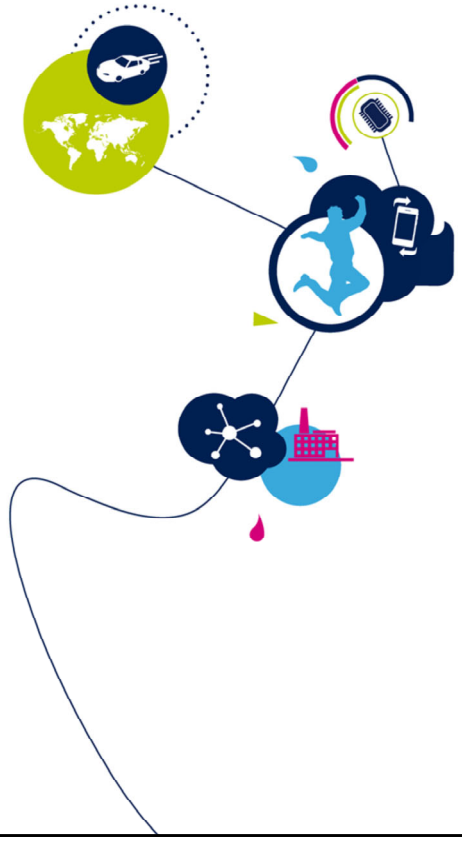
STM32WB - RSS

Root Security Services

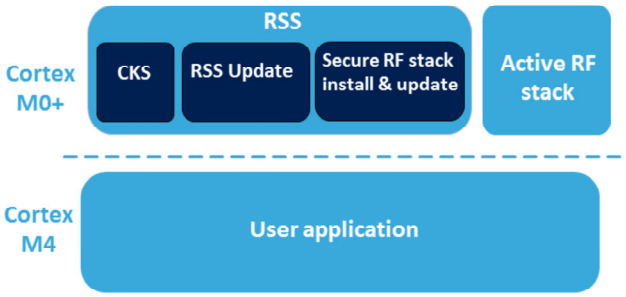
Revision 1.0



life.augmented

Hello, and welcome to this presentation of the STM32WB Root Security Services features.

Overview



- RSS services
  - Secure install and update of wireless stack
    - Dual ST & Customer authentication
  - Secure RSS update
  - Cryptographic Key Storage
    - Up to 100 AES keys stored in secure memory
    - Prevent direct key manipulation by user firmware

Application benefits

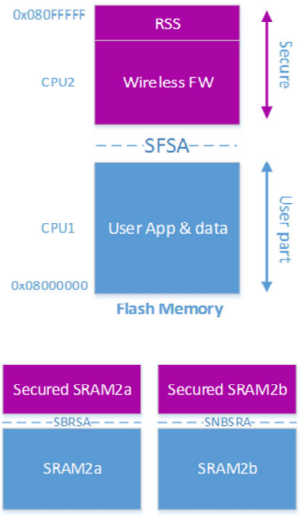
- Capability to keep best wireless performance and updated security level during device lifetime
- Storage of application keys on the secure CM0+ side.

life.augmented

The Root Security Services (RSS) is a protected firmware executed by the Cortex M0+ core. It is used to securely install and update the wireless stack using cryptographic mechanisms to ensure its integrity and authentication. The update of the wireless stack is key to guaranteeing the best performance and security level throughout the device's lifetime.
Authentication of the wireless firmware is granted by STMicroelectronics by default. However, a double authentication can be added by the customer.
The RSS firmware itself can be updated.
The RSS also provides a secure slot for storing cryptographic keys. This way, secret keys can be provisioned and used by user applications without direct access to their value.

Cortex M0+ code protection

- RSS and active wireless stack are part of the secure Flash memory
  - Their code and data cannot be accessed by Cortex M4 (CPU1) or debug port
  - Execution data are stored in protected part of SRAM2
  - They are executed by cortex M0+ (CPU2) only
- RSS and wireless stack are exclusive
  - RSS is active by default when no wireless stack is present
  - Else the wireless stack is under execution
  - CPU2 switches on RSS when requested either
    - By a bootloader command
    - By user application through IPCC message (mailbox)

