BGSW- SJCE Academy Connect

Secure Boot overview

Naveen Kumar S H

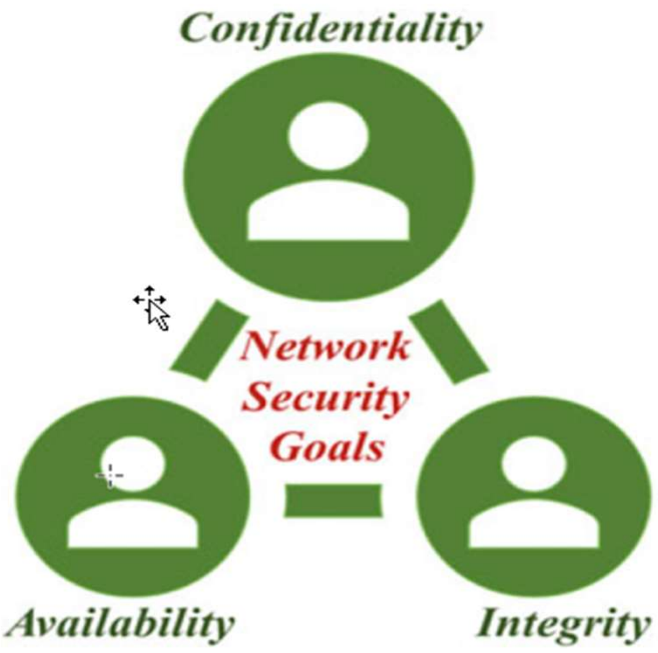
Bosch
Global
Software
Technologies
alt_future

Security Features overview

CIA - RECAP

Confidentiality

Confidentiality mechanism guarantees the secrecy of the transmitted information by guaranteeing that the message isn’t unveiled to an unapproved client/user.



Availability

Availability ensures information assets such as session key and applications are accessible by the authorized users.

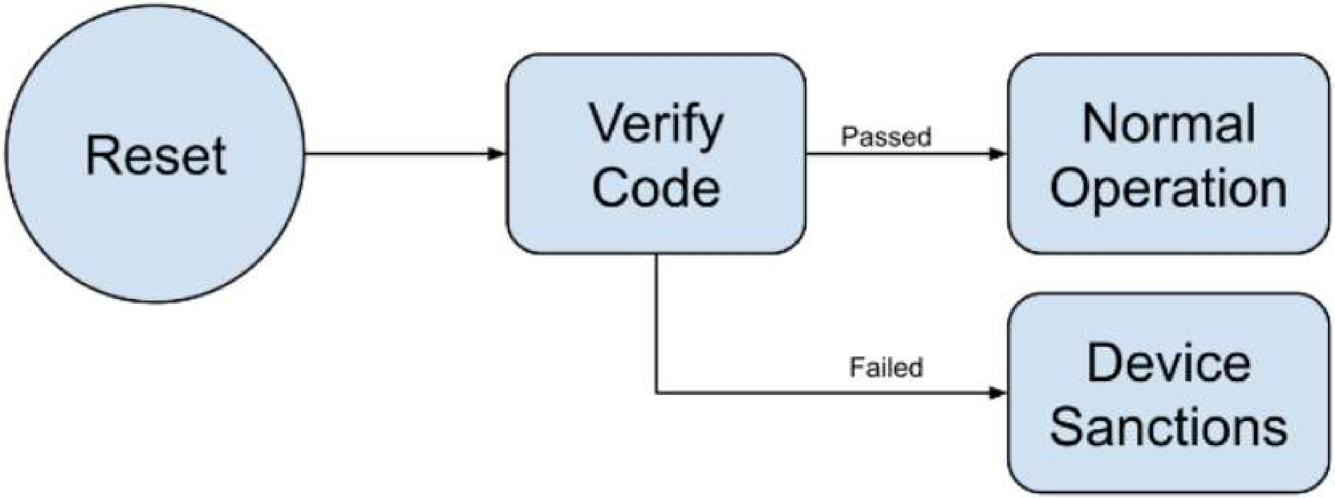
Integrity

Integrity assurance (often also referred as data integrity) of a message provides the receiver with an assurance that the data has not been modified during transmission.

Why Secure Boot?

- Secure Boot is a security feature which is used to verify the authenticity of a target.
- It is used to identify if the software is corrupted/tampered.

