IEEE P802.11Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed resolution to 11az LB249 CIDs on Secure LTF | | | | |
| Date: 2021-01-04 | | | | |
| Author(s): | | | | |
| Name | Company | Address | Phone | email |
| Tianyu Wu | Apple |  |  | tianyu@apple.com |
| Anuj Batra |  |  |  |
| Mithat Dogan |  |  |  |
| Ahmet Cepni |  |  |  |
| Qi Wang |  |  |  |
| SK Yong |  |  |  |
| Vinko Erceg | Broadcom |  |  |  |
| Nehru Bhandaru |  |  |  |
| Luis Gutierrez |  |  |  |
| Manas Deb |  |  |  |
| Mengchang Doong |  |  |  |
| Hossein Roufarshbaf |  |  |  |
| Gadi Shor | Intel |  |  |  |
| Robert Stacey |  |  |  |
| Steve Shellhammer | Qualcomm |  |  |  |
| Ali Raissinia |  |  |  |
| Bin Tian |  |  |  |

Abstract

This submission proposes the resolution to 11az LB#249 CID 3215, 3354, 3911, 3920 and 4018. All of these comments are related to the secure LTF feature specified in the draft 11az spec.

The page and line numbers for proposed changes refer to those in 11az Draft 2.6 [1].

**Introduction**

This submission proposes the resolution to 11az LB#249 CID 3215, 3354, 3911, 3920 and 4018. All of these comments are related to the secure LTF feature specified in the draft 11az spec.

The page and line numbers for proposed changes refer to those in 11az Draft 2.6 [1].

Comments:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page/Line | Clause | Comment | Proposed change | Resolution |
| 3215 | 205/21 | 27.3.17c | 8 PSK is introduced for HE-LTF modudation. Is it reilable at low SNR condition? Also 8 PSK is not defined anywhere in the draft. Note that clause 27 HE PHY baseline doesn't' have 8 PSK | Either remove 8 PSK Modulation or clearly define it in the draft | Revised.  TGaz Editor: Please make the text changes included in 21/40r0. |
| 3354 | 208/6 | 27.3.17c | Not enough random bits in secure ranging. This can create security problems. | increase the randomness in secure waveform creation. | Revised.  TGaz Editor: Please make the text changes included in 21/40r0. |
| 3911 | 153/22 | 11.22.6.4 | The Secure LTF mechanism for TB and NTB ranging needs to be improved. A submission will be provided. | As in comment. | Revised.  TGaz Editor: Please make the text changes included in 21/40r0. |
| 3920 | 211/9 | 27.3.17d | Construction of Secure HE-LTF For multi stream, the construction of the HE-LTF dictates that ,no CSD is applied to the space-time streams. Without CSD, thus duplication (or perfectly negated version of the LTF sequence gets transmitted on the two spatial streams. This will result in construtive addition or destructive combing, resulting in signal boost or partial signal loss. | The secure LTF design needs to be adapted to multi stream mode. Present design works for SISO, but overlooks the MIMO scenario. Either a CSD per stream or another design is needed to make the secure LTF robust in MIMO mode. | Revised.  TGaz Editor: Please make the text changes included in 21/40r0. |
| 4018 | 205 | 27.3.17c | Introducing 8PSK into LTFs will be unique to all the other amendments (11g, 11n, 11ac, FTM, 11ax). | Redesign the Randomized LTF sequences so that 8PSK is not used. Use QPSK. | Revised.  TGaz Editor: Please make the text changes included in 21/40r0. |

**Discussion**

Current 11az Draft 2.6 has an LTF protection mechanism used for secure ranging that has been shown to be insecure. Several submissions have been presented to the IEEE 802.11az task group related to secure LTF design to address the deficiencies [3][4][6].

This discussion presents some rationale and proposed specification changes based on those submissions and is focused on generating an octet stream, from an 802.11 PTKSA, and used for protecting the LTFs used in 802.11az NTB or TB secure ranging. The changes are complementary to other submissions which describe how the octet stream is used by the 802.11 HE PHY to protect the LTFs.

The modulation and per-stream phase rotation schemes to provide acceptable security levels utilize a large number of secure bits derived from the current PTKSA between an RSTA and ISTA. It is necessary to add a secure LTF bit generator to the PHY Layer since the number of bits required exceed the capacity of the MAC/PHY control interface in many implementations.

The current proposal is to provide a mechanism for the MAC to provide a fixed number of bits to the PHY layer and use a secure bit generator in the PHY to derive the bits needed for the modulation and per-stream phase rotation.

The proposed bit generator is based on AES-128 Counter mode. This type of generator has been approved by NIST [6] to a generate random bit stream based on a block cipher (AES) and an initial IV that is incremented as each block of the output stream is generated – See Figure 12 CTR-DRBG [6]

![Diagram

Description automatically generated]()

However, simplified versions of AES-Counter mode-based secure bit generators have been used in 802.11 and other standards – without mechanisms for internal (key and IV) update with generation of a set of output blocks, reseeding, personalization etc. UWB [7] uses a 128-bit secret IV (sent encrypted in an IE) *phyHrpUwbStsVUpper96* with a 32-bit block counter *phyHrpUwbStsVCounter*) with a secret key to generate random TD pulses. IEEE 802.11 uses 128-bit IV in counter-mode encryption (e.g., GCM) using a 96-bit (Transmitter Address(A2) || Packet Number (Nonce) and a 32-bit block counter. In general, IV/Counter inputs to AES-Counter mode need not be secret or derived from a KDF but need to be unique for a given key. With GCM, a pseudo-random bit stream of up to (2^39 – 256) bits can be generated and used for encryption (and IPSEC ESP IEFF RFC allows up to 2^64 blocks to be encrypted with a single key with counter mode)

**Note to reviewers:** Please comment on any security issues from using a non-repeating 128-bit counter to generate the bits stream. Also comment on why NIST DRBG construction (that is even more complicated than a simple counter used in IPSEC ESP IETF RFC 3686 and GCM and using a random IV) allows only 2^19 bits per generate function. Note that if unique non-secret counter is not sufficient for secure LTF octet stream generation, some of the privacy guarantees provided by 802.11 MAC layer security using AES-CCM-128 or AES-GCM-128 will be void.

Use of AES-128 provides a security strength of 128 bits and is a familiar algorithm and already used in IEEE 802.11 to provide MAC layer security. The parameters used for generation as follows

* Key length = 128 bits
* A unique IV of length of 128 bits –The maximum number of bits required to construct an NDP with 64 Secure LTFs at 160MHz with 64-QAM is about 512k bits (8 Streams \* 8 Repetitions \* 1024 tones \* 8 (bits/Octet)– see Slide 41 in [3], although one may only need fewer bits per NDP transmission in practice.