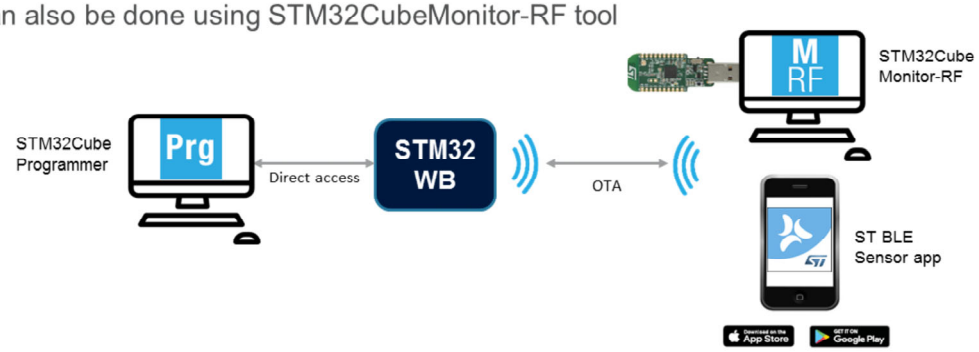


life.augmented

The RSS firmware, as well as the wireless stack, are sensitive code with embedded cryptographic data. Therefore it is stored in the secure part of the Flash memory accessible only by the Cortex M0+ core. Code and data stored in this memory part cannot be accessed by any application running on the Cortex M4 core. They cannot be accessed by the debug port either. Volatile data, needed for execution, are secured too in the secure part of SRAM2 memory.
The RSS firmware and wireless stack are exclusive; the Cortex M0+ core can execute either one or the other. By default, when no wireless stack is installed, the RSS is active. Then, after a first install, the wireless stack becomes active. New access to the RSS is granted either by a bootloader command or by an application message through the inter-processor communication controller (IPCC).

Direct access and OTA

- Wireless stack can be updated in two ways
  - Direct access
    - Saves user memory space: current stack removal before new stack install
    - Direct access is granted by bootloader and STM32CubeProgrammer tool
  - Over-the-Air (OTA)
    - Allows wireless update when no physical access to the device is possible
    - Allows usage of mobile application as update client: ST BLE Sensor application
    - Can also be done using STM32CubeMonitor-RF tool

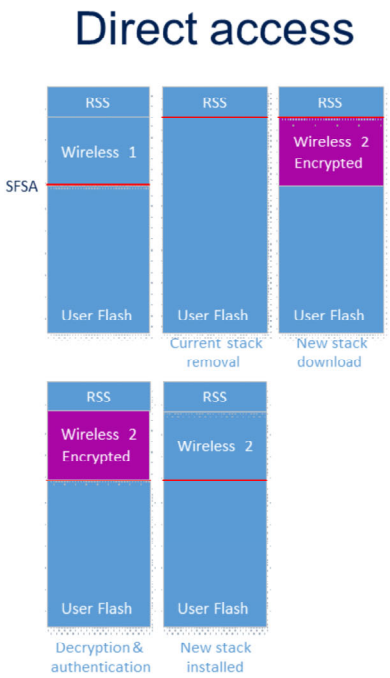


The wireless stack can be updated in one of two ways, depending on the availability of a physical (wired) link on the device.
A direct access through a physical link for using the bootloader allows you to perform the first install of the wireless stack.
Afterwards, the stack can be updated by first removing the current stack and hence reducing the amount of memory needed for the operation. There is only enough space for one stack in the Flash memory.
The STM32CubeProgrammer tool can be used to directly access the bootloader and dedicated commands.
The second way for updating the wireless stack is to do it over-the-air (OTA). This requires the use of the current wireless stack for downloading the new one. It means that an extra Flash memory area must be available for the update. The OTA application example is available in

the STM32Cube Firmware package. New firmware can be downloaded either using the ST BLE sensor mobile application or the STM32CubeMonitor-RF tool and a BLE dongle.

Wireless stack update

- Update procedure
  1. Current stack is removed by RSS command
     - Space of current stack is freed for next download
     - Wireless connection is no longer possible
  2. Encrypted stack is downloaded in available user area
     - By CPU1 application, SWD/JTAG or bootloader
  3. Protection is extended to wireless stack
  4. Encrypted stack is decrypted and authenticated by RSS
  5. Security option bytes are set by RSS according to new stack header
  6. CPU2 ready to execute new wireless stack

