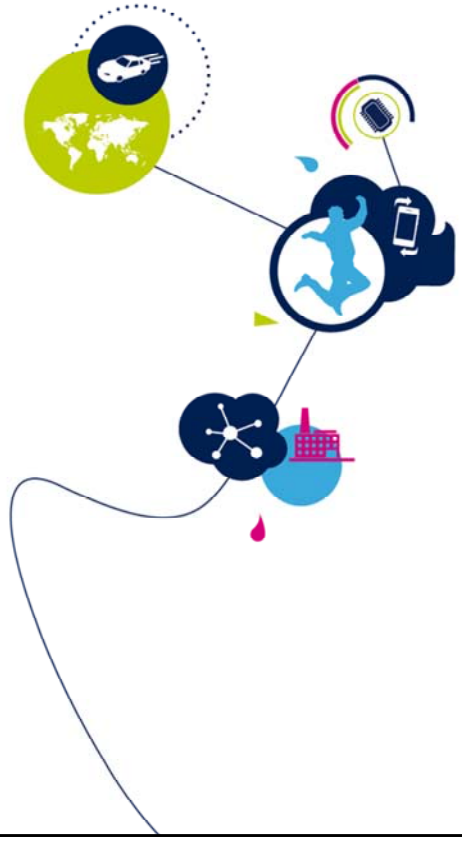
STM32H7B - OTFDEC

On-The-Fly Decryption Engine
Revision 1.0

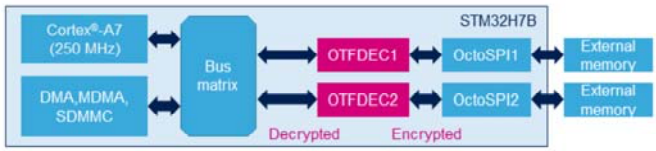


life.augmented

Hello, and welcome to this presentation of the OTFDEC, which is included in STM32H7B microcontrollers.

Overview

- On-the-fly decryption during OctoSPI memory-mapped read operations (single or multiple)
  - Use of AES in counter (CTR) mode to achieve lowest possible latency
- OTFDEC location:



Application benefits

- External flash protection
- Up to eight independent encrypted regions (four per OTFDEC)
- Region configuration write locking mechanism

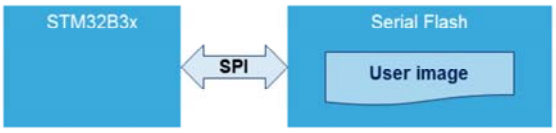
life.augmented

Original purpose of OTFDEC is to protect the confidentiality of read-only firmware libraries stored in external SPI NOR Flash devices.
The OTFDEC performs on-the-fly decryption during OCTOSPI memory-mapped read operation. Any read access size down to the byte is supported.
Two OTFDEC instances are located between the AXI bus matrix and one OctoSPI peripheral that controls the access to an external serial flash.
Advanced Encryption Standard (AES) -128-bit algorithm in counter mode is implemented, to achieve the lowest possible latency.
As a consequence, each time the content of one encrypted region is changed the entire region must be re-encrypted with a different cryptographic context (key or initialization vector).
Up to eight independent encrypted regions (four per OTFDEC) can be defined, each with their own 128-bit key

and initialization vector information (64-bit application nonce and 16-bit encrypted library version).
A write locking mechanism prevents any further reconfiguration of region parameters.

External Flash protection

- User wants to protect code and data stored in external Flash
  - External Flash can be unsoldered and then soldered again on new boards
  - External Flash standard SPI bus can be spied using probes



- User wants protection with a minimum impact on performances
  - AES algorithm consumes a lot of cycle to process blocks of data

life.augmented

The purpose of the OTFDEC peripheral is to protect the user code and/or data that are stored in the external serial flash memory.
If the image is stored unencrypted, it is easy to read it by either de-soldering the flash device then re-soldering it on another board or by spying the traffic on the SPI bus by using a logic analyzer or an oscilloscope.
Consequently the image stored in the flash memory should be encrypted then decrypted on the fly during run-time reads.
The latency caused by the decryption should be minimized.
The OTFDEC has been designed to tackle these objectives.

