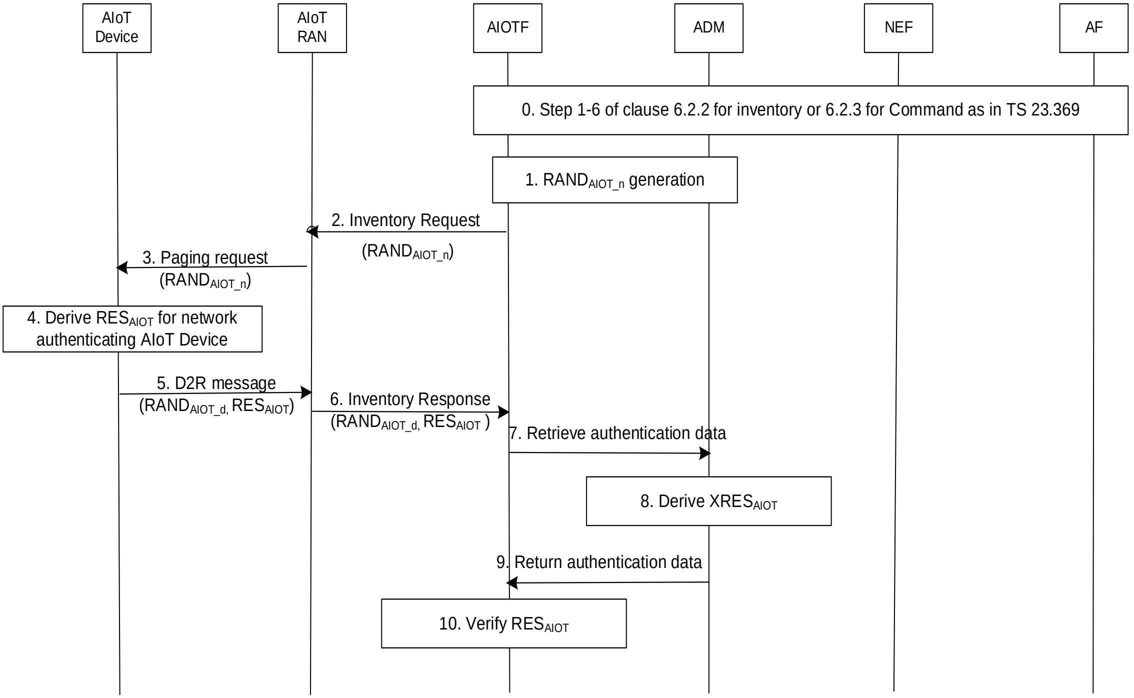
Authentication procedure for AIOT system

oppo

Resolving Inventory authentication ENs


AIoT Device
AIoT RAN
AIOTF
ADM
NEF
AF
0. Step 1-6 of clause 6.2.2 for inventory or 6.2.3 for Command as in TS 23.369
1. $RAND_{AIOT\_n}$ generation
2. Inventory Request ($RAND_{AIOT\_n}$)
3. Paging request ($RAND_{AIOT\_n}$)
4. Derive RES_{AIOT} for network authenticating AIoT Device
5. D2R message ($RAND_{AIOT\_d}$, RES_{AIOT})
6. Inventory Response ($RAND_{AIOT\_d}$, RES_{AIOT})
7. Retrieve authentication data
8. Derive $XRES_{AIOT}$
9. Return authentication data
10. Verify RES_{AIOT}