A summary of the scheme that is proposed is below:

* The IV construction scheme is similar to base 802.11
  + First 48 bits of the IV are the transmitter MAC address (A2)
  + Second 48 bits of the IV are the Secure LTF Counter exchanged in the ranging negotiations.
  + A 4-Octet block counter is used. It allows up to 2^39 bits (2^32 AES 128bit Blocks) to be generated with a given key.
* 256 bits of key material are derived from the security association – derived from Secure-LTF-key-seed already specified in 11az Draft 2.6 – and provided for each measurement viz. NDP exchange
  + 128 bits are used as ISTA key in generating the secure bits for the I2R NDP
  + Another 128 bits are used as RSTA key in generating the secure bits for the R2I NDP
  + The two-key approach (compared to the proposal in [3]) removes the requirement to store the generator state across NDPs.
* Everything related to security derivations related to deriving secure LTF bits up to the derivation of secure LTF key seed remains the same as in the current draft.
* The key seed is used to derive 272 bits – 16 bits of SAC used in the protocol, and additional 256 bits that comprise of a key used by ISTA for Tx (and RSTA for Rx) and a key used by RSTA for Tx (and ISTA for Rx)
* The Secure LTF counter and keys are populated into Tx and Rx vectors as appropriate.

List of text changes:

1. Replace the section 11.21.6.4.5.4 Secure LTF Generation Information’ with a new section that describes secure LTF Octet Stream Generation.
2. Multiple KDKs may exist if extended key id features are implemented. In such as case, the most recent KDK is used in LTF key derivation.
3. Update to 11.21.6.4.5 Use of Secure LTF in the TB and Non-TB Ranging Measurement Exchange Protocol
   1. 11.21.6.4.5.2 TB ranging measurement exchange with Secure LTF
      1. p162.20 to p162.36 and p163.20 to p164.3
   2. Also, Non-TB – p168.17 to p168.47 and p169.6 to p169.22
4. Remove TXVECTOR.LTF\_SEQUENCE and RXVECTOR.LTF\_SEQUENCE and add LTF\_KEY and LTF\_IV as input – Table 27-1 Tx and Rx Vector Parameters
5. Remove J.14 remove LTF generation text vectors from p257.22 to p258 – keep the ltf key seed – add a test vector for the key derivation of 272 bits.
6. Modify subclause 27.3.2 and 27.3.3 to specify how PHY uses the random bits stream.

**Proposed Changes**

**TGaz Editor: Replace the section title for § 11.21.6.4.5.4 Secure LTF Generation Information with ‘Secure LTF Octet Stream Generation’.**

**TGaz Editor: Replace the text in § 11.21.6.4.5.4 Secure LTF Generation Information with the following text – it includes two new sub-clauses, one for ISTA and another for RSTA**

This clause describes mechanisms for generating the SAC, LTF protection keys, counters, and the pseudo random octet stream used to randomize the input to modulation and per-stream phase rotation for constructing Secure LTFs. The mechanism is illustrated in Figure 11-xx (ISTA Secure LTF Octet Stream Generation, and Figure 11-yy (RSTA Secure LTF Octet Stream Generation ).

For each secure measurement (e.g. NDP exchanges), a SAC and two secret keys *ista-ltf-key* and *rsta-ltf-key* shall be derived by the ISTA and the RSTA independently as follows.

SAC-and-LTF-Keys = KDF-Hash-Length(Secure-LTF-Key-Seed, “Secure LTF Expansion”, Secure-LTF-Counter)

Where

* KDF and Hash are the key derivation function and hash function determined by the AKM used to derive the PTKSA from which the Secure-LTF-Key-Seed was derived.
* Length is equal to 272 (bits)
* SAC = L(SAC-and-LTF-Keys, 0, 16)
* *ista-ltf-key* = L(SAC-and-LTF-Keys, 16, 128)
* *rsta-ltf-key* = L(SAC-and-LTF-Keys, 144, 128)

The *ista-ltf-key* is used to generate the pseudo random octet stream to protect all of the LTFs in PPDUs transmitted by the ISTA. The *rsta-ltf-key* is used to generate the pseudo random octet stream to protected all of the LTFs in PPDUs transmitted by the RSTA. ISTA and RSTA use the same derivation and derive identical keys.

With the SAC constructed as above, an attacker not knowing Secure-LTF-Key-Seed would not be able to predict the SAC that would be used for given measurement.

Integer to octet string conversion (MSB first) specified in 12.4.7.2.2 shall be used to encode the Secure-LTF-Counter input to the KDF as well as in the transmitted LTF sequence information. The counter shall be padded with leading (MSB) 0s to be exactly 6 octets.

The 16-octet IV input *ltf-iv* to the Secure LTF Bit Generator shall be constructed as follows

* First 6 octets shall be the transmitter MAC address (A2)
* Next 6 octets shall encode the Secure-LTF-Counter with the encoding convention described above.
* Final 4 octets shall be used as a 32-bit block counter. The block counter is initialized to 0 before NDP with secure LTFs is transmitted or received. The counter is incremented by 1 each time an AES block is output by the generator.

Each time pseudo random bits are required to protect the LTFs in an NDP, the Secure LTF Bit Generator generates the required number of AES blocks using the AES algorithm (see FIPS PUB 197) with the corresponding 128-bit key and 128-bit IV inputs. The output of the AES-128 counter encryption is a 128-bit integer. It shall be converted using the conventions specified in  12.4.7.2.2 (Integer to octet string conversion) to obtain the octet stream used for secure LTF generation.

NOTE— A 6 octet Secure LTF counter is sufficient because a unicast protected management frame that uses a 6 octet PN is used to convey the LTF sequence information that carries the counter.

NOTE—The pseudo random bit generator is based on AES-128 Counter mode approved by NIST (CTR-DRBG - NIST SP 800 90Ar1 - Recommendations for Random Number Generation).

The number of pseudo random bits that can be generated without violating security guarantees of the scheme is 2^39 without updating the key or reseeding the generator with additional entropy.

Secure LTF measurement requires variable number of octets from the pseudo random octet stream depending on the TXVECTOR or RXVECTOR parameters for the LTFs as described in Generation of Randomized LTF Sequence (27.3.18c Generation of Randomized LTF Sequence). The initial block counter value used to construct the *ltf-iv* for setting the TXVECTOR and RXVECTOR parameters shall be 0.

**11.21.6.4.5.4.1 Secure LTF Octet Stream Generation on an ISTA**

Figure 11-xx (ISTA Secure LTF Octet Stream Generation) illustrates the scheme for generating the SAC, LTF protection keys, and counter values used to generate a pseudo random octet stream on an ISTA.

Diagram

Description automatically generated

**Figure 11-xx ISTA Secure LTF Octet Stream Generation**

For each secure measurement on an ISTA, the following parameters are provided for generating the pseudo random octet stream used to construct the LTFs for NDPs

* the *ista-ltf-key* and *rsta-ltf-key* for transmitted NDP (TXVECTOR parameter LTF\_KEY) and received NDP (RXVECTOR parameter LTF\_KEY) respectively.
* the *ltf\_iv* for transmitted NDP (TXVECTOR parameter LTF-IV) and received NDP (RXVECTOR parameter LTF-IV) with corresponding ISTA MAC address and RSTA MAC address respectively together with the Secure LTF Counter and the block counter

**11.21.6.4.5.4.2 Secure LTF Input Octet Stream Generation on an RSTA**

The following Figure 11-yy (RSTA Secure LTF Octet Stream Generation) illustrates the scheme for generating the SAC, LTF protection keys, and counter values used to generate a pseudo random octet stream on an RSTA.