External Flash protection with OTFDEC

4

- The OTFDEC is a peripheral within STM32H7B that is able to decrypt with low latency code or data stored within external SPI Flash
  - OTFDEC also supports a global encryption mode
- Code + data are protected within external flash up to OTFDEC output to STM32H7B masters
- Each OTFDEC stands before one OctoSPI from STM32 master perspective, intercepting all data read and instruction fetch transactions targeting external Flash (in memory mapped mode)
- After OTFDEC setup, read/fetch transactions to external flash via OTFDEC are transparent from STM32 masters point of view (no decryption needed)

life.augmented

The OTFDEC is a peripheral implemented in the STM32H7B line, able to decrypt with low latency code and/or data stored within an external flash. It also supports an encryption mode. The encryption process must follow the sequence described in the reference manual. When the encryption mode is selected, on-the-fly decryption for all regions is de-activated. Since the decryption is done internally by the microcontroller, the data transferred over the OctoSPI bus is encrypted. This is a countermeasure against flash unsoldering and bus spying.
The OTFDEC is a companion IP of the OctoSPI peripheral. It intercepts any data read and instruction fetch that targets the external flash.
Decryption is transparent to the Cortex-A7 core. Data and/or instructions that the processor receives have been decrypted in hardware by the OTFDEC unit.

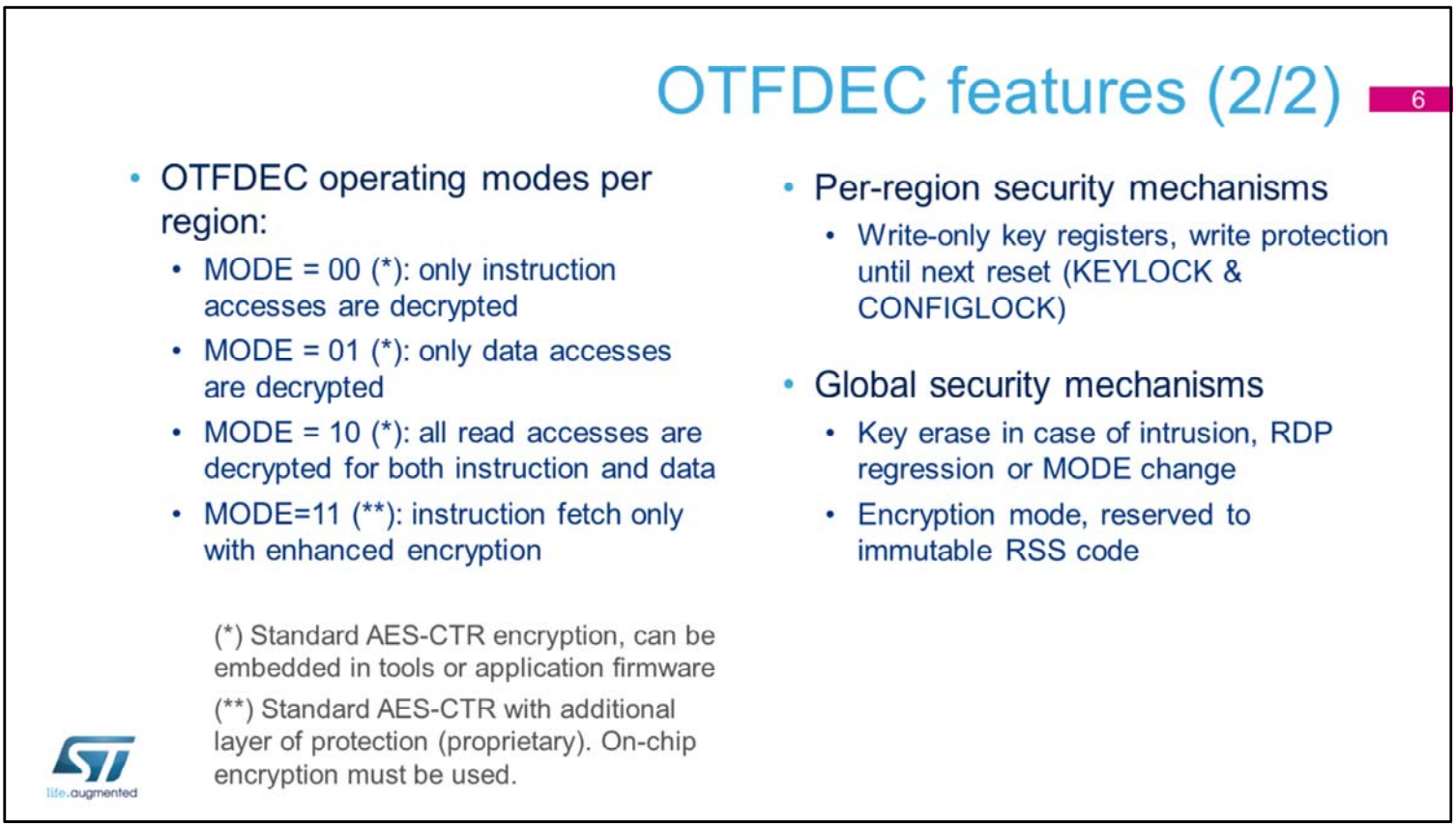
OTFDEC features (1/2)

