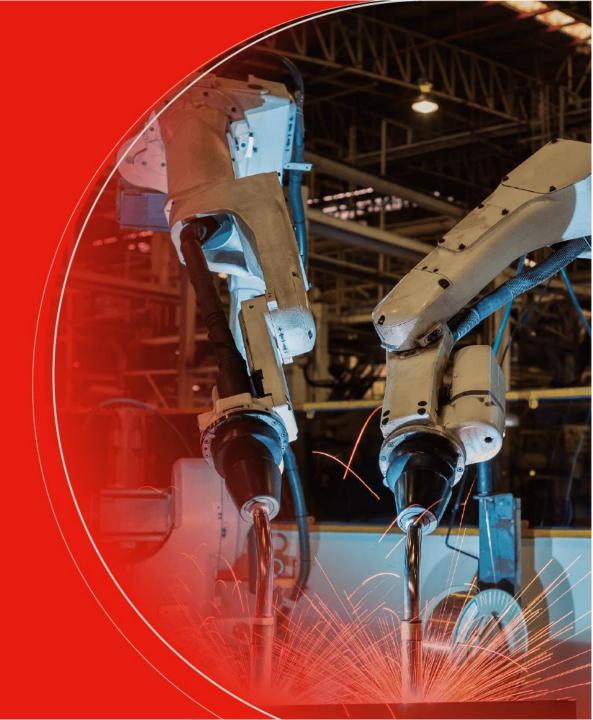
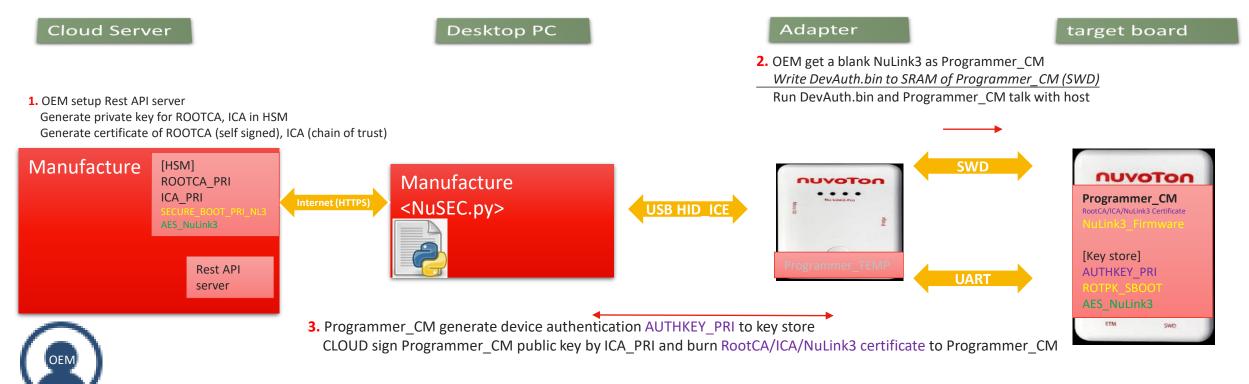
Manufacture stage



Prepare NuLink3 programmer for CM (In an OEM's secure environment)

