



life.augmented



STM32Trust

STM32H5 Security Secure Manager - Part 5

Hands On: Debug Authentication

Presenter: Rishi Shukla

Agenda

I

Debug Authentication
Overview

II

Debug Authentication
Ecosystem

III

Hands-On: Debug Re-Opening
and Regression

IV

Resources

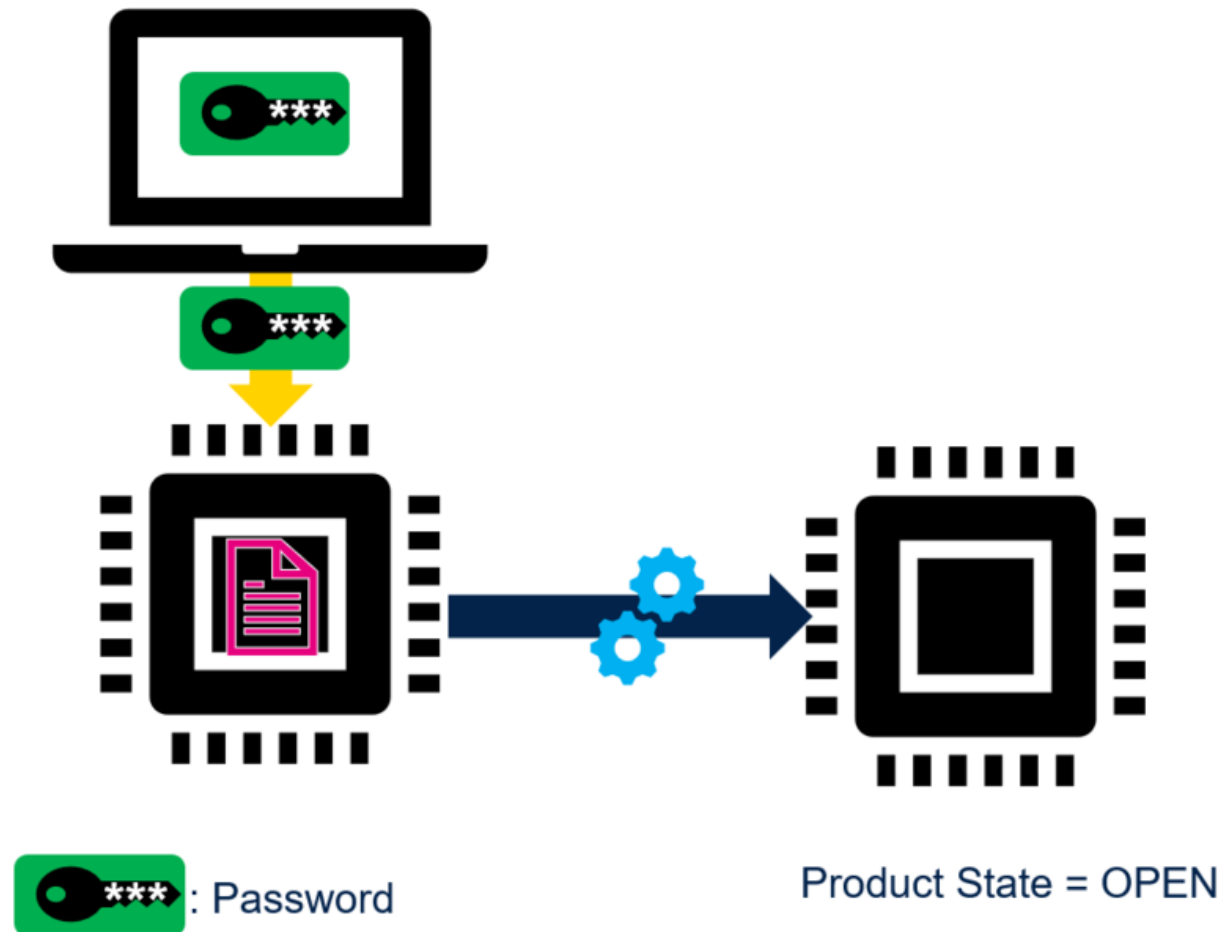
Debug Authentication Overview

What is Debug Authentication?

- The user leverages the debug authentication security feature to perform:
 - Secure regression to OPEN (full regression) or TZ-CLOSED (partial regression) product states, erasing user data in user flash memory, SRAM and OBKeys.
 - Safely re-open debug access on the STM32
- Debug Authentication usage is different depending on the TrustZone activation:
 - Password based (In TZ disabled applications)
 - Certificate based (In TZ enabled applications)

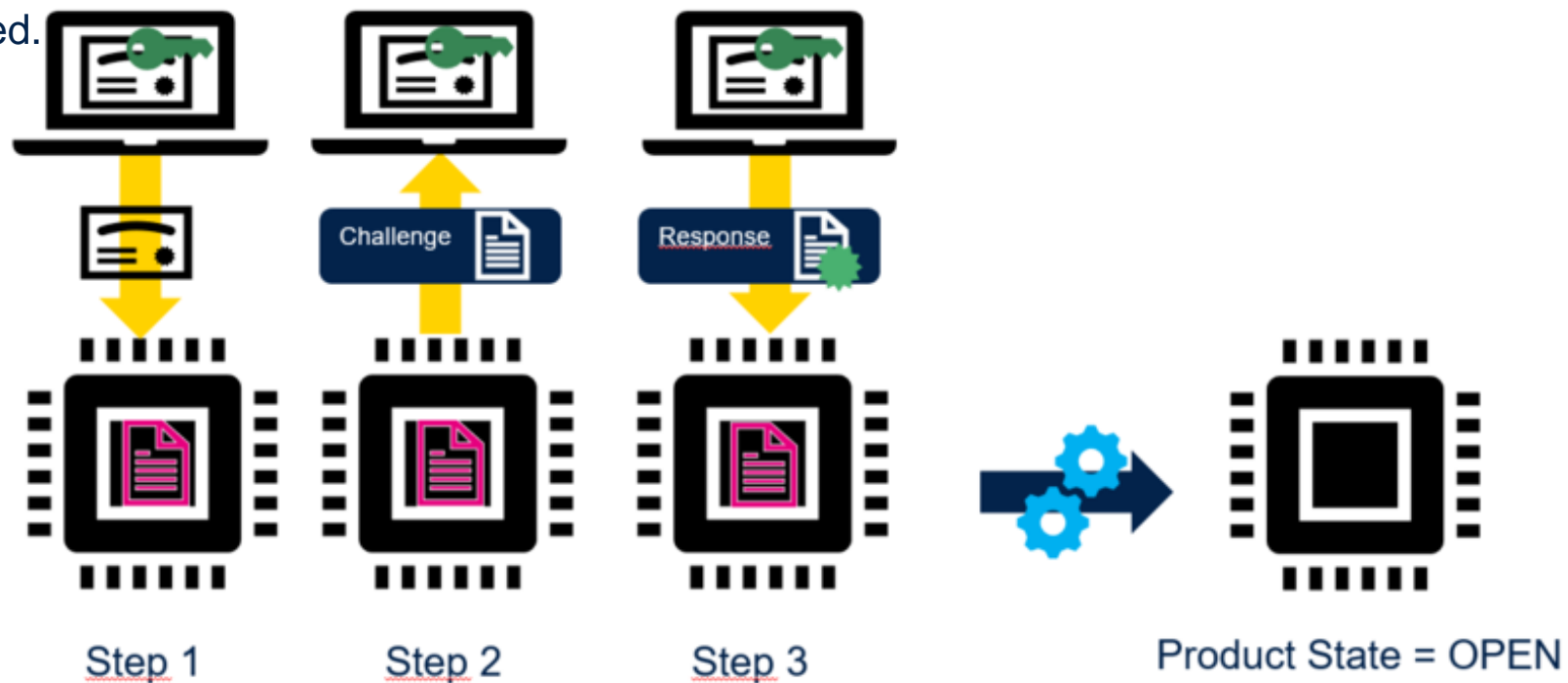
DA: Password based

- In order to access the debug authentication feature, the host sends the debug authentication password to the STM32. When the STM32 receives the hash of the password, it verifies that it corresponds to the one that is provisioned.



DA: Certificate based

- When the user access the debug authentication feature (regression or debug re-opening), he sends first a certificate and a debug authentication action request to the STM32. When the STM32 receives the certificate, it verifies that:
 - Certificate fits the one that is provisioned.
 - The authorized actions sent with the certificate match the ones provisioned.
 - The action request matches the authorized action list carried by the certificate.