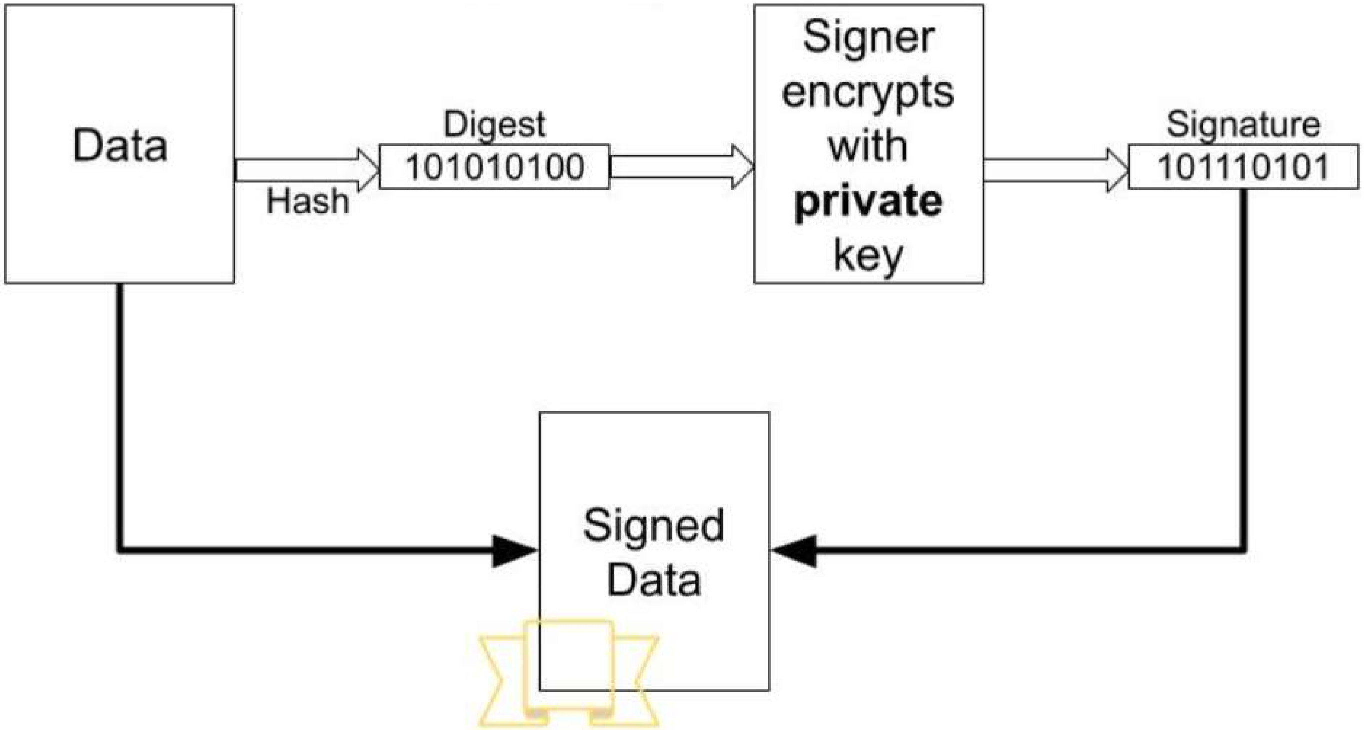
How it works?


Reset
Verify Code
Passed
Normal Operation
Failed
Device Sanctions

Signature generation

- To generate a signature, a message digest (hash) is calculated from the input data.
- This is typically created in an enterprise setting outside of the embedded device.
- The signer encrypts the message digest with their private key.
- The encrypted digest is called the signature.

