



life.augmented

Secure boot

Presenter name

Agenda

1 Bootrom

5 BSEC and OTP

2 Activation

3 Bootrom Authentication

4 FSBL Authentication

The first code executed after the STM32MP15x reset is the code in the processor itself, the bootrom.

This code is hardcoded in the MPU and can't be updated.

There are two ways to configure the bootrom :

- via the One Time Programmable fuse (OTP).
- Via the FSBL header (mandatory for the FSBL).

The bootrom provides authentication processing with ECDSA verification algorithm, based on ECC. ECDSA offers better result than RSA with a smaller key. STM32 MPU relies on a 256 bits ECDSA key.

Two algorithms are supported for ECDSA calculation (selection via STM32 header):

- P-256 NIST
- Brainpool 256

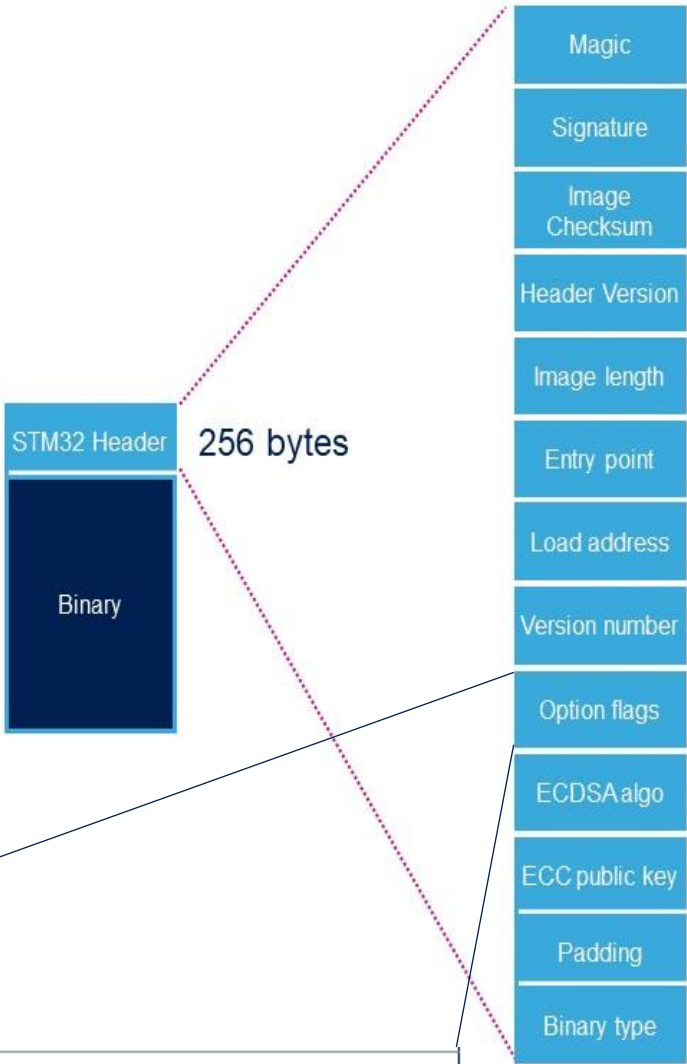


Activation

Without any other modification, the device can perform image authentication or use non authenticated images via the FSBL header option flags field: the device is still opened, let's see this option as a kind of test mode to check that the PKH is properly set.