Diagram

Description automatically generated

**Figure 11-yy RSTA Secure LTF Octet Stream Generation**

For each secure measurement on an RSTA, the following parameters are provided for generating the pseudo random octet stream to construct the LTFs for NDPs

* the *ista-ltf-key* and *rsta-ltf-key* for received NDP (RXVECTOR parameter LTF\_KEY) and transmitted NDP (TXVECTOR parameter LTF\_KEY) respectively.
* the *ltf\_iv* for received NDP(RXVECTOR parameter LTF-IV) and for transmitted NDP (TXVECTOR parameter LTF-IV) with corresponding ISTA MAC address and RSTA MAC address respectively together with the Secure LTF Counter and the block counter.

NOTE—In a R2I NDP used for range measurement, LTFs assigned to each of the recipient STAs would use their corresponding pseudo random octet stream derived from the key material from the corresponding PTKSA for the recipient.

The Secure-LTF-Counter shall be maintained for the lifetime of a PTKSA used for Secure LTF measurements. It shall not be reset between measurements and shall not be reset for multiple FTM negotiations using the same PTKSA.

The Secure-LTF-Counter value used for each measurement protected bits derived from a given Secure-LTF-Key-Seed (and its KDK and the PTKSA) and shall be unique. When the derived SAC is equal to 0, the RSTA shall increment the Secure-LTF-Counter by 1 and derive the SAC until a nonzero SAC value is obtained. An RSTA shall also increment the Secure-LTF-Counter by 1 each time an *ista-ltf-key* and *rsta-ltf-key* are derived.

**TGaz Editor: Modify 11.21.6.3.4 Negotiation for Secure LTF in the TB and Non-TB Ranging measurement exchange p130.1 as follows**

**…**

When dot11SecureLTFImplemented is true, prior to generating the new Secure LTF Counter (#2289) for a given PTKSA, the RSTA initializes a monotonically increasing 48-bit counter Secure-LTF-Counter to 0. The RSTA also derives a Secure-LTF-Key-Seed as follows

Secure-LTF-Key-Seed = HMAC-Hash(KDK, “Secure LTF key seed”)

where KDK is derived as part of PTKSA establishment, Hash is the hash determined by the AKM and used to derive the PTK. If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is 1 for both the Authenticator and the Supplicant, there may be multiple KDKs each identified by a different Key ID (0 or 1). In such a case, the KDK chosen to derive the Secure-LTF-Key-Seed shall be the most recent.

**TGaz Editor: Change definition of ‘Null-SAC-HE-LTF’ as follows**

**p21.1**

**Null-SAC-HE-LTF** : An HE LTF present in I2R NDP or R2I NDP in the Ranging frame exchange where the SAC subfield in the STA Info field of Ranging NDP Announcement frame or the SAC subfield in the Trigger Dependent User Info field in the Ranging Secure Sounding Trigger frame doesn’t either match the value of the LTF Generation SAC subfield in the Secure LTF Parameters field in the last transmitted Fine Timing Measurement frame or last transmitted Location Measurement Report frame to the ISTA or is equal to 0 (#**3124**). The TXVECTOR parameters LTF\_KEY and LTF\_IV ~~LTF\_SEQUENCE~~ corresponding to this LTF ~~is~~ are set to generate ~~either the Secure-LTF-bits-I2R for generating~~ any secure HE-LTF or null

**TGaz Editor: Change 11.21.6.4.5.2 TB Ranging Measurement Exchange with Secure LTF, Figure 11-37o, Figure 11-37o, Figure 11-37p, Figure 11-37q and Figure 11-37r as follows:**

**Figure11-37o:**

**replace “LTF Sequence of R2I NDP\_1 is derived from SEC\_LTF\_CTR1” with “LTF Sequence of R2I NDP\_1 is derived from SEC\_LTF\_CTR1 & *rsta-ltf-key*”.**

**replace “LTF Sequence of R2I NDP\_2 is derived from SEC\_LTF\_CTR2” with “LTF Sequence of R2I NDP\_2 is derived from SEC\_LTF\_CTR2 & *rsta-ltf-key*”.**

**replace “LTF Sequence of I2R NDP\_1 is derived from SEC\_LTF\_CTR1 & LTF\_GEN\_SAC1” with “LTF Sequence of I2R NDP\_1 is derived from SEC\_LTF\_CTR1 & *ista-ltf-key*”.**

**replace “LTF Sequence of I2R NDP\_2 is derived from SEC\_LTF\_CTR2 & LTF\_GEN\_SAC2” with “LTF Sequence of I2R NDP\_2 is derived from SEC\_LTF\_CTR2 & *ista-ltf-key*”.**

**Figure 11-37p**

**replace “LTF Sequence of R2I NDP\_2 derived from SEC\_LTF\_CTR2” with “LTF Sequence of R2I NDP\_2 derived from SEC\_LTF\_CTR2 & *rsta-ltf-key*”.**

**replace “LTF Sequence of I2R NDP\_1 derived from SEC\_LTF\_CTR1 & LTF\_GEN\_SAC1” with “LTF Sequence of I2R NDP\_1 derived from SEC\_LTF\_CTR1 & *ista-ltf-key*”.**

**Figure11-37q:**

**replace “LTF Sequence of R2I NDP\_1 is derived from SEC\_LTF\_CTR1” with “LTF Sequence of R2I NDP\_1 is derived from SEC\_LTF\_CTR1 & *rsta-ltf-key*”.**

**replace “LTF Sequence of R2I NDP\_2 is derived from SEC\_LTF\_CTR2” with “LTF Sequence of R2I NDP\_2 is derived from SEC\_LTF\_CTR2 & *rsta-ltf-key*”.**

**replace “LTF Sequence of I2R NDP\_1 is derived from SEC\_LTF\_CTR1 & LTF\_GEN\_SAC1” with “LTF Sequence of I2R NDP\_1 is derived from SEC\_LTF\_CTR1 & *ista-ltf-key*”.**

**replace “LTF Sequence of I2R NDP\_2 is derived from SEC\_LTF\_CTR2 & LTF\_GEN\_SAC2” with “LTF Sequence of I2R NDP\_2 is derived from SEC\_LTF\_CTR2 & *ista-ltf-key*”.**

**Figure 11-37r**

**replace “LTF Sequence of R2I NDP\_1 is derived from SEC\_LTF\_CTR1” with “LTF Sequence of R2I NDP\_2 is derived from SEC\_LTF\_CTR1 & *rsta-ltf-key*”.**

**replace “LTF Sequence of I2R NDP\_1 is derived from SEC\_LTF\_CTR1 & LTF\_GEN\_SAC1” with “LTF Sequence of I2R NDP\_1 is derived from SEC\_LTF\_CTR1 & *ista-ltf-key*”.**

**TGaz Editor: Change 11.21.6.4.5.2 TB Ranging Measurement Exchange with Secure LTF, p162.20 to p162.36 and p163.20 to p164.3 as follows**

**p162.20**

After transmission of the Ranging Secure Sounding Trigger frame to the ISTA, the RSTA’s MAC sublayer shall issue a PHY-RXLTFSEQUENCE.request primitive with LTFVECTOR parameters ~~LTF\_SEQUENCE~~ LTF\_KEY and LTF\_IV that ~~is~~ are set to *ista-ltf-key* and *ltf-iv* for generating secure HE-LTF~~the Secure-LTF-bits-I2R~~ based on (#**1830**, #**1832, #3124, #3754**) Secure LTF Counter (#**2289**) in the Secure LTF Parameters field in the last transmitted Fine Timing Measurement frame or last transmitted Location Measurement Report frame to the ISTA; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation ~~Information~~).