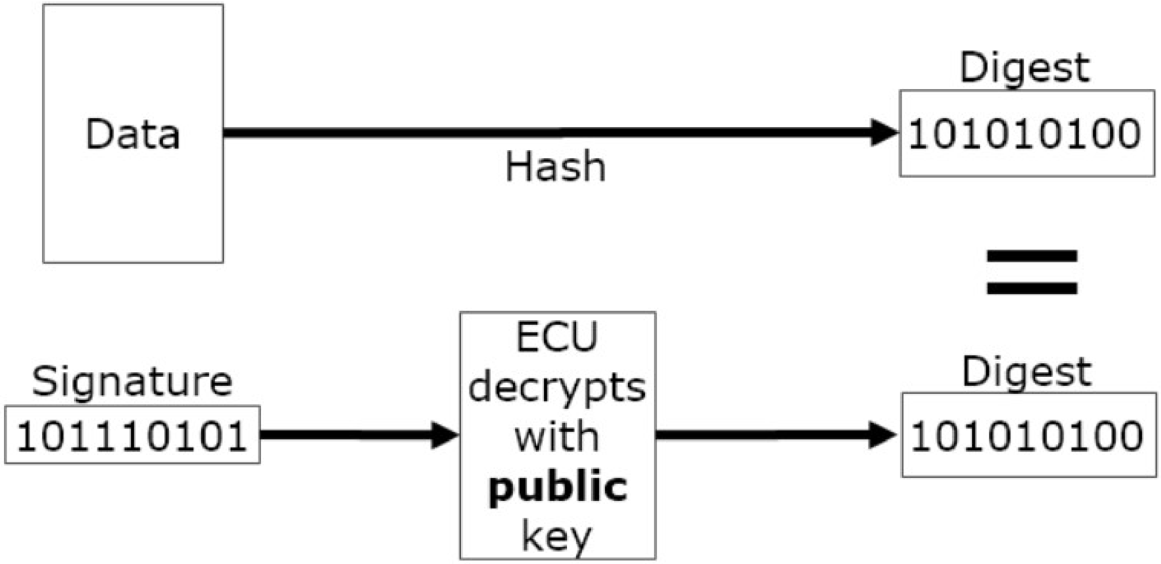
Signature generation



Signature verification

- Signature verification is the process of validating data against the code signature for integrity and authenticity.
- Verification involves calculating the message digest (hash) of the data and comparing it to the digest received in the decrypted signature.