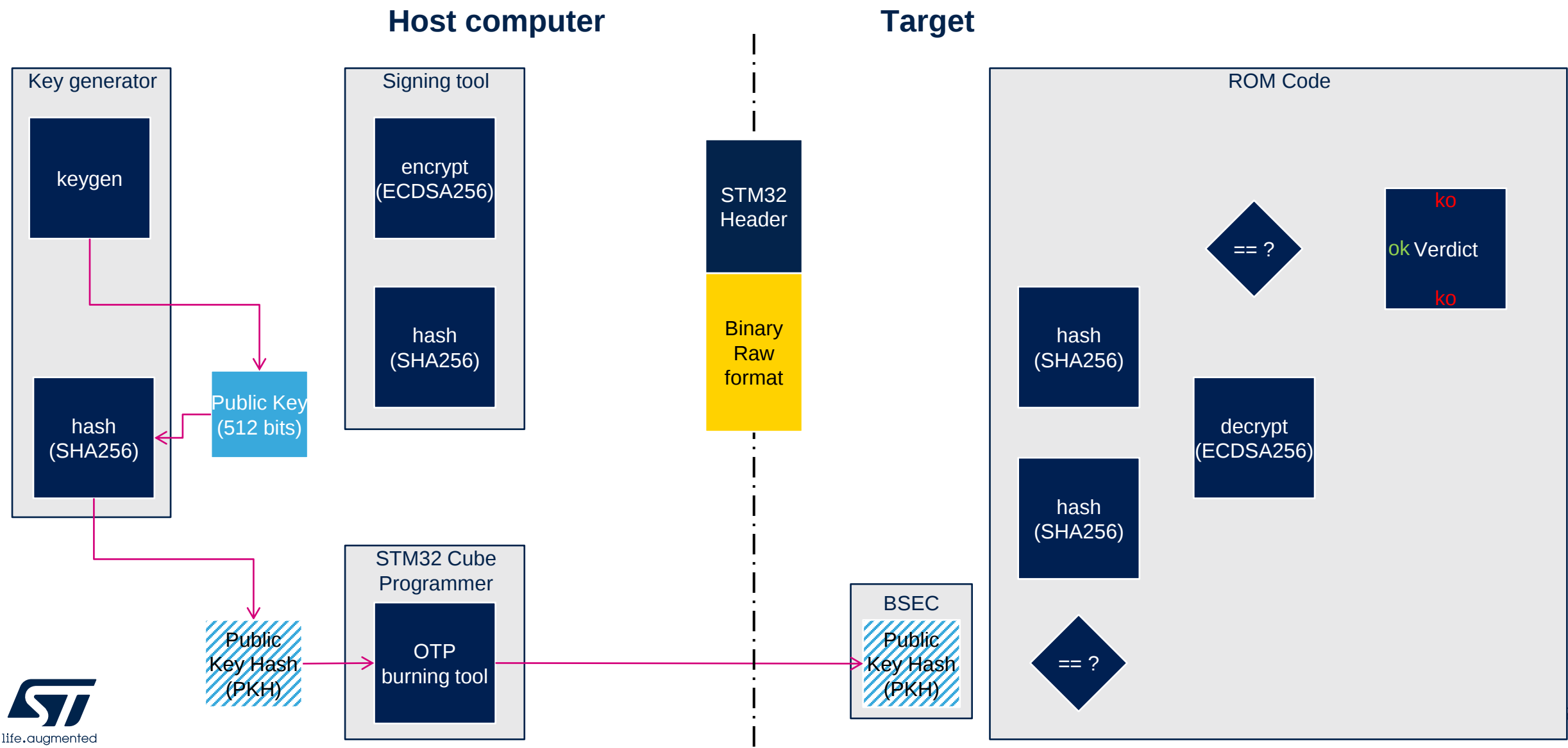
As soon as the authentication process is confirmed, the device can be closed and the user forced to use signed images.

OTP WORD0 bit 6 is the OTP bit that closes the device. Burning this bit will lock authentication processing and force authentication from the Boot ROM. Non signed binaries will not be supported anymore on the target.

