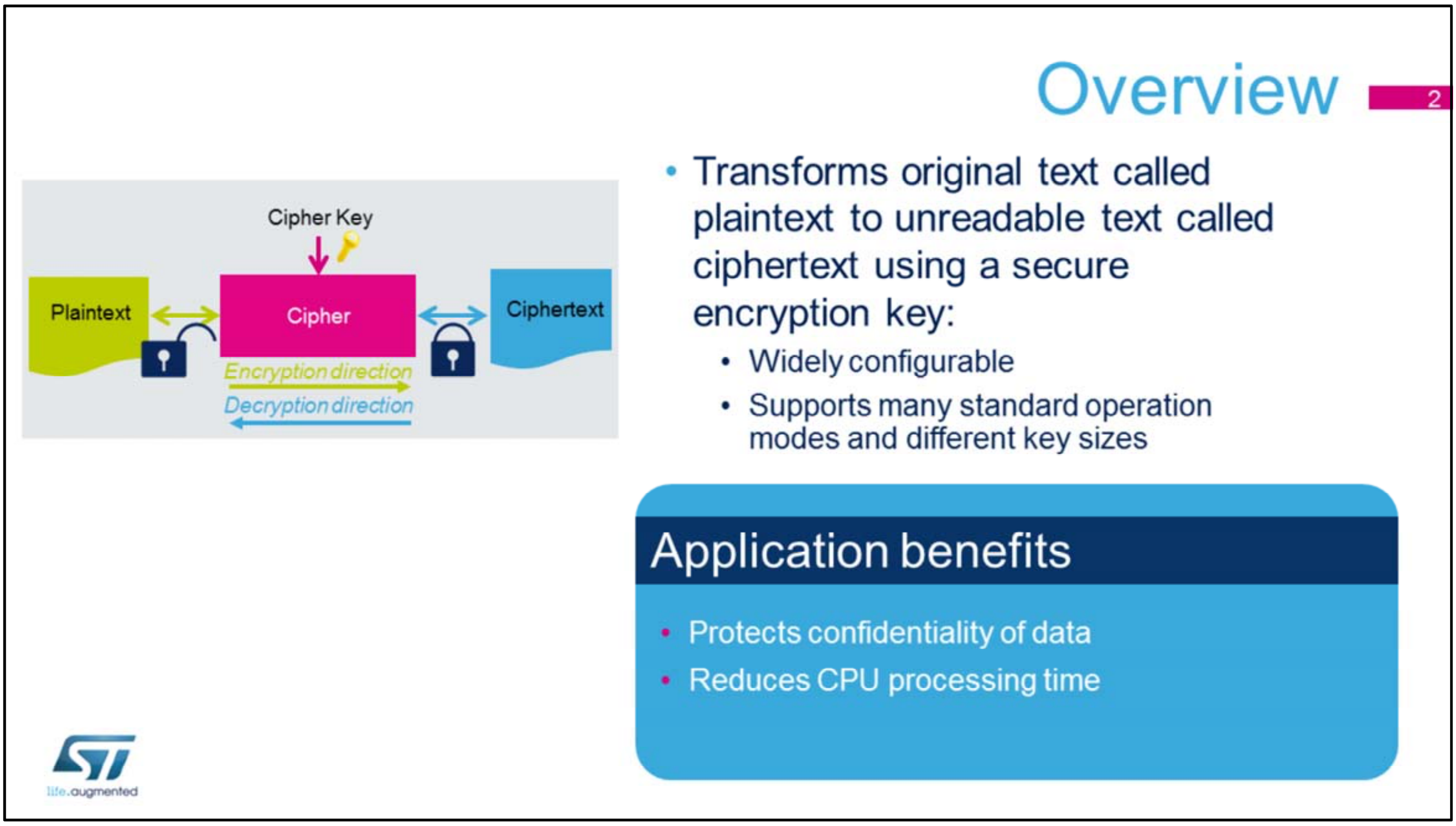
STM32L4 – AES

Advanced Encryption Standard hardware accelerator

Revision 2.0

Hello and welcome to this presentation of the STM32 Advanced Encryption Standard hardware accelerator. It covers the features of the AES interface, which is widely used for cryptographic applications.


Cipher Key
Plaintext
Cipher
Ciphertext
Encryption direction
Decryption direction
life.augmented

Overview

2

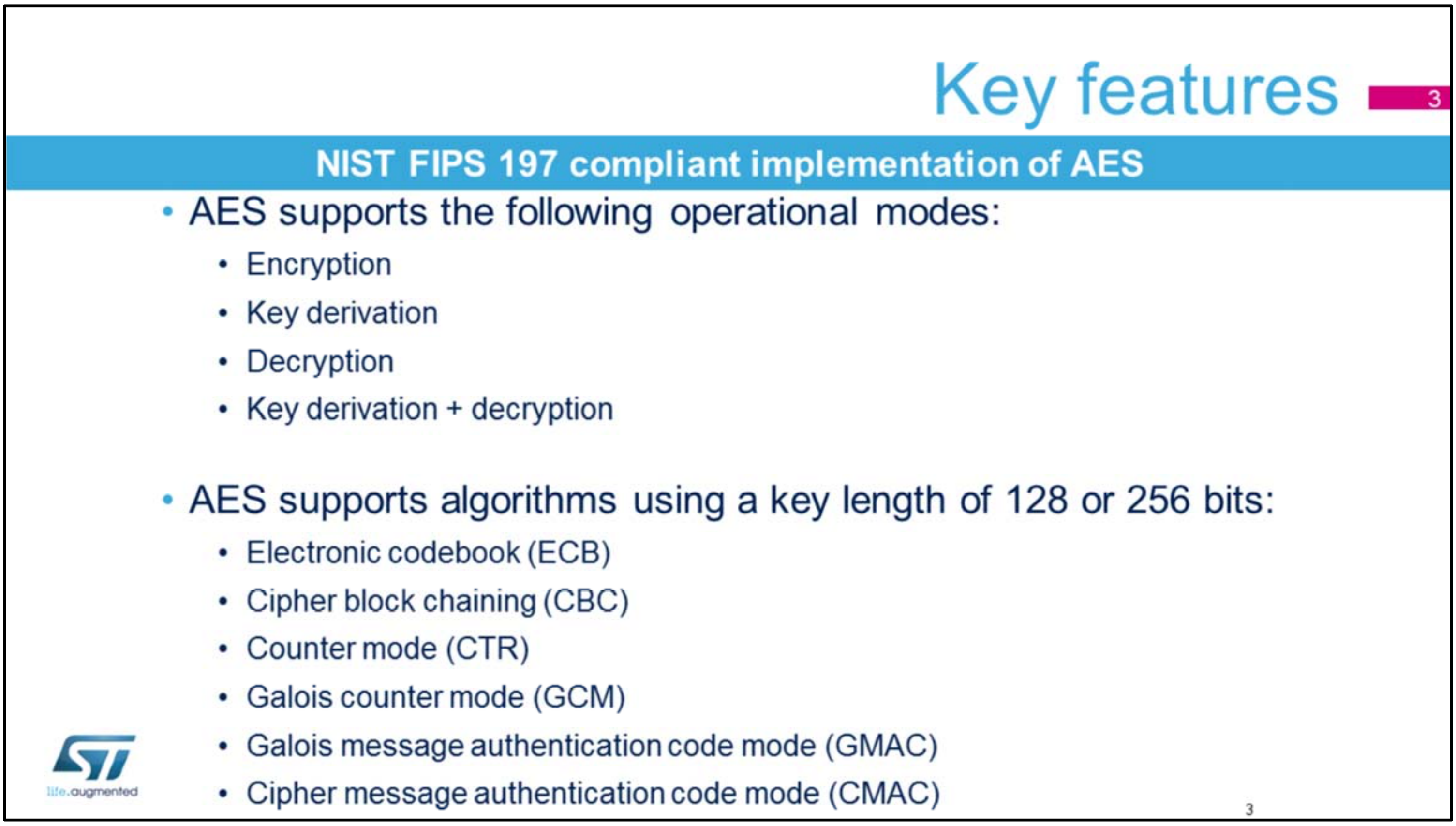
- Transforms original text called plaintext to unreadable text called ciphertext using a secure encryption key:
  - Widely configurable
  - Supports many standard operation modes and different key sizes