When the RSTA receives the HE TB Ranging NDP from the ISTA, the RSTA shall:

(a) Send a Ranging NDP Announcement frame.

(b) Send an HE Ranging NDP with the TXVECTOR parameter LTF\_KEY and LTF\_IV ~~LTF\_SEQUENCE~~ set to the *rsta-ltf-key* and *ltf-iv* ~~Secure-LTF-bits-R2I~~ for generating secure HE-LTF based on (#**1830**, #**1832**) Secure LTF Counter (#**2289**) in the Secure LTF Parameters field in the last transmitted Fine Timing Measurement frame or last transmitted Location Measurement Report frame to the ISTA; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation ~~Information~~). (#**3754, #1828, #1831**)

(c) Send a Location Measurement Report frame that includes the Secure LTF Parameters field to the ISTA.

..

**P163.20**

When an ISTA receives a Ranging Secure Sounding Trigger frame from an RSTA in which the value of the SAC subfield in the Trigger Dependent User Info field is equal to the value of the LTF Generation SAC subfield in the Secure LTF Parameters field in the last Fine Timing Measurement frame received or last Location Measurement Report frame received from the RSTA, the ISTA shall:

— Send an HE TB Ranging NDP with the TXVECTOR parameter ~~LTF\_SEQUENCE~~ LTF\_KEY and LTF\_IV that are set to *ista-ltf-key* and *ltf-iv* for generating secure HE-LTF ~~the Secure-LTF-bits-I2R~~ based on (#**1830**, #**1832**) the Secure LTF Counter (#**2289**) and the corresponding SAC (#**3123**) in the Secure LTF Parameters field in the last Fine Timing Measurement frame received, or last Location Measurement Report frame received from the RSTA; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation);

— Issue a PHY-RXLTFSEQUENCE.request primitive with a LTFVECTOR parameter ~~LTF\_SEQUENCE~~ LTF\_KEY and LTF\_IV that ~~is~~ are set to the *rsta-ltf-key* and *ltf-iv* for generating secure HE-LTF ~~Secure-LTF-bits-R2I~~ based on (#**1830**, #**1832)** the Secure LTF Counter (#**2289**) in the Secure LTF Parameters field in the last Fine Timing Measurement frame received, or last Location Measurement Report frame received from the RSTA; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation).

When an ISTA receives a Ranging Secure Sounding Trigger frame from an RSTA in which the value of the SAC subfield in the Trigger Dependent User Info field is not equal to the value of the LTF Generation SAC subfield in the Secure LTF Parameters field in the last Fine Timing Measurement frame received or last Location Measurement Report frame received from the RSTA, the ISTA shall:

1. Send an HE TB Ranging NDP with the TXVECTOR parameter ~~LTF\_SEQUENCE~~ LTF\_KEY and LTF\_IV that are set to (#**2289**) the *ista-ltf-key* and *ltf-iv* ~~Secure-LTF-bits-I2R~~ for generating any secure HE-LTF (#**3124**) l (#**1828**, #**1831**);
2. Issue a PHY-RXLTFSEQUENCE.request primitive with a LTFVECTOR parameter ~~LTF\_SEQUENCE~~ LTF\_KEY and LTF\_IV that ~~is~~ are set to (#**2289**) the *rsta-ltf-key* and *ltf-iv* ~~Secure-LTF-bits-R2I~~ for receiving any secure HE-LTF. (#**3124**#**1828**, #**1831**)

**TGaz Editor: Change 11.21.6.4.5.3 Non-TB Ranging Measurement Exchange with Secure LTF as follows – p168.17 to p168.47 and p169.6 to p169.22**

**P168.17**

An ISTA that sends a Ranging NDP a SIFS after transmission of the Ranging NDP Announcement frame shall set the TXVECTOR parameters LTF\_KEY and LTF\_IV that are set ~~LTF\_SEQUENCE~~ as follows:

* Either (#**3754**) ~~null~~ Null-SAC-HE-LTF (#**1828**, #**1831**) if the SAC subfield in the STA Info field with AID equal to 2043 in the Ranging NDP Announcement frame is set to a value of 0 (#**3124**);

— Or the *ista-ltf-key* and *ltf-iv* for generating secure HE-LTF ~~Secure-LTF-bits-I2R~~ based on (#**1830**, #**1832**) the Secure LTF Counter (#**2289**) and the corresponding SAC in the Secure LTF Parameters field in the last Fine Timing Measurement frame received or last Location Measurement Report frame received from the RSTA; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation). (#**3123**)

After transmission of the Ranging NDP Announcement frame to the RSTA, the ISTA’s MAC sublayer shall issue a PHY-RXLTFSEQUENCE.request primitive with a LTFVECTOR parameters LTF\_KEY and LTF\_IV that are ~~is~~ set (#**2289**) as follows:

— Either (#3754) ~~null~~ Null-SAC-HE-LTF (#**1828**, #**1831**) if the SAC subfield in the STA Info field with AID equal to 2043 in the Ranging NDP Announcement frame is set to 0;

—Or the *rsta-ltf-key* and *ltf-iv* for generating secure HE-LTF ~~Secure-LTF-bits-R2I~~ based on (#**1830**, #**1832**) the Secure LTF counter in the Secure LTF 33 Parameters field in the last Fine Timing Measurement frame received or last Location Measurement 34 Report frame received from the RSTA; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation). (#**3123**)

When an RSTA receives a Ranging NDP Announcement frame from an ISTA frame in which the SAC subfield in the STA Info field with AID equal to 2043 is ~~neither~~ not equal to ~~value 0 nor~~ the value of the LTF Generation SAC subfield in the Secure LTF Parameters field in the last transmitted Fine Timing Measurement frame or last transmitted Location Measurement Report frame to the ISTA, the RSTA shall:

—Issue a PHY-RXLTFSEQUENCE.request primitive with a LTFVECTOR parameter *ista-ltf-key* and the corresponding *ltf-iv* ~~that is set to the Secure-LTF-bits-I2R~~ for receiving any secure HE-LTF; (**#1828, #1831**)

— Send an HE Ranging NDP to ISTA with the TXVECTOR parameter r*sta-ltf-key* and *ltf-iv* for generating secure HE-LTF ~~LTF\_SEQUENCE set to the Secure-LTF-bits-R2I~~ for generating any secure HE-LTF (#**1828**, #**1831**) to the ISTA, if the RSTA receives an HE Ranging NDP from the ISTA a SIFS after the ranging NDP Announcement frame;

**P169.6**

When an RSTA receives a Ranging NDP Announcement frame from an ISTA in which the value of the SAC subfield in the STA Info field with AID equal to 2043 is ~~set~~ equal to the value of the LTF Generation SAC subfield in the Secure LTF Parameters field in the last transmitted Fine Timing Measurement frame or last transmitted Location Measurement Report frame to the ISTA, the RSTA shall:

