the ICV fails and increment **d**ot11RSNAStatsTKIPLocal-MicFailures

else if entry has a WEP key then

decrypt the frame using WEP decryption

discard the frame if the ICV fails and increment dot11WEPICVErrorCount

else

discard the frame body and increment dot11WEPUndecryptableCount

 endif

 else if GTK for the Key ID does not exist then

 discard the frame body and increment dot11WEPUndecryptableCount

 else if GTK for the Key ID is null then

 discard the frame body and increment dot11WEPUndecryptableCount

 else if the GTK for the Key ID is a CCM key then

 decrypt frame using AES-CCM key

 discard the frame if the integrity check fails and increment dot11RSNAStatsCCMPDecryptErrors

 else if the GTK for the Key ID is a GCM key then

 decrypt frame using AES-GCM key

else if the GTK for the Key ID is a TKIP key then

 prepare a temporal key from the TA, TKIP key and PN

 decrypt the frame using ARC4

 discard the frame if the ICV fails and increment dot11RSNAStatsTKIPICVErrors

 else if the GTK for the Key ID is a WEP key then

 decrypt the frame using WEP decryption

 discard the frame if the ICV fails and increment dot11WEPICVErrorCount

 endif

else

 else

 MLME-PROTECTEDFRAMEDROPPED.indication

 discard the frame body and increment dot11WEPUndecryptableCount

 endif

endif

### 11.8.2.7 Per-MPDU Rx pseudo-code for an MMPDU

else // STA with frame TA advertised MFPC = 1

if (MMPDU has an individual RA) then

if (Pairwise key does not exist) then

if (frame is a Disassociation or Deauthentication) then

If (Protected Frame subfield of the Frame Control field is equal to 0) then

Make the MPDU available for further processing

 else // encrypted

Discard the frame

 endif

else // frame is not a Disassociation or Deauthenticate

Discard the frame

endif

else if (security association has an AES-CCM key) then

if (Protected Frame subfield of the Frame Control field is equal to 0) then

//unprotected frame

Discard the frame

 else // frame is encrypted

 if (PN is not sequential) then

Discard the MPDU as a replay

Increment dot11RSNAStatsCCMPReplays

else

Decrypt frame using AES-CCM key

 if (the integrity check fails) then

Discard the frame

Increment dot11RSNAStatsCCMPDecryptErrors

else

Make the MPDU available for further processing

endif

endif

endif

else if (security association has AES-GCM key) then

 if (Protected Frame subfield of the Frame Control field is equal to 0) then

 //unprotected frame

 Discard the frame

 else //frame is encrypted

 if (PN is not sequential) then

 Discard the MPDU as a replay

 Increment dot11RSNAStatsGCMPReplays

 else

 Decrypt frame using AES-GCM key

 If (the integrity check fails) then

 Discard the frame

 else

 Make the MPDU available for further processing

 endif

 endif

 endif

else // key for some other cipher—for future expansion

 endif

else // MMPDU has a group RA

if (IGTK does not exist) then

if (Disassociation or Deauthentication) then

Make frame available for further processing

else

Discard the frame

endif

else // IGTK exists

if (MME is not present) then

Discard the frame

else // MME is present

if (AES-128-CMAC IGTK) then

if (IPN is not valid) then

Discard the frame as a replay

Increment dot11RSNAStatsCMACReplay

else if (integrity check fails) then

Discard the frame

Increment dot11RSNAStatsCMACICVError

else

Make frame available for further processing

endif

else // some other kind of key—for the future

endif

endif

endif

### 11.8.2.8 Per-MSDU/Per-A-MSDU Rx pseudo-code

if dot11RSNAActivated = TRUE then

if the frame was not protected then

Receive the MSDU or A-MSDU unprotected

Make MSDU(s) available to higher layers

else if address1 has an individual RA then // Have a protected MSDU or A-MSDU

