Hello, and welcome to this presentation of the STM32WB Cortex M0+ security features.
The Cortex M0+ security manages the firmware and peripheral security, and is used to authenticate the ST radio firmware and allows the secure handling of cryptographic keys. The Cortex M0+ security uses secure options to control Flash memory, SRAM2 and Debug security. The AES encryption machine, Private Key Accelerator, and True Random Number Generator are peripherals whose security is managed dynamically by the secure Cortex-M0+ core through secure register bits in the System Configuration block.
The Cortex-M0+ security is based on giving exclusive access to a secure area in Flash memory and in SRAM2a and SRAM2b. Additionally, peripherals such as AES1, AES2, Private Key Accelerator and True Random Number Generator can be made secure, to allow secure cryptography and key generation. The secure memory areas and peripherals are not accessible by the Cortex-M4 and neither through the debugger.
The Cortex M0+ security is completely handled by the Cortex M0+ itself. At STM32WB production, the Cortex M0+ security is enabled after the Root Security Service (RSS) firmware has been programmed into the User Flash memory. Any subsequent Cortex M0+ firmware update (Connectivity stack, or RSS) is handled by the RSS and modifies the Cortex M0+ security parameters as needed.

The AES2, PKA, and RNG security is fully handled by the Cortex M0+ whenever needed by the Cortex M0+ firmware. The AES1 key security is also managed by the Cortex M0+ when requested by the Cortex M4 application firmware.
Secure Options Registers

- Cortex-M0+ security is configured by secure User Options.

<table>
<thead>
<tr>
<th>Register</th>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTR (*)</td>
<td>User Options</td>
</tr>
<tr>
<td>SFR (*)</td>
<td>res.</td>
</tr>
<tr>
<td>SRRVR (*)</td>
<td>C2OPT</td>
</tr>
</tbody>
</table>

*OPTR: Options Register
*SFR: Secure Flash Register
*SRRVR: Secure Ram and Reset Vector Register

- When the Cortex-M0+ security is enabled, the secure User Options are exclusively writable by the Cortex-M0+.
  - The non-secure Cortex-M4 can read the secure User Options for example to determine the size of the secure areas.

The Cortex-M0+ security is controlled through secure user options loaded at device startup in the Secure Flash Register and Secure RAM and Reset Vector Register. The secure user options can only be modified by the secure Cortex-M0+, i.e. to change parameters when a secure Cortex-M0+ software is updated. The non-secure Cortex-M4 has read access to the secure user options to be able to determine the start of the secure areas.
Memory security is enabled and configured by secure user options.
The Flash Security Disable bit enables the Global Cortex-M0+ security.
The Secure Flash Start Address defines the start address from which the Flash memory is secure.
The Backup RAM Security Disable bit controls the security on the backup RAM, and the Secure Backup RAM Start Address defines the start address from which the backup RAM is secure.
The Non Backup RAM Security Disable bit is used to enable security on the non-backup RAM, and the Secure Non Backup RAM Start Address defines the start address from which the non-backup RAM is secure.
The Debug access to the secure areas is controlled by the Debug Disable Security bit.
The Enable Security Environment bit is a read-only bit as
the security is always enabled on the Cortex-M0+ core.
The top of the memories can be secured for exclusive Cortex-M0+ access.
The top of the Flash memory, starting from the Secure Flash Start Address, is secure when the Flash Security Disable bit (FSD) is set to “0”.
The top of the backup SRAM2a, starting from the Secure Backup RAM Start Address (SBRSA), is secure when both the Flash Security Disable and Backup RAM Security Disable (BBRSD) bits are set to “0”.
The top of the non-backup SRAM2b, starting from the Secure Non-Backup RAM Start Address (SNBRSA), is secure when both the Flash Security Disable and Non-Backup RAM Security Disable (NBRSD) bits are set to “0”.
It is possible to only secure the Flash memory without any RAM security; however, it is recommended to secure both the Flash memory and RAM used by the Cortex-
M0+ software.
The Cortex-M0+ boot reset vector is programmed in the Secure Boot Reset Vector (SBRV) option.
- Word-aligned value.

- The Cortex-M0+ may boot from Flash memory or SRAM2 (a or b) as selected by the Secure CPU2 option (C2OPT).

- At production, the Cortex-M0+ boot reset vector is set in Flash memory at the RSS boot reset vector.