Application benefits

- Protects confidentiality of data
- Reduces CPU processing time

The AES algorithm is a symmetric block cipher used to encrypt and decrypt information using a secret cryptographic key that is 128 or 256 bits long. Encryption converts data to an unintelligible format called ciphertext; decrypting the ciphertext converts the data back into its original format, called plaintext.

Applications benefit from the NIST FIPS 197 compliant implementation of the AES algorithm to protect the confidentiality of data as well as its low processing time.



Key features

NIST FIPS 197 compliant implementation of AES

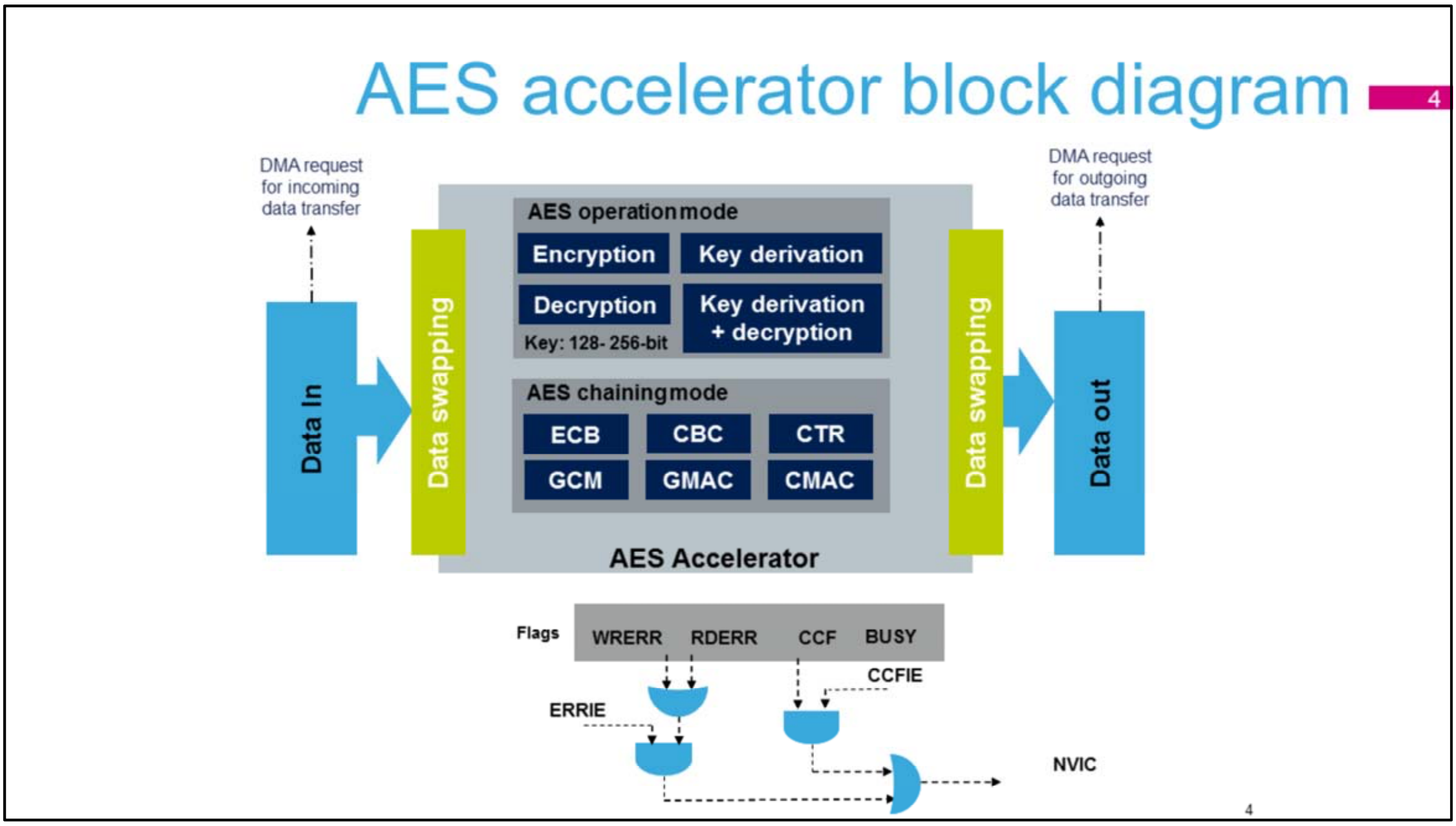
- AES supports the following operational modes:
  - Encryption
  - Key derivation
  - Decryption
  - Key derivation + decryption

- AES supports algorithms using a key length of 128 or 256 bits:
  - Electronic codebook (ECB)
  - Cipher block chaining (CBC)
  - Counter mode (CTR)
  - Galois counter mode (GCM)
  - Galois message authentication code mode (GMAC)
  - Cipher message authentication code mode (CMAC)

The AES accelerator supports four operation modes: Encryption, Key derivation, Decryption and Key derivation plus decryption.

It processes 128-bit data blocks using an encryption key that is either 128 or 256 bits long, based on the selected chaining mode as shown on the next slide.

