**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | Comment resolution on Security CIDs | |
| Date Submitted | 16, March, 2016 | |
| Source | Jae Seung Lee ETRI  218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, Korea | Voice: +82 42 860 1326 Fax: +82 42 860 1326 E-mail: jasonlee@etri.re.kr |
| Re: | LB114\_Consolidated\_Comments | |
| Abstract | Proposes comment resolution on Security CIDs. | |
| Purpose | To be used by the technical editor to apply the necessary changes to the draft. | |
| Notice | This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. | |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | |

**List of contributors**

|  |  |
| --- | --- |
| **Name** | **Affiliation** |
| Jae Seung Lee | ETRI |
| Moon-Sik Lee | ETRI |
| Yeong Jin Kim | ETRI |
| Itaru Maekawa | Japan Radio Co., Ltd |
| Lee Doohwan | NTT Corporation |
| Ken Hiraga | NTT Corporation |
| Hideki Toshinaga | NTT Corporation |
| Keitarou Kondou | Sony Corporation |
| Hiroyuki Matsumura | Sony Corporation |
| Makoto Noda | Sony Corporation |
| Masashi Shinagawa | Sony Corporation |
| Ko Togashi | Toshiba Corporation |
| Kiyoshi Toshimitsu | Toshiba Corporation |
|  |  |

Security CIDs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CID** | **Page** | **Sub-clause** | **Line #** | **Comment** | **Proposed Change** | **Resolution Status** |
| 74 | 76 | 9a.2.4 | 21 | CGM IV nonce ("IV") rquirements differ widely from the CCM Nonce format. Although the GCM IV format specified here is likely to guarentee uniqueness, it does not conform to the GCM specification (NIST SP800-38D). SP800-38D has very specific rules in how the nonce can be constructed. Due to this variance, an implementation that does not follow those rules is highly unlikely to be usable by any organization requiring implementations to have a NIST FIPS 140 certificate. Please note that NISP FIPS 140 guidance is often carried into other security requirements, such as Common Critiera profiles, so this is an international consideration (This is an issue the 802.1 group has recently dealt with regarding MACsec frame security.) | To meet the deterministic IV construction requirements in Clause 8.2.1 of SP800-38D, the IV should be formed of just two fields: a fixed field and an invocation field. The SrcID could meet the requirement of the fixed field and the invocation field could be the secure frame counter (but together they would need to compose 12 octets). Additionally, the SrcID should be chosen by the station itself and be guarenteed to be globally unique. The Time Token should be placed within the authenticated portion of the frame (in either the AAD or the encrypted data, whichever is appropriate). | Revised.  Changed the text accordingly. See the modified text in 15-16-0264r2 |
| 75 | 77 | 9a.3.1 | 9 | Encrypting a key with GCM requires a unique nonce, and network security systems typically send the nonce with the key. The Encrypted Key value here only consists of the 16 octets of key. | There should be a note describing how the nonce is guarenteed to be unique OR that the key encrypting the key is never used more than once (which makes nonce reuse moot). (I cannot find this note anywhere else in 802.15.3 or this document). | Revised.  The nonce specified in 9a.2.4 is also used for secure command frames including key distribution related command frames, and the management key is used to for the frame. The nonce design in 9a.2.4 is guaranteed to be unique.  It is better to add some notes on it.  Changed the text accordingly. See the modified text in 15-16-0264r2 |
| 76 | 79 | 9a.4.1 | 3 | Is the heading for the table intended to be inputs for "GCM" rather than "CCM"? | Change CCM to GCM, if applicable | Accepted |
| 77 | 79 | 9a.4.2 | 26 | This comment is from this line until the end of thes section. The text seems to re-state the text in SP800-38D describing how GCM works. It would be much easier to analyze for correctness if it simply pointed to the definitive algorithm specification (either SP800-38D or an equivilent ISO document) rather than re-writing the algorithm. | Replace the description of how the encyption and decyrption algorithms work with a reference to an authoritative crypto algorithm reference. | Revised.  The baseline also re-states the text in CCM spec to describe how CCM works. It is better to clearly indicate that the algorithm specified in SP800-38D shall be used for GCM, and indicate that the re-stated part is just informative overview of the GCM algorithm.  Changed the text accordingly. See the modified text in 15-16-0264r2 |
| 79 | 36 | 6.3.1.2a | 22 | The SEC field is very badly named, as there is already SEC field in the Frame Control Field. In other places this field is called SFC or Secure Frame Counter | Change “SEC” with “SFC” | Accepted |
| 80 | 40 | 6.3.4a.2 | 13 | In the aggregated frame we use the default MAC subheader, which contains 9-bit sequence number, and I think that sequence number is used as 8-bit secure subframe counter when generating the nonce.  This is not really specified anywhere, the secure subframe counter is specified to be incremented for every subframe, but nothing is said whether it needs to match the sequence number of the frame. Sequence number in the subframe is NOT specified to be strictly increasing or anything like, it is just sequence number of the subframe.  Also the sequence number is 9-bit field, but the secure subframe counter is only 8-bit field, so there can be only 256 subframes in secure subframe, even when the sequence number field would allow 512 subframes. | Add text limiting the number of subframes in the secure aggregated data frames to 256 subframes. Specify whether the sequence number and secure subframe number needs to match or not. | Revised.  The number of subframes in the aggregated frame is limited to 256 in the current spec, so it is not necessary to mention the limit for secure subframes.  Secure subframe counter has been removed in the revised draft.  It is better to describe that the SFC and the sequence number do not need to match.  Changed the text accordingly. See the modified text in 15-16-0264r2 |
| 104 | 75 | 9a | 6 | SFC has been increased to 4 octets, but it may not be sufficient for 100 Gbps peak throughput. | Consider to increase the SFC to 6 octets | Revised.  Changed the text accordingly. See the modified text in 15-16-0264r2 |