The Cortex-M0+ boot reset vector is to be programmed in the secure boot reset vector option and secure CPU2 option. At production, the Cortex-M0+ boot reset vector points to the Root Secure Service start address in Flash memory. In Secure mode, the Cortex-M0+ boot reset vector can only be changed by the secure Cortex-M0+ side.
Cortex-M0+ debug access is controlled by the Debug Disable Option bit. It is independent from security and can be enabled and disabled in both Secure and Non-secure modes. In Secure mode, debug access control can only be changed by the secure Cortex-M0+ side.
The STM32WB has a single Flash memory for both the Cortex-M4 and Cortex-M0+ software. The Cortex-M0+ security prevents secure Flash memory pages from being erased by the non-secure Cortex-M4. A Cortex-M4 Flash Mass Erase operation will be rejected, and a Multiple Block Erase has to be used to erase the Cortex-M4 software.

When regressing the ReaD Protection from Level 1 to Level 0, only the non-secure part of the Flash memory will be erased. The secure Cortex-M0+ software will be retained.

The complete Flash memory is mass erased and the security is removed only when regressing the ReaD Protection from Level 1 to Level 0. In this case, the ST radio stack authentication and security is lost and can no longer be programmed.
Secure System Configuration bits

- Secure System Configuration bits are used to handle peripheral security
- Peripheral security is only available when Security is enabled in (FSD)
  - AES1 key security, enabled by SAES1
  - A secure application key storage feature is provided by the Cortex M0 software.
  - AES2 full security, enabled by SAES2
  - PKA full security, enabled by SPKA
  - True RNG full security, enabled by SRNG
- Peripheral security
  - can be managed dynamically.
  - is enabled/disabled by the secure Cortex-M0+.
  - security enable bits can be read by the Cortex-M4.

The AES accelerator 1, AES accelerator 2, Public Key Accelerator and True Random Number Generator peripherals can dynamically be made secure by Cortex-M0+ firmware through secure register bits in the System Configuration block. The AES 2, Public Key Accelerator and True Random Number Generator peripherals provide full peripheral security. The AES 1 provides only key security, which allows the application running on the Cortex-M4 to use cryptography with a secure key. Secure key storage is provided by the Cortex-M0+ firmware. The Cortex-M4 may read the peripheral security bit to determine its security status.
The radio stack running on the Cortex-M0+ provides cryptographic key management to the Application. The cryptographic keys are generated and stored on the secure Cortex-M0+ side using the Cryptographic Key Storage (CKS).
The STM32WB includes a preprogrammed RSS which allows the secure Cortex-M0+ software to be updated. Both the Radio stack software and the RSS itself can be updated. Secure software can be downloaded via In-Circuit Programming by the system bootloader or via In-Application Programming by an application bootloader including Over The Air (OTA). Secure Cortex-M0+ software update is possible in all ReaD Protection levels (0, 1, and 2).
This slide lists the events generated by the Cortex-M0+ security feature. Events are only generated to the non-secure Cortex-M4. Depending on the Cortex-M4 access type, a bus error is generated to the non-secure Cortex-M4. Reading secure areas returns zeros. Only the secure user options and system configuration peripheral security enable bits can be read by the non-secure Cortex-M4.

<table>
<thead>
<tr>
<th>Cortex-M4 action:</th>
<th>Generated event:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex-M4 write access to secure peripheral registers</td>
<td>Bus error</td>
</tr>
<tr>
<td>Cortex-M4 requesting a mass erase of secure flash</td>
<td>Bus error</td>
</tr>
<tr>
<td>Cortex-M4 requesting a page erase of a secure flash page</td>
<td>Bus error</td>
</tr>
<tr>
<td>Cortex-M4 requesting a secure flash page</td>
<td>Bus error</td>
</tr>
<tr>
<td>Cortex-M4 data write operation to secure flash</td>
<td>Bus error</td>
</tr>
<tr>
<td>Cortex-M4 read access to secure flash memory area</td>
<td>Bus error + read zero value</td>
</tr>
<tr>
<td>Cortex-M4 read access to secure RAM memory area</td>
<td>Read zero value</td>
</tr>
<tr>
<td>Cortex-M4 read access to secure peripheral registers</td>
<td>Read zero value</td>
</tr>
</tbody>
</table>

* When RDP level is 1 and booting from SRAM1 no bus error is generated. (SRAM2 is looked)
In addition to this training, you may find the flash memory interface and system configuration modules useful.

- Refer to these trainings linked to this feature:
  - STM32WB Power control (Flash memory interface)
    - Secure user options
  - STM32WB System configuration (SYSCFG)
    - Secure peripheral enable bits
  - BLE stack
    - Secure Cortex-M0+ BLE stack