.



AES accelerator block diagram

4

DMA request for incoming data transfer

DMA request for outgoing data transfer

Data In

Data swapping

AES operation mode

Encryption

Key derivation

Decryption

Key derivation + decryption

Key: 128- 256-bit

AES chaining mode

ECB

CBC

CTR

GCM

GMAC

CMAC

AES Accelerator

Data swapping

Data out

Flags

WRERR

RDERR

CCF

BUSY

CCFIE

ERRIE

NVIC

4

This simplified block diagram of the AES shows the basic functional and control modules.
The AES accelerator processes 128-bit data blocks using an encryption key with a length of either 256 bits or 128 bits, with or without a data swapping option.

The AES accelerator has 4 operating modes:
• Mode 1: Encryption using the encryption key stored in the AES Key registers.
• Mode 2: Key derivation which derives a new key based on the value stored in the AES Key registers before enabling the AES accelerator. This mode is independent from the AES chaining mode selection.
• Mode 3: Decryption using a given (pre-computed) decryption key stored in the AES Key registers.
• Mode 4: Key derivation + decryption using an encryption key stored in the AES Key registers (not used

when the AES is configured in Counter mode for perform a chaining algorithm).
The AES accelerator supports six chaining algorithms or modes:

Electronic codebook (ECB)→This is the default mode. This mode does not use the AES_IVR register. There are no chaining operations. The message is divided into blocks and each block is encrypted separately.

Cipher block chaining (CBC)→ Each block of plaintext is XORed with the previous ciphertext block before being encrypted. To make each message unique, an initialization vector is used when processing the first block.

Counter mode (CTR)→ A 32-bit counter is used in addition to a nonce value for the XOR operation with the ciphertext or plaintext.

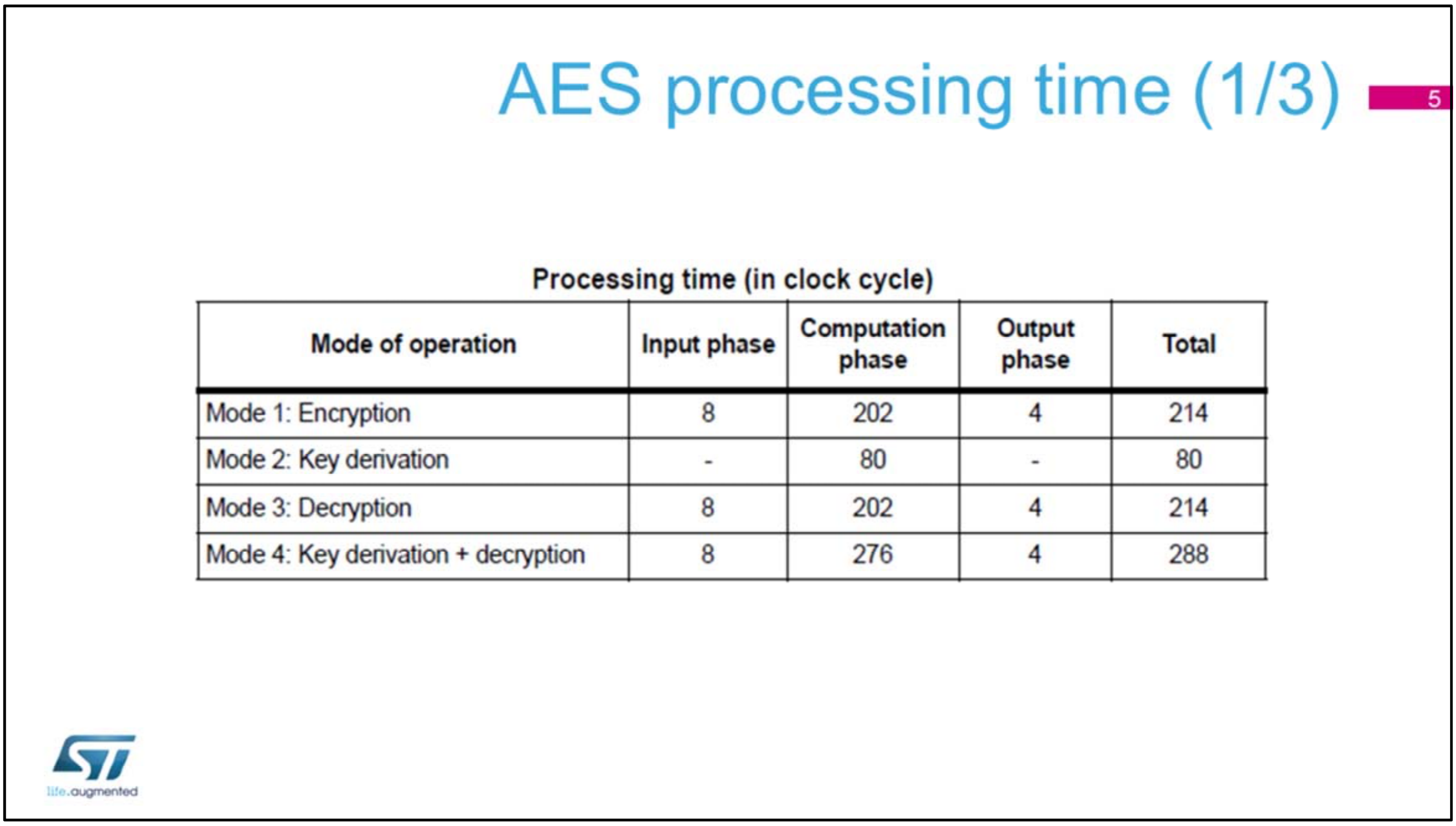
Galois counter mode (GCM)→ Used to encrypt and authenticate the plaintext, generating the corresponding ciphertext and the TAG (also known as message authentication code or message integrity check). It is based on the AES's counter mode for confidentiality and uses a multiplier over a fixed finite field for generating the TAG. It requires an initialization vector at the beginning.

Galois message authentication code mode (GMAC)→ GMAC is the same as GCM applied on a message composed of only the header. All steps and settings are the same except the payload phase will not be used.

Cipher message authentication code mode (CMAC).→ CMAC is used to authenticate the plaintext, generating the corresponding TAG. The message is composed of only the header phase and the tag phase. The CCM standard defines specific encoding rules for the first authentication block (called B0 in the standard). In particular, the first block includes flags, a nonce and the payload length expressed in bytes.

The Error Flags block checks the behavior of the AES accelerator via two different flags:
The Read Error flag (**RDERR**) is set in the AES Status register when an unexpected read operation is detected during the computation phase or during the input phase.
The Write Error flag (**WRERR**) is set in the AES Status register when an unexpected write operation is detected during the output phase or during the computation phase.
An interrupt can be generated when one of these two error flags is set if the Error Interrupt Enable (**ERRIE**) bit in the AES Control register was previously set.