Signature verification


Data

Hash

Digest

101010100

=

Signature

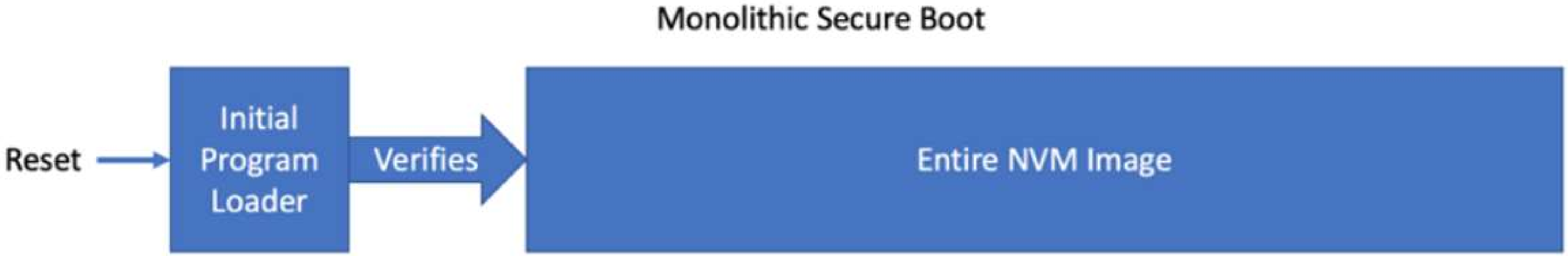
101110101

ECU decrypts with **public** key

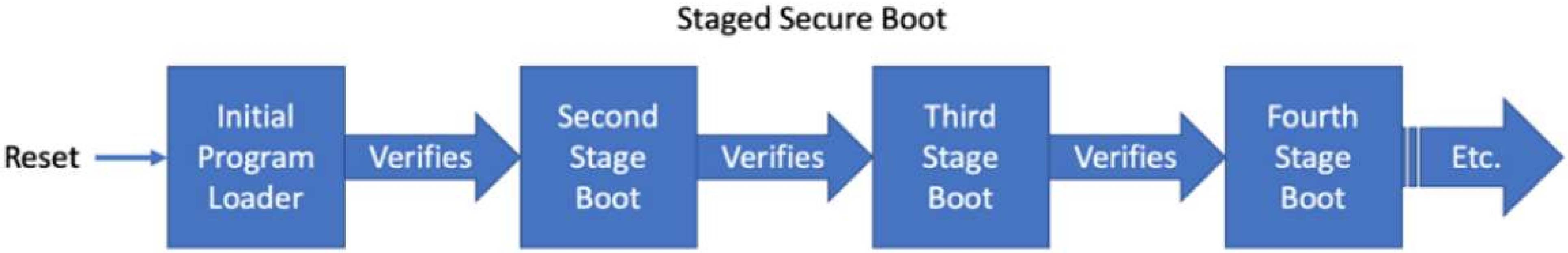
Digest

101010100

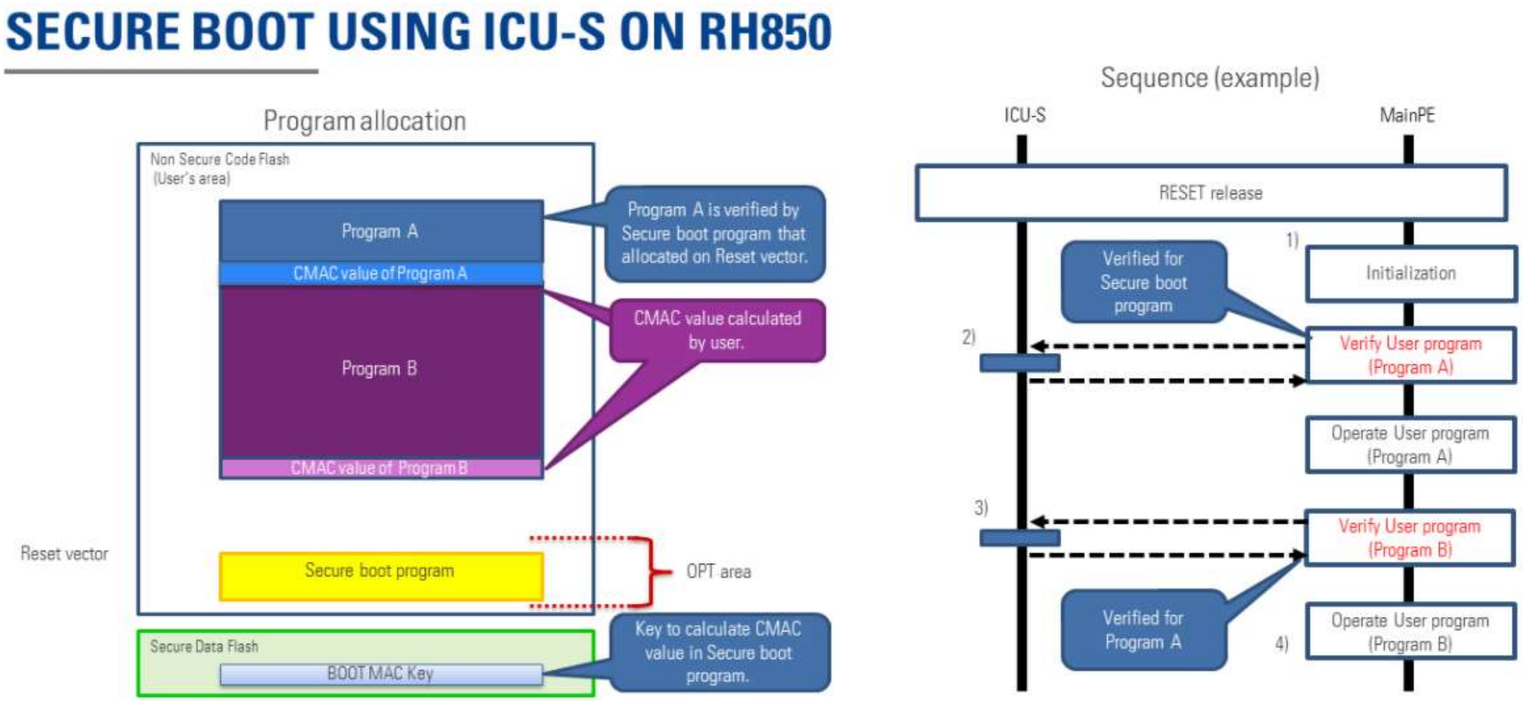
Monolithic Approach



Staged Approach



Secure Boot in Renesas RH850 – ICU S



SECURE BOOT USING ICU-S ON RH850

Program allocation

Non Secure Code Flash
(User's area)

Program A

CMAC value of Program A

Program B

CMAC value of Program B

Program A is verified by Secure boot program that allocated on Reset vector.

CMAC value calculated by user.

Reset vector

Secure boot program

OPT area

Secure Data Flash

BOOT MAC Key

Key to calculate CMAC value in Secure boot program.

Sequence (example)

ICU-S

MainPE

RESET release

1)

Initialization

Verified for Secure boot program

2)

Verify User program (Program A)

Operate User program (Program A)

3)