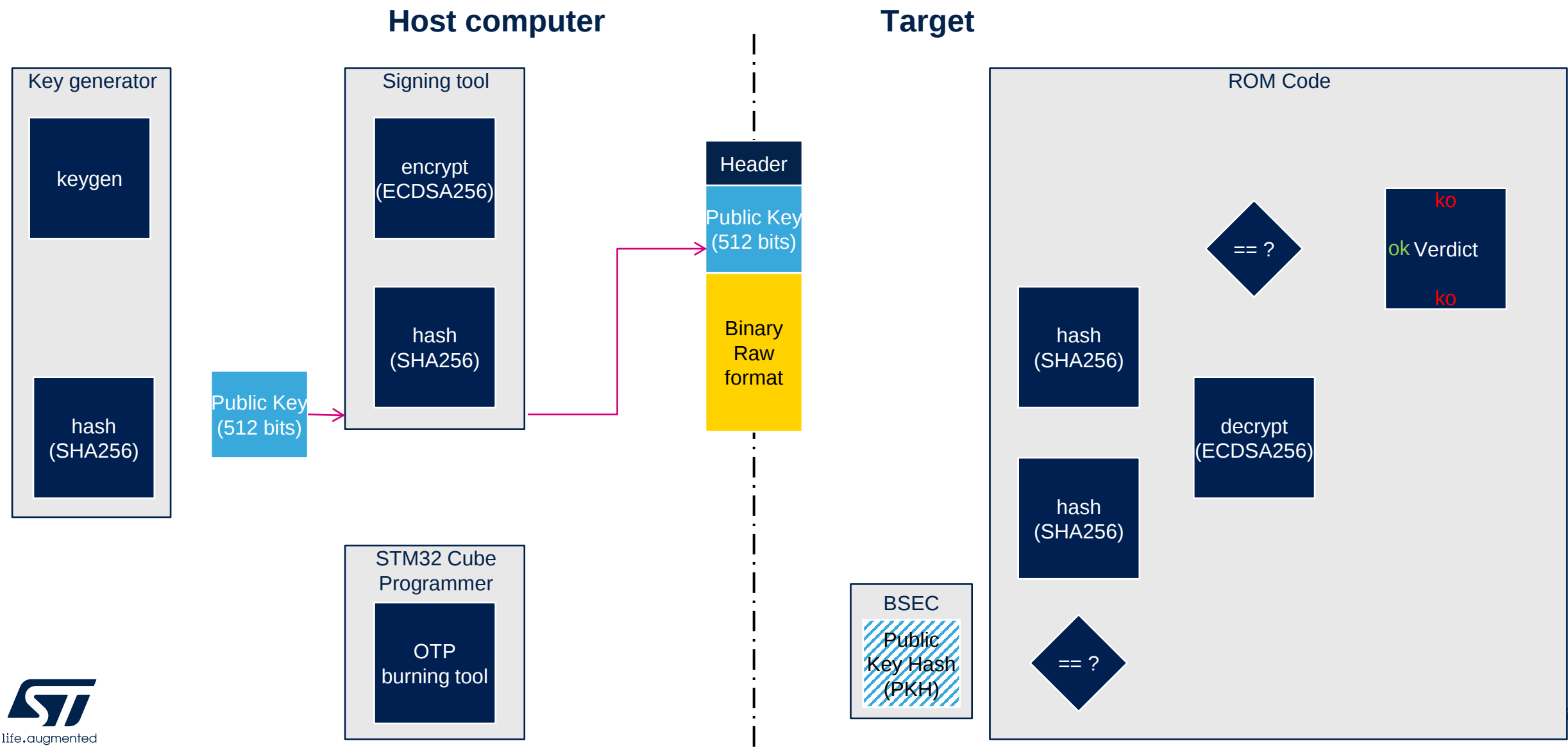


Option flags	32 bits	100	b0=1: no signature verification
--------------	---------	-----	---------------------------------