Two extra flags are available for the AES accelerator to give the status of current operation:
The Computation Complete flag (**CCF**) is set by hardware when the computation is complete. An interrupt is generated if the CCF Interrupt Enable bit was previously set.
The Busy flag, used only with GCM mode, indicates that a higher priority message can interrupt the current message during GCM payload phase for encryption mode.

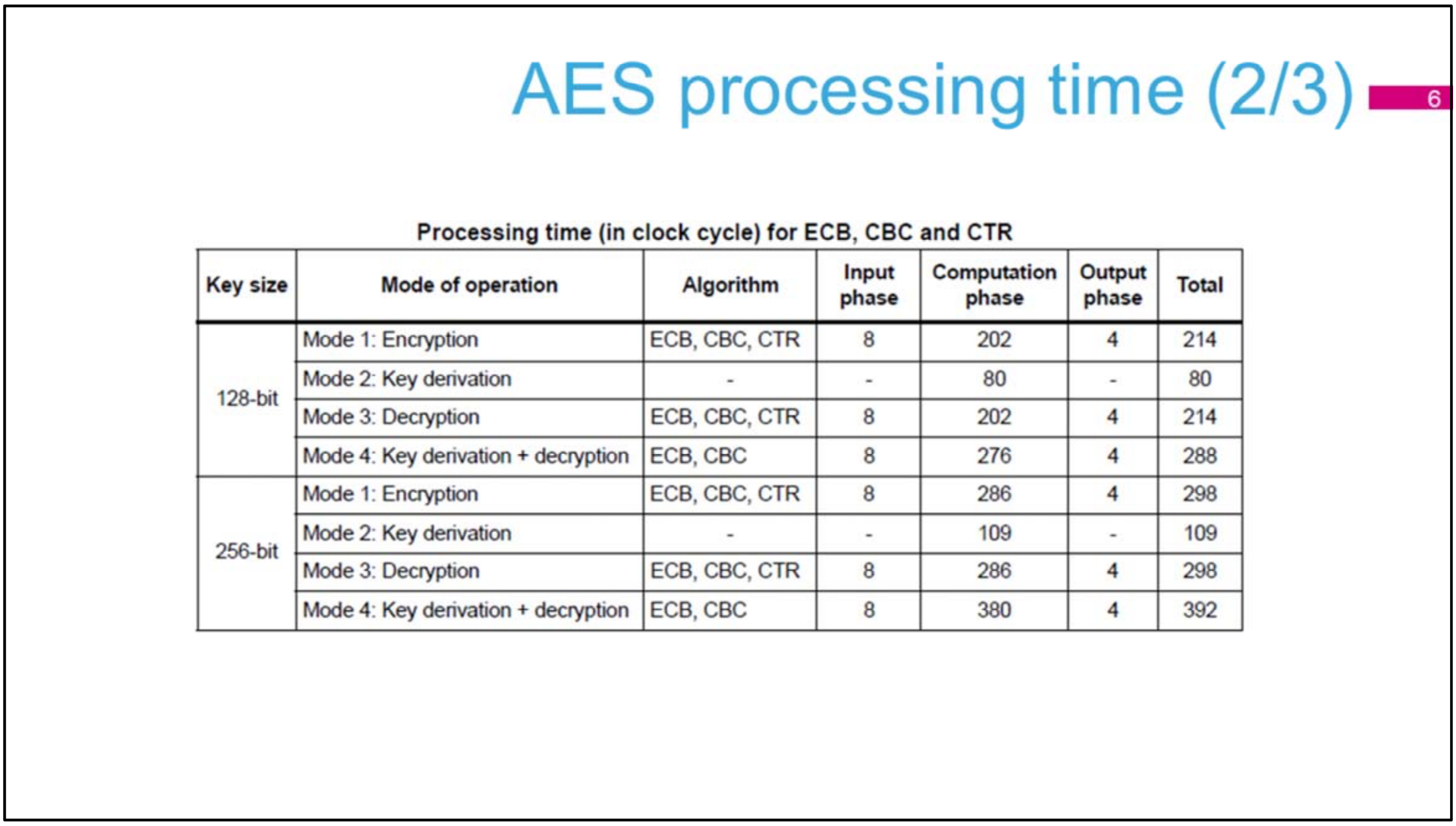


AES processing time (1/3)

Processing time (in clock cycle)

Mode of operation	Input phase	Computation phase	Output phase	Total
Mode 1: Encryption	8	202	4	214
Mode 2: Key derivation	-	80	-	80
Mode 3: Decryption	8	202	4	214
Mode 4: Key derivation + decryption	8	276	4	288

The following slides give the processing times for each of the operating modes according to the selected chaining mode.

