X-CUBE-SBSFU
Expansion software for STM32Cube
Secure boot & secure firmware update
In the News
Today’s Connected World

20 Billion

Operating system based solutions

65%
Our concern for tomorrow

48 Billion

65%

Embedded solutions
What will embedded solutions protect?

- **Their Services**
  What the end customer pays

- **Their Networks**
  Avoid DoS
  Provide quality/reliability

- **Their Brand**
  What we sell every day!
Categories of attacks

- **Logical attack**
  From outside the box

- **Board-level attack**
  From inside the box

- **Chip-level attack**
  From inside the chip
Secure Boot / Root of Trust Target

- Immutable Secure Boot code
- Executed first at reset
- Verify the platform integrity
  - Clock settings
  - Register configurations
  - Memory Protection, ….
- Launch Root of Trust services
  - Authentication of code
  - Uses cryptographic keys and encryption functions
Root of Trust General Process

- Performed at each RESET, using a Key stored in the device
- It is a predictable process
Current Market approach

• Not a single standardized Secure Boot / Root of Trust model
How to support such approach

• Embedded ROMed code

<table>
<thead>
<tr>
<th>SB / RoT approach</th>
<th>feasibility</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>One code on all STM32</td>
<td>😊</td>
<td>May not be market acceptable</td>
</tr>
<tr>
<td>Multiple code on STM32</td>
<td>😞</td>
<td>Diversify products Increase development, qualification, certification, cost</td>
</tr>
</tbody>
</table>

• ST proposal
  • Allow Industries to develop their own Secure Boot / Root of Trust method
  • Propose a way to securely load it into the STM32
  • Propose a way to isolate and securely execute it within STM32
Secure Firmware Update

- Server sends FW update
- Device receives, stores new FW update and executes it
Secure Firmware Update

- Complete process performed in a secure way
- Prevent Unauthorized updates
- Access to secret code and key
- Access to confidential on-device data
- Developed in several software modules
SBSFU software package overview

- **Secure Boot (SB) module**
  - Execution with Root of trust service
  - Application authentication and Integrity check before execution

- **Secure Firmware Update (SFU) module**
  - Detect new FW version to install
    - From local download service
    - Pre-downloaded OTA via User application from previous execution…
  - Manage FW version (check unauthorized updates or unauthorized installation)
  - Secure FW update:
    - FW Authentication and Integrity check
    - FW decryption
    - FW installation
  - In case of any error occurring during new image installation, possibility to rollback to the previous valid version…
  - Execute new installed FW (once Authenticated and integrity checked)

- **Secure Engine (SE) module**
  - Code isolated from main Firmware → Secure execution
  - Dedicated to executing cryptographic algorithms
  - Manage secure key storage
Secure Firmware Update Process

- Performed when a new firmware update is available by the using a public key stored in the device
- Each new firmware update is authenticated before being installed
• MCU Security Features
  • Used to establish a robust platform on which trusted processes and associated cryptographic functions can be performed

• Cryptographic Functions
  • Preserve confidentiality, verify integrity, authenticity

• Secure Boot (SB) and Secure Firmware Update (SFU)
  • Establishing a Root-of-Trust
  • Building a system that can evolve to counter new threats, add new functionality, fix bugs in a controlled and secure way once the device is in the field
STM32 Static Protections

**Readout Protection (RDP)**
- Level 0: no readout protection
- Level 1: memory readout protection
- Level 2: chip readout protection

**Proprietary code Read Out Protection (PcROP)**
- Specific configurable area
- 1 each per Flash memory bank

**Write protection (WRP)**
- 1 each per Flash memory / SRAM sector

Flash memory code and registers (+ SRAM2 in L4) cannot be dumped through JTAG/SWD or by the CPU itself booted from other memories than internal Flash.

Flash memory code is only executable, cannot be read and dumped by the CPU.