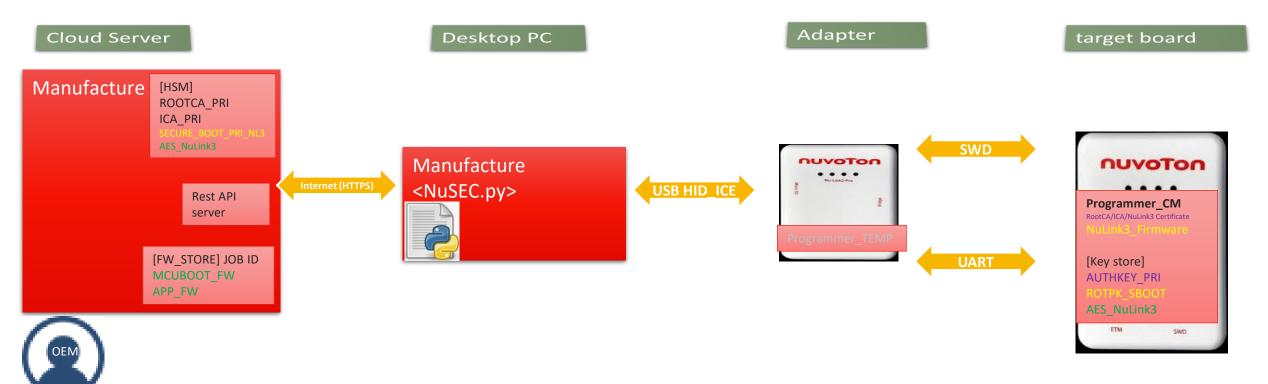


The RootCA is owned by OEM ICAs belong to different projects of OEM

- 4. NuSEC.py ask CLOUD server to generate SECURE_BOOT_PRI_NL3 for Programmer_CM NuSEC.py get SECURE_BOOT public key from CLOUD and burn to Programmer_CM ROTPK_SBOOT NuSEC.py send NuLink3 firmware to cloud server and HSM signed it by SECURE_BOOT_PRI_NL3 NuSEC.py install NuLink3 firmware (signed) to Programmer_CM Enable secure lock of Programmer CM (Restricting debugger access)
- 5. Run NuLink3 firmware, Generate AES NuLink3 by ECDH between HSM and Programmer CM



EXPORT PROJECT for CM (by OEM)



- 7. OEM use PROJECT EXPORTER (function of NuSEC.py) Generating a package for the CM, the package include:
- Production count
- JOB ID (AS A TOKEN)
- IDs to check (e.g. UID list provided by chip vendor)
- 8. Encrypt package by AES_NuLink3, OEM send encrypted package to CM OEM upload target firmware (MCUBOOT: BL2, APP_FW: BL3) to FW_STORE server Bind uploaded firmware with corresponding Job ID



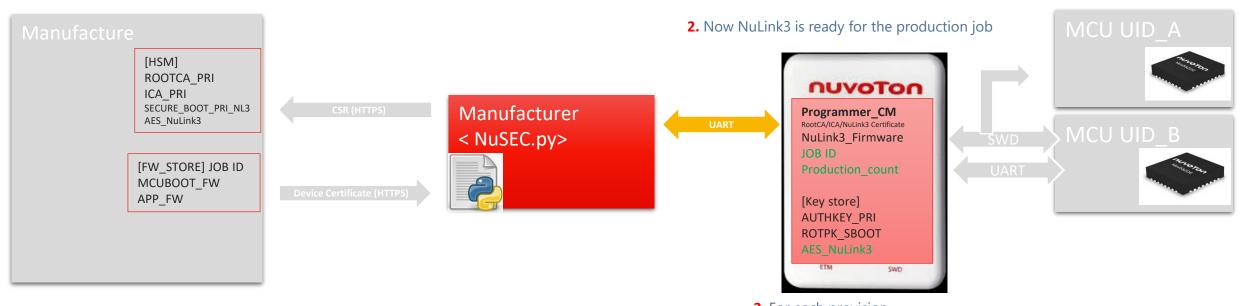
Import OEM package (CM)

Cloud Server Adapter (NuLink3) target board

0. CM use package importer function of NuSEC.py to import OEM package to NuLink3



1. The package is processed in NuLink3 SECURE ENVIRONMENT to ensure authenticity and integrity after transporting from OEM to NuLink3. The package is then decrypted (by AES_NuLink3) in NuLink3 SECURE ENVIRONMENT and production_count/Job_ID/.. are programmed in the NuLink3



3. For each provision. NuLink3 decrease the production count number Read device UID and communicate with cloud server Device authentication provision (the following slide) Secure boot key provision and FW installation (the following slide) Firmware installation will be rejected if device authentication fail

Device authentication key provision

Cloud Server Desktop PC Adapter (NuLink3) target board

0. Trigger Mass Production flow by NuSEC.py, BLE/WIFI connect to Programmer_CM and it's the **start point** of target device provisioning procedure

2. Program NuLink3 Certificate, ROOTCA CERT (SWD)
Program DevAuth_MCU.bin to SRAM and run it (SWD)

SWD/UART



1. Programmer_CM "mutual TLS" with Cloud, from authentication to get SESSION KEY https://builtin.com/sites/www.builtin.com/files/styles/ckeditor_optimize/public/inline-images/2_mutual-tls-tutorial.jpg

- 4. Target MCU send "pubkey+UID" to NuLink3
- 5. NuLink3 send CSR hash to Target MCU
- 6. Target MCU sign the CSR hash with private key of NuLink3