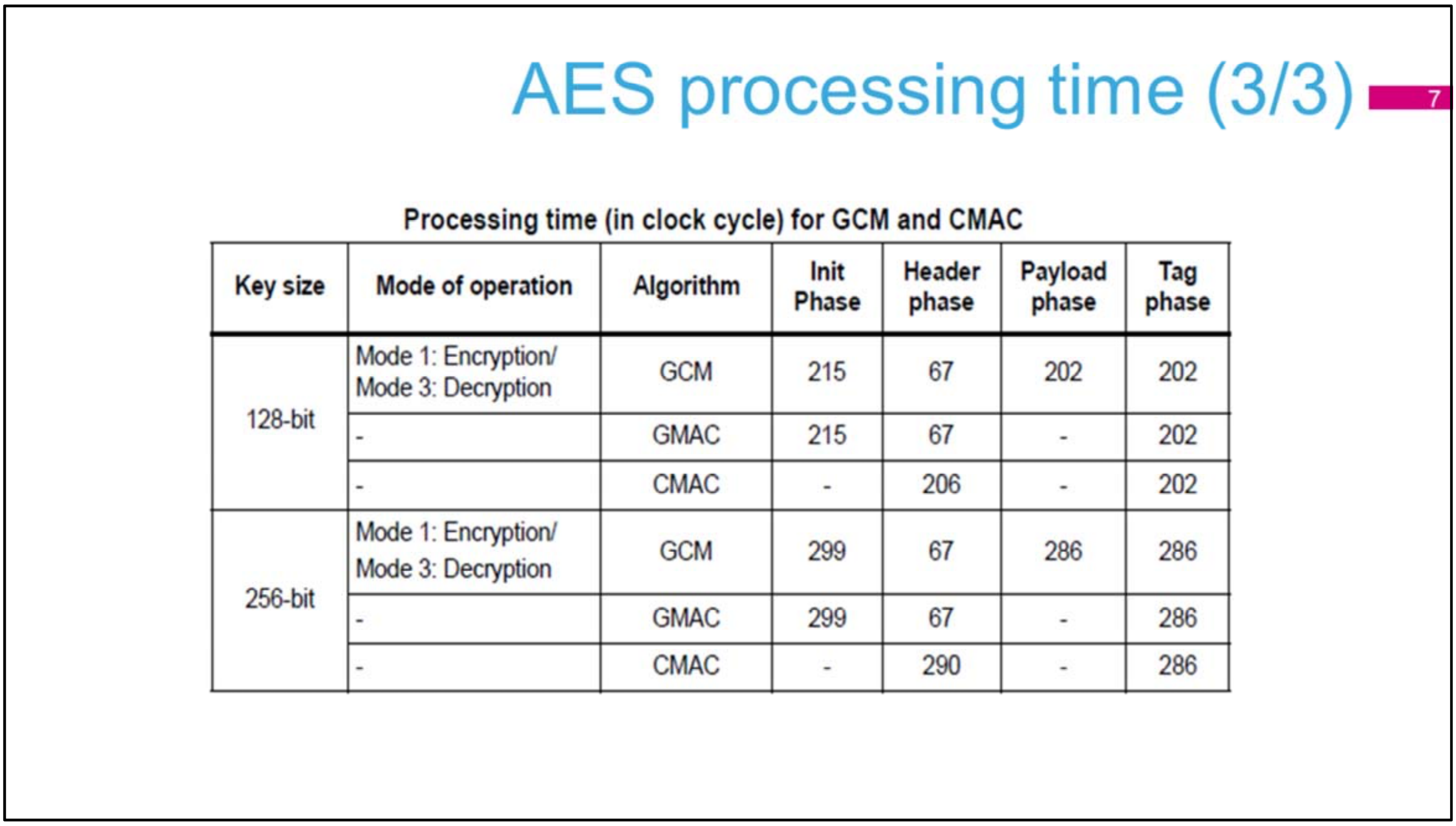


AES processing time (2/3)

Processing time (in clock cycle) for ECB, CBC and CTR

Key size	Mode of operation	Algorithm	Input phase	Computation phase	Output phase	Total
128-bit	Mode 1: Encryption	ECB, CBC, CTR	8	202	4	214
	Mode 2: Key derivation	-	-	80	-	80
	Mode 3: Decryption	ECB, CBC, CTR	8	202	4	214
	Mode 4: Key derivation + decryption	ECB, CBC	8	276	4	288
256-bit	Mode 1: Encryption	ECB, CBC, CTR	8	286	4	298
	Mode 2: Key derivation	-	-	109	-	109
	Mode 3: Decryption	ECB, CBC, CTR	8	286	4	298
	Mode 4: Key derivation + decryption	ECB, CBC	8	380	4	392

Here it is the processing times depending on the key size and algorithms

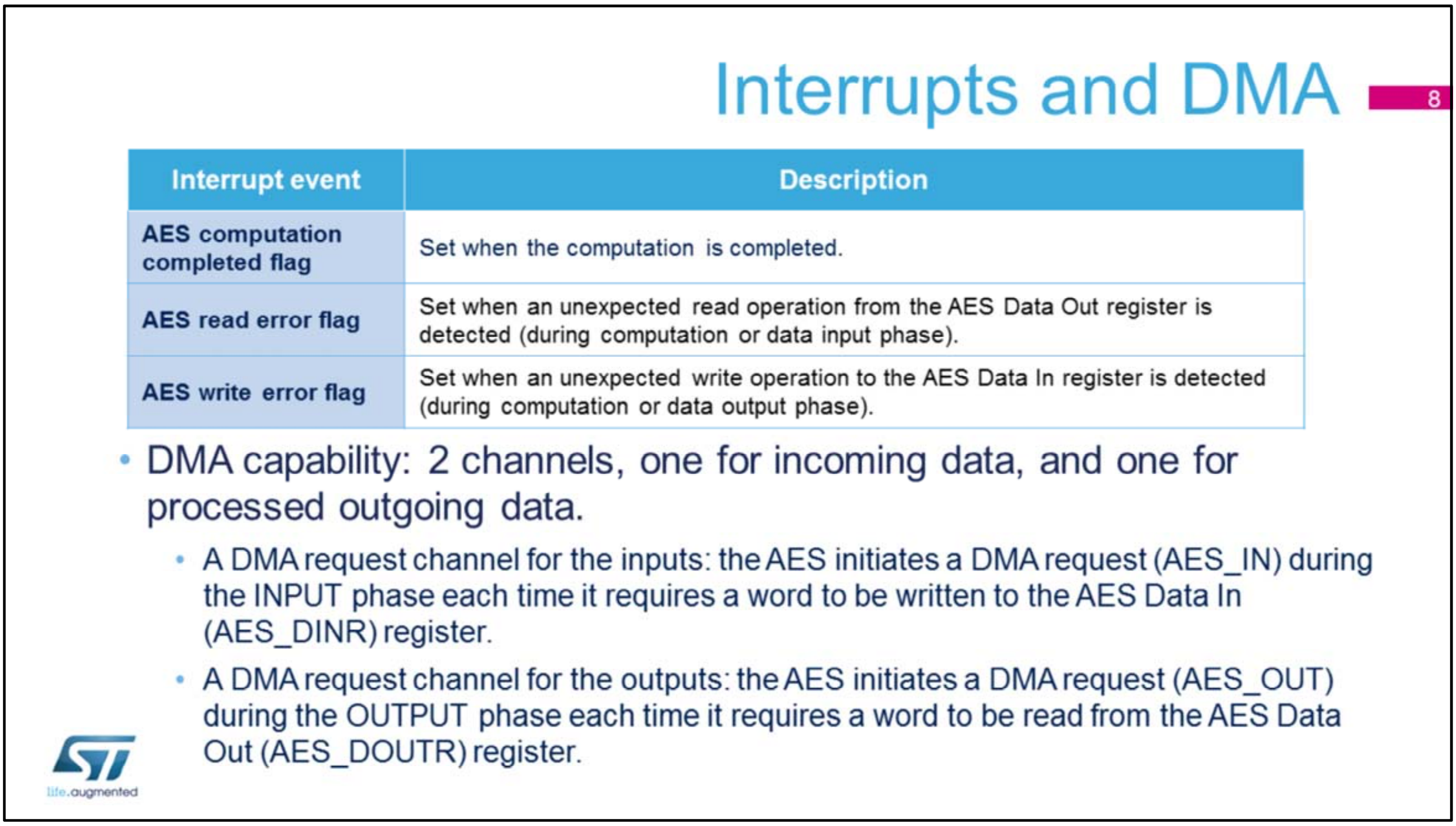


AES processing time (3/3)

Processing time (in clock cycle) for GCM and CMAC

Key size	Mode of operation	Algorithm	Init Phase	Header phase	Payload phase	Tag phase
128-bit	Mode 1: Encryption/ Mode 3: Decryption	GCM	215	67	202	202
	-	GMAC	215	67	-	202
	-	CMAC	-	206	-	202
256-bit	Mode 1: Encryption/ Mode 3: Decryption	GCM	299	67	286	286
	-	GMAC	299	67	-	286
	-	CMAC	-	290	-	286

To complete the feature, here it is the processing time for GCM and GMAC algorithms.