**Discussion on CID 74 and 104 (Nonce)**

* Nonce format in 15.3e D1.0 (12 octets)
  + For non-aggregated frame

|  |  |  |  |
| --- | --- | --- | --- |
| **Octets: 1** | **6** | **4** | **1** |
| SrcID | Time Token | Secure Frame  Counter (SFC) | Padding |

* + For aggregated frame

|  |  |  |  |
| --- | --- | --- | --- |
| **Octets: 1** | **6** | **4** | **1** |
| SrcID | Time Token | Secure Frame  Counter (SFC) | Secure  Subframe Counter |

* The Secure Subframe Counter is incremented for each subframe in an aggregated frame, starting from 0
* The nonce should be formed of just two fields: a fixed field and an invocation field
  + Current fixed field: SrcID which is assigned by the PPC which is unique during one associated phase, but it is not globally unique.

🡪 MAC address (6 octet) should be used as a fixed field since it is guaranteed to be globally unique

* + Current invocation field: Time Token + SFC ( + Secure Subframe Counter)
  + Analysis on current 4 octet SFC during one associated phase in the worst case scenario:
    - The worst case scenario: aggregation is not used, one frame consists of just one AES block (128 bit)
    - Total number of octets that can be encrypted using a single Key without SFC roll-over:
      * 232 \* 24 octets = 236 octets
    - Duration with 100 Gbps throughput (worst case):
      * 236 \* 8 bit / 100000000000 sec = 239 / 1011 sec = 5.5 sec
        + SFC should be increased to 6 octet
* If SFC is 6 octet,
  + Total number of octets that can be encrypted using a single Key without SFC roll-over:
    - 248 \* 24 octets = 252 octets
  + Duration with 100 Gbps throughput (worst case):
    - 252 \* 8 bit / 100000000000 sec = 255 / 1011 sec = 360,288 sec = 100 hours = 4.17 days
  + Including Time Token in the nonce is not necessary.
    - Since the key should be updated when 6 octet SFC reaches its max value.
    - It is not recommended to use a single AES key for more than 248 frames. Re-keying is necessary if the AES key is used for more than 248 frames.
      * + If a single AES key is used 248 times in the above scenario, the maximum duration that can be covered by the AES key is 4.17 days
* Proposed Nonce format

|  |  |
| --- | --- |
| **Octets: 6** | **6** |
| **Source MAC address** | Secure Frame Counter (SFC) |

* + Time Token is not used in the nonce.
    - SFC is not reset when the time token is updated by the PPC.
  + Secure Subframe Counter is not used.
    - The SFC for the first subframe is included in the security header of the secure aggregated frame.
    - SFC for other subframes can be determined internally.
      * The same approach is used in the baseline, but Secure Subframe Counter is not used and the SFC is increased instead even for the subframes in the aggregated frame.