5

- Protect confidentiality of external :
  - Read-only code, read-only data or read-only code+data areas, all decrypted on the-fly
  - Four independent and non-overlapping encrypted regions can be defined per OTFDEC unit
- AES 128-bit cipher in counter mode (CTR) is used to achieve the lowest possible latency
  - Access minimum granularity: 8-bit
- Each region is defined by:
  - A secret key and its public 8-bit CRC
  - Public diversification data: 64-bit application info + 16-bit library version
- AHB interface for register programming

life.augmented

The OTFDEC protects the confidentiality of external read-only code, read-only data or read-only {code + data} areas. They are decrypted on the fly.
Four independent and non-overlapping encrypted regions can be defined.
The AES 128-bit cipher in counter mode is used to achieve the lowest possible latency.
Access minimum granularity is 8 bits.
Each regions is defined by a 128-bit secret key, and its public 8-bit CRC.
Initialization vector of each region is built by the OTFDEC unit using a 64-bit application information and a 16-bit library version. The user can define this information as public diversification data.
The OTFDEC unit has an AHB slave interface, used to access control and status registers.



For each region, the operating mode has to be selected.
More specifically:

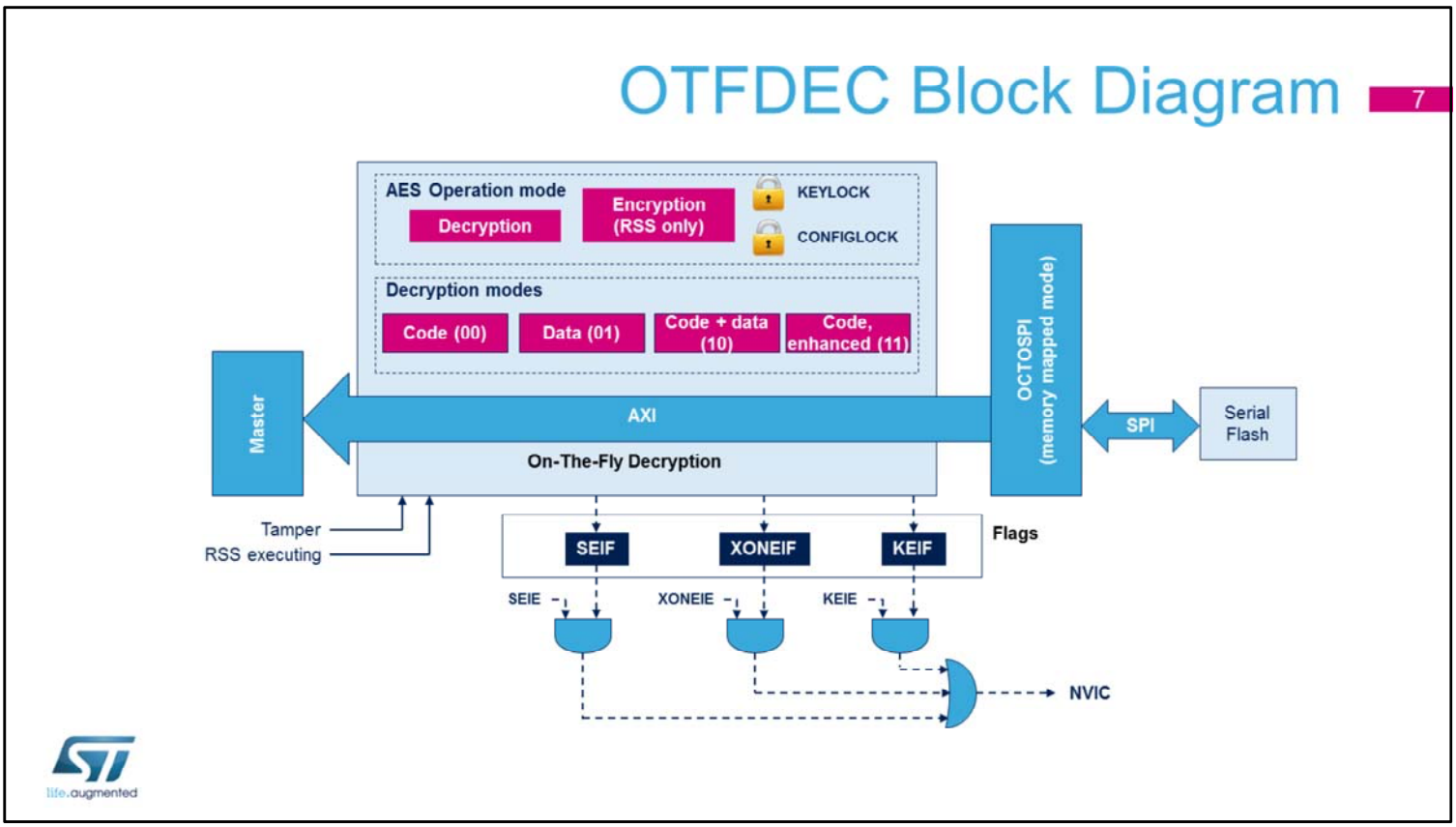
- If the region contains both code and data, the MODE field of the region configuration register has to be set to binary value 10.
- If the region contains only data, the MODE field of the region configuration register has to be set to binary value 01.
- If the region contains only code that can be encrypted externally, the MODE field of the region configuration register has to be set to binary value 00.

For those three modes, standard AES encryption algorithm is used, hence encryption process can be embedded in code generation tools or application firmware for run-time encryption.

If the region only contains instruction, the MODE field of the region configuration register could be set to binary value 11.

In this case an additional layer of protection is added on top of the standard AES encryption algorithm, hence encryption process cannot be embedded in software tools (OTFDEC must be used to perform the encryption, using dedicated RSS function).

The configuration of each region can be independently locked to prevent any further modification. Both the 128-bit key and the configuration parameters can be locked.
All key registers are write only, and are automatically erased in case of intrusion detected by tampers, Readout Protection (RDP) regression or MODE field change.