- ~~Editor's Note: Whether ADM or AIOTF generates $RAND_{AIOT\_n}$ is FFS.~~ （ADM, to align with 5G AKA, if for implementation, KAIOT is in ADM, it would be more convenient if both are in the ADM）
- ~~Editor's Note: The inclusion of $RAND_{AIOT\_n}$ in Paging Request and the size of $RAND_{AIOT\_n}$ needs RAN confirmation.~~ （128bit）
- ~~Editor's Note: Whether replay attack is possible is FFS.~~ (align with 33501, the first RAND sent from network to device is in plain text)
- ~~Editor's Note: How RES_{AIOT} is derived and whether it is derived from K_{AIoT} or intermediate key is FFS.~~ RES = fx (K_{AIoT} ,$RAND_{AIOT\_n}$)； same as in 33501
- No current RAN2/SA2 procedure could support key bootstrapping
- ~~Editor's Note: Where the authentication credentials are processed in AIOT device is FFS.~~ （delete according to SA plenary LS）
- ~~Editor's Note: The security requirements of generating $RAND_{AIOT\_d}$ are FFS.~~
- ~~Editor's Note: Whether $RAND_{AIOT\_d}$ is required for inventory procedure is FFS.~~ （No $RAND_{AIOT\_d}$ is needed for inventory only case）For inventory only case, since there is no following command message and no command protection, derive intermediate key would be redundant. Considering the limited power and energy for Device 1, the RES should be generated based on the long-term key KAIoT.
- RANDAIOT_d is not needed for RES derivation. Furthermore, using RANDAIOT_n for RES derivation is align with 33501.
- ~~Editor's note: The impact of interaction between AIOTF and ADM is FFS. If the authentication is expected to be run more often than normal UE, (e.g., during each inventory procedure), the analysis of load of ADM is FFS.~~ Only after inventory response is received, ADM needs to derive authentication vectors, hence ADM load is lower than derive authentication vectors before sending inventory request.
- No current RAN2/SA2 procedure could support key bootstrapping. No guarantee the key can be stored in device.
- According to SA plenary LS, R19 AIOT service is in private network, so ADM load is not like in public network
- ~~Editor's Note: Where the authentication credential is processed in AIOT device is FFS.~~
- Note: Where the authentication credential is processed in AIOT device is out of scope for the present document.
- ~~Editor's note: How and where to derive keys is FFS.~~ （No intermediate key is needed for inventory only case）

Command authentication using $\text{RAND}_{\text{AIOT\_n2}}$

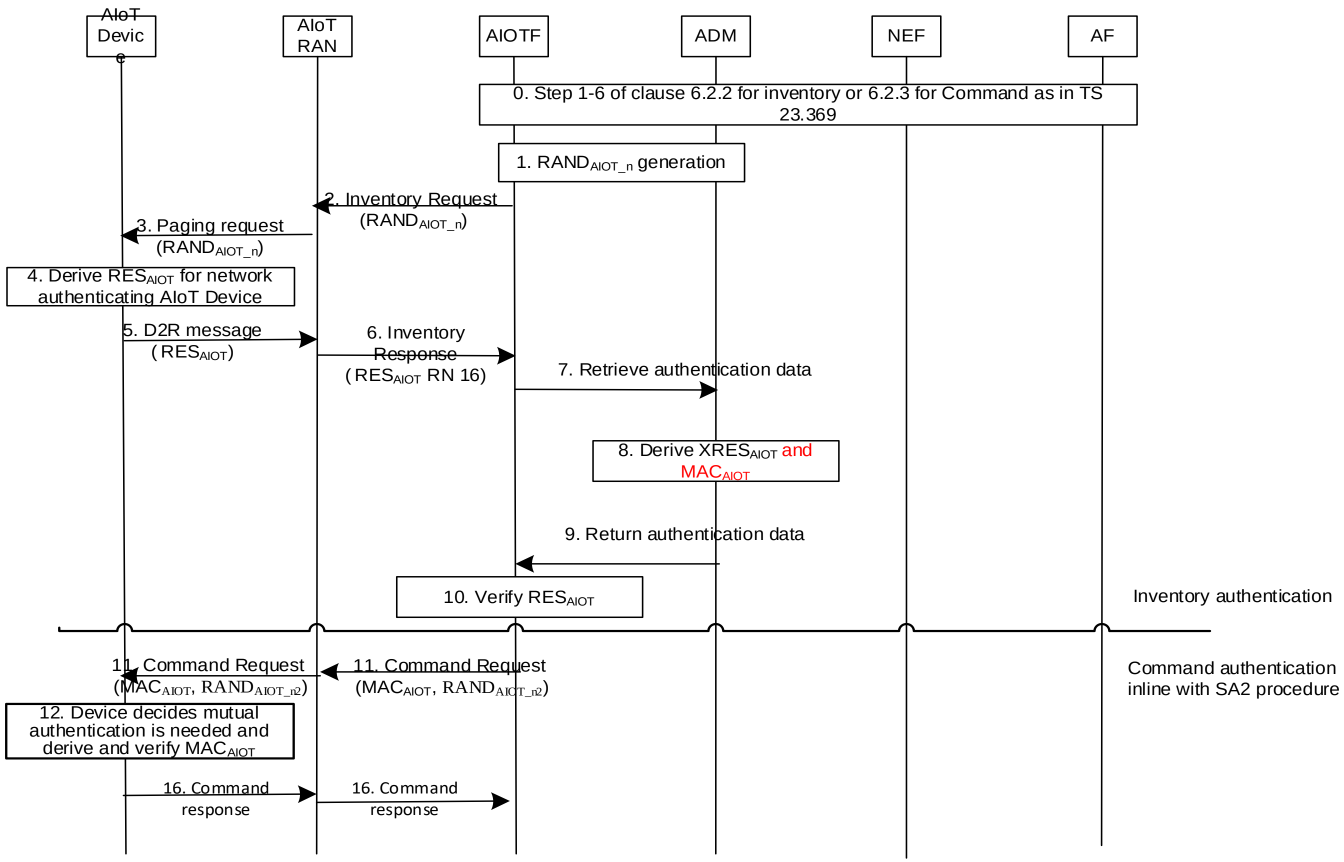
- According to RAN and SA2 procedure, AIOT device has no way to know if it's inventory only or inventory and command
- Prefer to send a second network random number $\text{RAND}_{\text{AIOT\_n2}}$ for 3 reasons
  1. Not sure if device could store the 1st network random number $\text{RAND}_{\text{AIOT_}}$n1 until command procedure is complete.
  2. 2nd random number $\text{RAND}_{\text{AIOT\_n2}}$ is fresh, provide freshness.
  3. 5G AKA use only network side RAND for authentication