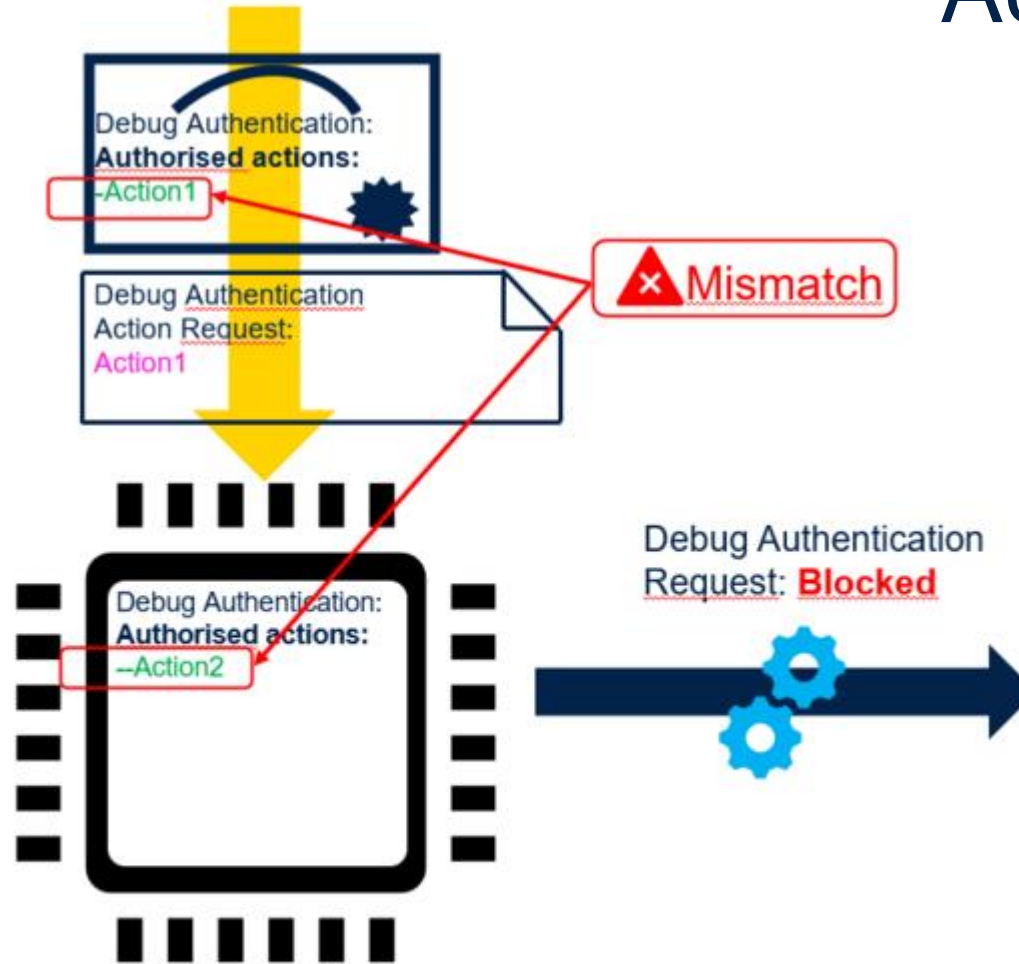


: Debug Authentication Certificate



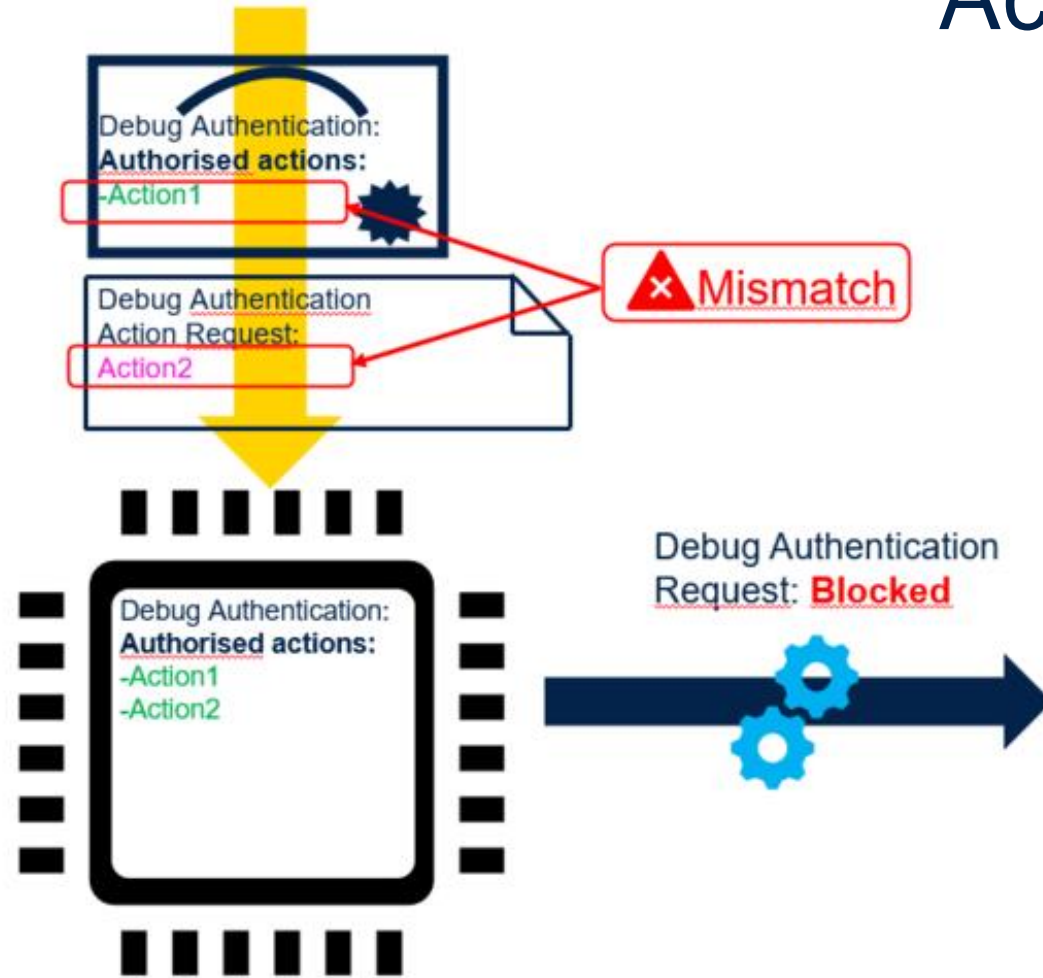
: Debug Authentication Private Key

Actions Request 1/2



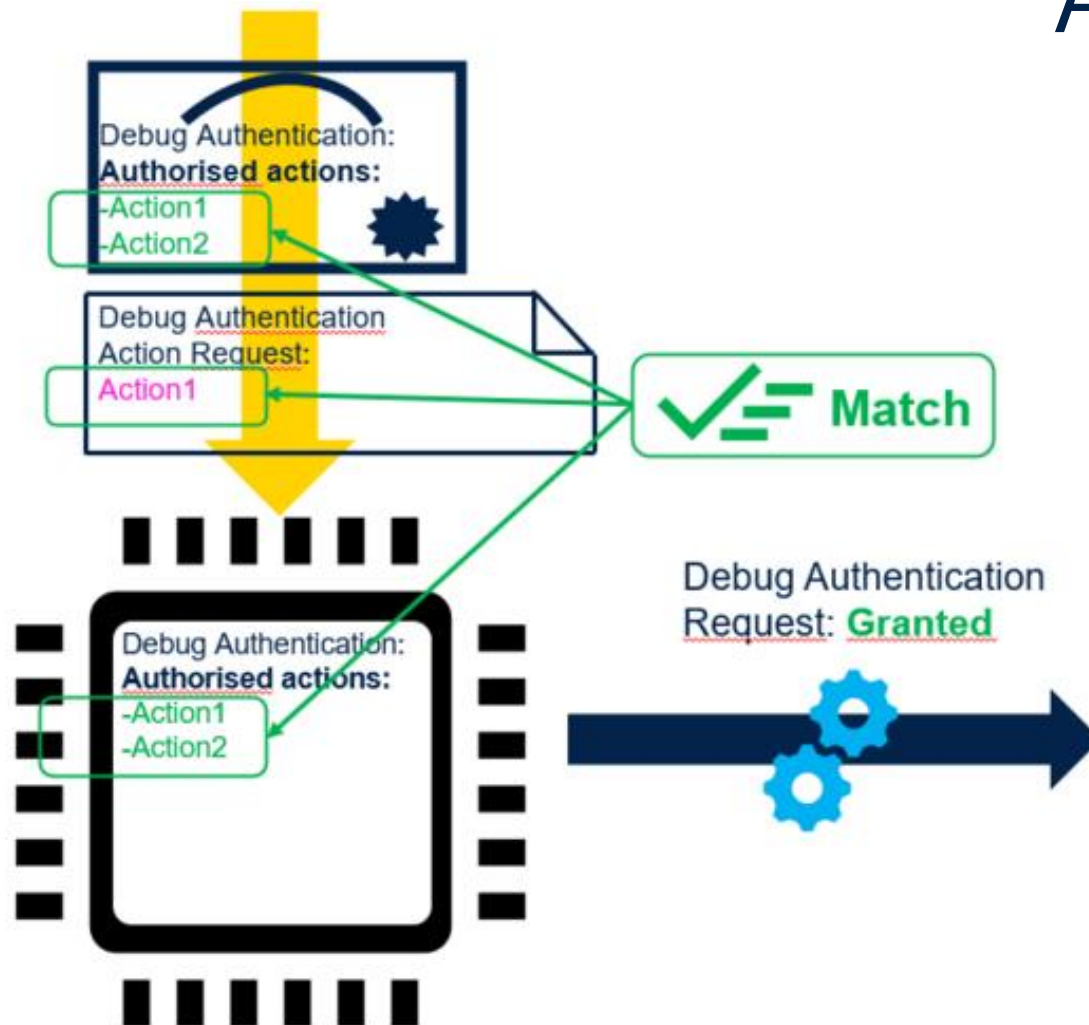
: Debug Authentication Certificate

Actions Request 2/2



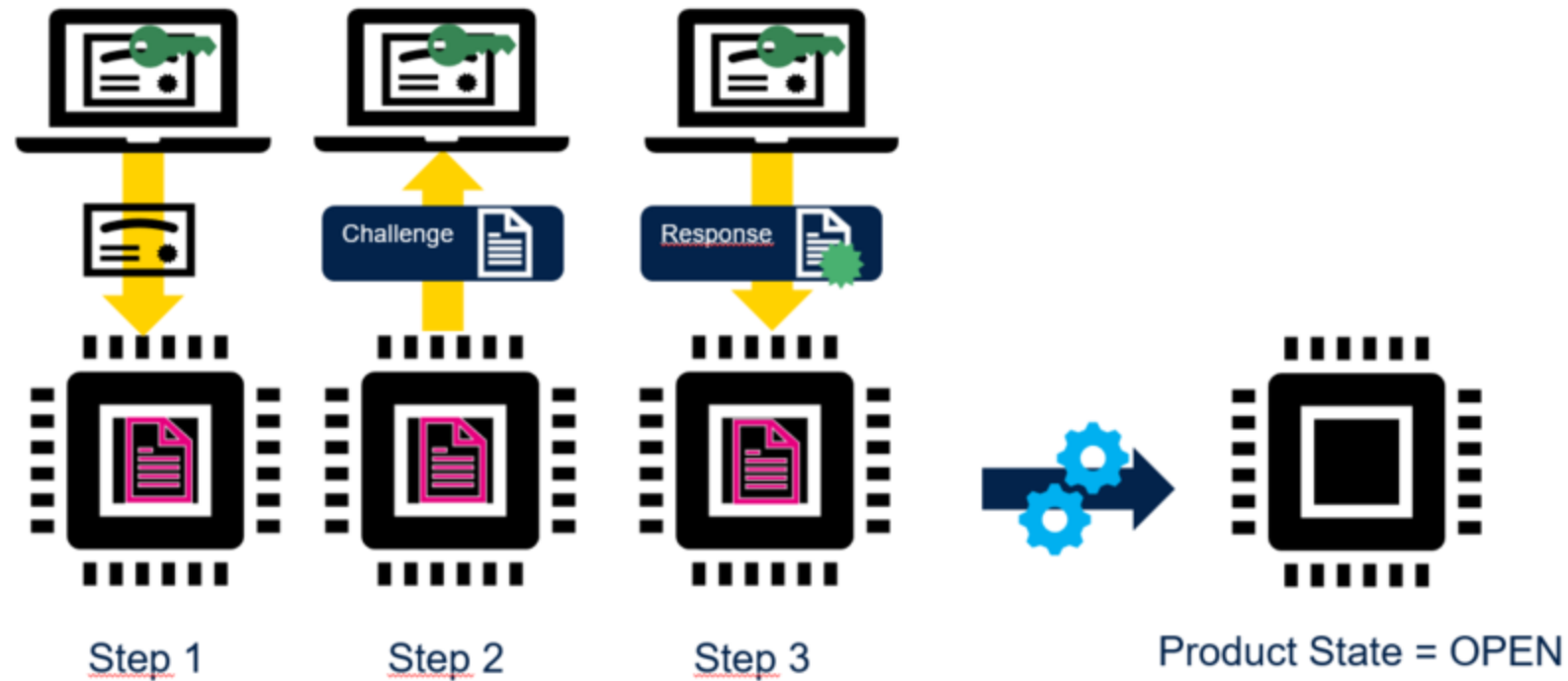
: Debug Authentication Certificate



Actions Request



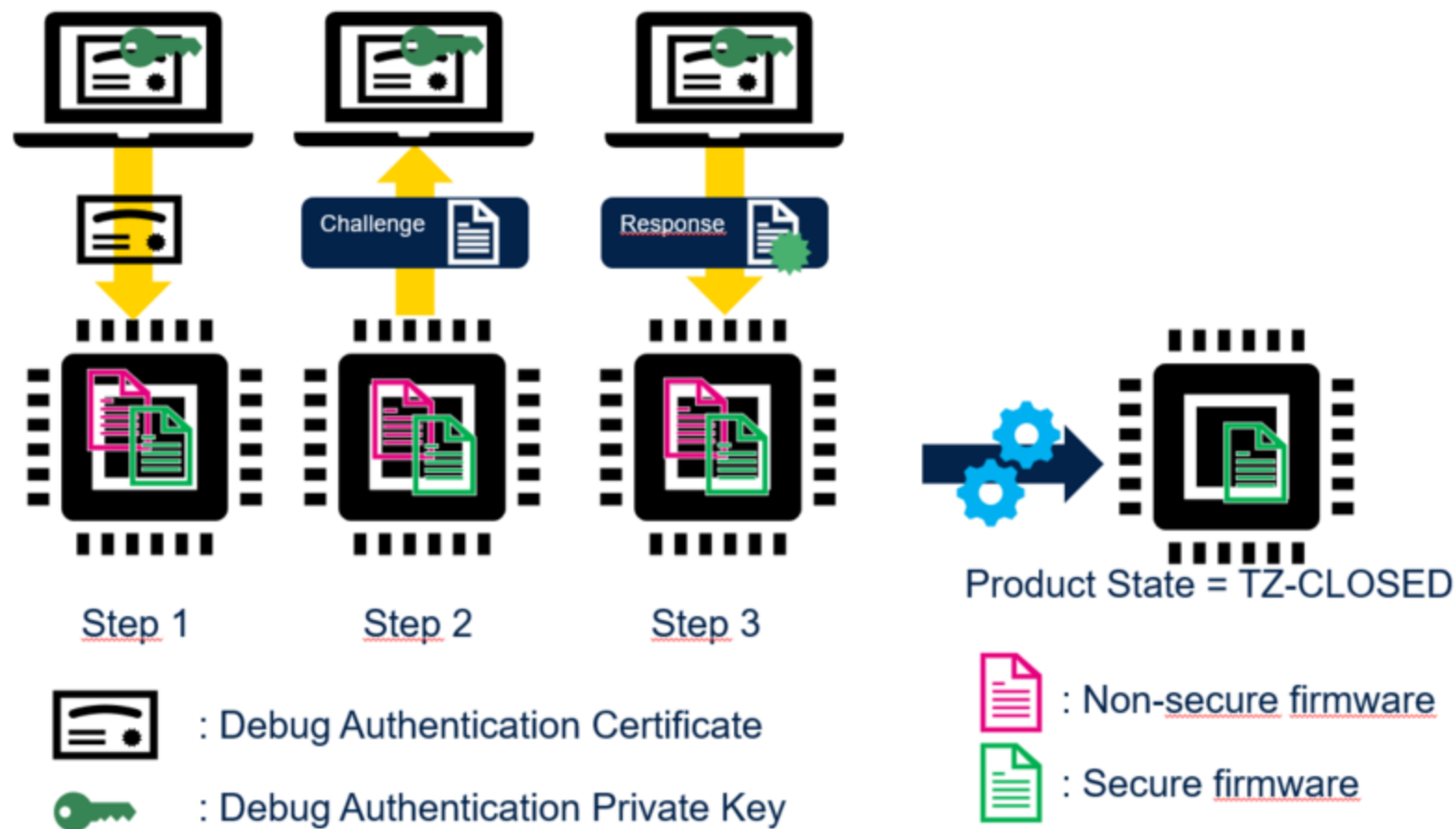
: Debug Authentication Certificate

Regression: Full

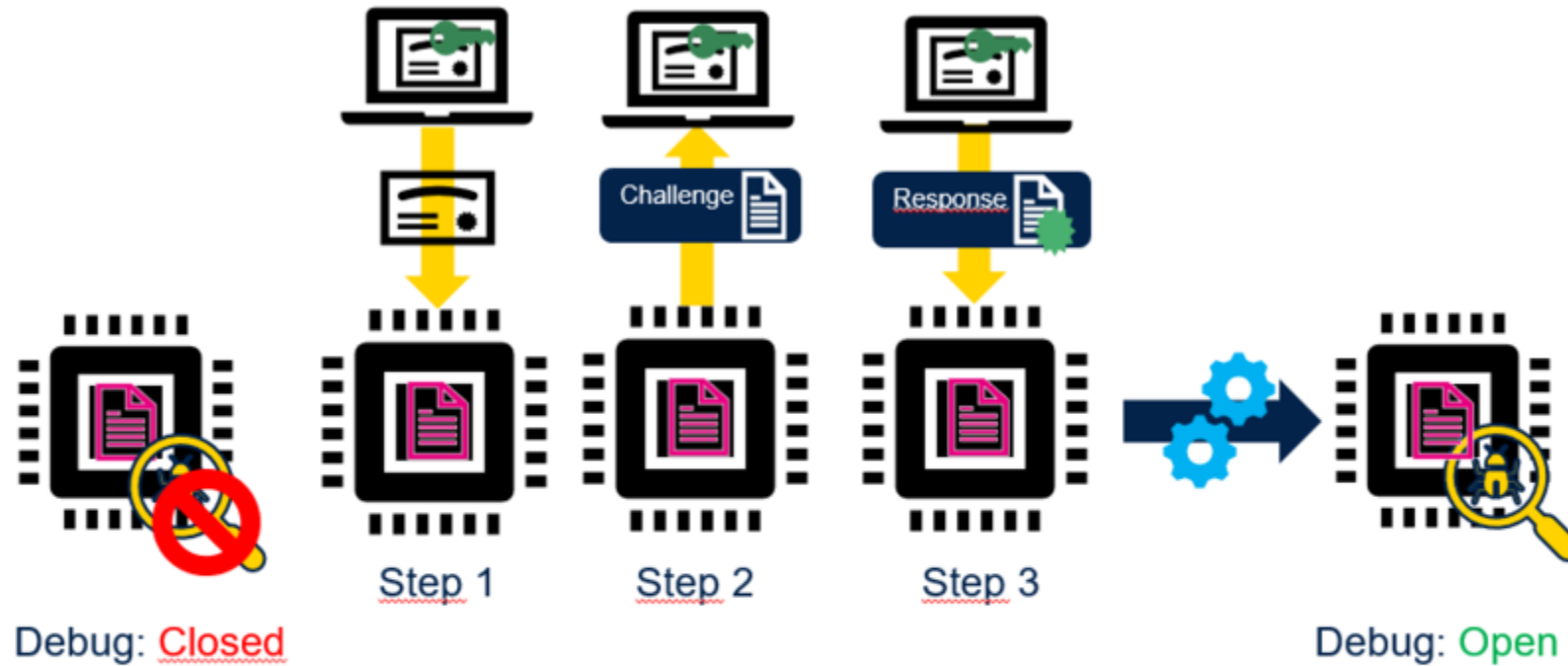


 : Debug Authentication Certificate
 : Debug Authentication Private Key

Regression: Partial



Debug Re-opening



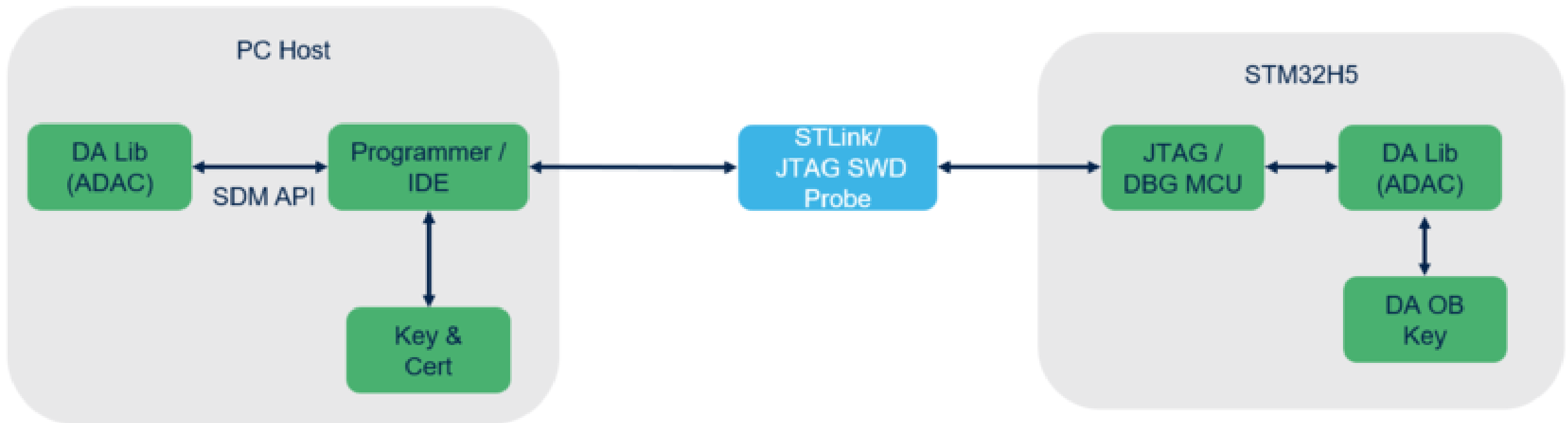
: Debug Authentication Certificate



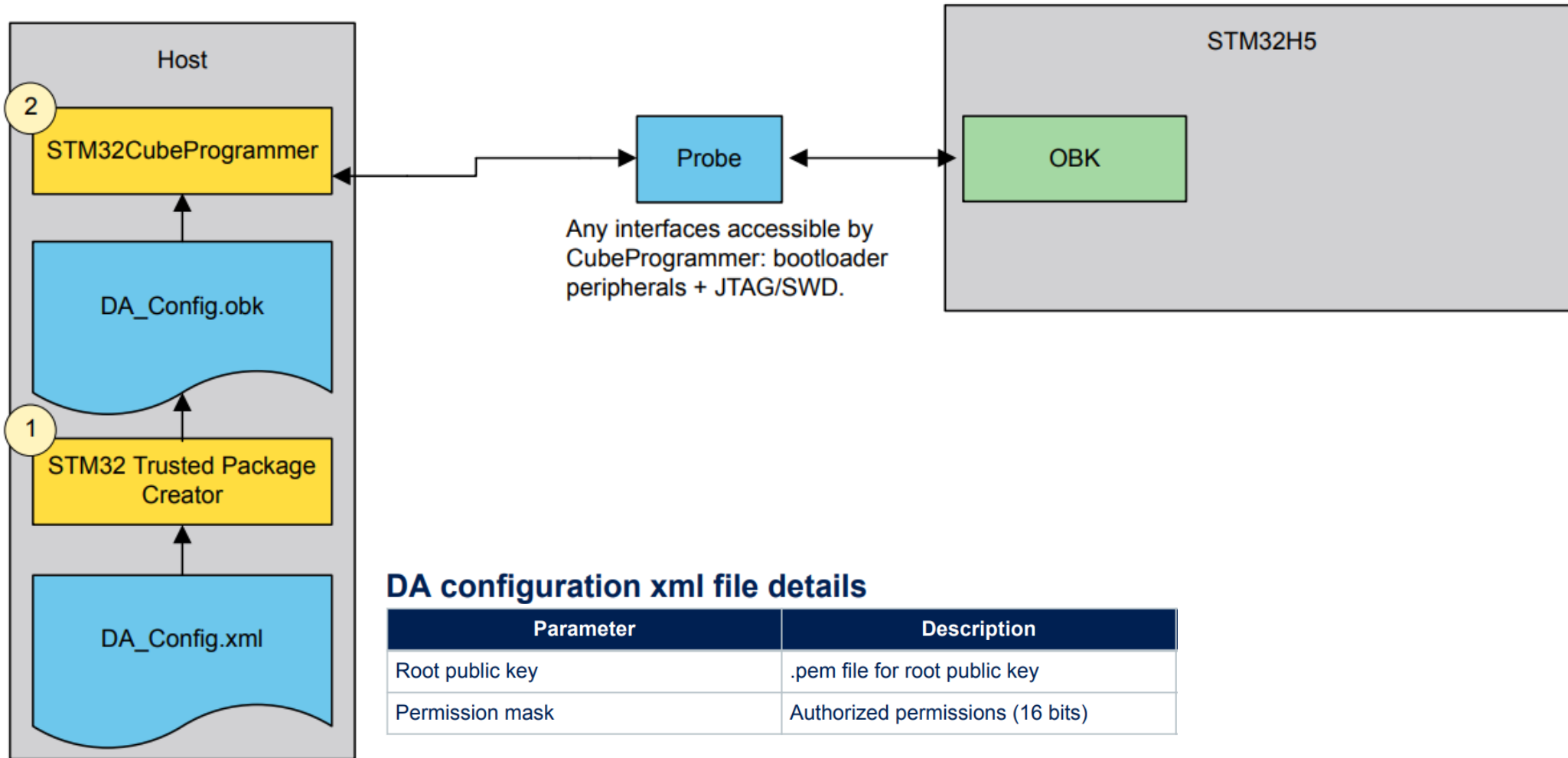
: Debug Authentication Private Key