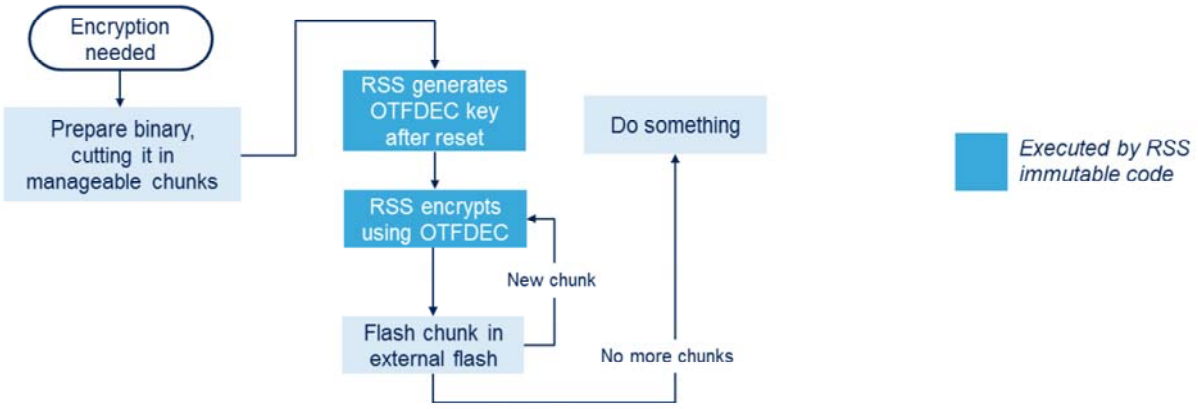


The principle of OTFDEC is to analyze all AXI read transfers on the associated AXI bus.
If the read request is within one of the four regions programmed in OTFDEC, the control logic triggers a keystream computation based on the AES algorithm in counter mode.
This keystream is then used to on-the-fly decrypt the data present in the read transfer from the OCTOSPI AXI master, tying low the RREADY signal of this master while the keystream information is being computed (this takes up to 11 cycles).
Any access outside the enabled OTFDEC regions belongs to a non-encrypted region.
As OTFDEC is used in conjunction with OCTOSPI, it is mandatory to access the flash memory using the memory map mode of the flash controller.
In the region configuration register, the MODE bits define the OTFDEC operating mode (standard or enhanced

encryption).
The RSS can use OTFDEC for encrypting data using either the standard AES algorithm or the enhanced encryption algorithm.
A Tamper detection, a RDP regression or a MODE bits change automatically erases the keys.
The OTFDEC can assert an interrupt to the NVIC for three possible causes: Security error, Key error and Execute-only or execute while encryption error. Each of these causes has a dedicated flag and interrupt enable bit.

OTFDEC encryption

- User or ST firmware can OTFDEC RSS service `resetAndEncrypt()` to manage OTFDEC encryption. User firmware then perform external Flash programming
  - Refer to Application note AN5281 for details



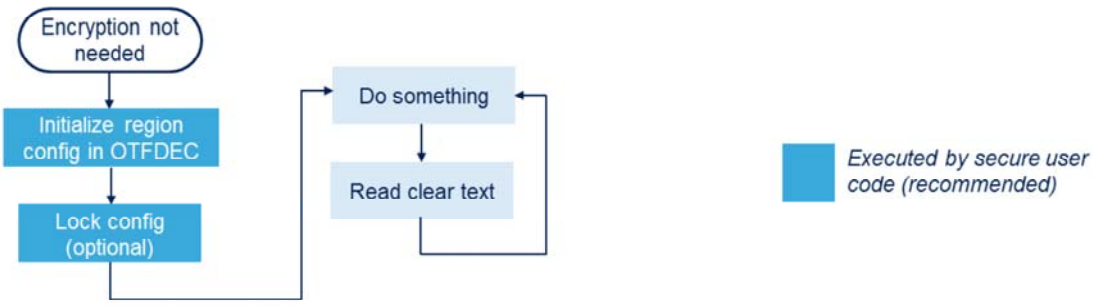
RSS resetAndEncrypt() service to application (ST or user) encrypts code loaded in the dedicated SRAM area.
Depending on the size of the code to encrypt, several calls to this service can be requested.
User firmware is responsible for external Flash Programming.