- For command authentication, use $\text{RAND}_{\text{AIOT\_n2}}$ to provide freshness, better than repeating RAND $_{\text{AIOT\_n1}}$

Plain text $\text{RAND}_{\text{AIOT\_d}}$ maybe redundant in inventory and may not be secure

- In 5G-AKA, no UE side RAND is used. Not sure if it’s secure enough to send $\text{RAND}_{\text{AIOT\_d}}$ in inventory response and D2R, since inventory response is not protected and tamper attack is possible.
- Maybe Redundant to send $\text{RAND}_{\text{AIOT\_d}}$

Inventory and command authentication- use $\mathrm{RAND}_{\mathrm{AIOT\_n2}}$ directly



AIoT Device

AIoT RAN

AIOTF

ADM

NEF

AF

0. Step 1-6 of clause 6.2.2 for inventory or 6.2.3 for Command as in TS 23.369

1. $\mathrm{RAND}_{\mathrm{AIOT\_n}}$ generation

2. Inventory Request ($\mathrm{RAND}_{\mathrm{AIOT\_n}}$)

3. Paging request ($\mathrm{RAND}_{\mathrm{AIOT\_n}}$)

4. Derive $\mathrm{RES}_{\mathrm{AIOT}}$ for network authenticating AIoT Device

5. D2R message ($\mathrm{RES}_{\mathrm{AIOT}}$)

6. Inventory Response ($\mathrm{RES}_{\mathrm{AIOT}}$ RN 16)

7. Retrieve authentication data

8. Derive $\mathrm{XRES}_{\mathrm{AIOT}}$ and $\mathrm{MAC}_{\mathrm{AIOT}}$

9. Return authentication data

10. Verify $\mathrm{RES}_{\mathrm{AIOT}}$

Inventory authentication

11. Command Request ($\mathrm{MAC}_{\mathrm{AIOT}}$, $\mathrm{RAND}_{\mathrm{AIOT\_n2}}$)

11. Command Request ($\mathrm{MAC}_{\mathrm{AIOT}}$, $\mathrm{RAND}_{\mathrm{AIOT\_n2}}$)

12. Device decides mutual authentication is needed and derive and verify $\mathrm{MAC}_{\mathrm{AIOT}}$

16. Command response

16. Command response

Command authentication inline with SA2 procedure

8. knowing it's command procedure and mutual authentication procedure is needed ADM shall derive $\mathrm{MAC}_{\mathrm{AIOT}}$ using $\mathrm{K}_{\mathrm{AIoT}}$ and $\mathrm{RAND}_{\mathrm{AIOT\_n2}}$ for device authenticating network. This step could happen after step 7 and before step 9.

MAC = fx (KAIoT, $\mathrm{RAND}_{\mathrm{AIOT\_n2}}$);

Communication protection key $\mathrm{K}_{\mathrm{AIoTF}}$ derivation is performed in this step. The method of $\mathrm{K}_{\mathrm{AIoTF}}$ derivation is descried in 5.3.X

Editor's Note: Whether F1 is as same as RES derivation function is FFS.

11, AIOTF shall construct a NAS Command Request including $\mathrm{MAC}_{\mathrm{AIOT}}$ and $\mathrm{RAND}_{\mathrm{AIOT\_n2}}$. AIOTF sends NAS Command Request to AIoT RAN.

12. AIoT RAN shall send NAS Command Request includes $\mathrm{MAC}_{\mathrm{AIOT}}$ and $\mathrm{RAND}_{\mathrm{AIOT\_n2}}$ to AIoT Device as specified in as specified in TS 38.300 [x3] and TS 38.391 [x4] .