— Issue a PHY-RXLTFSEQUENCE.request primitive with a LTFVECTOR parameters *ista-ltf-key* and *ltf-iv* for receiving secure HE-LTF ~~that is set to the Secure-LTF-bits-I2R~~ based on (#**1830**, #**1832**) the Secure LTF Counter and corresponding SAC (#**2289**) in the Secure LTF Parameters field in the last transmitted Fine Timing Measurement frame, or last transmitted Location Measurement Report frame to the ISTA; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation);

— Send an HE Ranging NDP with the TXVECTOR parameter *rsta-ltf-key* and *ltf-iv* for generating secure HE-LTF ~~LTF\_SEQUENCE set to the Secure-LTF-bits-R2I~~ based on (#**1830,** #**1832**) the Secure LTF Counter (#**2289**) in the Secure LTF Parameters field in the last transmitted Fine Timing Measurement frame, or last transmitted Location Measurement Report frame to the ISTA, if the RSTA receives an HE Ranging NDP from the ISTA a SIFS after the ranging NDP Announcement frame; see 11.21.6.4.5.4 (Secure LTF Octet Stream Generation);

**TGaz Editor: Modify subclause 12.7.1.3 as follows:**

**12.7 Keys and key distribution**

**12.7.1 Key hierarchy**

* + - 1. **Pairwise key hierarchy**

…

The PTK is partitioned into KCK, KEK, a temporal key, and a KDK ~~if WUR frame protection is negotiated; otherwise the PTK is partitioned into KCK, KEK, and a temporal key~~,. A KDK is derived *if and only if* any of the following are true:

* WUR frame protection is negotiated
* dot11SecureLTFImplemented is true and the peer STA has advertised Secure LTF Support capability in its RSNXE (see 9.4.2.241 (RSN Extension element (RSNXE))

The temporal key~~which~~ is used by the MAC to protect individually addressed communication between the Authenticator's and Supplicant's respective STAs. If WUR frame protection is negotiated, the KDK is used to derive a WTK, which is used by the MAC of the WUR AP to protect and by the MAC of the WUR non-AP STA to validate individually addressed WUR Wake-up frames. PTKs are used between a single Supplicant and a single Authenticator.

If a KDK is derived to support secure LTF, it is recommended that KDK is updated periodically for improved ranging security.

**TGaz Editor: Replace the row for LTF\_SEQUENCE (image of shown below) in Table 27-1—TXVECTOR and RXVECTOR parameters – p212**

**27 High Efficiency (HE) PHY specification**

**27.2 HE PHY service interface**

**27.2.2 TXVECTOR and RXVECTOR parameters**

**Replace**

~~![Table

Description automatically generated]()~~

**With**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LTF\_KEY | FORMAT is either HE\_SU or HE\_TB and RANGING\_FLAG is 1 | Set to the *rsta-ltf-key* (See 11.21.6.4.5.4 (Secure LTF Octet Stream Generation)) when the secure HE-LTFs are used and the UPLINK\_FLAG parameter is set to 0 (see 11.21.6.4.6 (Secure Non-TB and -TB Ranging Measurement Exchange Protocol)).  Set to the *ista-ltf-key* (See 11.21.6.4.5.4 (Secure LTF Octet Stream Generation)) when the secure HE-LTFs are used and the UPLINK\_FLAG parameter is set to 1 (see 11.21.6.4.6 (Secure Non-TB and -TB Ranging Measurement Exchange Protocol)).  Set to null if the insecure HE-LTFs are used. (#2289, #1828, #1831) | O | | N |
| LTF\_IV | FORMAT is either HE\_SU or HE\_TB and RANGING\_FLAG is 1 | Set to *ltf-iv* (See 11.21.6.4.5.4 (Secure LTF Octet Stream Generation)) for secure HE-LTFs or null otherwise. Must be non-null if LTF\_KEY is not null. | | O | N |

**TGaz Editor: Replace the row for LTF\_SEQUENCE (image of shown below) in Table 27-2a—LTFVECTOR parameters – p215**

**Replace**

**![Text

Description automatically generated]()**

**With**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| LTF\_KEY | Set to the *rsta-ltf-key* (See 11.21.6.4.5.4 (Secure LTF Octet Stream Generation)) when receiving the secure HE-LTFs sent by an RSTA; see 11.21.6.4.6 (Secure Non-TB and -TB Ranging Measurement Exchange Protocol).  Set to the *ista-ltf-key* (See 11.21.6.4.5.4 (Secure LTF Octet Stream Generation)) when receiving the secure HE-LTFs sent by an ISTA; see 11.21.6.4.6 (Secure Non-TB and -TB Ranging Measurement Exchange Protocol).  Set to null if receiving the insecure HE-LTFs. (#2289, #1828, #1831). |
| LTF\_IV | Set to *ltf-iv* (See 11.21.6.4.5.4 (Secure LTF Octet Stream Generation)) for secure HE-LTFs or null otherwise. Must be non-null if LTF\_KEY is not null. |

**TGaz Editor: Replace the remaining term ‘LTF\_SEQUENCE’ in the draft with the term ‘LTF\_KEY’ throughout the draft**

**TGaz Editor: Remove the subclause title for § Appendix J.14.1 as the following text includes more than LTF Key Seed information - p257.8**

**TGaz Editor: Revise the text in pp222 ln3 of § 27.3.18a HE Ranging NDP in 11az D2.6 as shown below:**

**27.3.18a HE Ranging NDP**

The HE Ranging NDP has the following properties:

* Uses the HE SU PPDU format but without the Data field.
* No beamforming steering matrix is applied to the waveform, the Beamformed field in HE- SIG-A of a Ranging NDP is always set to 0. For transmission of HE-LTFs, if NSTS = NTx, Q matrix shall be an Identity matrix, and if NSTS < NTx, Q matrix shall be based on antenna selection matrix with no antenna swapping. Q matrix becomes an Identity matrix when all 0 rows are removed.
* Can use insecure HE-LTFs or Secure HE-LTFs.
* Secure HE-LTFs shall use randomized LTF sequence, pseudo random and deterministic per stream phase rotation. Applying a frequency domain flat top window is recommended to enhance the security. See 27.3.18d (Construction of Secure HE-LTF).
* Has a Packet Extension (PE) field that is 4 μs in duration; when using Secure HE-LTFs with randomized LTF sequence, the PE will start with a zero-power GI.
* When the TXVECTOR parameter NUM\_USER is more than 1, the TXVECTOR parameter NUM\_STS[1] is used to encode the NSTS And Mid-amble Periodicity field of the HE-SIG-A1. Otherwise, the TXVECTOR parameter NUM\_STS is used to encode the NSTS And Mid-amble Periodicity field of the HE-SIG-A1.
* The TXVECTOR parameter LTF\_REP that indicates the number of repetitions of the HE- LTF symbols. For decoding the HE-LTF fields, a PHY-RXLTFSEQUENCE.request primitive issued from the MAC provides the LTF\_REP parameter and LTF\_OFFSET parameter, which are not encoded in the HE-SIG-A, but included in the preceding Ranging NDP Announcement frame. The LTF\_OFFSET parameter indicates the number of secure HE-LTF symbols to skip for receiving the corresponding user’s HE-LTF field, e.g., in Figure 27-46d the LTF\_OFFSET for the first and second user would be 0 and 4 respectively.