. To fully utilize 6 octet counter space.

* + No change to other part in the current spec.

**Proposed text changes (based on 802.15RevA-D01 and P802.15.3e D0.1)**

4.3.5 Security overview

*(Page 20, at the end of 4.3.4)*

The payload protection protocol, as described in 9.2.2 for piconets and 9a.2.2 for P2PLinks, uses a symmetric key that is generated by the key originator and is securely distributed to DEVs that have established secure membership or a secure relationship with the key originator, as described in 8.4.2. For non-HRCP, ~~T~~the symmetric key encryption algorithm used is the advanced encryption standard (AES) 128 in counter mode encryption and cipher block chaining-message authentication code (CCM). For HRCP, GCM (Galois/Counter Mode) mode of the AES-128 is used.

1. MAC frame formats

**6.2 General frame format**

*(Page 31, line 1)*

|  |  |
| --- | --- |
| **Octets: 8** | **variable** |
| Security Header | Frame Payload |
| Secure MAC frame body | |

**Figure 6-5a —Secure MAC frame body format for HRCP**

*(Page 33, line 27)*

**6.2.7a.1 Security Header**

|  |  |
| --- | --- |
| **Octet: 2** | **6** |
| SECID | SFC |

**Figure 6-8a—Security Header**

*Change the paragraph in 6.2.7.2 as follows:*

6.2.7.2 Secure session ID (SECID) field

The Key Originator field for all keys except the piconet group data key or the P2Plink group data key shall be set to the DEVID of the key originator in the relationship. The Key Originator field for the piconet group data key or the P2Plink group data key shall be set to the BcstID.

*Change the paragraph in 6.2.7.3 as follows:*

6.2.7.3 Secure Frame Counter (SFC) field

The Secure Frame Counter field contains a counter that is used to ensure the uniqueness of the nonce in a secure frame. For non-HRCP, a~~A~~ DEV shall not reuse a frame counter with the same time token, as described in 6.3.1.1, and key, as described in 8.3.5. For HRCP, a DEV shall not reuse a frame counter with the same key, as described in 8.3.5. For non-HRCP, t~~T~~he DEV shall initialize the SFC to zero for the first frame sent and increment it for each successive secure frame sent. For HRCP, the DEV shall initialize the SFC to zero for the first frame sent and increment it for each successive secure frame sent and each successive subframe sent in the aggregated frame. For non-HRCP, w~~W~~hen the time token, as described in 6.3.1, is updated, the DEV shall reset the SFC to zero. For HRCP, the SFC shall be increased even when the time token is updated. In the case where the DEV receives a new key, the DEV shall set the SFC to zero.

*(Page 36, Line 19 and 22)*

**6.3.1.2a Secure beacon frame for HRCP**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Octets: 2** | **6** | **6** | **14** | **variable** | **…** | **variable** | **16** | **4** |
| SECID | SFC | Time Token | P2P synchronization  parameters | Information  element-1 |  | Information  element-n | Integrity Code | FCS |

**Figure 6-51a—Secure beacon frame format for HRCP**

The SFC field is used by the DEV for this frame to ensure uniqueness of the nonce, as defined in 6.2.7.3.

*(Page 38, Line 4)*

**6.3.3a.2 Secure command frame**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Octet: 2** | **6** | **4** | **variable** | **16** | **4** |
| SECID | SFC | MAC Subheader | Command block | Integrity code | FCS |

**Figure 6-56c—Secure command frame format**

*(Page 40, Line 8)*

**6.3.4a.2 Secure Aggregated HRCP Data frame**

|  |  |  |
| --- | --- | --- |
| **Octets: 2** | **6** | **variable** |
| SECID | SFC | Secure HRCP Aggregated data Frame |

**Figure 6-58g— Frame Payload field format for Secure Aggregated Data Frame**

The SFC is defined in 6.2.7.3. Only the SFC of the first subframe is included in the secure aggregated data frame.

*Change the paragraph in 6.5.2.3 as follows:*

6.5.2.3 Distribute Key Request command

The Distribute Key Request command is used to transmit a key to another DEV. The SEC field in the Frame Control field shall be set to one. This command may have the ACK Policy field set to no-ACK only if the source ID is the PNCID or PPCID. This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The Distribute Key Request command Payload filed shall be formatted as illustrated in Figure 6-131.