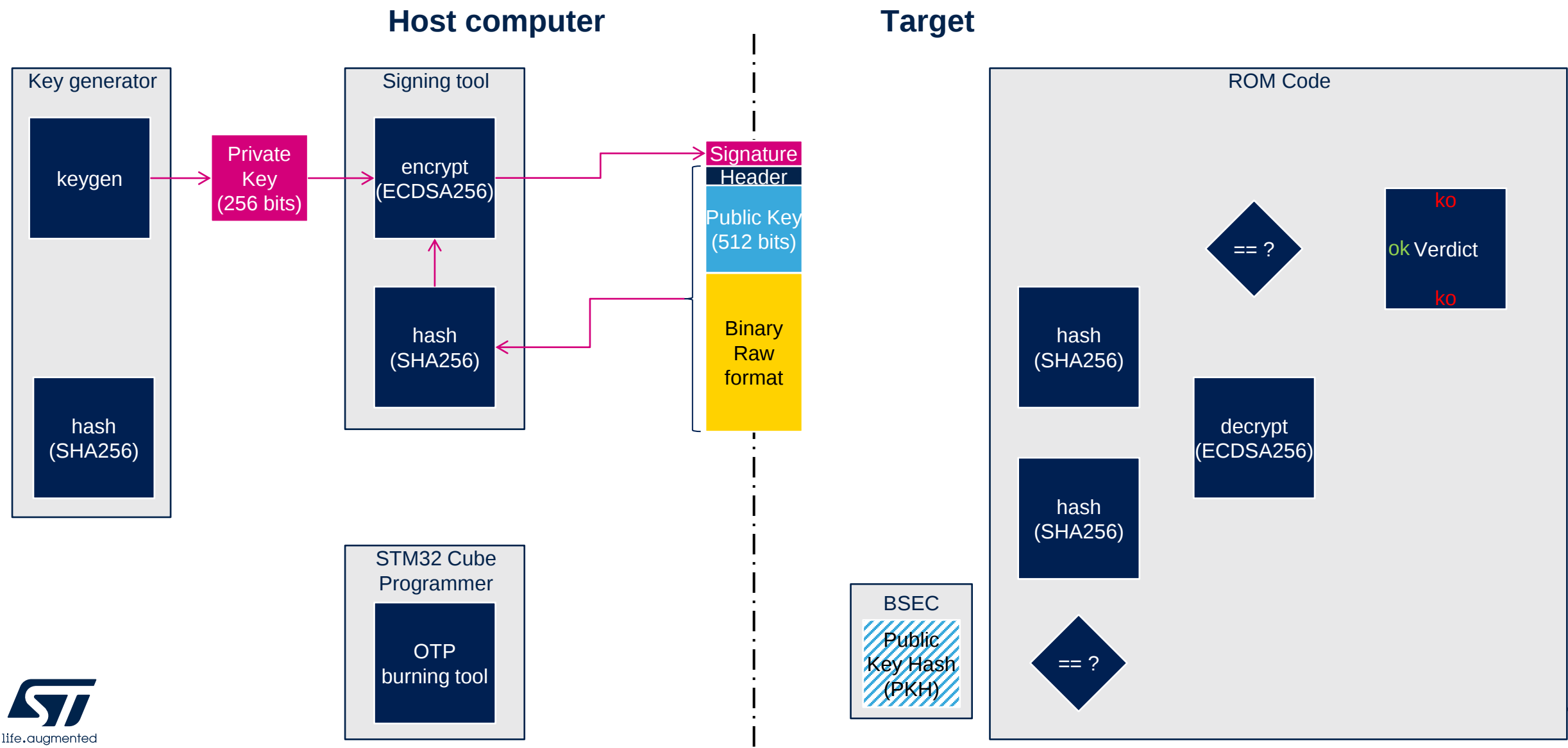
Bootrom Authentication



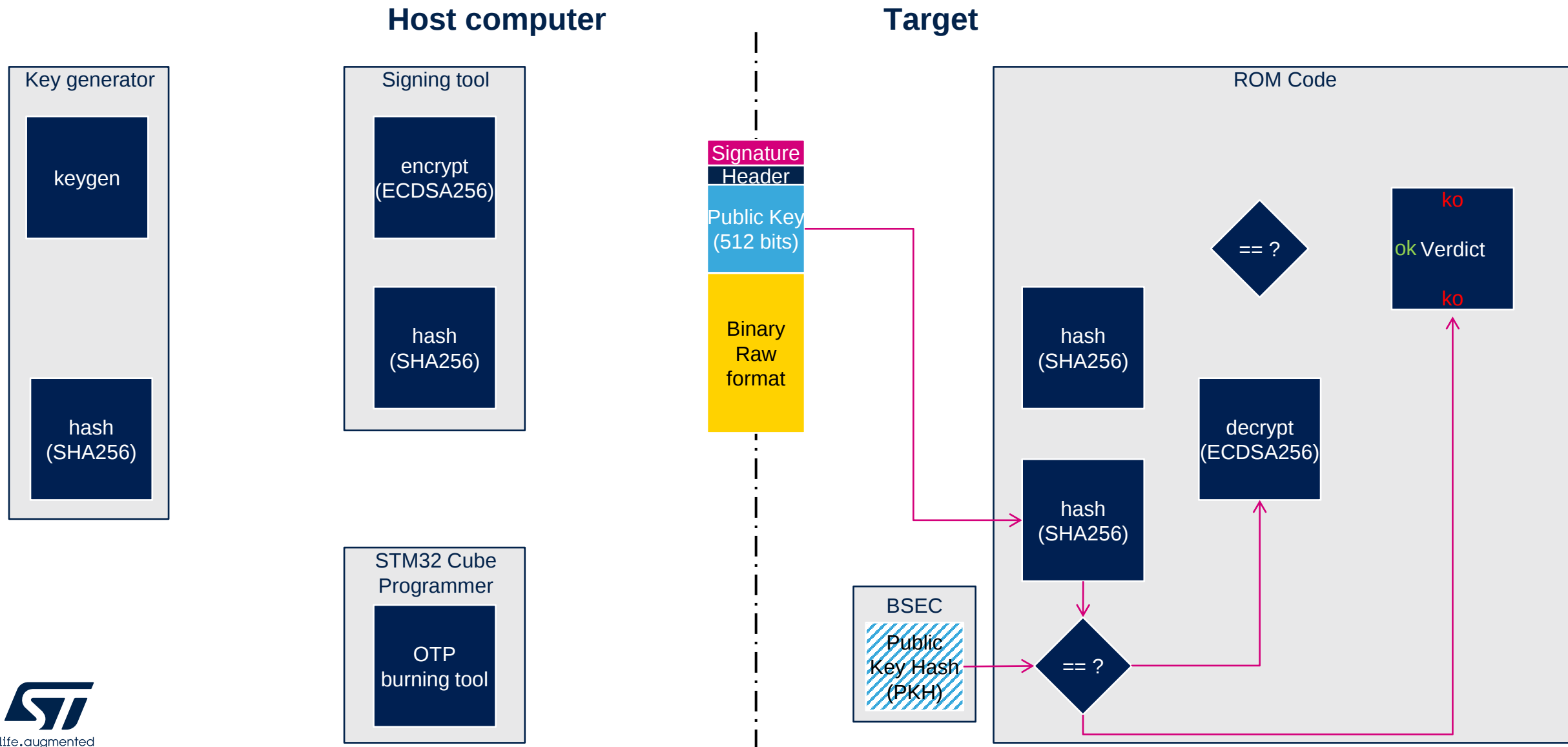
Bootrom Authentication



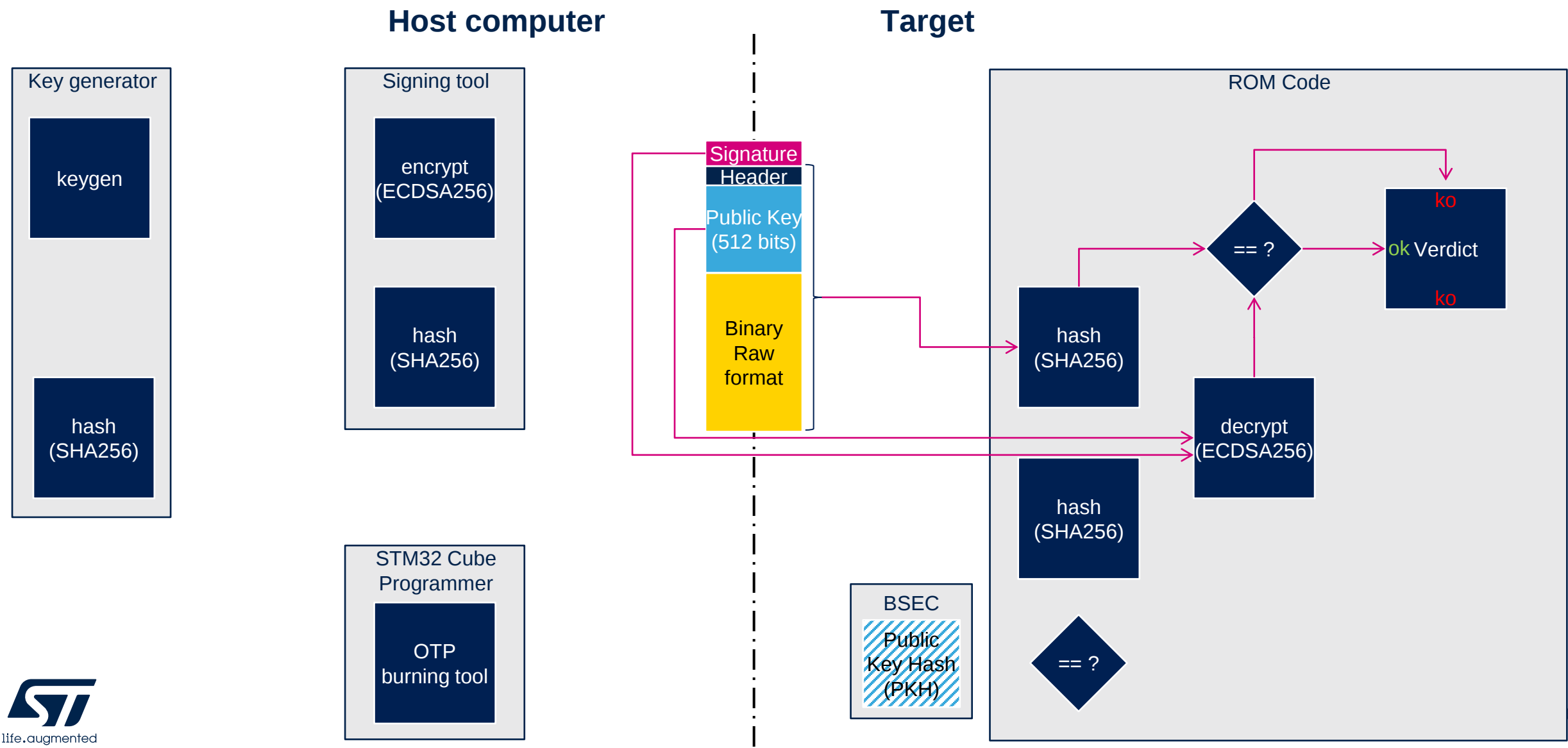
Bootrom Authentication



Bootrom Authentication



Bootrom Authentication



FSBL and SSBL authentication

FSBL in OpenSTLinux < 3.0.0 :

FSBL (TF-A) use the bootrom secure services to do the image authentication with the same ECC pair of keys, so there is no need to support ECDSA algorithm in FSBL.

FSBL is able to do the authentication of SSBL and also of the OP-TEE image (if OPTEE is used instead of BL32).

FSBL in OpenSTLinux > 3.0.0 (not yet delivered):

TF-A will use the FIP architecture see `/docs/getting_started/user-guide.rst` in TF-A source code.

If an authentication is needed from SSBL to authenticate the Linux part, ST advice to use the FIT format well described in the `doc/ulmage.FIT` in uboot source folder.