Interrupts and DMA

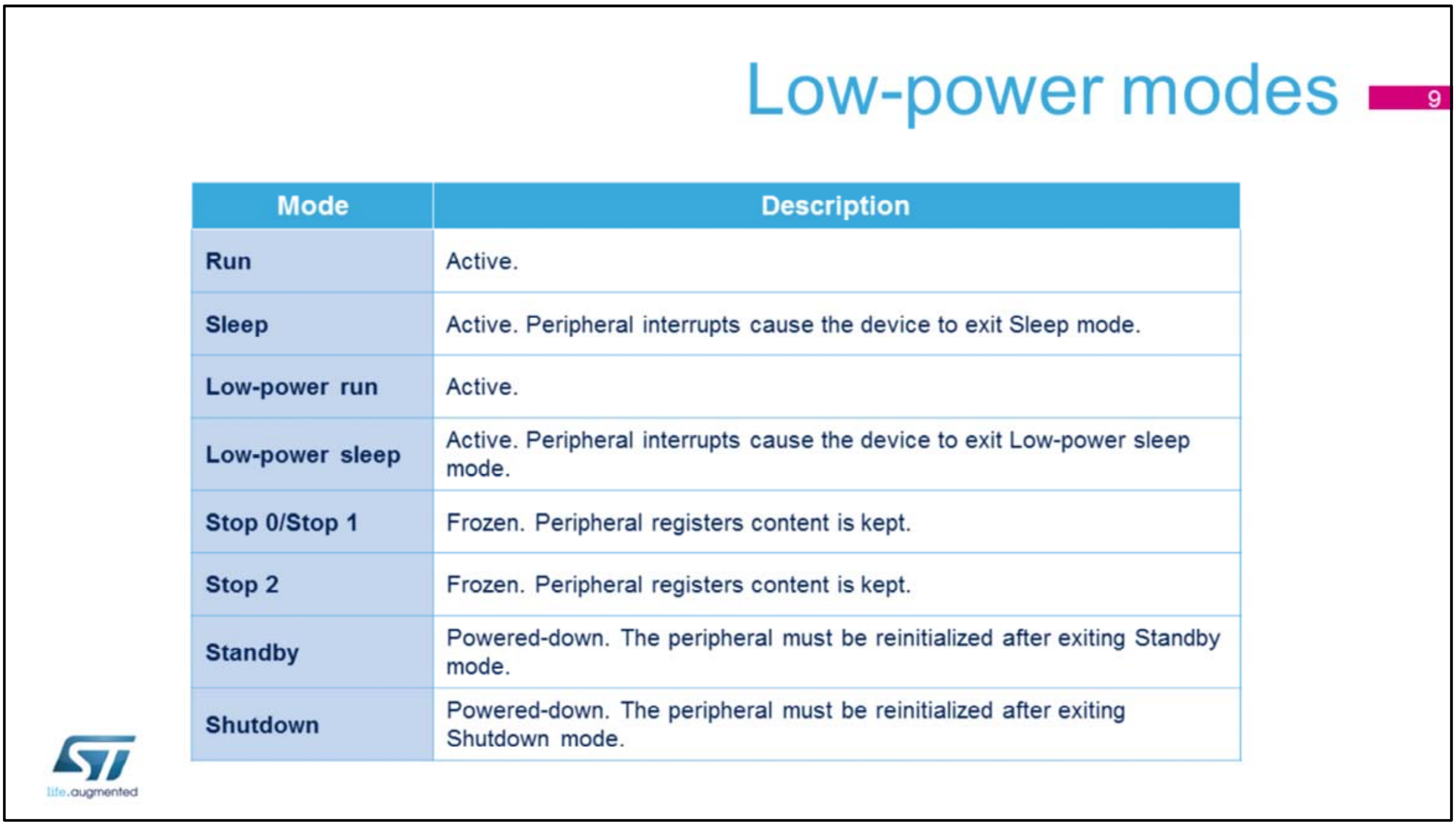
Interrupt event	Description
AES computation completed flag	Set when the computation is completed.
AES read error flag	Set when an unexpected read operation from the AES Data Out register is detected (during computation or data input phase).
AES write error flag	Set when an unexpected write operation to the AES Data In register is detected (during computation or data output phase).

- DMA capability: 2 channels, one for incoming data, and one for processed outgoing data.
  - A DMA request channel for the inputs: the AES initiates a DMA request (AES_IN) during the INPUT phase each time it requires a word to be written to the AES Data In (AES_DINR) register.
  - A DMA request channel for the outputs: the AES initiates a DMA request (AES_OUT) during the OUTPUT phase each time it requires a word to be read from the AES Data Out (AES_DOUTR) register.

life.augmented

Here is a summary of the events able to trigger an interrupt in the nested vectored interrupt controller: AES computation completed, AES read error, and AES write error.

Direct memory access requests are generated internally for both incoming and outgoing data. The DMA channel must be configured in Memory-to-peripheral or Peripheral-to-memory mode with a data size equal to 32 bits.