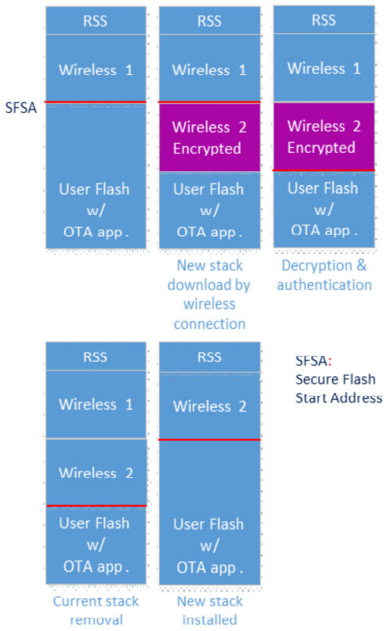


This slide presents the Direct Access procedure for updating the wireless stack.
The procedure is automatically performed by the RSS firmware once the update command has been sent by the bootloader.
First, the current wireless stack is removed. Then the encrypted and signed new wireless stack is downloaded in the Flash memory.
Sensitive operations of decryption, integrity and authentication checks are done by RSS firmware inside the protected Flash memory area.
In the last step of the procedure, option bytes related to the wireless execution configuration are set with the proper values.
At the end of the procedure, the wireless stack becomes active.

Wireless stack update

Over-the-air (OTA)

- Update procedure
  1. Download of new wireless stack requires
     - A user application with OTA service
     - Current stack active (wireless 1)
     - Enough user memory available to store new stack (wireless 2)
  2. Protection is extended to new stack
  3. Encrypted stack is decrypted and authenticated by RSS
  4. After authentication and integrity checks, current stack is removed and replaced by the new one
  5. Security option bytes are set by RSS according to new stack header
  6. CPU2 ready to execute new wireless stack

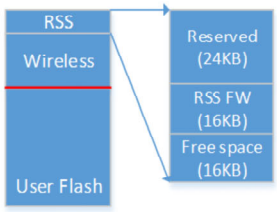


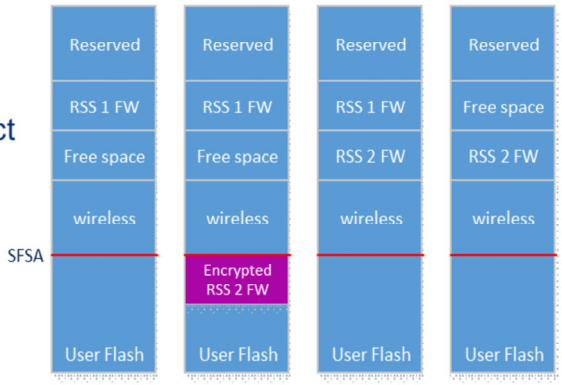
This slide presents the over-the-air procedure for updating the wireless stack.
This procedure requires a wireless application from the user side able to provide a download service to a remote client.
On request from this client, the user application downloads the new wireless stack to the device's Flash memory. Then, it sends a message to the RSS to start the update procedure.
The encrypted and signed new wireless stack is downloaded in the Flash memory before being decrypted and authenticated.
After all checks have been passed, the current stack is removed and replaced by the new one.
In the last step of the procedure, option bytes related to the wireless execution configuration are set with the proper values.

At the end of the procedure, the new wireless stack becomes active.

RSS update

- RSS is composed of
  - Reserved data (up to 24 Kbytes)
  - RSS firmware (up to 16 Kbytes)
  - Free space for RSS firmware update (16 Kbytes)
    - This area allows RSS update with rollback capability
- Update procedure
  - Current RSS firmware is active during update
  - New RSS firmware download can be done by Direct access or OTA
  - Former RSS memory space is now available for future updates



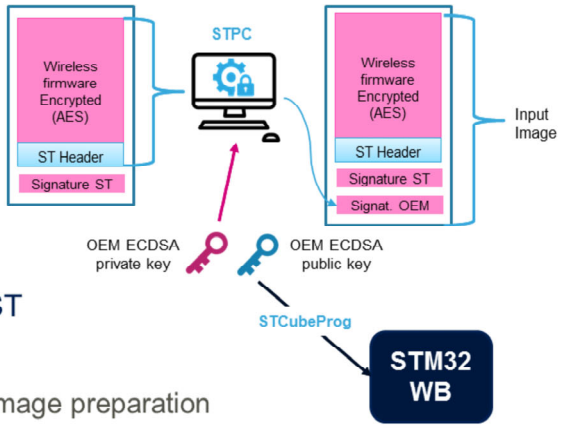


life.augmented

This slide presents the RSS firmware update procedure. The RSS firmware itself can be updated. Two slots are available for the RSS firmware. One for the active RSS code and one for the installation of a new one.
The new encrypted and signed RSS firmware is downloaded either via a wired connection or over-the-air. Once decrypted and authenticated, the new RSS is active. Then, the old one is removed and its slot becomes free for a future update.