*Change the paragraph in 6.5.4.4 as follows:*

6.5.4.4 Security Information command

The DEVID field contains the ID assigned to the DEV by the PNC or PPC. If the DEV is not currently associated in this piconet or P2Plink, the field shall be set to the UnassocID. This field shall not contain the broadcast or multicast DEVIDs.

*Change the paragraph in 6.5.9.1 as follows:*

6.5.9.1 Security Message command

The Security Message command is used to send security-related information to another DEV in the piconet or P2Plink.

*Change the paragraph in 8.1.4 as follows:*

8. Security

8.1.4 Data integrity

Data integrity uses an integrity code to protect data from being modified by parties without the cryptographic key. It further provides assurance that data came from a party with the cryptographic key. Integrity may be provided using a key shared by all piconet DEVs or using a key shared by all P2Plink DEVs, or using a key shared between only two DEVs. All secure data frames that fail integrity checks are passed to the DME using MLME-SECURITY-ERROR.indication, and no other action is taken on the frame by the MLME.

**8.1.7 Freshness protection**

***Change the first paragraph in 8.1.7 as shown:***

To prevent replay of old messages, a strictly-increasing time token is included in the beacon. A DEV shall reject as invalid a received beacon with a time token less than or equal to the current time token. For HRCP, a DEV shall further check the SFC and the SECID included in the beacon and shall reject as invalid the beacon if the SFC in the beacon is not strictly greater than the last SFC received from that DEV corresponding to the key identified by the SECID. In addition, for non-HRCP, the time token is included in the nonce, as described in 9.2.4, for each secure frame, as described in 6.2, so the integrity check will fail if a frame is replayed in a different superframe. For HRCP, a DEV shall check the SFC and the SECID included in each secure frame, and shall reject as invalid the received frame if the SFC in the frame is not strictly greater than the last SFC received from that DEV corresponding to the key identified by the SECID to detect wheter the frame is replayed or not. A DEV maintains two values for freshness. The CurrentTimeToken is the time token value found in the beacon for the current superframe and is used to protect all messages sent and check all messages received during that superframe. For HRCP, the values are used only to check beacon freshness and the SFC is used to check freshness of other frames. The LastValidTimeToken is used by the DEV to ensure that the security of the beacons have not been compromised.

**8.3.5 Secure frame generation**

*(page 70, line 18)*

If the non-HRCP DEV is able to obtain the appropriate keying material, the DEV shall use the CurrentTimeToken and secure frame counter for the corresponding SECID to construct the CCM nonce, Figure 9-1, used to protect the secure frame. If the HRCP DEV is able to obtain the appropriate keying material, the DEV shall use the secure frame counter for the corresponding SECID to construct the GCM nonce, Figure 9a-1, used to protect the secure frame. The SECID included in the frame shall be the value corresponding to the keying material being used.

*Change the last paragraph of 8.3.5 of the baseline as follows: (Insert at the end of clause 8.3.5 of 802.15.3e D1.0)*

A non-HRCP DEV shall send only frames that have increasing SFCs in a superframe, except for frames that are retransmitted with the same SFC without any intervening frames having been sent. An HRCP DEV shall send only frames that have increasing SFCs for a single key corresponds to the SECID indicated in the transmitted frames.

**8.3.7 Secure frame reception**

(Page 71, line 10)

When a DEV receives a secure beacon frame, as defined in 6.3.1.2, the DEV shall determine if the received time token is greater than the CurrentTimeToken and less than the LastValidTimeToken + mMaxTimeTokenChange. If not, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to BAD-TIME-TOKEN and shall not perform any additional operations on the received beacon. The DEV shall also determine if the SECID matches the SECID of the piconet group data key or P2Plink group data key stored in the MAC/MLME, or the SECID of a valid old piconet group data key or old P2Plink group data key, as described in 8.3.5. If the SECID matches, an HRCP DEV shall further check the SFC included in the beacon and the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to BAD-TIME-TOKEN and shall not perform any additional operations on the received beacon if the SFC in the beacon is not strictly greater than the last SFC received from that DEV corresponding to the key identified by the SECID. If the SECID does not match, the DEV may request a new piconet group data key or new P2Plink group data key, as described in 8.3.2. If ~~both of~~ these checks succeed, the DEV shall check the integrity code on the beacon using the piconet group data key or P2Plink group data key. If this succeeds, the DEV shall accept the beacon and set the LastValidTimeToken and CurrentTimeToken to be the time token in the beacon.