Manufacture

[HSM]
ROOTCA_PRI
ICA_PRI
SECURE_BOOT_PRI_NL3
AES_NuLink3

Rest API
Server

[FW_STORE] JOB ID
MCUBOOT_FW
APP_FW

7. Send CSR (certificate signing request) to CLOUD

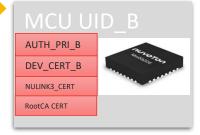


8. CLOUD verify signature, create device certificate DEV_CERT (UID, pubkey, ICA sign), and then provision it to the MCU





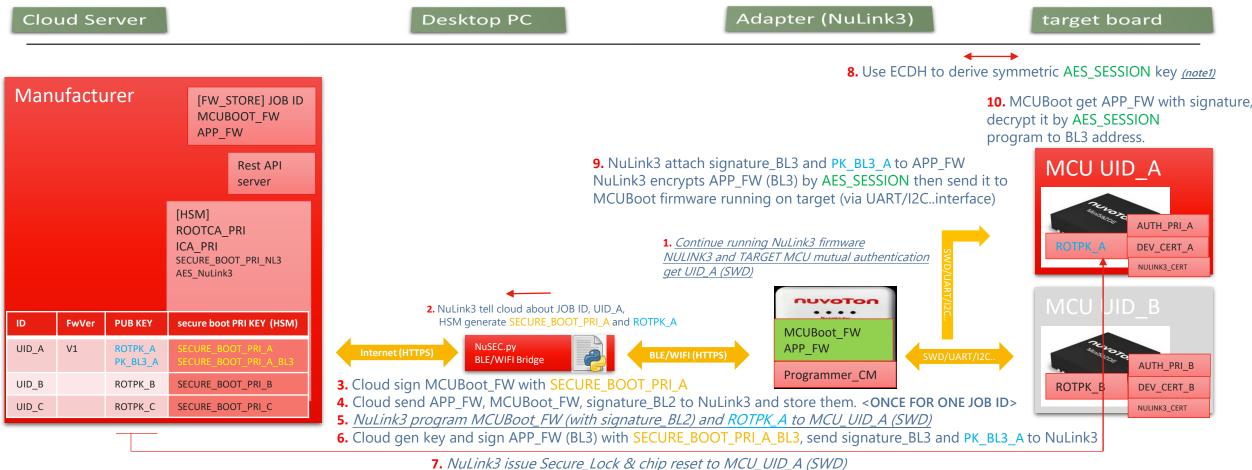
3. Generate private key and store it in key store



Public key: 512 bits (64B) Unique ID: 128 bits (16B)

Certificate: 16000 bits (2000bytes)

Firmware attestation - secure boot key and FW install

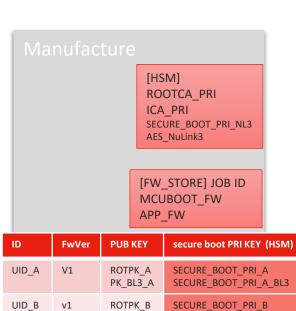


- 7. NuLink3 issue Secure_Lock & chip reset to MCU_UID_A (SWD)

 MCU_UID_A run from SecureBoot -> MCUBoot (BL2 with firmware upgrade function)
- 11. NuLink3 notify the could MCU UID_A has been successful provisioned with FwVer v1. End point 🛇
- 12. Loop until production counter decrease to zero, each target MCU has being provisioned from start point to the end point by the NuLink3 system.
- 13. MCUBoot_FW and APP_FW on NuLink3 will be erased. Traceability log table can be found on the cloud