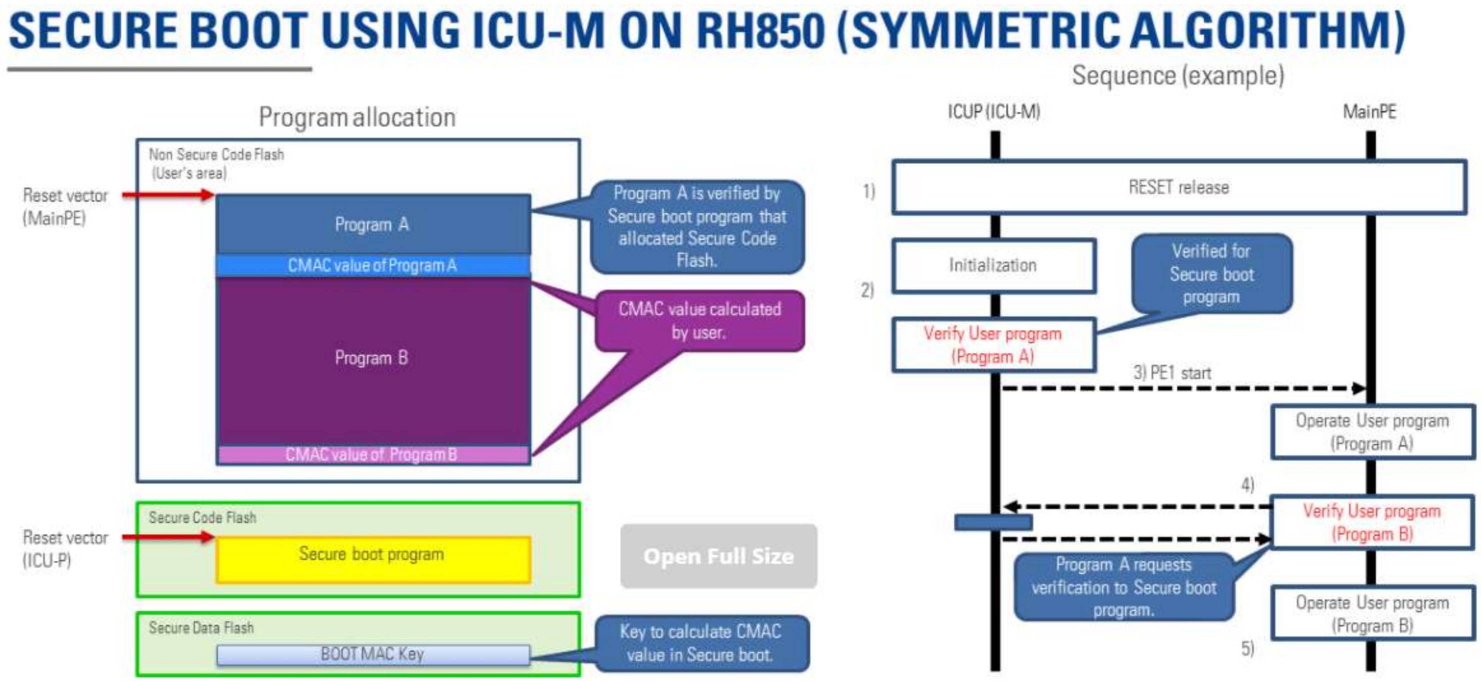
Verify User program (Program B)

Verified for Program A

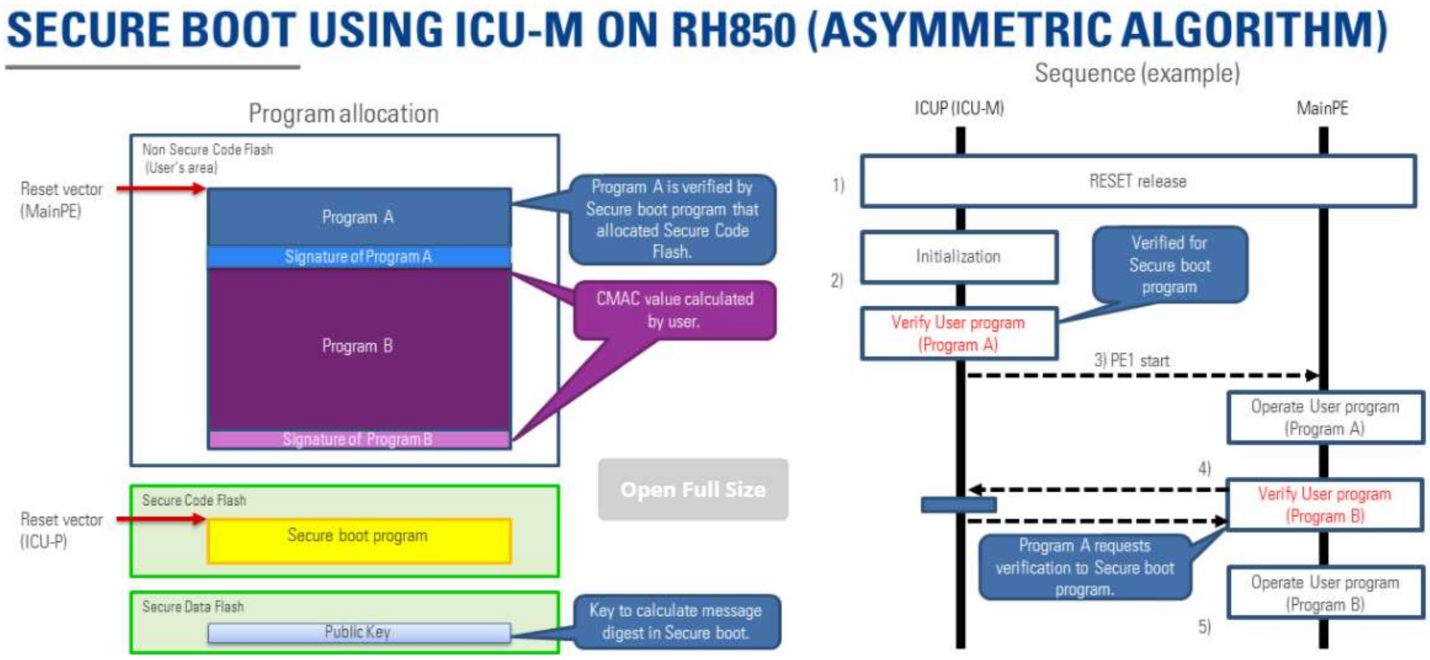
4)