*Change the last paragraph of 8.3.7 of the baseline as follows: (Insert at the end of clause 8.3.7 of 802.15.3e D1.0)*

When a DEV receives a secure non-Beacon frame, it shall use the appropriate keying material depending on the type of frame, SECID, and SrcID found in the frame. If the SECID in the frame does not correspond to known keying material in the receiving DEV, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to UNAVAILABLE-KEY and shall not perform any additional operations on the received frame. For non-HRCP, a~~A~~ DEV shall reject all frames that do not have an SFC that is strictly greater than the last SFC received from that DEV in that superframe. For HRCP, a DEV shall reject all frames that do not have an SFC that is strictly greater than the last SFC received from that DEV corresponding to the key identified by the SECID in the received frames.

**9a. Security specifications for HRCP**

**9a.2.4 Nonce value**

In order to preserve the security of the symmetric algorithms, the nonce used for GCM encryption and authentication shall be unique for a given key. As a result, the DEV shall not reuse any Secure Frame Counter (SFC) field value with a given key that is intended for a particular DEV address (as this would cause a repeated nonce).

This uniqueness is guaranteed by the use of the DEV address of the source DEV and the Secure Frame Counter (SFC). The DEV address is globally unique and guarantees that different two DEVs sharing the same key will use a different nonce.

The DEV address of the source DEV and the secure frame counter guarantee uniqueness of the nonce for a given key as long as a DEV does not send more than 248 frames to the other DEV in the P2Plink.

If a frame is retransmitted and a single bit in the header or frame body has been changed, a new nonce shall be used. To ensure this, each time a frame is retransmitted, the value of the Secure Frame Counter field shall be incremented.

The nonce that is input to the GCM algorithm shall be formatted as illustrated in Figure 9a-1.







.

|  |  |
| --- | --- |
| **Octets: 6** | **6** |
| DEV address (source DEV) | Secure Frame Counter (SFC) |

**Figure 9a-1—GCM nonce format**

The DEV address field shall be set to the DEV address of the source DEV.

The Secure Frame Counter field is set to the value of the SFC field that is included in the transmitted frame that is being protected. SFC field is defined in 6.2.7.3. If the transmitted frame is an aggregated frame, only the Secure Frame Counter of the first subframe is explicitly included in the aggregated frame, and the Secure Frame Counter value for other subframes shall be incremented for each subframe in the aggregated frame, starting from the value explicitly indicated in the SFC field of the received frame.

Note – The value of the Secure Frame Counter field is independent from the value of the sequence number in the MAC header and they do not need to match.

**9a.3.1 Symmetric cryptography data formats**

Table 9a-1— Symmetric cryptography frame object formats for GCM

|  |  |  |  |
| --- | --- | --- | --- |
| **Notation** | **Length** | **Value** | **Description** |
| Encrypted keya | 16 | Variable | The encrypted key consists of the result of the encryption of a 16-octet key (not including the integrity code) using GCM encryption, as specified in 9a.2.2. |
| Integrity code | 16 | Variable | The integrity code consists of the encrypted integrity code that is the result of a GCM computation, as specified in 9a.2.2, that is computed along with the encrypted seed. |
| Encrypted data | Variable | Variable | The encrypted data consists of the result of the encryption of the specified data (not including the integrity code) using GCM encryption, as specified in 9a.2.2. |

aNote: Encrypting a key with GCM requires a unique nonce. The Encrypted key is transmitted in secure command frames using a management key, and the Nonce defined in 9a.2.4, which is guaranteed to be unique, is used for secure command frames. A group data key and a management key may use separate Secure Frame Counter.