Note: The RSS service resetAndEncrypt() always triggers a system reset.

OTFDEC decryption

9

- At STM32H7B reset, during the boot sequence, the user firmware must:
  - Load keys within OTFDEC key registers for each OTFDEC region
    - Eight available regions with two OTFDEC instances four per OTFDEC)
  - Lock OTFDEC configuration (e.g. keys)
- Then on-the-fly decryption is ready



life.augmented

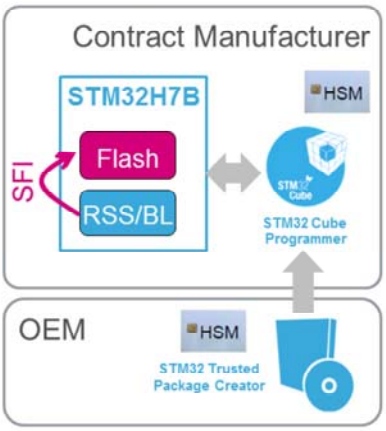
The user firmware is in charge of the following initializations during the boot sequence:

- Loading keys within OTFDEC key registers for each OTFDEC region
- Loading nonce, version, address start and address end information for each OTFDEC region
- Set REG_EN bits
- Locking OTFDEC configuration above (recommended)

Then on-the-fly decryption is ready.

Secure firmware install with OTFDEC (1)

- Secure firmware install (SFI) is a global solution for STM32H7x Series of microcontrollers, allowing secure and counted installation of OEM firmware in untrusted production environment (such as OEM contract manufacturer)
- When external Flash memory is targeted by SFI, OEM firmware is encrypted with an external firmware and data AES key
- OTFDEC can be used to encrypt the external firmware, for example with a device unique key
  - This option is mandatory when MODE=11 (enhanced) is selected for the region. It is illustrated on the next slide
- Refer to AN4992 for more details



Secure firmware install (SFI) is a global solution for STM32H7B Series of microcontrollers, allowing secure and counted installation of OEM firmware in untrusted production environment (such as OEM contract manufacturer). OEM firmware protected by SFI can be stored in the device's embedded flash or encrypted in external flash connected via OCTOSPI.

When external Flash memory is targeted by SFI, OEM firmware code must be encrypted with an external firmware and data AES key. This key can be:

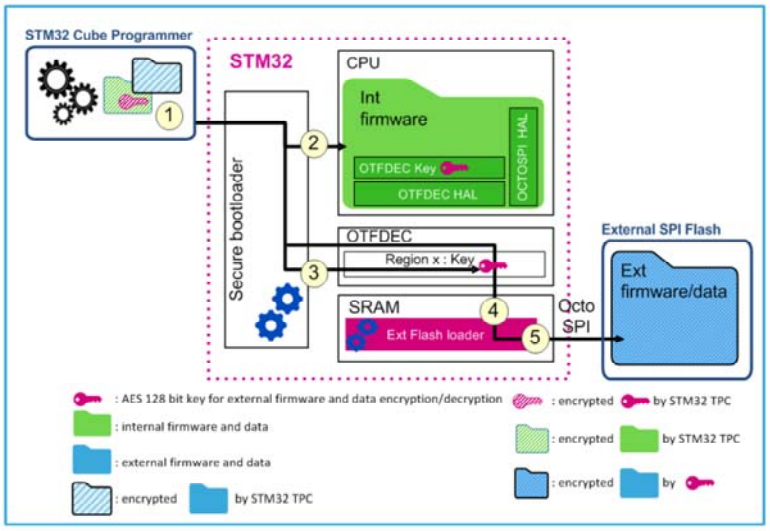
- Common to all devices (in this case tools could perform the encryption if OFTDEC MODE=10), or
- Unique per device (in this case firmware is encrypted inside the device, mandatory if OTFDEC MODE=11)

On-chip encryption using OTFDEC is illustrated in the next slide.