13. AIoT Device receives NAS Command Request and shall decides mutual authentication is needed, and derive $\mathrm{MAC}_{\mathrm{AIOT}}$ as the same way as ADM. AIoT Device shall verify the $\mathrm{MAC}_{\mathrm{AIOT}}$ to authenticate the network.

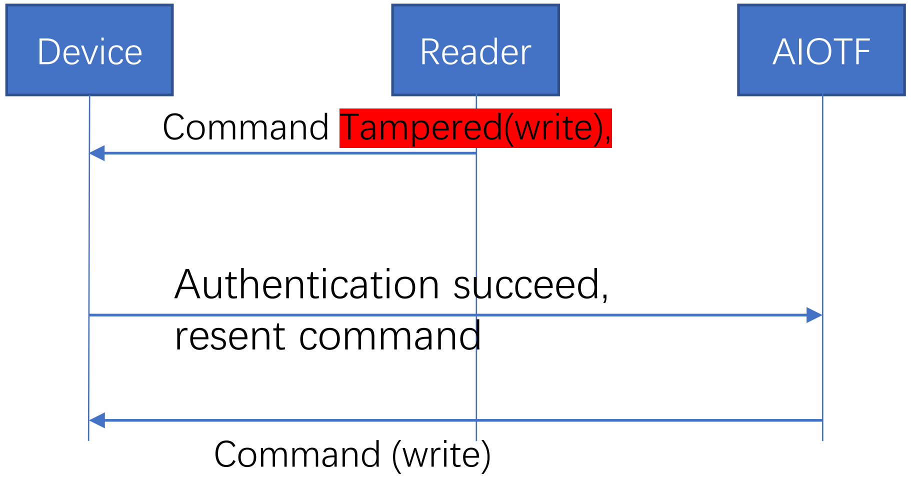
14. AIoT device sends Command response to RAN

15.RAN sends Command response to AIOTF.

Step 11-15 is inline with clause 6.2.3 Procedure for Command in TS 23.369 [2].

Command authentication- implicit authentication NOT inline with 5G AKA and cause complicated failure case

- No implicit authentication is used before in 3GPP, and no security guarantee for implicit authentication. We need the same security level as in 33501 authentication, and the security requirement for authentication and integrity protection is different. MAC for Authentication is 64 bit according to 33102, and MAC for integrity protection is only 32 bit.
- Also for security consideration and future proofing, in the future, session key could be possible used for multiple command. So for authentication purpose, session key is not as secure as root key Kaiot. Authentication MAC needs to be derive from root key Kaiot in ADM. We need the same security level as in 33501 authentication. MAC for Authentication is 64 bit according to 33102, and MAC for integrity protection is only 32 bit
- For example, tampering attack to command (write) will lead to authentication failure for implicit authentication
- With explicit authentication, tampering attack to command (write) will not affect the authentication result, and the device could still successfully authenticate the network. Thus the network could re send the command (and send subsequent command in future release)



Device

Reader

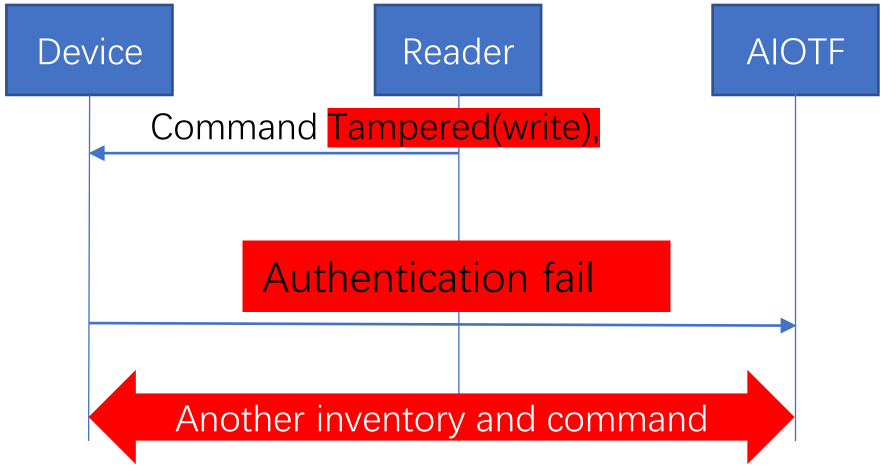
AIOTF

Command Tampered(write),

Authentication succeed, resent command

Command (write)

explicit authentication



Device

Reader

AIOTF

Command Tampered(write),

Authentication fail

Another inventory and command

implicit authentication