Debug Authentication Ecosystem

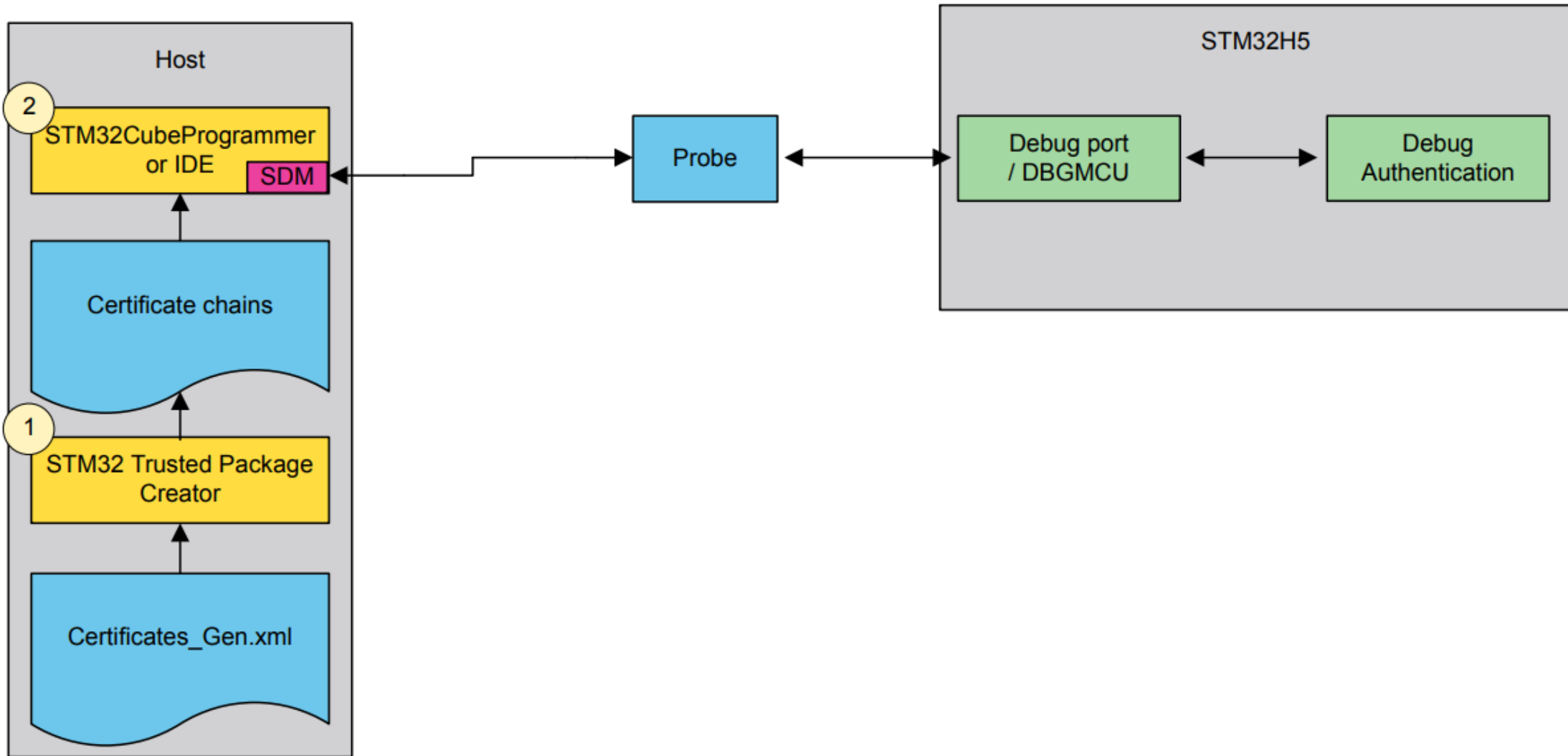
Debug Authentication Ecosystem



Ecosystem: Provisioning phase



Ecosystem: Launch DA (certificate method)



How To: Create new certificate

- Instructions available in section 6.2 of the following [Wiki page](#).

- Development is completed and device is released in closed state
- Our product is on the field, unfortunately some devices are not working fine...
- Customer returned the product to the support team

Let's see what the support team can do

Hands-on: Debug re-opening and regression

First possibility: re-open the debug

- What do we need ?
 - A certificate that allows reopening the device for debug and regression (leaf certificate)

C:\...\ROT_Provisioning\DA\Certificates\cert_leaf_chain.b64

- The associated private ECC key

C:\...\ROT_Provisioning\DA\Keys\key_3_leaf.pem

Let's reopen the device for debug !

Debug Reopening

Key File Path

C:\ST_SM_Workshop\STM32Cube_FW_H5_V1.2.0\Projects\STM32H573I-DK\ROT_Provisioning\DA\Keys\key_3_leaf.pem

Discover

Certificate File Path

C:\ST_SM_Workshop\STM32Cube_FW_H5_V1.2.0\Projects\STM32H573I-DK\ROT_Provisioning\DA\Certificates\cert_leaf_chain.b64

Continue

Permissions

Permission	Select
Non-Secure Intrusive Debug (Level1)	<input type="radio"/>
Non-Secure Intrusive Debug (Level2)	<input type="radio"/>
Non-Secure Intrusive Debug (Level3)	<input checked="" type="radio"/>
Secure Intrusive Debug (Level1)	<input type="radio"/>
Secure Intrusive Debug (Level2)	<input type="radio"/>
Secure Intrusive Debug (Level3)	<input type="radio"/>