**TGaz Editor: Revise the text in § 27.3.18b HE TB Ranging NDP as shown below:**

**27.3.18b HE TB Ranging NDP**

The HE TB Ranging NDP has the following properties:

* Uses the HE TB PPDU format without the Data field.
* No beamforming steering matrix is applied to the waveform.
* Can use insecure HE-LTFs or Secure HE-LTFs.
* Secure HE-LTFs shall use randomized LTF sequence, pseudo random and deterministic per stream phase rotation. Applying a frequency domain flat top window is recommended to enhance the security. See 27.3.18d (Construction of Secure HE-LTF).
* Has a Packet Extension (PE) field that is 4 μs in duration; when using Secure HE-LTFs with randomized LTF sequence, the PE will start with a zero-power GI.
* For transmission of HE-LTFs, if NSTS = NTx, Q matrix shall be an Identity matrix, and if NSTS < NTx, Q matrix shall be antenna selection matrix with no antenna swapping. Q matrix becomes an Identity matrix when all 0 rows are removed.

**TGaz Editor: Please replace subclause 27.3.18c in 11az\_D2.6 with the following new text:**

**27.3.18c Generation of Randomized LTF Sequence**

Each element of the secure HE-LTF sequence is a pseudo random 64-QAM value. Pseudo random octets 11.21.6.4.5.4 (Secure LTF Octet Stream Generation) are used in the construction of the pseudo random 64-QAM values.

The first seven pseudo random octets (-) in the secure NDP are used for per-stream phase rotations, TBD (Pseudo Random and Deterministic Per-Spatial Stream Phase Rotations). Starting with these pseudo random octets are used for construction of pseudo random 64-QAM values in the secure LTF sequences.

**27.3.18c.1 Randomized LTF Sequence for 20-MHz secure NDP**

There are 122 non-zero entries in the 20-MHz secure 2x LTF sequence. This subclause describes the mapping of pseudo random octets to the non-zero entries of the 20-MHz secure 2x LTF sequence, and then the construction of the 64-QAM values for each non-zero entry of the secure LTF sequence.

The indices of the non-zero entries of the 20-MHz secure 2x LTF sequence are those of the nonzero entries in Equation (27-42). The indices for the nonzero entries are provided in Equation (27-E1)

|  |  |
| --- | --- |
| = {-122, -120, -118, -116, -114, -112, -110, -108, -106, -104, -102, -100, -98, -96, -94, -92, -90, -88, -86, -84, -82, -80, -78, -76, -74, -72, -70, -68, -66, -64, -62, -60, -58, -56, -54, -52, -50, -48, -46, -44, -42, -40, -38, -36, -34, -32, -30, -28, -26, -24, -22, -20, -18, -16, -14, -12, -10, -8, -6, -4, -2, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122} | (27-E1) |

There are up to sixty four secure LTF sequences in an NDP, since there are up to eight repetitions and up to eight secure LTF sequences within a repetition. For notational convenience we indicate the LTF sequence number with the integer , which is an integer between one and sixty four. Table 27-T1 provides the pseudo random octet index for each nonzero subcarrier index in the k-th 20-MHz secure LTF sequence.

Table 27-T1: Pseudo Random Octet Index for each Non-Zero Subcarrier Index in the k-th 20-MHz Secure LTF Sequence

|  |  |
| --- | --- |
| **Secure LTF Index** | **Pseudo Random Octet Index** |
| -122 |  |
| -120 |  |
| -118 |  |
|  |  |
| -4 |  |
| -2 |  |
| 2 |  |
| 4 |  |
|  |  |
| 118 |  |
| 120 |  |
| 122 |  |

All entries in the 20-MHz secure LTF sequence other than the non-zero entries shall be set to zero.

The six least significant bits () of an octet are used in the construction of the 64-QAM value, as specified in Table 17-15 (64-QAM Encoding Table).

**27.3.18c.2 Randomized LTF Sequence for 40-MHz secure NDP**

There are 242 non-zero entries in the 40-MHz secure 2x LTF sequence. This subclause describes the mapping of pseudo random octets to the non-zero entries of the 40-MHz secure 2x LTF sequence, and then the construction of the 64-QAM values for each non-zero entry of the secure LTF sequence.

The indices of the non-zero entries of the 40-MHz secure 2x LTF sequence are those of the nonzero entries in Equation (27-45). The indices for the nonzero entries are provided in Equation (27-E2)

|  |  |
| --- | --- |
| = { -244, -242, -240, -238, -236, -234, -232, -230, -228, -226, -224, -222,  -220, -218, -216, -214, -212, -210, -208, -206, -204, -202, -200, -198, -196, -194,  -192, -190, -188, -186, -184, -182, -180, -178, -176, -174, -172, -170, -168, -166,  -164, -162, -160, -158, -156, -154, -152, -150, -148, -146, -144, -142, -140, -138,  -136, -134, -132, -130, -128, -126, -124, -122, -120, -118, -116, -114, -112, -110,  -108, -106, -104, -102, -100, -98, -96, -94, -92, -90, -88, -86, -84, -82, -80, -78, -76,  -74, -72, -70, -68, -66, -64, -62, -60, -58, -56, -54, -52, -50, -48, -46, -44, -42, -40, -38, -36, -34, -32, -30, -28, -26, -24, -22, -20, -18, -16, -14, -12, -10, -8, -6, -4, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244} | (27-E2) |

There are up to sixty four secure LTF sequences in an NDP, since there are up to eight repetitions and up to eight secure LTF sequences within a repetition. For notational convenience we indicate the LTF sequence number with the integer , which is an integer between one and sixty four. Table 27-T2 provides the pseudo random octet index for each nonzero subcarrier index in the k-th 40-MHz secure LTF sequence.

Table 27-T2: Pseudo Random Octet Index for each Non-Zero Subcarrier Index in the k-th 40-MHz Secure LTF Sequence

|  |  |
| --- | --- |
| **Secure LTF Index** | **Pseudo Random Octet Index** |
| -244 |  |
| -242 |  |
| -240 |  |
|  |  |
| -6 |  |
| -4 |  |
| 4 |  |
| 6 |  |
|  |  |
| 240 |  |
| 242 |  |
| 244 |  |

All entries in the 40-MHz secure LTF sequence other than the non-zero entries shall be set to zero.

The six least significant bits () of an octet are used in the construction of the 64-QAM value, as specified in Table 17-15 (64-QAM Encoding Table).

**27.3.18c.3 Randomized LTF Sequence for 80-MHz secure NDP**

There are 498 non-zero entries in the 80-MHz secure 2x LTF sequence. This subclause describes the mapping of pseudo random octets to the non-zero entries of the 80-MHz secure 2x LTF sequence, and then the construction of the 64-QAM values for each non-zero entry of the secure LTF sequence.