Operate User program (Program B)

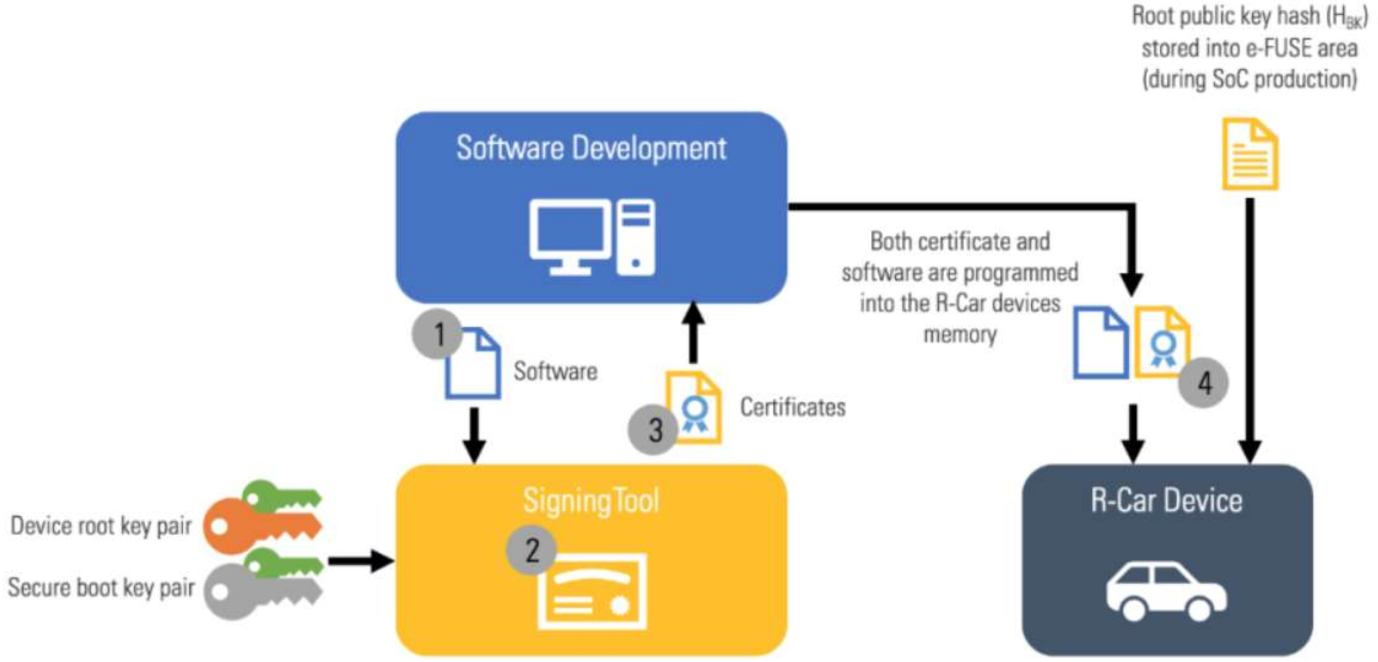
Secure Boot in Renesas RH850 – ICU M Sym


SECURE BOOT USING ICU-M ON RH850 (SYMMETRIC ALGORITHM)
Program allocation
Non Secure Code Flash (User's area)
Reset vector (MainPE)
Program A
CMAC value of Program A
Program B
CMAC value of Program B
Program A is verified by Secure boot program that allocated Secure Code Flash.
CMAC value calculated by user.
Secure Code Flash
Reset vector (ICU-P)
Secure boot program
Secure Data Flash
BOOT MAC Key
Key to calculate CMAC value in Secure boot.
Open Full Size
Sequence (example)
ICUP (ICU-M)
MainPE
1)
RESET release
2)
Initialization
Verified for Secure boot program
Verify User program (Program A)
3) PE1 start
Operate User program (Program A)
4)
Verify User program (Program B)
Program A requests verification to Secure boot program.
Operate User program (Program B)
5)