For more information please refer to application note AN4992 for secure firmware install (SFI) solutions.

Secure firmware install with OTFDEC (2)

1. Create an SFI image with STM32 Trusted Package Creator (TPC)
   - Internal firmware and data (including external Flash memory drivers)
   - AES key for the External firmware and data
   - External firmware and data
2. Internal Flash memory programming
3. External firmware and data *AES key* programming in OTFDEC peripheral
   - Alternatively such key(s) can be managed locally to the device, not globally in the flashing tools.
4. External Flash memory chunk encryption
5. External Flash memory programming by the user's firmware



life.augmented

This slide represents the sequence where the STM32 secure bootloader handles both internal firmware installation and external firmware installation with an AES key for the global external Flash memory and the help of an external Flash memory loader. The numerical steps are represented on the schematic.

(1) Create an SFI image using STM32 Trusted Package Creator (TPC), with a) internal firmware and data (including external Flash memory drivers), b) the AES key for the external firmware and data, and c) external firmware and data

(2) Internal Flash memory programming, as described in the STM32H7B RSS training.

(3) External firmware and data AES key programming in OTFDEC peripheral. Alternatively to what is drawn on the slide, this key can be managed locally to the device, not globally in the flashing tools.

(4) External Flash memory chunk encryption

(5) External Flash memory programming by the user’s firmware

Afterward, during each boot, the secure internal firmware in RSS first copies the AES firmware and data key(s) in write-only OTFDEC key registers, then activates the OTFDEC region tied to those keys. At this point the CPU can seamlessly read/fetch data/code from external Flash memory (encrypted or not), once the OCTOSPI driver has been initialized.

Interrupts

Interrupt event	Description
Security error	Illegal read to key registers Illegal write to key registers while KEYLOCK=1 Illegal write to a region's configuration while CONFIGLOCK=1
Execute-only Execute-never	Read access to an execute-only region (MODE[1:0]=00 or 11) Execute access to a execute-never data region (MODE[1:0]=01).
Key error	Read request to an encrypted region while its key registers are null or not properly initialized (KEYCRC=0x0). ➢ Source of the error can be an incorrect key loading sequence (see KEYCR in OTFDEC_REGIONxCFGR) or erased in case of intrusion detected by tampers, Readout Protection (RDP) regression or MODE field change Such read requests return 0x0, without bus error.