BSEC and OTP

Bsec is in charge :

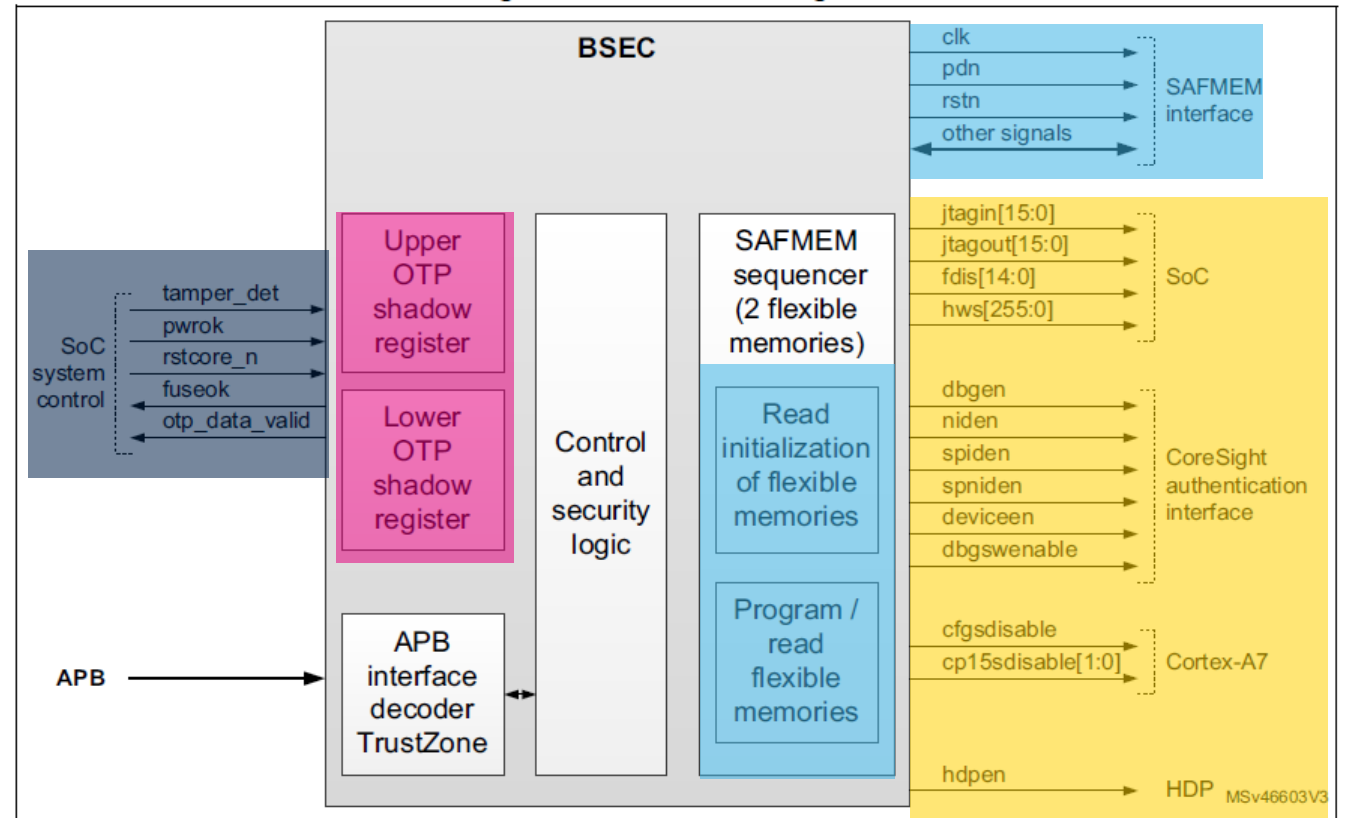
Notify the system in case of problem with the OTP

Manage the debug signals in the SOC.

Read and Write in the OTP (SAFMEM)

Manage the Shadow memory.