Secure Boot in Renesas RH850 – ICU M A-Sym


SECURE BOOT USING ICU-M ON RH850 (ASYMMETRIC ALGORITHM)
Program allocation
Non Secure Code Flash (User's area)
Reset vector (MainPE)
Program A
Signature of Program A
Program B
Signature of Program B
Program A is verified by Secure boot program that allocated Secure Code Flash.
CMAC value calculated by user.
Secure Code Flash
Reset vector (ICU-P)
Secure boot program
Secure Data Flash
Public Key
Key to calculate message digest in Secure boot.
Open Full Size
Sequence (example)
ICUP (ICU-M)
MainPE
1)
RESET release
Initialization
Verified for Secure boot program
2)
Verify User program (Program A)
3) PE1 start
Operate User program (Program A)
4)
Verify User program (Program B)
Program A requests verification to Secure boot program.
Operate User program (Program B)
5)

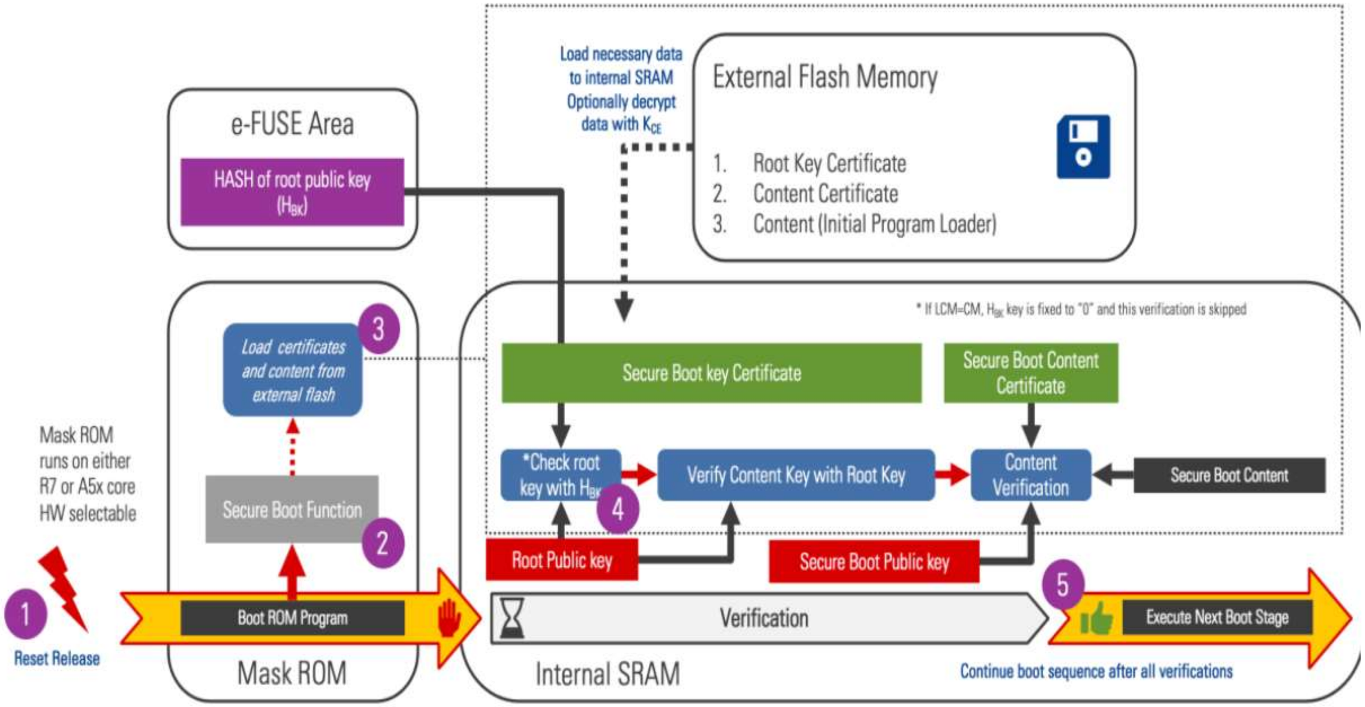
Secure Boot Environment



Root public key hash (H_{BK}) stored into e-FUSE area (during SoC production)