The indices of the non-zero entries of the 80-MHz secure 2x LTF sequence are those of the nonzero entries in Equation (27-48). The indices for the nonzero entries are provided in Equation (27-E3)

|  |  |
| --- | --- |
| = {-500, -498, -496, -494, -492, -490, -488, -486, -484, -482, -480, -478,  -476, -474, -472, -470, -468, -466, -464, -462, -460, -458, -456, -454, -452, -450,  -448, -446, -444, -442, -440, -438, -436, -434, -432, -430, -428, -426, -424, -422,  -420, -418, -416, -414, -412, -410, -408, -406, -404, -402, -400, -398, -396, -394,  -392, -390, -388, -386, -384, -382, -380, -378, -376, -374, -372, -370, -368, -366,  -364, -362, -360, -358, -356, -354, -352, -350, -348, -346, -344, -342, -340, -338,  -336, -334, -332, -330, -328, -326, -324, -322, -320, -318, -316, -314, -312, -310,  -308, -306, -304, -302, -300, -298, -296, -294, -292, -290, -288, -286, -284, -282,  -280, -278, -276, -274, -272, -270, -268, -266, -264, -262, -260, -258, -256, -254,  -252, -250, -248, -246, -244, -242, -240, -238, -236, -234, -232, -230, -228, -226,  -224, -222, -220, -218, -216, -214, -212, -210, -208, -206, -204, -202, -200, -198,  -196, -194, -192, -190, -188, -186, -184, -182, -180, -178, -176, -174, -172, -170,  -168, -166, -164, -162, -160, -158, -156, -154, -152, -150, -148, -146, -144, -142,  -140, -138, -136, -134, -132, -130, -128, -126, -124, -122, -120, -118, -116, -114,  -112, -110, -108, -106, -104, -102, -100, -98, -96, -94, -92, -90, -88, -86, -84, -82,  -80, -78, -76, -74, -72, -70, -68, -66, -64, -62, -60, -58, -56, -54, -52, -50, -48, -46, -44, -42, -40, -38, -36, -34, -32, -30, -28, -26, -24, -22, -20, -18, -16, -14, -12, -10, -8, -6,  -4, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314, 316, 318, 320, 322, 324, 326, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 362, 364, 366, 368, 370, 372, 374, 376, 378, 380, 382, 384, 386, 388, 390, 392, 394, 396, 398, 400, 402, 404, 406, 408, 410, 412, 414, 416, 418, 420, 422, 424, 426, 428, 430, 432, 434, 436, 438, 440, 442, 444, 446, 448, 450, 452, 454, 456, 458, 460, 462, 464, 466, 468, 470, 472, 474, 476, 478, 480, 482, 484, 486, 488, 490, 492, 494, 496, 498, 500} | (27-E3) |

There are up to sixty four secure LTF sequences in an NDP, since there are up to eight repetitions and up to eight secure LTF sequences within a repetition. For notational convenience we indicate the LTF sequence number with the integer , which is an integer between one and sixty four. Table 27-T3 provides the pseudo random octet index for each nonzero subcarrier index in the k-th 80-MHz secure LTF sequence.

Table 27-T3: Pseudo Random Octet Index for each Non-Zero Subcarrier Index in the k-th 80-MHz Secure LTF Sequence

|  |  |
| --- | --- |
| **Secure LTF Index** | **Pseudo Random Octet Index** |
| -500 |  |
| -498 |  |
| -496 |  |
|  |  |
| -6 |  |
| -4 |  |
| 4 |  |
| 6 |  |
|  |  |
| 496 |  |
| 498 |  |
| 500 |  |

All entries in the 80-MHz secure LTF sequence other than the non-zero entries shall be set to zero.

The six least significant bits () of an octet are used in the construction of the 64-QAM value, as specified in Table 17-15 (64-QAM Encoding Table).

**27.3.18c.4 Randomized LTF Sequence for 160-MHz secure NDP**

This subclause describes the mapping of pseudo random octets to the non-zero entries of the 160-MHz secure 2x LTF sequence, and then the construction of the 64-QAM values for each non-zero entry of the secure LTF sequence.

The construction of the 160-MHz secure LTF sequence uses a segment parser to divide the pseudo random octets between the sequence for the lower 80 MHz segment and the sequence for the upper 80 MHz segment. Figure 27-F1 illustrates the segment parser distribution of pseudo random octets between the sequence for the lower 80 MHz segment and the sequence for the upper 80 MHz segment.

Figure 27-F1: Segment Parser distributing Pseudo Random Octets to the sequences for the Lower and Upper 80 MHz Segments in the 160-MHz Secure LTF

* 

There are up to sixty four secure LTF sequences in an NDP, since there are up to eight repetitions and up to eight secure LTF sequences within a repetition. For notational convenience we indicate the LTF sequence number with the integer , which is an integer between one and sixty four. Table 27-T4 provides the pseudo random octet index for each nonzero subcarrier index for the k-th pair of lower and upper 80-MHz segments.

Table 27-T4: Pseudo Random Octet Index for each Non-Zero Subcarrier Index in the k-th Pair of Lower and Upper 80-MHz Segments

|  |  |  |
| --- | --- | --- |
| **80-MHz Segment** | **Secure LTF Index** | **Pseudo Random Octet Index** |
| Lower | -500 |  |
| Upper | -500 |  |
| Lower | -498 |  |
| Upper | -498 |  |
| Lower | -496 |  |
| Upper | -496 |  |
|  |  |  |
| Lower | -6 |  |
| Upper | -6 |  |
| Lower | -4 |  |
| Upper | -4 |  |
| Lower | 4 |  |
| Upper | 4 |  |
| Lower | 6 |  |
| Upper | 6 |  |
|  |  |  |
| Lower | 496 |  |
| Upper | 496 |  |
| Lower | 498 |  |
| Upper | 498 |  |
| Lower | 500 |  |
| Upper | 500 |  |

All entries in the 160-MHz secure LTF sequence other than the non-zero entries shall be set to zero.

The six least significant bits () of an octet are used in the construction of the 64-QAM value, as specified in Table 17-15 (64-QAM Encoding Table).

**TGaz Editor: Revise the text in § 27.3.18d Construction of Secure HE-LTF as shown below:**

**27.3.18d Construction of Secure HE-LTF**

The Secure HE-LTF field is largely like the insecure HE-LTF field defined in 27.3.11.10 (HE- LTF), the main differences are as follows:

* The HE-LTF sequence is replaced by the randomized LTF sequence described in 27.3.18c (Generation of Randomized LTF Sequence).
* The conventional GI is replaced by a zero-power GI.
* There are no single stream pilot subcarriers in the secure HE-LTFs, all subcarriers are mapped using the 𝑃 matrix
* No CSD is applied to the space-time streams.
* Each spatial stream has a per stream pseudo random and deterministic phase rotation applied to all the subcarriers.
* A frequency domain flat top window is recommended to be applied to the secure HE-LTF to enhance the security.

The construction of the Secure HE-LTF field is as follows:

1. Sequence generation: Generate the randomized LTF sequence in frequency domain over the bandwidth indicated by CH\_BANDWIDTH as described in Subclause 27.3.18c (Generation of Randomized LTF Sequence).
2. Apply per spatial stream phase rotation: Generate the pseudo random phase rotation for each spatial stream. Apply the pseudo random phase rotation along with the deterministic phase rotation to the spatial streams as described in Subclause TBD (Pseudo Random and Deterministic Per Spatial Stream Phase Rotations).
3. 𝐴HE-LTF matrix mapping: Apply the PHE-LTF matrix to all tones of the secure HE-LTF sequence.
4. Frequency domain windowing: A frequency domain window function is applied to all the tones of the secure HE-LTF sequence. The default is the Rectangular window where for all the tones in all channel bandwidths.   
   The optional flat top window is defined as:   
   where and the impulse response p(n) is given by:  
   where  
   a0 = 0.21557895,   
   a1 = -0.41663158,   
   a2 = 0.277263158,  
   a3 = -0.083578947,  
   a4 = 0.006947368 and  
   NWinFT = 20.  
     