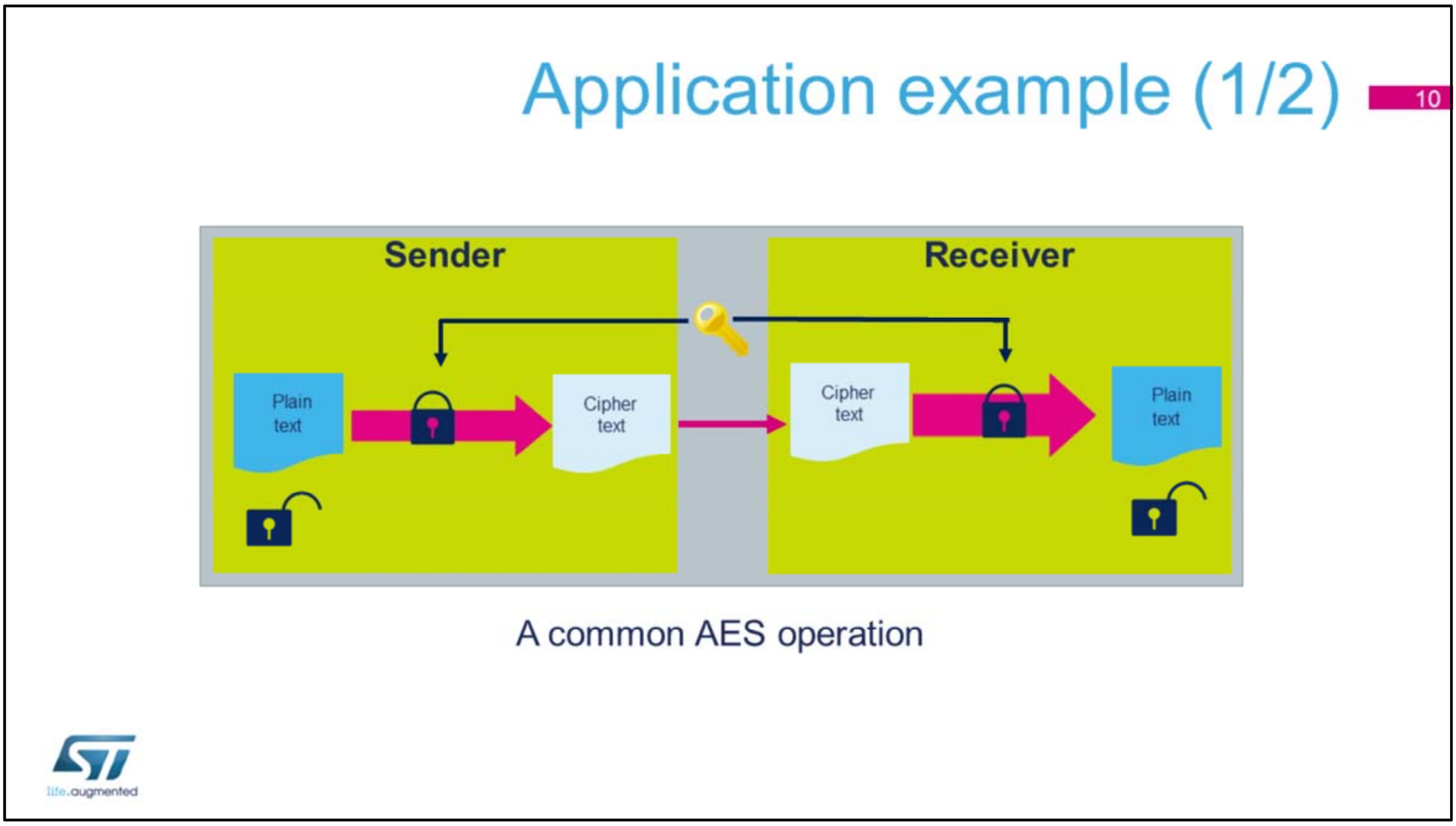


Low-power modes

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Low-power run	Active.
Low-power sleep	Active. Peripheral interrupts cause the device to exit Low-power sleep mode.
Stop 0/Stop 1	Frozen. Peripheral registers content is kept.
Stop 2	Frozen. Peripheral registers content is kept.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The peripheral must be reinitialized after exiting Shutdown mode.

Here is an overview of the status of the AES accelerator in each of the low-power modes.

AES operations are not possible when the device is in Stop mode.



Application example (1/2)

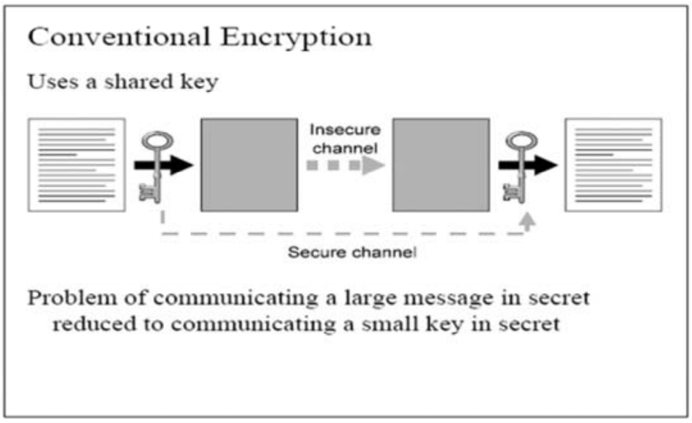
A common AES operation

The AES encryption and decryption algorithms are suitable for a variety of applications such as secure networking routers, wireless communications, encrypted data storage including secure smartcards, secure video surveillance systems, secure electronic financial transactions, etc.

The sender sends a plaintext message encrypted with a secret-key. And the receiver decrypts the message with the same secret key.

Application example (2/2)

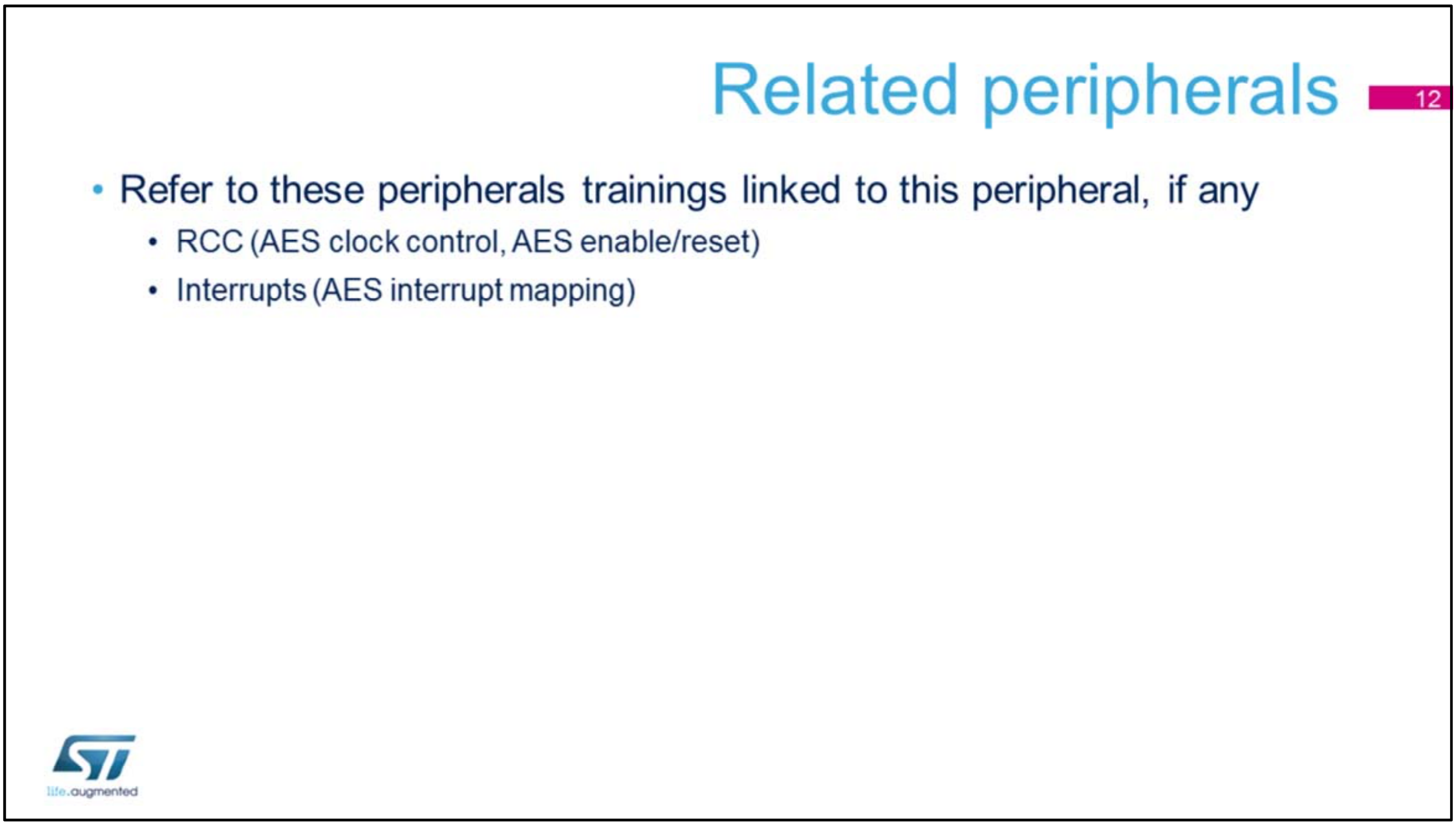
- Examples
  - AES-128, -256
- Advantage
  - Easy/fast way to compute using a normal processor and very, very fast when using coprocessors
- Disadvantage
  - Key distribution
    - Transmitted using some other encryption
    - Sent using secure hardware
    - Key ceremony