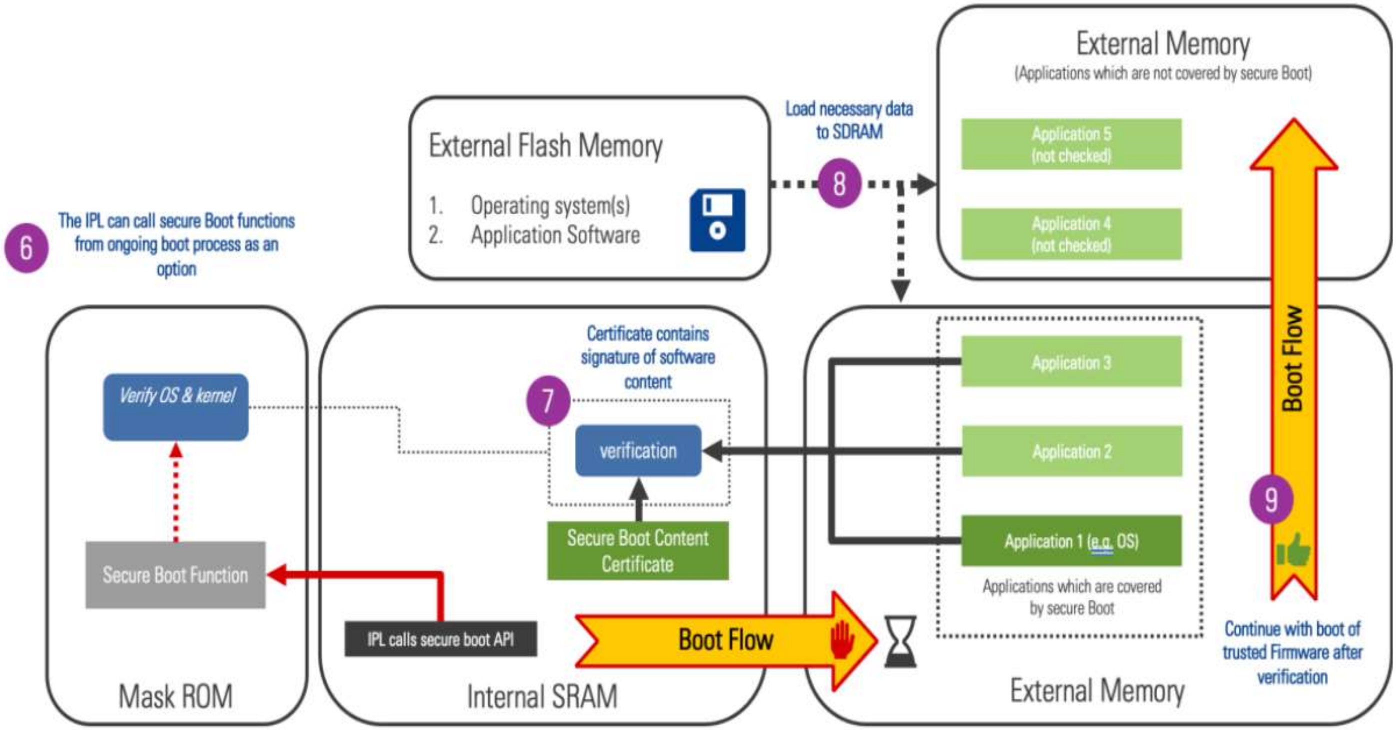
Both certificate and software are programmed into the R-Car devices memory

R-CAR sequence



e-FUSE Area
HASH of root public key (H_{BK})
Load necessary data to internal SRAM Optionally decrypt data with K_{CE}
External Flash Memory
1. Root Key Certificate
2. Content Certificate
3. Content (Initial Program Loader)
* If LCM=CM, H_{BK} key is fixed to "0" and this verification is skipped
3
Load certificates and content from external flash
Secure Boot key Certificate
Secure Boot Content Certificate
Mask ROM runs on either R7 or A5x core HW selectable
Secure Boot Function
2
*Check root key with H_{BK}
4
Verify Content Key with Root Key
Content Verification
Secure Boot Content
Root Public key
Secure Boot Public key
1
Reset Release
Boot ROM Program
Verification
5
Execute Next Boot Stage
Mask ROM
Internal SRAM
Continue boot sequence after all verifications

R-CAR sequence



6
The IPL can call secure Boot functions from ongoing boot process as an option
External Flash Memory
1. Operating system(s)
2. Application Software
Load necessary data to SDRAM
8
External Memory
(Applications which are not covered by secure Boot)
Application 5 (not checked)
Application 4 (not checked)
Verify OS & kernel
Certificate contains signature of software content
7
verification
Secure Boot Content Certificate
Secure Boot Function
IPL calls secure boot API
Boot Flow
Mask ROM
Internal SRAM
Application 3
Application 2
Application 1 (e.g. OS)
Applications which are covered by secure Boot
External Memory
Boot Flow
9
Continue with boot of trusted Firmware after verification

Chain of Trust – What it is ?

- To perform secure boot, a “root-of-trust" is required.
- Before accessing any memory region, the content of that region should be verified, so that unverified/untrusted region is not accessed. This is called chain of trust.
- Chain of trust starts with hardware trust anchor.

Thanks..!