Status report

Cloud Server Desktop PC Adapter (NuLink3) target board



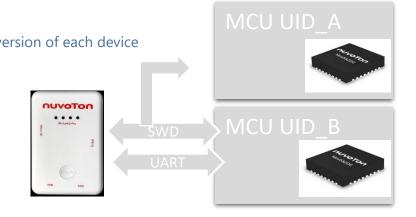
ROTPK C

UID C

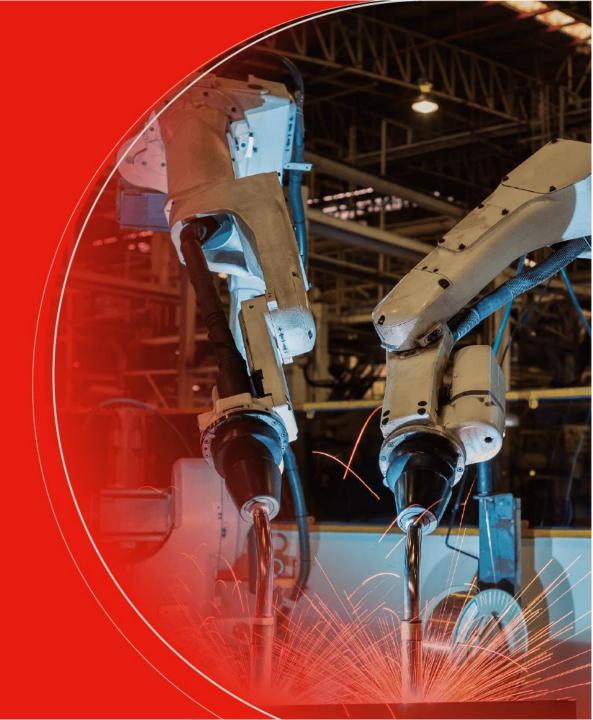
v1

SECURE BOOT PRI C

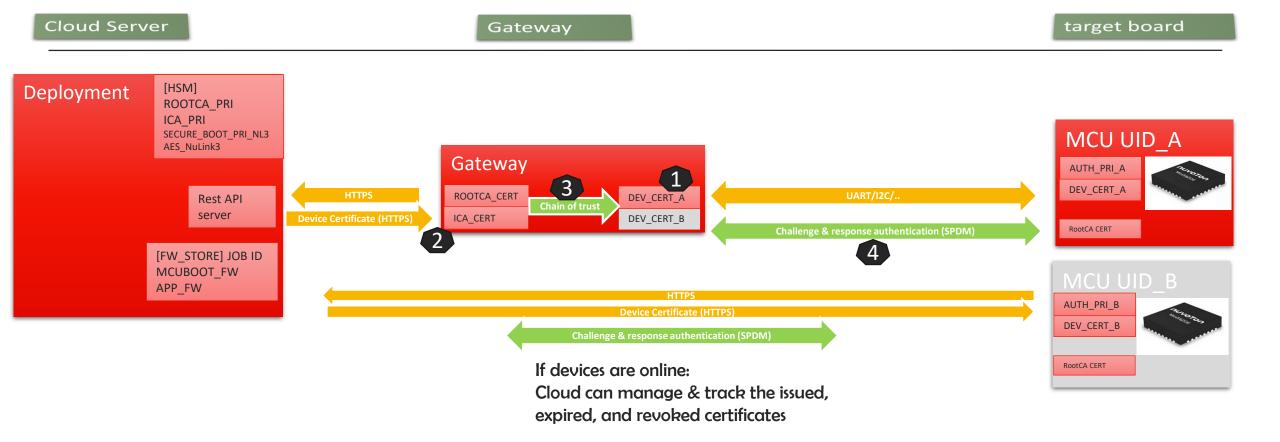




Deployment stage



Device authentication



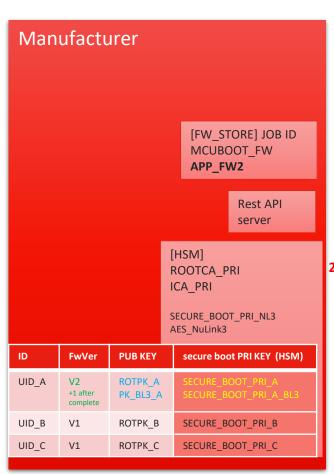
Maintenance stage



Firmware upgrade (NuSEC.py + NuLink3 bridge)

Adapter (NuLink3) Cloud Server Desktop PC

4. Use ECDH to derive symmetric AES_SESSION key



NuLink3 and Target MCU mutual authenticate (by DEV_CERT/ROOTCA/ICA cert, NULINK3_CERT) NuLink3 check firmware rollback NuLink3 attach signature BL3 and PK BL3 A to APP FW2 NuLink3 encrypts APP_FW2 by AES_SESSION then send it to MCUBoot firmware running on target (via UART/I2C..interface)

6. MCUBoot get APP_FW with signature decrypt it by AES_SESSION program to BL3 address.

target board





- Put upgrade APP_FW2 to FW_STORE server
 - 1. Trigger FW upgrade by NuSEC.py, NuLink3 run FW upgrade application, Programmer_CM "mutually_TLS" with cloud get UID A and firmware version from device
- 2. NuLink3 tell cloud about UID A. NuSEC know UID A FwVer is V1 and need update



NuSEC.py BLE/WIFI Bridge



.... APP FW2 Programmer CM RootCA CERT [Kev store] **AUTHKEY PRI**

3. Cloud send APP FW2 < ONCE, if batch update > Cloud sign APP FW2 with SECURE BOOT PRI A BL3, send signature BL3 and PK BL3 A to NuLink3.