Conventional Encryption

Uses a shared key

Problem of communicating a large message in secret reduced to communicating a small key in secret



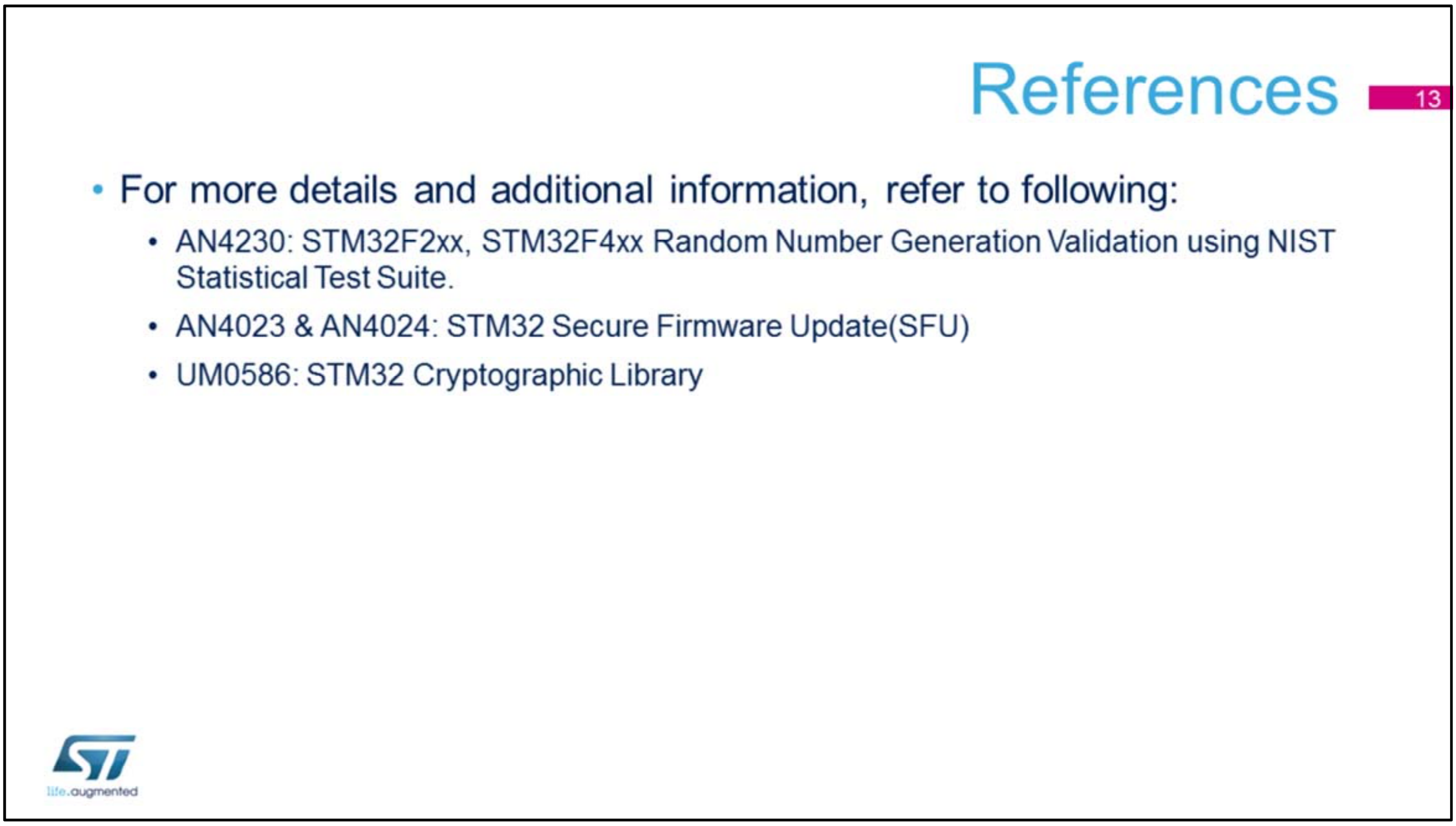
Related peripherals

12

- Refer to these peripherals trainings linked to this peripheral, if any
  - RCC (AES clock control, AES enable/reset)
  - Interrupts (AES interrupt mapping)

life.augmented

This is a list of peripherals related to the AES accelerator. Please refer to these peripheral trainings for more information if needed.



References

- For more details and additional information, refer to following:
  - AN4230: STM32F2xx, STM32F4xx Random Number Generation Validation using NIST Statistical Test Suite.
  - AN4023 & AN4024: STM32 Secure Firmware Update(SFU)
  - UM0586: STM32 Cryptographic Library

For more details, please refer to these application notes and user manual available on our website.