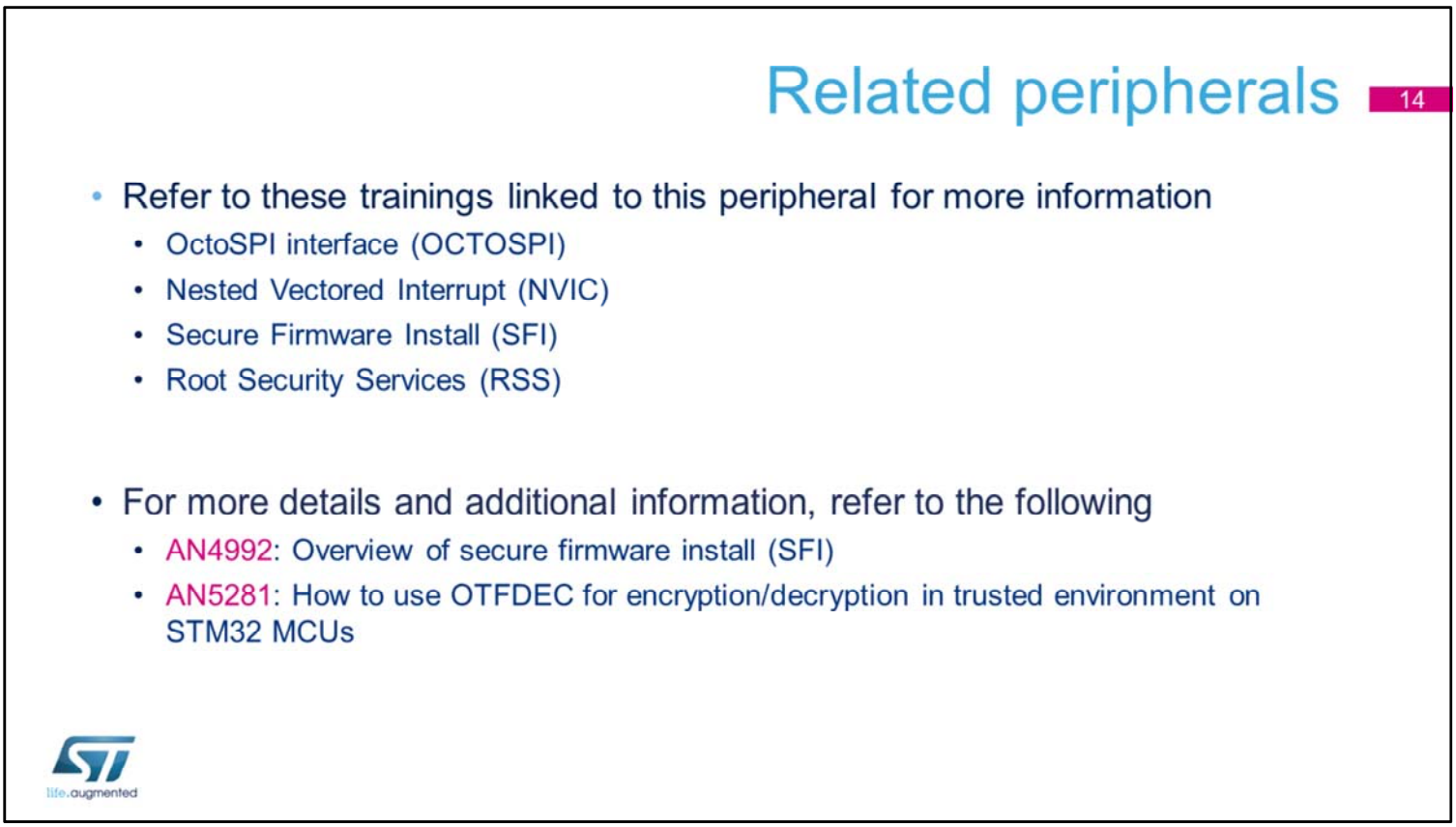
The OTFDEC has 3 interrupt sources.
The security error is raised when an attempt to read key registers is detected or when an attempt to write keys while the KEYLOCK bit is set or when an attempt to reconfigure a region while the CONFIGLOCK bit is set.
When execute-only mode (MODE=00) or enhanced encryption mode (MODE=11) is selected, the execute-only error is raised when a read access is attempted to this protected region.
When data-only mode (MODE=01) is selected the execute-never error is raised when an execute access is attempted to this protected region.
The key error is raised when a read request is attempted to a region whose key registers are null or not properly programmed (KEYCRC=0x0). Key error can happen due to an incorrect key register writing sequence. It can also

occur in case of intrusion detected by tampers, Readout Protection (RDP) regression or MODE field change.

Low-power Modes

Mode	Description
DRun	Active.
DStop	Frozen. Peripheral registers content is kept.
DStop2	
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The peripheral must be reinitialized after exiting Shutdown mode.

The OTFDEC is active in DRun mode.
In DStop, or DStop2 mode, the OTFDEC is frozen, and its registers content is maintained.
In Standby or Shutdown mode, the OTFDEC is powered-down and it must be reinitialized afterward.



The OTFDEC module has relationships with the following other module:

- OctoSPI interface
- Nested Vectored Interrupt Controller
- Secure Firmware Install (SFI)
- Root Security services (with SFI information)

For more details on SFI, please refer to application note AN4992 about Overview of secure firmware install (SFI). For more details (and code example) of the usage of OTFDEC in encryption and decryption please refer to application note AN5281.