   Note that the shall be normalized to have unit RMS power.  
   In Equations (27-3) and (27-4), the data tones , where is 11az secure LTF sequence constructed after step c).
5. There is no CSD per space-time stream.
6. There is no spatial mapping, the Q matrix is a block identity matrix.
7. IDFT: Compute the inverse discrete Fourier transform.
8. Insert zero-power GI and apply windowing: Prepend values of zero of length indicated by the TXVECTOR parameter GI\_TYPE and apply windowing as described in 27.3.10 (Mathematical description of signals).
9. Analog and RF: Upconvert the resulting complex baseband waveform associated with each transmit chain to an RF signal according to the center frequency of the desired channel and transmit. Refer to 27.3.9 (Mathematical description of signals) and 27.3.11 (HE preamble) for details.

**TGaz Editor: Create an new subclause after the exiting 27.3.18e in 11az\_D2.6 as follows. Renumber the existing 27.3.18e as 27.3.18f.**

**27.3.18e Pseudo Random and Deterministic Per-Spatial Stream Phase Rotations**

In each repetition of the secure HE-LTF symbols, each spatial stream has a phase rotation applied to all the subcarriers. The same phase rotation that is applied to all of the subcarriers. The phase rotation can vary in different repetitions of the secure HE-LTF symbols.

The phase rotation for a given spatial stream consists of two components: a pseudo random phase rotation and a deterministic phase rotation. For each spatial stream, the pseudo random phase rotation is the same for all the repetitions of HE-LTF symbols within the secure NDP. As specified in 11.21.6.4.5.4 (Secure LTF Octet Stream Generation) a stream of pseudo random octets are provided by the AES-128 CTR Mode. The first six pseudo random octets are used in the construction of the pseudo random phase rotations.

The deterministic phase rotation is a function of the repetition number within the secure NDP and the spatial stream number.

The total phase rotation applied to spatial stream within repetition is given by,

(27-E1)

In (27-E1) the term is the pseudo random phase rotation that is the same for the entire secure NDP and the term is the deterministic phase rotation that depend on the rotation value .

The pseudo random phase is constructed using one of the pseudo random octets 11.21.6.4.5.4 (Secure LTF Octet Stream Generation). The index of spatial streams, , is a number between one and eight. For the phase rotation is zero: . The first octet () is used to construct , the second octet () is used to construct Table 27-T1 provides a summary of pseudo random octets used in construction of each of the pseudo random phase rotations.

Table 27-T1: Pseudo Random Octets for Generating Per-Stream Pseudo Random Phase Rotations

|  |  |
| --- | --- |
| **Pseudo Random Octet** | **Spatial Stream Pseudo Random Phase Rotation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

For STAs with fewer than eight spatial streams not all of the pseudo random octets up to are used, and in that case the unused Octets may be discarded.

The three most significant bits of the pseudo random octet are used to construct the pseudo random phase rotation. The integer in Equation (27-E2) is derived from the three MSBs of the pseudo random octet.

(27-E2)

Equation (27-E3) specifies the pseudo random phase rotation as a function of the integer ,

(27-E3)

The deterministic phase rotation , which depends on both the spatial stream number and the repetition number , is provided in Table 27-T2.

Table 27-T2: Deterministic Phase Rotation (radians)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |

Clause 27.3.17d (Construction of Secure HE-LTF) describes the steps in the process of the construction of the secure HE-LTF, including the application of the spatial stream phase rotation.

**TGaz Editor: Remove the lines from p257.22 to end of the Appendix J.14 as the secure bit generation has changed.**

**J.14 LTF Sequence Generation Test Vectors**

**~~J.14.1 Secure-LTF-Key-Seed~~**

~~Downlink Secure LTF bits are derived as follows, 176 bits that comprise of 156 bits for 80 MHz 22 Bandwidth, two symbols for repetition and two repetitions, plus 16 bits for SAC rounded to nearest 23 multiple of 8 bits.~~

~~…~~

P258.1

~~Secure-LTF-Counter: 0x000000000100 22~~

~~SAC: e5 b9 23~~

~~Secure-LTF-UL-Bits: 24~~

~~da ac 92 58 02 f1 d7 d4 1e 62 3f 5d a0 a8 3e 7a 25~~

~~10 5a d4 71 (#~~**~~2148~~**~~, #~~**~~1090~~**~~)~~

**TGaz Editor: Add the following text after p257.21**

SAC || *ista-ltf-key* || *rsta-ltf-key* = KDF-Hash-Length(Secure-LTF-Key-Seed,

“Secure LTF Expansion”, Secure-LTF-Counter)

Hash: SHA-256

Length: 272 (bits)

Secure-LTF-Key-Seed:

07 60 6f 7b 0d 98 ca 03 ec 2d 61 e1 7c 6b df d3

0e 2f 20 30 e3 47 02 22 55 1a 05 ec 55 d1 35 b9

Secure-LTF-Counter: 0x000000000100

SAC: 23 cf

ista-ltf-key:

d2 a8 a2 b7 6c 3c 29 2d 81 e1 82 a4 69 fd e8 3c

rsta-ltf-key:

65 02 7a 83 8d 58 59 3c 57 b9 41 6f 17 24 e6 c4

Transmitter MAC address:

00 10 18 32 76 54

LTF Octet Stream Generator Inputs and Outputs:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Octet** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| **LTF\_KEY** | d2 | a8 | a2 | b7 | 6c | 3c | 29 | 2d | 81 | e1 | 82 | a4 | 69 | fd | e8 | 3c |
| **LTF\_IV** | 00 | 10 | 18 | 32 | 76 | 54 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 |
| **AES Counter [0]** | 00 | 10 | 18 | 32 | 76 | 54 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 |
| **Output Block [0]** | aa | f6 | 2c | 30 | 6b | cd | 8a | 5d | 89 | 80 | 8b | 03 | 8e | da | 43 | f1 |
| **AES Counter [1]** | 00 | 10 | 18 | 32 | 76 | 54 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 01 |
| **Output Block [1]** | 54 | 15 | f0 | 5c | 7f | c7 | ee | f5 | 9b | c4 | 58 | d2 | f4 | 6b | 5b | 5a |
| **...** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**References**

[1] IEEE P802.11az™/D2.6 November 2020

[2] IEEE P802.11ax™/D8.0 November 2020

[3] IEEE 802.11 document 11-20-0836r0 – 11az Secure LTF design, Bin Tian et al.

[4] IEEE 802.11 document 11-20-1855r0 – Further updates on Secure LTF Design, Anuj Batra et al.

[5] IEEE 802.11 document 11-20-1959r2 – Tx FD Window Design for Secure LTF, Anuj Batra et al.

[6] IEEE 802.11 document 11-20-1863r0 – Secure LTFs – Additional Design Details, Steve Shellhammer et al.

[7] NIST SP 800 90Ar1 - Recommendations for Random Number Generation - <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf>

[8] IEEE 802.15.4z-2020 – Enhanced UWB PHY and Associated Ranging Techniques