OEM wireless stack authentication

- RSS cryptographic scheme
  1. Encryption
     - Wireless stack is encrypted by ST with AES algorithm
  2. Authentication
     - First level: ST authentication
     - Second optional level: OEM authentication



- Tools
  - Wireless stack is delivered encrypted and signed by ST
  - Additional authentication is provided by
    - STM32TrustedPackageCreator tool for signature and image preparation
    - STM32CubeProgrammer for public key download inside the device in secure area

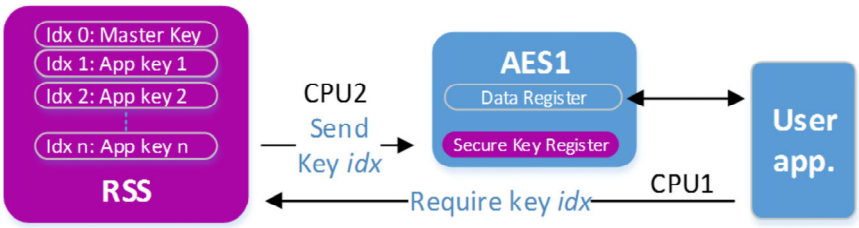
life.augmented

The wireless stack is developed, encrypted and signed by ST. Its encryption is ensured by the AES algorithm and its signature relies on an elliptic curve algorithm.
A second authentication level can be added by customer. This second level of authentication can be required to ensure that only specified versions of the wireless stack can be downloaded.
Customer authentication uses the same elliptic curve algorithm.
A private-public key pair is used. The private key is used to sign the stack image using the STM32TrustedPackageCreator. The public key is stored in the secure part of the Flash memory using the STM32CubeProgrammer and a specific RSS service.

Cryptographic Key Storage - CKS

Usage with AES1 HW IP

- Secure storage of AES cryptographic keys
  - AES1 hardware IP has a secure key register
    - Only Cortex M0 (CPU2) can access (Read/Write) to the key value
    - AES Data is fed by Cortex M4 (CPU1) for user application
  - Application keys (128- or 256-bit AES) are stored inside the secure Flash memory, in RSS area
  - After provisioning, application keys are referenced by a simple index from user point of view



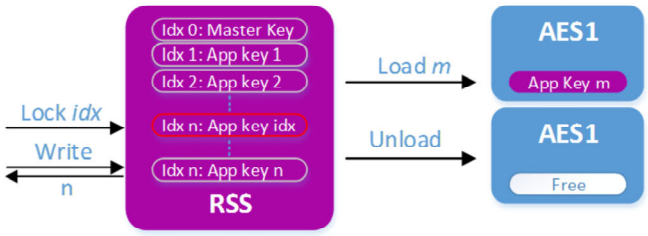
life.augmented

The RSS offers a secure slot for storing the AES cryptographic keys.
These keys are intended for use with the AES1 IP. The Key register of this hardware block is only accessible by the Cortex M0+ in a secure configuration.
The AES1 block of data are sent and fetched by the user application while the key register is loaded or unloaded by the RSS upon user request.
From the user point of view, the key is referenced by its index inside the RSS and its value can never be accessed.

Cryptographic Key Storage - CKS

Provisioning and commands

- Key provisioning
  - External with STM32CubeProgrammer
    - In clear format
    - In encrypted format
      - A master key shall be provisioned first
  - Internal with direct call to RSS from user code



- CKS runtime services

CKS functions	Usage	Arguments
Write	Store key inside RSS	In: Key type Out: Key index
Load	Load specific key inside AES1 key register	In: Key index
Unload	Remove current key from AES1 key register	None
Lock	Prevent reuse of key until next reset	In: Key index

life.augmented

Keys can be provisioned using the STM32CubeProgrammer tool through the device bootloader.
Keys can be sent in the clear or can be first encrypted using a master encryption key. This master key is previously provisioned in a safe place.
All Root Secure Services, including provisioning, are available at application runtime. Four services are available to the user application for key management:

- Write service for storing additional keys
- Load service for AES1 key register programming
- Unload service for cleaning the AES1 key register after application completion
- Lock service to prevent any reuse by another process of a given key until the next reset. This service can be used in a user secure boot application for example.

Related documentation

- Refer to these trainings and application notes related to this feature
  - On-line training modules
    - “STM32WB-System-CM0+ security” module
    - “STM32WB-Security-Memories Protections“ module
  - Application notes
    - AN5185 “STM32WB ST Firmware Upgrade Services”
    - AN5247 “STM32WB: application and wireless firmware update on-the-air (OTA)”

life.augmented

In addition to this training, you may find the Flash memory interface and system configuration trainings useful.