Execute

Close Debug

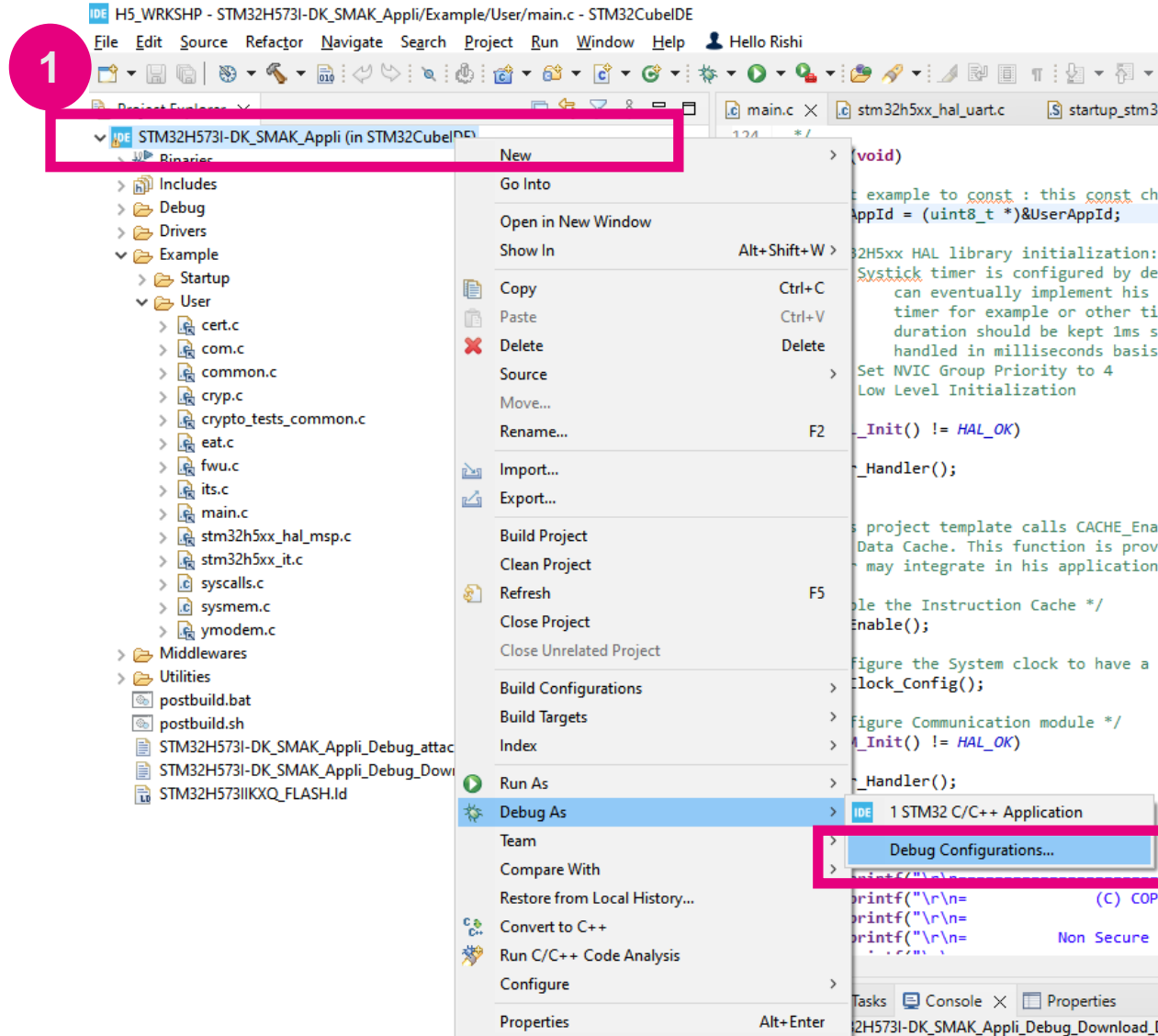
Used to lock device once the debug is opened through Debug Authentication. Applicable only when the feature is enabled.

Legend:

- Step 1: Path selection.
- Step 2: Permission selection.
- Step 3: Execution.

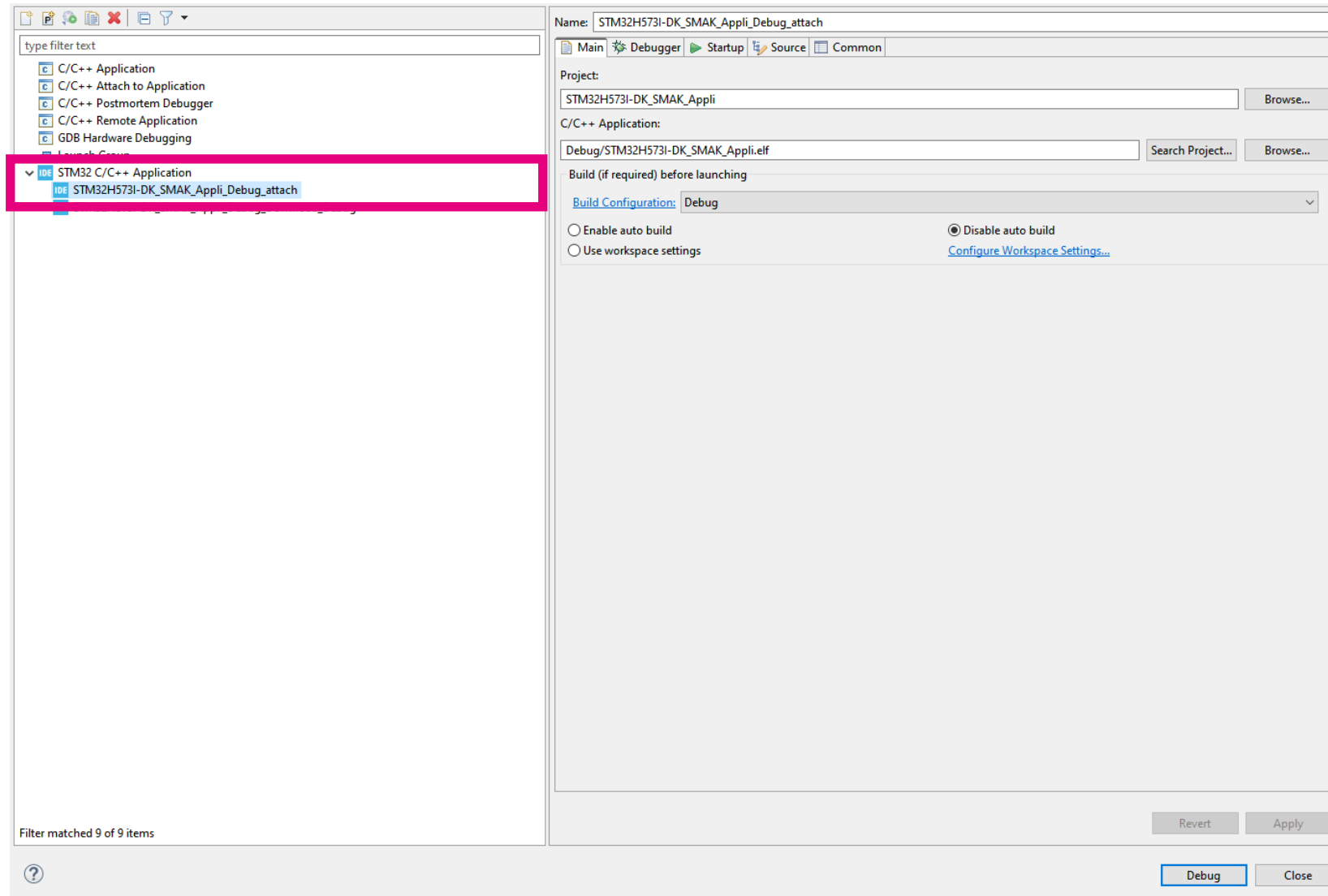
Modify the debug configuration

Right click

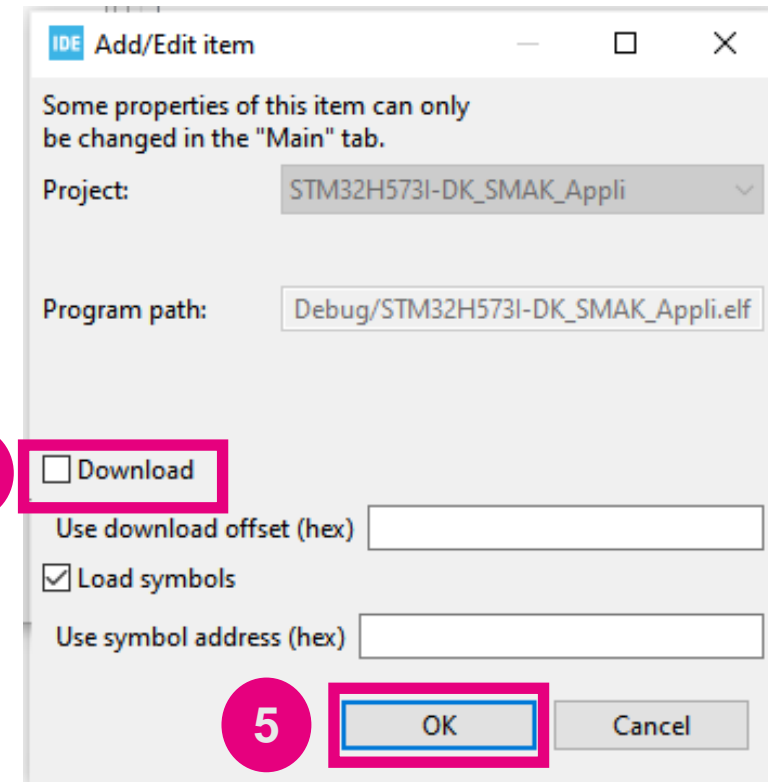
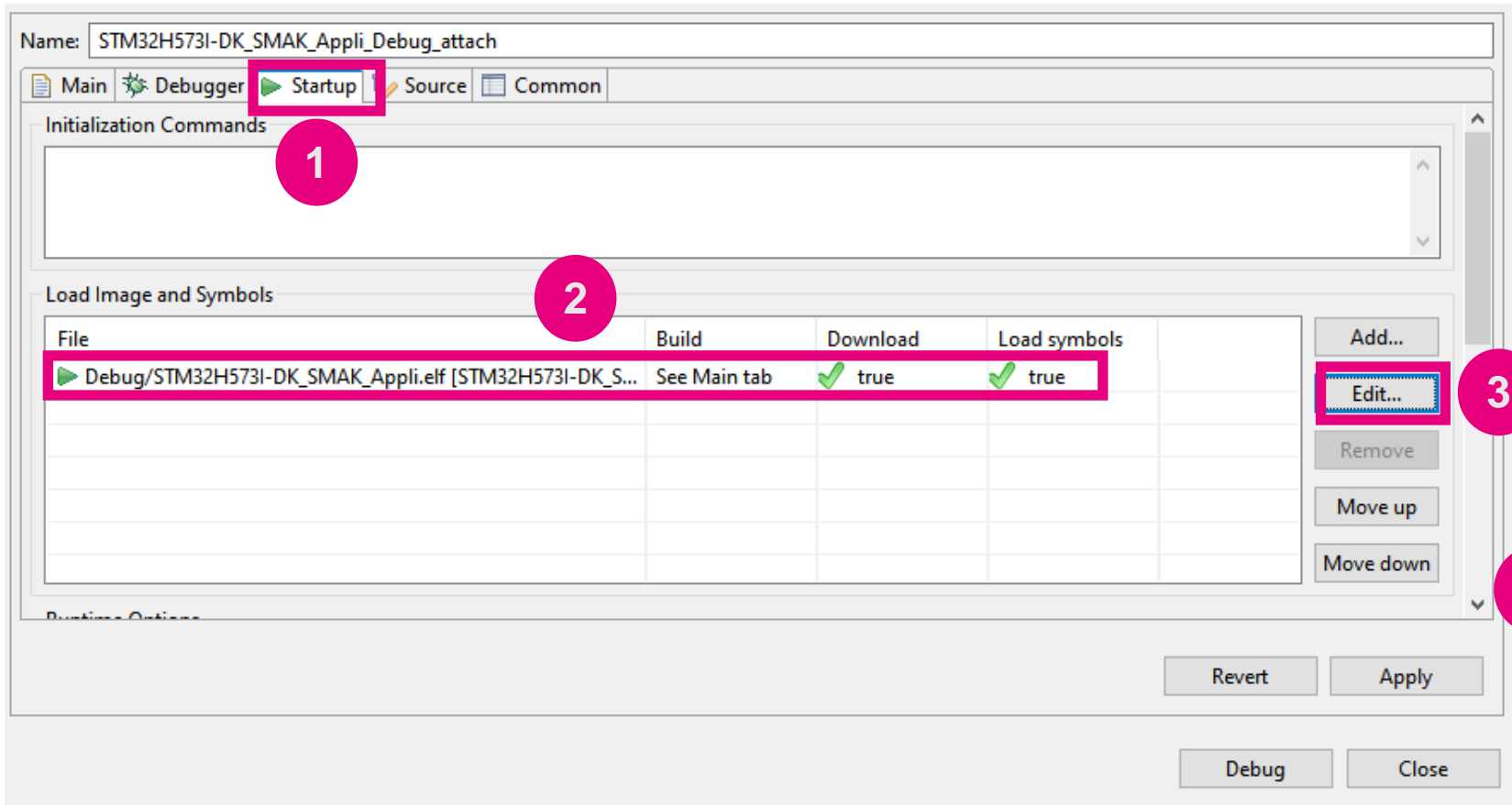