Flash memory code is protected from unwanted write/erase operations.
STM32 Dynamic Protections

- **Firewall**
  - Code or data protection in Flash memory or SRAM

- **MPU**
  - Memory isolation
  - Hard-fault or core lock-up in case of violation

- **Backup domain and Anti-Tamper**
  - Independent voltage
  - RTC, Backup SRAM
  - Tamper detection pin

- Single call-gate interface
- Trusted execution region
- Ideal to protect sensitive function and IP from the rest of the application
- Read, Write, eXecute attribute per region
- Prevent stack overflows
- System protection against unintended modification
- Detection of tamper event
- Reset of all backup register
- Time stamp event
SBSFU-based product architecture overview
Security Mindset / Security

• First, know what you want
  • What do you want to protect?
  • What do you want to protect it against?

• Then look at how to protect it

• Examine the protection we just added
  • Is it fully protecting?
  • Is it adding some weakness
  • Does it require additional elements to protect

Look at all the elements of the system
Incremental process
Protecting The Chain Of Trust

RDP–L2
- Disable external access
- Protect boot options
- Lock Option bytes
  - WRP
  - PCROP

WRP PCROP
- Protect the code enabling the MPU/Firewall
- Protect the code considered trusted
- Protect part of the Flash memory

MPU
- Execution allowed only inside the chain of trust

Firewall
- Protect RAM and Flash memory at runtime

Crypto
- Verify the integrity and authenticity of the user application

SB / SFU
SBSFU covered use cases

• **Industrial Firmware Update**
  - Usually performed by a human
  - Requires a physical connection between the updater tool and the MCU
    - UART, SPI, USB, ….. Wired connection
  - Able to stop the application running during the update
  - In case of update error, retry is manually managed

• **Over The Air Firmware Update (FOTA)**
  - Stand alone update operation
  - Uses device connectivity to receive and manage the update
    - Wi-Fi, LPWAN, BT/BLE, ….
  - Running application manages its own firmware update
  - Retry may be difficult to support
SBSFU : 2 implementations

• 1) Non isolated modules
  • Secure Boot and Secure Firmware Updates form a single immutable code protected the same way
  • It includes cryptographic and SFU keys
  • Introduce Root of Trust protection mechanism into STM32

• 2) Isolated modules
  • Secure Boot module is immutable code
  • Secure Engine is isolated from rest of code
  • Secure Firmware Update includes Root of Trust verification runtime code
## SBSFU roadmap on STM32

### X-CUBE-SBSFU

Expansion software for STM32Cube

<table>
<thead>
<tr>
<th></th>
<th>F4</th>
<th>F7</th>
<th>H7</th>
<th>L0</th>
<th>L4/L4+</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High-performance MCUs</strong></td>
<td><img src="image" alt="STM32 F4" /></td>
<td><img src="image" alt="STM32 F7" /></td>
<td><img src="image" alt="STM32 H7" /></td>
<td><img src="image" alt="STM32 L0" /></td>
<td><img src="image" alt="STM32 L4" /></td>
</tr>
<tr>
<td>Secure Boot</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Secure FW Update</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Secure Engine</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Ultra-low-power MCUs</strong></td>
<td><img src="image" alt="STM32 F4" /></td>
<td><img src="image" alt="STM32 F7" /></td>
<td><img src="image" alt="STM32 H7" /></td>
<td><img src="image" alt="STM32 L0" /></td>
<td><img src="image" alt="STM32 L4" /></td>
</tr>
</tbody>
</table>
SBSFU package now available for STM32L4 series

- X-CUBE-SBSFU expansion software available from ST website
  - www.st.com/x-cube-sbsfu
Recommendations

• Reduce risk
  • Ensures products are strengthened against attacks throughout their entire lifetime

• Understand the value of your assets
  • Perform threat analysis
  • Confidentiality, availability and integrity are key

• Use good design practices to develop and maintain secure products
  • Use security features and tools to achieve robust products
  • Work with trusted and experienced partners
Releasing Your Creativity

STM32 L4

www.st.com/stm32l4