CHANGE REQUEST

LoRaWAN	Version	1.1	CR		rev	

CR Title:	FOpts encryption, usage of FCntDwn Compliance to IEEE 802.15.4		
Submitter:	Name: Joris Delclef Contact Email: joris.delclef@st.com Contact Other: Affiliation: STMicroelectronics		
Work Item Ref:	Technical workgroup/main	***Date:***	~~10~~26 January 2018
Category:	**F (correction)**		

Reason for change #1:

The text states in section 4.3.1.6 figure 15 that **N**FCntDwn must be used in the FOpts encryption process of downlinks.

I think that this is not correct.

I consider 2 cases; FPort is present and FPort is absent

When FPort is present in a downlink frame that carries FOpts it means that FPort is different from 0 and that FRMPayload (containing applicative data) is present (it is forbidden to have MAC commands both in FOpts and in FRMPayload (see section 4.3.1.6 line 695))

- When FPort is different from 0, **A**FCntDwn must be used (see section 4.3.1.5 line 646). This is in contradiction with 4.3.1.6 figure 15 that states that **N**FCntDwn must be used.
- FHDR can only contain one FCnt, which is the **A**FCntDwn in this case
- As a consequence, FOpts must be encrypted with **Nwk**SEncKey using **A**FCntDwn in the formatting of A (see figure 15).
- Remember that FRMPayload will be encrypted with **App**SEncKey and with **A**FCntDwn in the formatting of Ai (see figure 17)
- **Nwk**SEncKey is now linked to 2 different counters; AFCntDwn and NFCntDwn so there is a risk that A-frames collide when counters have the same value. To mitigate the risk we introduce a constant in the A-frame that is different for both counters
- The constants are chosen to be coded in the 4-bytes field as 0x00000001 and 0x00000002. We deliberately didn't choose for 0x00000000 because this one is used in the A-frames of FRMPayload encryption so that both domains (FOpts encryption and FRMPayload encryption) are isolated.
- ~~. One could think that this could induce security risks because both counters could have the same value and thus reveal the same keystream but this is not~~

~~the case because the rightmost byte in A is always 0x00 (see figure 15) while in Ai the rightmost byte is never 0x00 (see figure 17); so the key streams of FOpts encryption and FRMPayload encryption are isolated.~~

When FPort is absent in a downlink frame that carries FOpts; **N**FCntDwn must be used (see section 4.3.1.5 line 645).

Reason for change #2:

FOpts encryption uses the IEEE 802.15.4 Annexe B scheme; it defines the encryption transformation in B.4.1.3

- A_i = Flags (1-byte) || Nonce *N* (15-bytes) || Counter i, for i = 0,1,2,… (2-bytes)
- C_i = E(Key, A_i) xor M_i for i = 1,2,…,t > please note that i starts at 1 (not 0); so A_1 must be used for FOpts encryption (max. 15 bytes, so only 1 block)
- As a result, the last byte of the A-field must be 0x01 (not 0x00)

On notation; we should use "A_1" (instead of "A")

Please note that A_0 (i = 0) is used in IEEE in the encryption of the authentication tag *U*. But LoRaWAN doesn't use U, it uses CMAC instead. This is also the reason why the last byte in the A_i fields of the FRMPayload encryption starts with 1.

Summary of change:

Introduce 2 constants to assure that A-frames are always distinct (even when NFCntDwn and AFCntDwn have the same value)

Introduce AFCntDwn as the counter to use when FPort > 0

Set the last byte of the A-frame to 0x01 (instead of 0x00) to comply to IEEE802.15.4

Use notation "A_1"rather than "A"

Clauses affected:

4.3.1.6

Other deliverables affected:

Other comment:

Proposed changes (normative/informative)

********** *Start of proposed change 1* **********

4.3.1.6 Frame options (FOptsLen in FCtrl, FOpts)

The frame-options length field (**FOptsLen**) in **FCtrl** byte denotes the actual length of the frame options field (**FOpts**) included in the frame.

FOpts transport MAC commands of a maximum length of 15 octets that are piggybacked onto data frames; see Chapter 5 for a list of valid MAC commands.

If **FOptsLen** is 0, the **FOpts** field is absent. If **FOptsLen** is different from 0, i.e. if MAC commands are present in the **FOpts** field, the port 0 cannot be used (**FPort** must be either not present or different from 0).

MAC commands cannot be simultaneously present in the payload field and the frame options field. Should this occur, the device SHALL ignore the frame.

If a frame header carries FOpts, FOpts MUST be encrypted before the message integrity code (MIC) is calculated.

The encryption scheme used is based on the generic algorithm described in IEEE 802.15.4/2006 Annex B [IEEE802154] using AES with a key length of 128 bits.

The key K used is the NwkSEncKey for FOpts field in both the uplink and downlink direction.

The fields encrypted are: pld = FOpts

For each message, the algorithm defines a single Block A_1~~A~~:

Size (bytes)	1	4	1	4	4	1	1
A_1~~A~~	0x01	0x00000001 when FCntUp or NFCntDwn is used 0x00000002 when AFCntDwn is used ~~4 x 0x00~~	Dir	DevAddr	FCntUp or NFCntDwn when FPort is absent or AFcntDwn when FPort > 0	0x00	0x01 ~~0x00~~

Figure 1 : Encryption block format

The direction field (**Dir**) is 0 for uplink frames and 1 for downlink frames.

The block A_1~~A~~ is encrypted to get a block *S*:

S = aes128_encrypt(K, A_1~~A~~)

Encryption and decryption of the **FOpts** is done by truncating (pld | pad_{16}) xor S to the first len(pld) octets.

********** *End of proposed change 1* **********