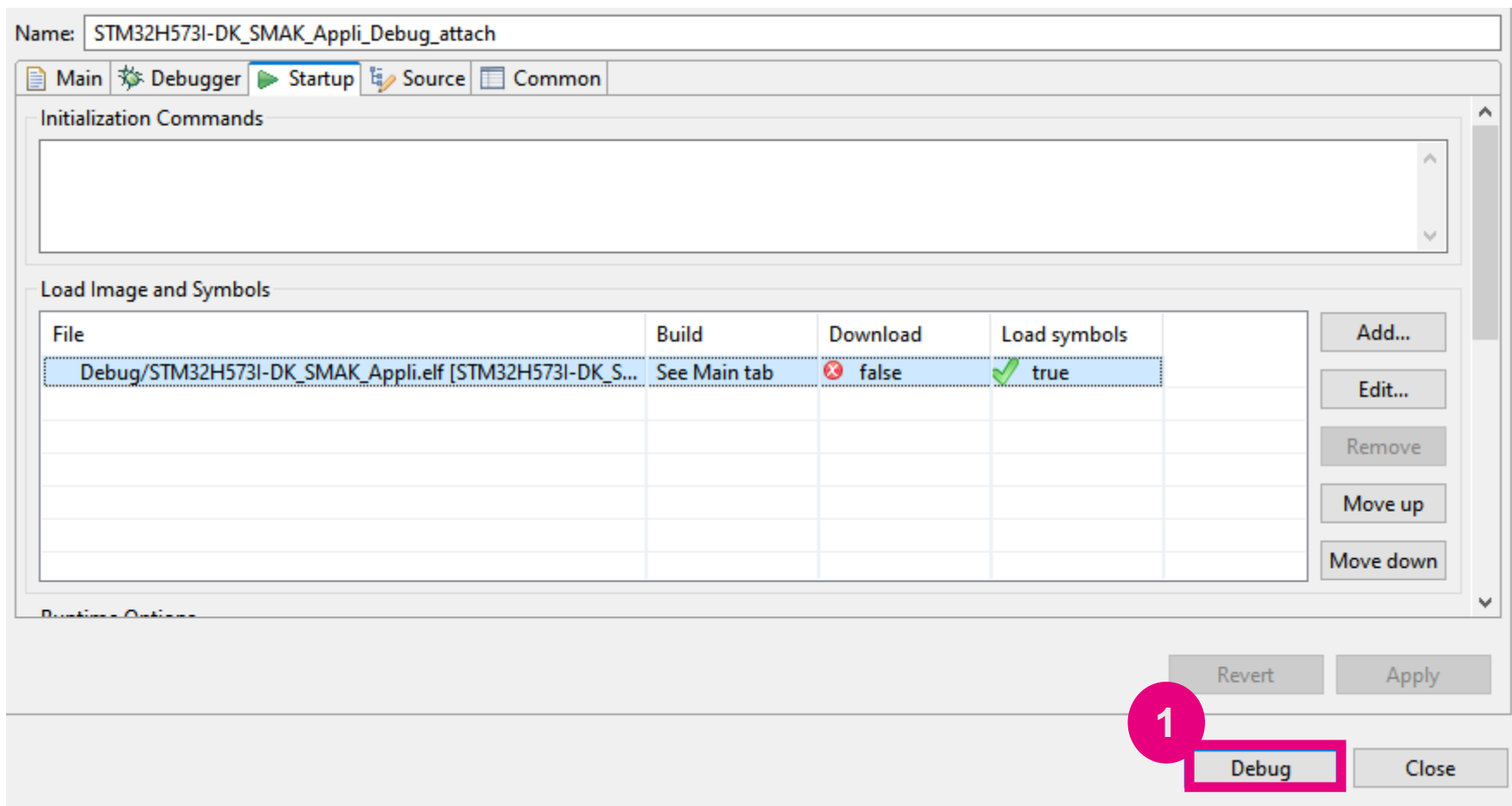
Debug Configuration

1



Remove 'download' option of the firmware





workspace_1.13.2 - TimeStamped_Event_Detection_NonSecure/Drivers/STM32H5xx_HAL_Driver/stm32h5xx_hal_uart.c - STM32CubeIDE

File Edit Source Refactor Search Project Run Window Help myST

Debug X Project Explorer

TimeStamped_Event_Detection_NonSecure [STM32 C/C++ Application]
 TimeStamped_Event_Detection_NonSecure.elf [cores: 1]
 Thread #1 [main] 1 [core: 1] (Running: User Request)
 arm-none-eabi-gdb (12.1.90.20220802)
 ST-LINK (ST-LINK GDB server)

main.c update_fw.c psa_fwu.c detection.c stm32h5xx_hal_u

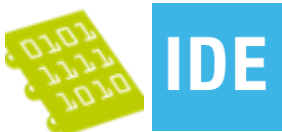
```

3455     if ( __HAL_UART_GET_FLAG(huart, UART_FLAG_RTOF) == SET)
3456     {
3457         /* Clear Receiver Timeout flag*/
3458         __HAL_UART_CLEAR_FLAG(huart, UART_CLEAR_RTOF);
3459
3460         /* Blocking error : transfer is aborted
3461          Set the UART state ready to be able to start again the process,
3462          Disable Rx Interrupts if ongoing */
3463         UART_EndRxTransfer(huart);
3464
3465         huart->ErrorCode = HAL_UART_ERROR_RTO;
3466
3467         /* Process Unlocked */
3468         __HAL_UNLOCK(huart);
3469
3470         return HAL_TIMEOUT;
3471     }
3472 }
3473 }
3474 }
3475 return HAL_OK;
3476 }
3477
3478 /**
3479  * @brief Start Receive operation in interrupt mode.
3480  * @note This function could be called by all HAL UART API providing reception in Interrupt
3481  * @note When calling this function, parameters validity is considered as already checked,
3482  * i.e. Rx State, buffer address, ...
3483  * UART Handle is assumed as locked
  
```

Console X Problems Executables Debugger Console Memory

TimeStamped_Event_Detection_NonSecure [STM32 C/C++ Application] [pid: 106]
 Listen Port Number : 61254
 Status Refresh Delay : 15s
 Verbose Mode : Disabled
 SWD Debug : Enabled

Waiting for debugger connection...
 Debugger connected
 Waiting for debugger connection...
 Debugger connected
 Waiting for debugger connection...



You can set breakpoint and debug...

IDE H5_WRKSH - STM32H573I-DK_SMAK_Appli/Example/User/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help Hello Rishi

Project Explorer X

STM32H573I-DK_SMAK_Appli (in STM32CubeIDE)

- Binaries
- Includes
- Debug
- Drivers
- Example
 - Startup
 - User
 - cert.c
 - com.c
 - common.c
 - cryp.c
 - crypto_tests_common.c
 - eat.c
 - fwu.c
 - its.c
 - main.c
 - stm32h5xx_hal_msp.c
 - stm32h5xx_it.c
 - syscalls.c
 - sysmem.c
 - ymodem.c
- Middlewares
- Utilities
 - postbuild.bat
 - postbuild.sh
 - STM32H573I-DK_SMAK_Appli_Debug_attach.launch
 - STM32H573I-DK_SMAK_Appli_Debug_Download_Debug.launch
 - STM32H573I-IKXQ_FLASH.ld

main.c X

```
102     return len;
103 }
104
105 #endif /* __GNUC__ */
106
107 /* Private variables -----*/
108 uint8_t *p_UserAppId;
109 const uint8_t UserAppId = 'A';
110
111 /* Private function prototypes -----*/
112 static void FW_APP_MAIN_PrintMenu(void);
113 static void FW_APP_MAIN_Run(void);
114 static void SystemClock_Config(void);
115 static void CACHE_Enable(void);
116 static void Error_Handler(void);
117
118 /* Private functions -----*/
119
120 /**
121  * @brief Main program
122  * @param None
123  * @retval None
124  */
125 int main(void)
126 {
127     /* set example to const : this const changes in binary without rebuild */
128     p_UserAppId = (uint8_t *)&UserAppId;
129
130     /* STM32H5xx HAL library initialization:
131      - SysTick timer is configured by default as source of time base, but user
132      can eventually implement his proper time base source (a general purpose
133      timer for example or other time source), keeping in mind that Time base
134      duration should be kept 1ms since PPP_TIMEOUT_VALUES are defined and
135      handled in milliseconds basis.
136      - Set NVIC Group Priority to 4
137      - Low Level Initialization
138     */
139     if (HAL_Init() != HAL_OK)
140     {
141         Error_Handler();
142     }
143 }
```

Let's close the debug

STM32CubeProgrammer

STM32CubeProgrammer

Data Information Notice

Secure programming

1

2

3

Debug Authentication

This interface supports only STM32H5!

Discover

name	value
Locking Mechanism	--
Soc ID	--
Life Cycle	--
Device ID	--

Close Debug

Used to lock device once the debug has been opened through Debug Authentication process. Applicable only when the feature is available.