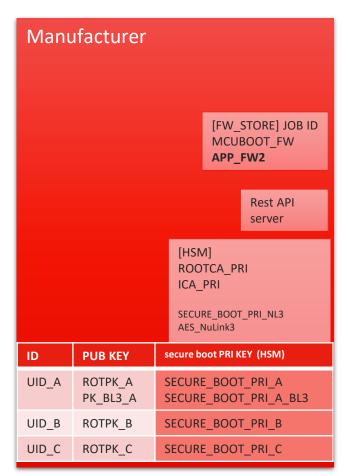
Firmware OTA upgrade (direct)

Cloud Server

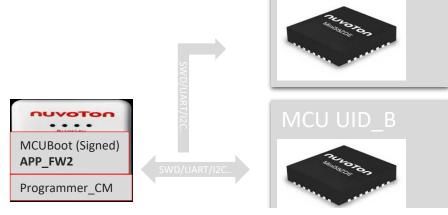
Desktop PC

Adapter (NuLink3)

target board



Put upgrade APP FW2 to FW STORE server



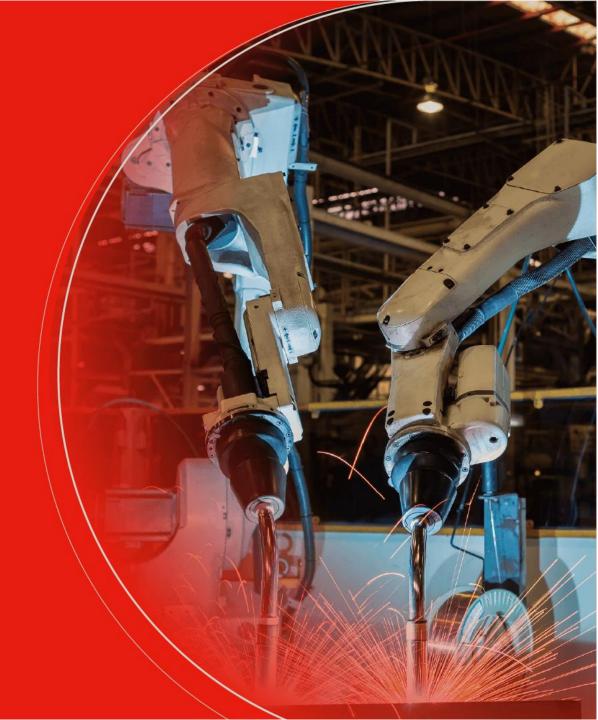
MCUBoot get APP_FW with signature, decrypt it by AES_SESSION program to BL3 address.

HTTPS/MQTTS

Device and cloud do mutual_TLS and derive AES_SESSION
Cloud prevent firmware rollback
Cloud attach signature_BL3 and PK_BL3_A to APP_FW2
Cloud encrypts APP_FW2 by AES_SESSION then send it to MCUBoot