Figure 5. BSEC block diagram



BSEC and OTP

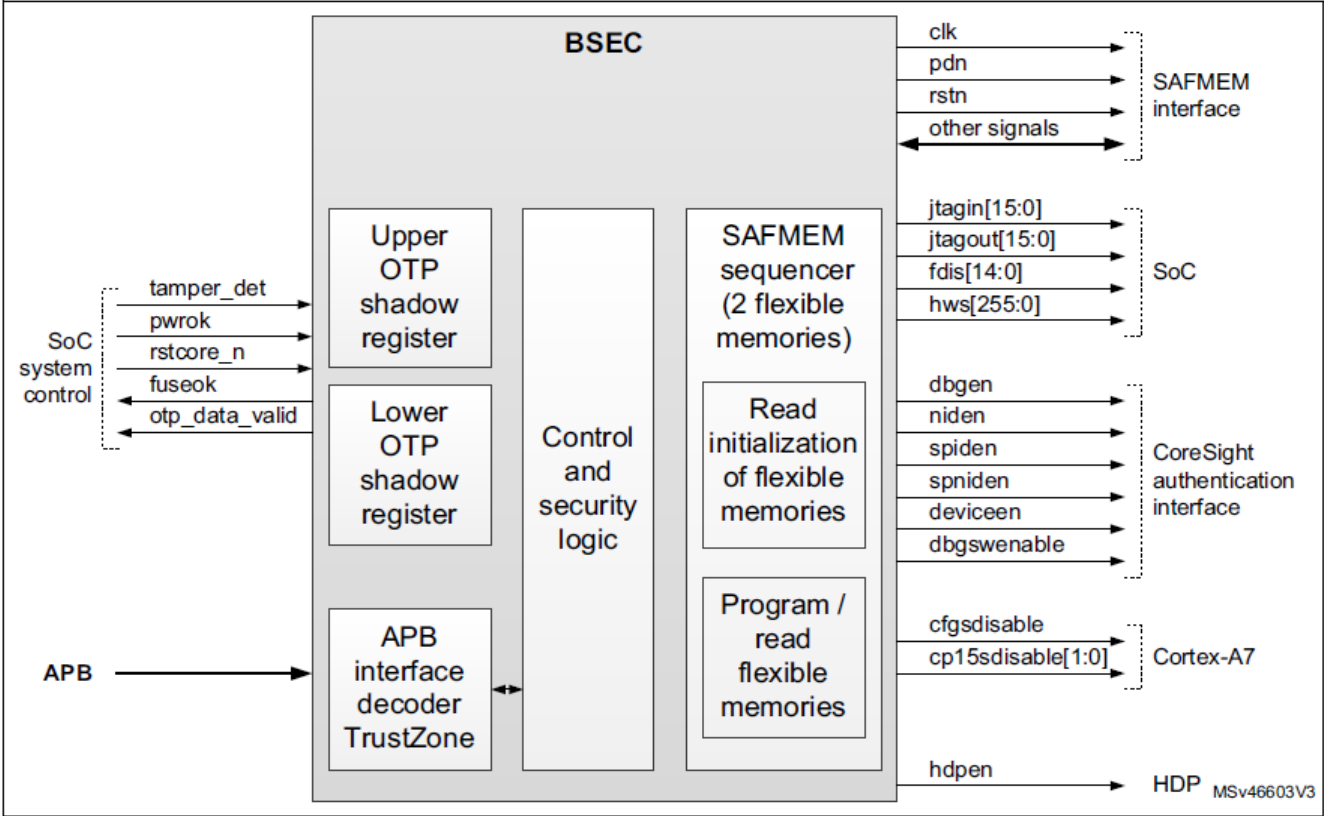
The shadow registers are loaded with the OTP values by the SOC after a hardware reset.

The values loaded are checked and use to enable/disable the different debug feature.

Programmer can read/write the OTP.

U-boot can read/write the OTP and the shadow memory.

Figure 5. BSEC block diagram



References



Secure boot :

https://wiki.stmicroelectronics.cn/stm32mpu/wiki/STM32MP15_secure_boot

Programmer :

<https://wiki.stmicroelectronics.cn/stm32mpu/wiki/STM32CubeProgrammer>

KeyGen tool :

https://wiki.stmicroelectronics.cn/stm32mpu/wiki/KeyGen_tool

Signing tool :

https://wiki.stmicroelectronics.cn/stm32mpu/wiki/Signing_tool

BSEC :

https://wiki.stmicroelectronics.cn/stm32mpu/wiki/BSEC_device_tree_configuration

Thank you