Close Debug Procedure

Secure programming

RDP REG

SFI/SFIx

PROV

DA

SSP

Debug Authentication

Key File Path

Select File

Browse

Certificate File Path

Select File

Browse

Continue

This interface supports only STM32H5!

1

Close Debug

Used to lock device once the debug has been opened through Debug Authentication process. Applicable only when the feature is available.

2

OK

3

Discover

name	value
Locking Mechanism	Certificate
Soc ID	0x003D005D 0x33325117 0x38363236 0x000000...
Life Cycle	ST_LIFECYCLE_TZ_CLOSED (Debug opened)
Device ID	0x484

✓

Step 1: Path selection.

○

Step 2: Permission selection

○

Step 3: Execution.

Message

Target successfully locked.

Debug is now closed

RDP REG
SFI/SFIx
PROV
DA
SSP

Debug Authentication

Key File Path

Select File

Browse

Certificate File Path

Select File

Browse

Continue

i This interface supports only STM32H5!

Discover

name	value
Locking Mechanism	Certificate
Soc ID	0x003A001A 0x3332510F 0x38363236 0x000000...
Life Cycle	ST_LIFECYCLE_CLOSED
Device ID	0x484

Close Debug

Used to lock device once the debug has been opened through Debug Authentication process. Applicable only when the feature is available.

✓

Step 1: Path selection.

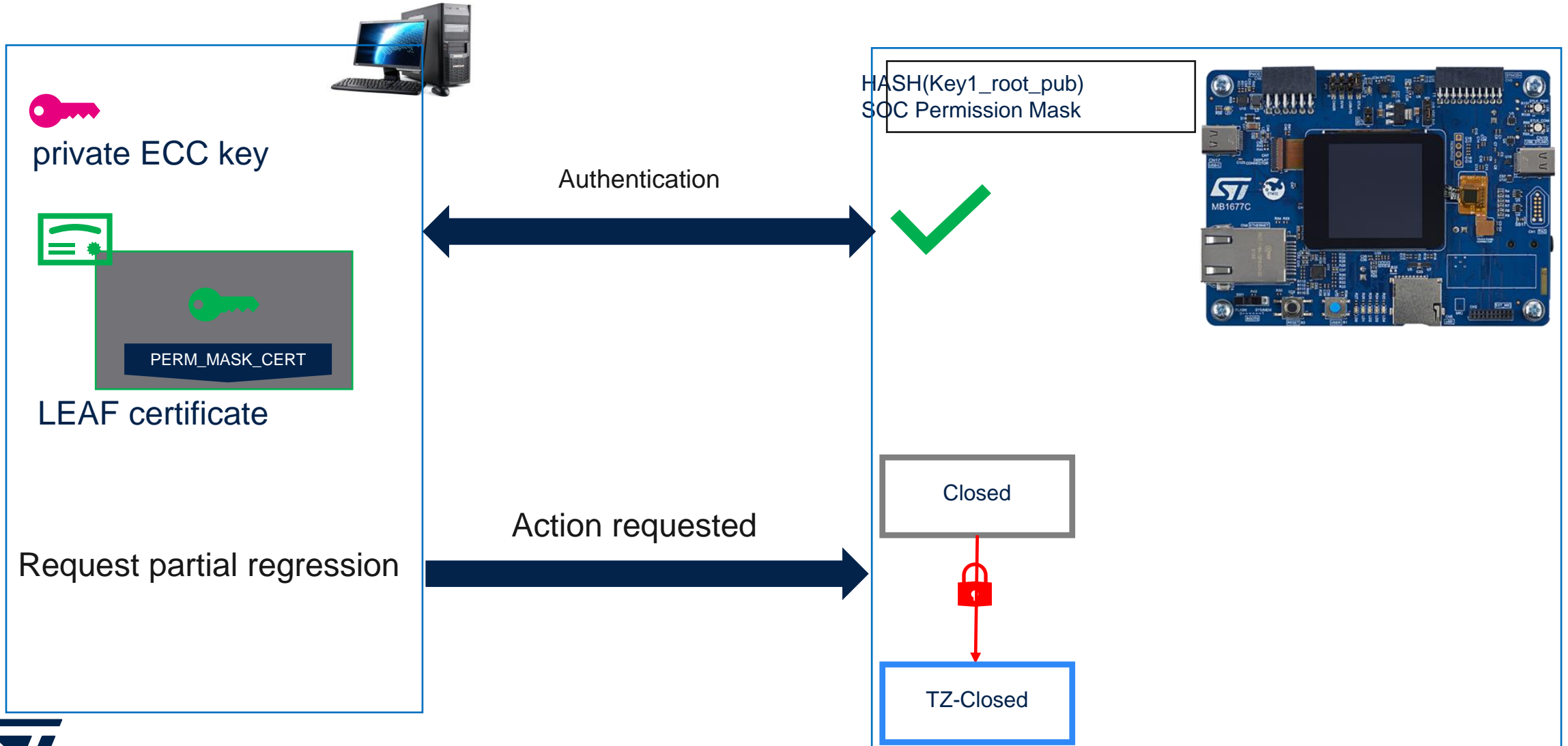
Step 2: Permission selection

Step 3: Execution.



STM32H5 Security

Debug Authentication partial regression



Partial Regression

STM32CubeProgrammer

STM32CubeProgrammer

Data Information Notice

Secure programming

1 [Icon]

2 [DA]

3 [Discover]

Debug Authentication

This interface supports only STM32H5!

name	value
Locking Mechanism	--
Soc ID	--
Life Cycle	--
Device ID	--

Close Debug

Used to lock device once the debug has been opened through Debug Authentication process. Applicable only when the feature is available.

Partial Regression: Keys and Certificates

Debug Authentication

Key File Path

1 **Key File Path**

Select File

Certificate File Path

2 **Certificate File Path**

Select File

3

name	value
Locking Mechanism	Certificate
Soc ID	0x004D0056 0x33325117 0x38363:
Life Cycle	ST_LIFECYCLE_CLOSED
Device ID	0x484

Used to lock device once the debug authentication is successful. This feature is available only when the feature is enabled in the device configuration.

This interface supports only STM32H5!

Legend:

- ☒ Step 1: Path selection.
- ☐ Step 2: Permission selection
- ☐ Step 3: Execution.

Partial Regression: Permission Selection

Permissions

Permission	Select
Non-Secure Intrusive Debug (Level3)	<input type="radio"/>
Secure Intrusive Debug (Level1)	<input type="radio"/>
Secure Intrusive Debug (Level2)	<input type="radio"/>
Secure Intrusive Debug (Level3)	<input type="radio"/>
Partial Regression	<input checked="" type="radio"/>
Full Regression	<input type="radio"/>

Device ID

0x484

Close Debug

Used to lock device once the debug is opened through Debug Authentication. Applicable only when the feature is enabled.

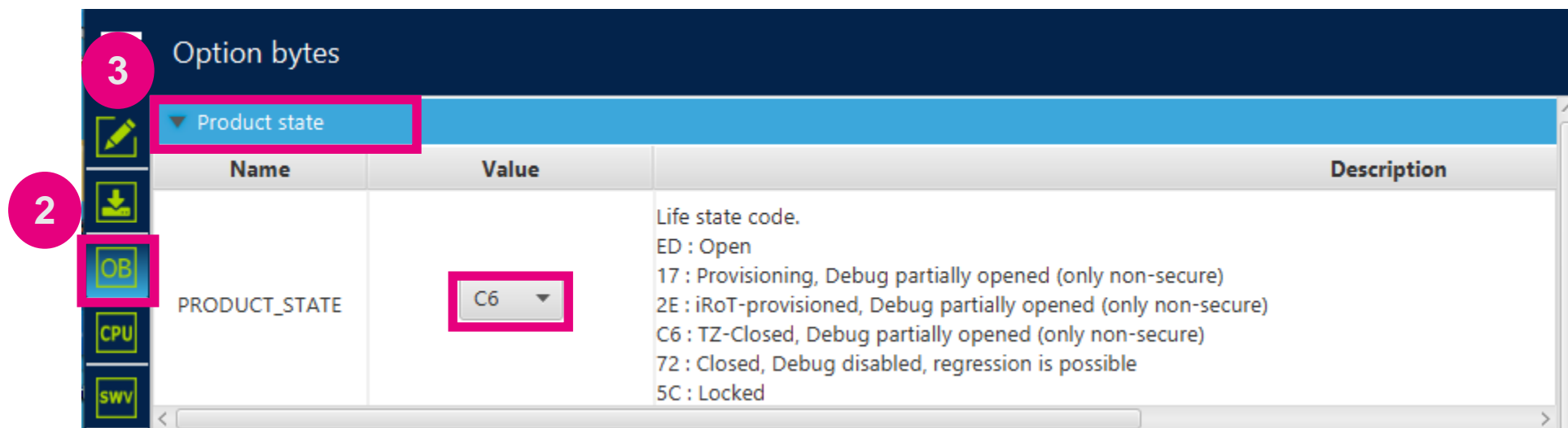
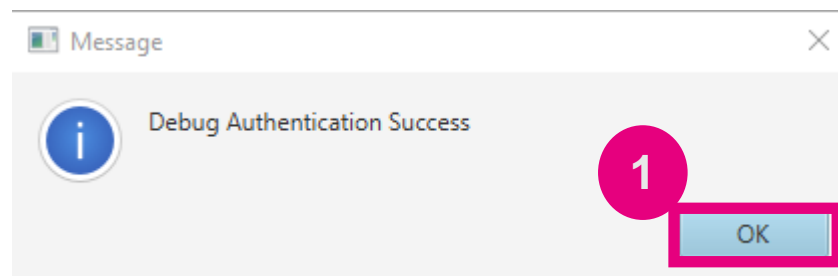
☐ Step 1: Path selection.

☒ Step 2: Permission selection

☐ Step 3: Execution.