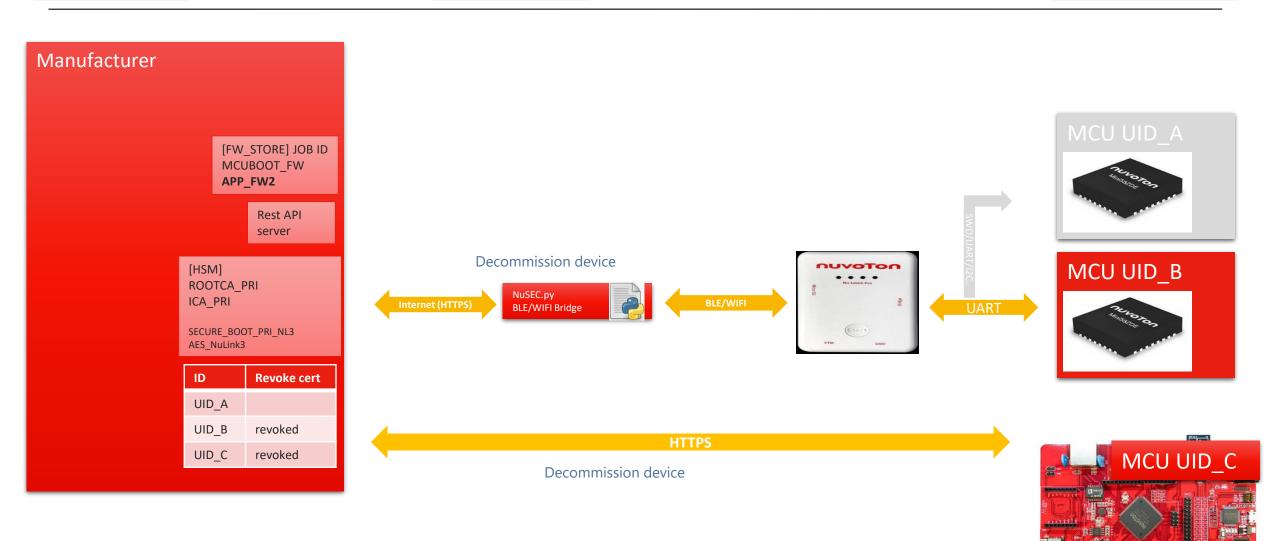


Decommission

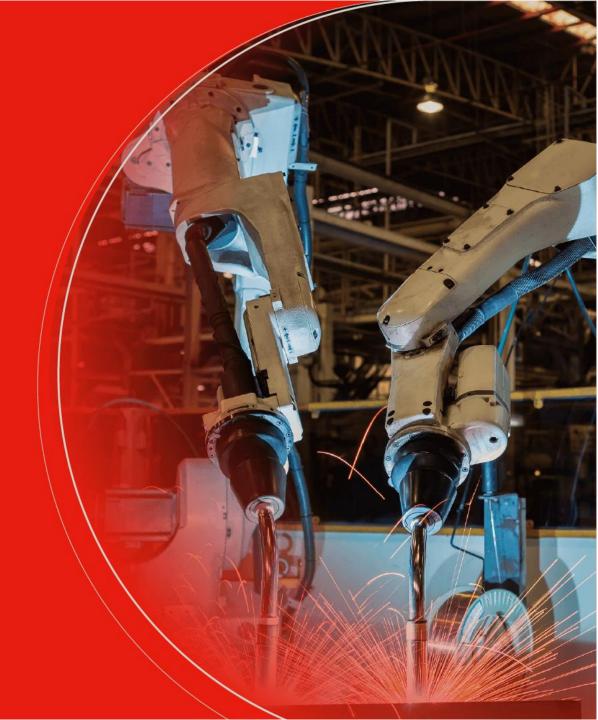


Decommission

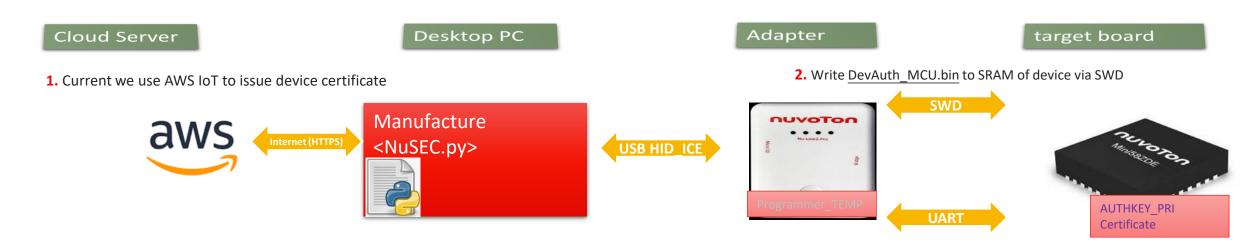
Cloud Server Desktop PC Adapter (NuLink3) target board



Current status



- Current status (Coworking with A008 JY33)
 - Device authentication: AWS <-> NuSEC.py <-> NuLink2 <-> (M2354,KM1M7C)





We setup server for next step (use Soft HSM as HSM simulator)

3. Target MCU generate device authentication AUTHKEY_PRI to key store (UART command) AWS sign Target MCU public key and burn certificate back to target MCU



UART protocol of DevAuth_MCU.bin is standardized. It can also communicates with other devices. (e.g. DATA IO)

Joy of innovation

NUVOTON

Thank You Danke Merci ありがとう Gracias Kiitos 감사합니다 धन्यबाद ك اركش הדות