if Pairwise key is an AES-CCM key then

Accept the MSDU or A-MSDU if its MPDUs had sequential PNs (or if it consists of

only one MPDU), otherwise discard the MSDU or A-MSDU as a replay attack and

increment dot11RSNAStatsCCMPReplays

Make MSDU(s) available to higher layers

else if Pairwise key is an AES-GCM key then

 Accept the MSDU or A-MSDU if its MPDUs had sequential PNs (or if it consists of

 only one MPDU), otherwise discard the MSDU or A-MSDU as a replay attack and

 increment dot11RSNAStatsGCMPReplays

else if Pairwise key is a TKIP key then

Compute the MIC using the michael algorithm

Compare the received MIC to the computed MIC

discard the frame if the MIC fails increment dot11RSNAStatsTKIPLocalMICFailures

and invoke countermeasures if appropriate

compare TSC to replay counter, if replay check fails increment dot11RSNAStatsTKIPReplays

otherwise accept the MSDU

Make MSDU available to higher layers

else if dot11WEPKeyMappings has a WEP key then

Accept the MSDU since the decryption took place at the MPDU

Make MSDU available to higher layers

endif

else // Have a group addressed MSDU or A-MSDU

if GTK for the Key ID does not exist then

discard the frame body and increment dot11WEPUndecryptableCount

else if GTK for the Key ID is null then

discard the frame body and increment dot11WEPUndecryptableCount

else if GTK for the Key ID is a CCM key then

Accept the MSDU or A-MSDU if its MPDUs had sequential PNs (or if it consists of

only one MPDU), otherwise discard the MSDU or A-MSDU as a replay attack and

increment dot11RSNAStatsCCMPReplays

Make MSDU(s) available to higher layers

else if GTK for the Key ID is a GCM key then

 Accept the MSDU or A-MSDU if its MPDUs have sequential PNs (or if it consists of

 only one MPDU), otherwise discard the MSDU or A-MSDU as a replay attack and

 increment dot11RSNAStatsGCMPReplays

else if GTK for the Key ID is a TKIP key then

Compute the MIC using the michael algorithm

Compare the received MIC to the computed MIC

discard the frame if the MIC fails increment dot11RSNAStatsTKIPLocalMICFailures and

invoke countermeasures if appropriate

compare TSC to replay counter, if replay check fails increment dot11RSNAStatsTKIPReplays

otherwise accept the MSDU

Make MSDU available to higher layers

else if GTK for the Key ID is a WEP key then

Accept the MSDU since the decryption took place at the MPDU

Make MSDU available to higher layers

endif

endif

endif

### 11.8.2.9 Per-MMPDU Rx pseudo-code

if (dot11RSNAActivated = TRUE) then

if (dot11RSNAProtectedManagmentFramesActivated = TRUE) then

if (the MPDU was not protected) then

Receive the MMPDU unprotected

Make the MMPDU available to higher layers

else //Have a protected MMPDU

if ((MMPDU has individual RA) and (security association has an AES-CCM key)) then

if (the MPDU has only one MPDU or multiple MPDUs with sequential PNs) then

Receive the MMPDU protected

Make the MMPDU available to higher layers

else

Discard the MMPDU as a replay

Increment dot11RSNAStatsRobustMgmtCCMPReplays

endif

else if ((MMPDU has individual RA) and (security association has an AES-GCM key)) then

 if (the MPDU has only one MPDU or multiple MPDUs with sequential PNs) then

 Receive the MMPDU protected

 Make the MMPDU available to higher layers

 else

 Discard the MMPDU as a replay

 Increment dot11RSNAStatsRobustMgmtGCMPReplays

else if ((MPDU has group addressed RA) and (security association has an AES-128-CMAC IGTK)) then

Receive the MMPDU

Make the MMPDU available to higher layers

else

if (any other cipher exists) then

Process the frame using other cipher

else

Discard the frame

endif

endif

endif

endif

endif **References:**