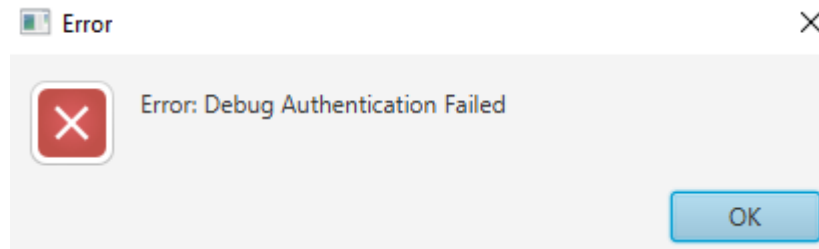
Execute

Partial Regression: Success

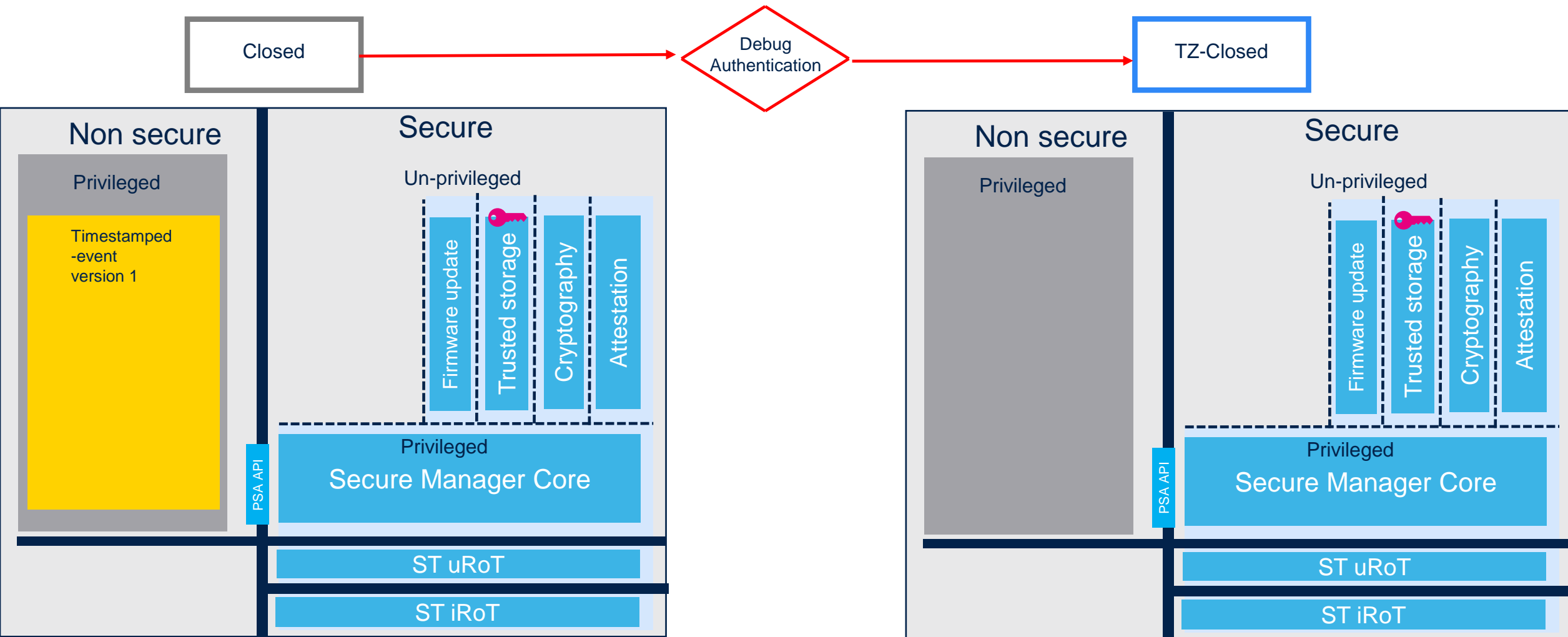




If you try any unauthorized action...



STM32H5 NS-Regression (partial) Result



STM32CubeProgrammer

STM32CubeProgrammer

Data Information Notice

Secure programming

RDP REG SFI/SFIx PA **2** DA SSP

1 Debug Authentication

3 Discover

This interface supports only STM32H5!

name	value
Locking Mechanism	--
Soc ID	--
Life Cycle	--
Device ID	--

Close Debug

Used to lock device once the debug has been opened through Debug Authentication process. Applicable only when the feature is available.

Full Regression: Keys and Certificates

Key File Path

1 **Key File Path**

Select File Browse

Certificate File Path

2 **Certificate File Path**

Select File Browse

3 **Continue**

Discover

name	value
Locking Mechanism	Certificate
Soc ID	0x004D0056 0x33325117 0x38363...
Life Cycle	ST_LIFECYCLE_CLOSED
Device ID	0x484

the debug Authenticat

Applicable only when the feature is

Step 1: Path selection.

Step 2: Permission selection

Step 3: Execution.

Full Regression: Permission Selection

RDP REG
SFI/SFIx
PROV
DA
SSP

Key File Path

Select File
C:\ST_SM_Workshop\STM32Cube_FW_H5_V1.1.0\Projects\STM32H573I-DK\ROT_Provisioning\I
Browse

Certificate File Path

Select File
C:\ST_SM_Workshop\STM32Cube_FW_H5_V1.1.0\Projects\STM32H573I-DK\ROT_Provisioning\I
Browse
Continue

Permissions

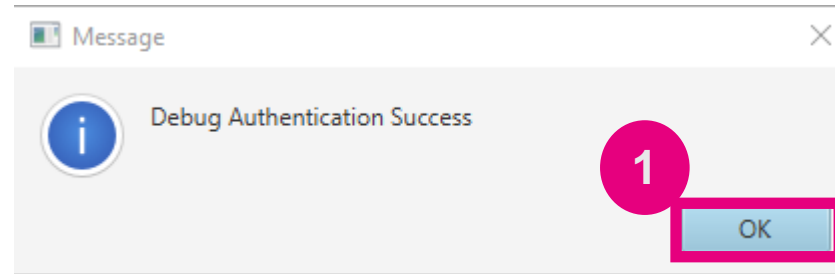
Permission	Select
Non-Secure Intrusive Debug (Level3)	<input type="radio"/>
Secure Intrusive Debug (Level1)	<input type="radio"/>
Secure Intrusive Debug (Level2)	<input type="radio"/>
Secure Intrusive Debug (Level3)	<input type="radio"/>
Partial Regression	<input type="radio"/>
Full Regression	<input checked="" type="radio"/>

1

2

Execute

Full Regression: Success



3

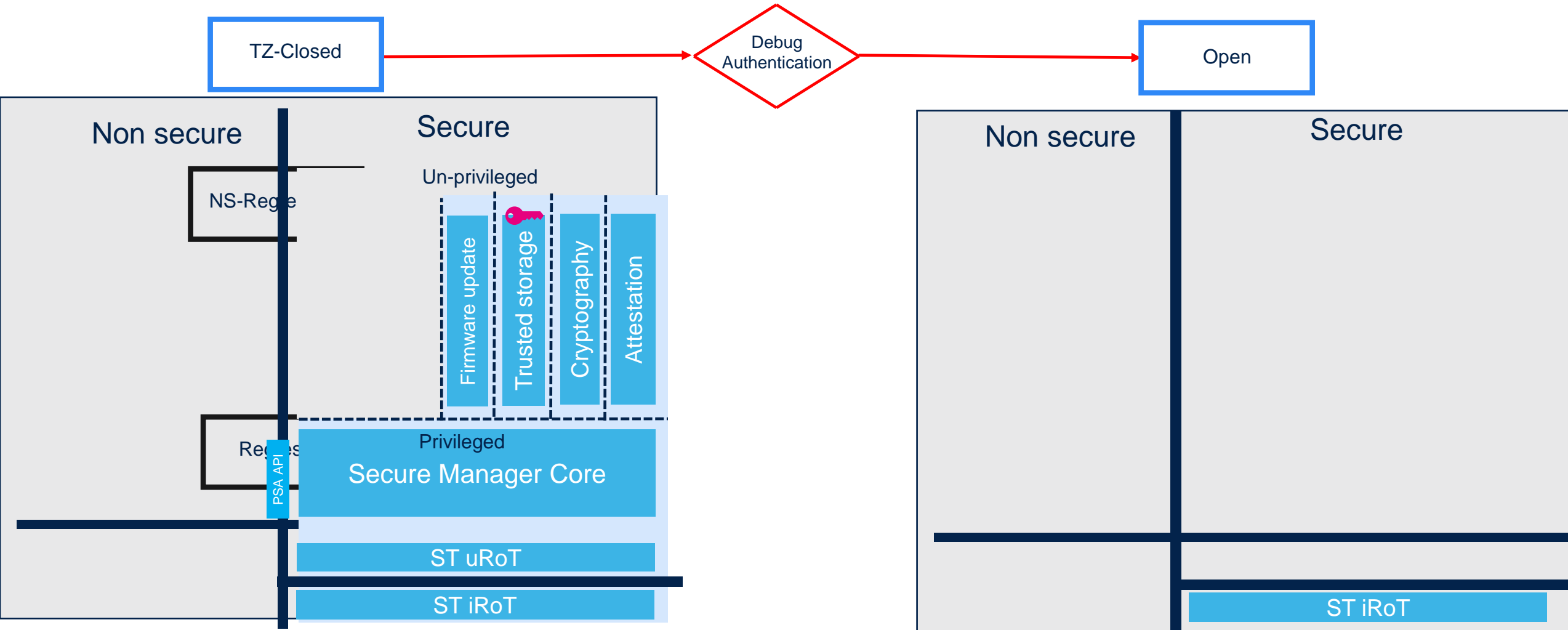
2

Option bytes

Product state

Name	Value	Description
PRODUCT_STATE	ED	Life state code. ED : Open 17 : Provisioning, Debug partially opened (only non-secure) 2E : iRoT-provisioned, Debug partially opened (only non-secure) C6 : TZ-Closed, Debug partially opened (only non-secure) 72 : Closed, Debug disabled, regression is possible 5C : Locked

STM32H5 Regression (Full) Result



- Debug Authentication is a new feature of STM32H5
- Certificates allow fine control on what user can do with the target
- This feature also simplifies the field return analysis

Resources

Links

- STM32Trust: [Web page](#)
- Security with STM32H5: [Wiki pages](#)
- Getting Started with STM32H5 security: [Wiki pages](#)
- Debug Authentication: How To Intro: [Wiki pages](#)
- STM32 Embedded Security Learning Journey: [Web page](#)

Videos

- STM32H5 Training: [Online Training](#)
- STM32 Security MOOC: [Online Course](#)
- Secure Manager MOOC: [Online Course](#)

Docs

- [AN5156](#) : Introduction to STM32 microcontrollers security
- [AN6007](#) : Getting Started with STiRoT for STM32H5 MCUs
- [AN6008](#) : Getting Started with Debug Authentication for STM32H5 MCUs
- [UM3254](#) : Secure manager for STM32H573xx microcontrollers
- [RM0481](#) : STM32H563/H573 Reference Manual

Agenda

1

Introduction

2

STM32H5 security features
overview

3

Hands-On: Getting started with
Secure Manager

4

Hands-On: SMAK
Develop and Debug

5

Hands-On: Debug Authentication

6

Conclusion & takeaways

Our technology starts with You



Find out more at www.st.com

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.



life.augmented