**9a.3.2 Symmetric cryptographic operations**



Figure 9a-2 specifies the length information and data input to the GCM operation for secure beacons. The Auth Data Length in octets, *l(a),* shall be set to the length of the Frame Header, SECID, SFC, Time Token, all of the P2P Synchronization Parameters field plus the sum of the lengths of the IEs that are included in the beacon. The Enc Data Length in octets, *l(p)*, shall be set to zero. The data input to GCM shall be taken in the order it is received in the frame, omitting the HCS, FCS and Integrity Code.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **6** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

**2**

Figure 9a-3 specifies the length information and data input to the GCM operation for secure commands. For all commands except for the Request Key Response command and Distribute Key Request command, the Auth Data Length, *l(a)*, shall be set to the length of all of the protected data including Frame Header, SECID, SFC, MAC Subheader, Command Type and Length plus the length of the Payload field in the command frame. The Enc Data Length, *l(p)*, shall be set to zero. For the Request Key Response command and Distribute Key Request command, the Auth Data Length, *l(a)*, shall be set to the length of all of the protected data minus 16 (the length of the key) and the Enc Data Length, *l(p)*, shall be set to 16 (the length of the key). The data input to GCM shall be taken in the order it is received in the frame, omitting the HCS for the Frame Header, FCS and Integrity Code.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Octets: 10** | **2** | **6** | **4** | **2** | **2** | **L1** | **L2** |  | **2** | **2** |
| Frame Header | SECID | SFC | MAC Subheader | Command Type | Length | Auth Data | Enc Data |  | Auth Data Length | Enc Data Length |

**Figure 9a-3—GCM input for secure commands**

Figure 9a-4 specifies the length information and data input to the GCM operation for secure data frames. The GCM operation is applied to each subframe in the data frame separately. For the first subframe, the Auth Data Length 1, *l1(a)*, which is the Auth Data Length for the first subframe, shall be set to 22 which is the length of the Frame Header, SECID, SFC, and the MAC Subheader of the first subframe, and the Enc Data Length 1, *l1 (p)*, which is the Enc Data Length for the first subframe, shall be set to the length of the Payload field in the first subframe.

For the *n*-th subframe, the Auth Data Length *n*, *ln(a)*, which is the Auth Data Length for the *n*-th subframe, shall be set to 4 which is the length of the MAC Subheader of the *n*-th subframe, and the Enc Data Length *n*, *ln (p)*, which is the Enc Data Length for the *n*-th subframe, shall be set to the length of the Payload field in the *n*-th subframe.

The data input to GCM for each subframe shall be taken in the order it is received in the frame, omitting the FCS and Integrity Code in the subframe.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Octets: 10** | **2** | **6** | **4** | **L1** |  | **2** | **2** |
| Frame Header | SECID | SFC | MAC Subheader 1 | Payload 1 |  | Auth Data Length 1 | Enc Data Length 1 |

***…………..***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Octets: 4** | **L*n*** |  | **2** | **2** |
| MAC Subheader *n* | Payload *n* |  | Auth Data Length *n* | Enc Data Length *n* |

**Figure 9a-4—GCM input for secure data frames**

Table 9a-2— Inputs for GCM

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Field size** | **Encoding of field** |
| *K* | Block cipher key | 16 octets | String of octets |
| *N* | Nonce | 12 octets | Not specified |
| *p* | Data to be encrypted | *l(p)* octets | String of octets |
| *a* | Additional authenticated data (AAD) | *l(a)* octets | String of octets |

**9a.4.2 Authenticated encryption**

*Insert the following sentence at line 23 in page 79 (before the sentence “The two main functions….”)*

The procedure defined in NIST Special Publication 800-38D shall be used for authenticated encryption. The procedure described below is provided as an informative overview of the authenticated encryption procedure.

*Insert the following sentence after line 21 in page 81*

**9a.4.5 Authenticated decryption**

The authenticated decryption operation is similar to the authenticated encryption operation, but with the order of the hash step and encrypt step reversed. The procedure defined in NIST Special Publication 800-38D shall be used for authenticated decryption. The procedure described below is provided as an informative overview of the authenticated decryption procedure.

*Change the figure number as follows: (line 6 of page 79)*

The inputs for encryption that are defined in 9a.3.2 shall be divided into 16-octet message blocks as shown in Figure 9a-5

**Figure 9a-5—Block ordering for encryption**

*Change the figure number as follows: (line 17 of page 79)*

Similarly, the inputs for AAD that are defined in 9a.3.2 shall be divided into 16-octet message blocks as 16 shown in Figure 9a-6.

**Figure 9a-6—Block ordering for AAD**