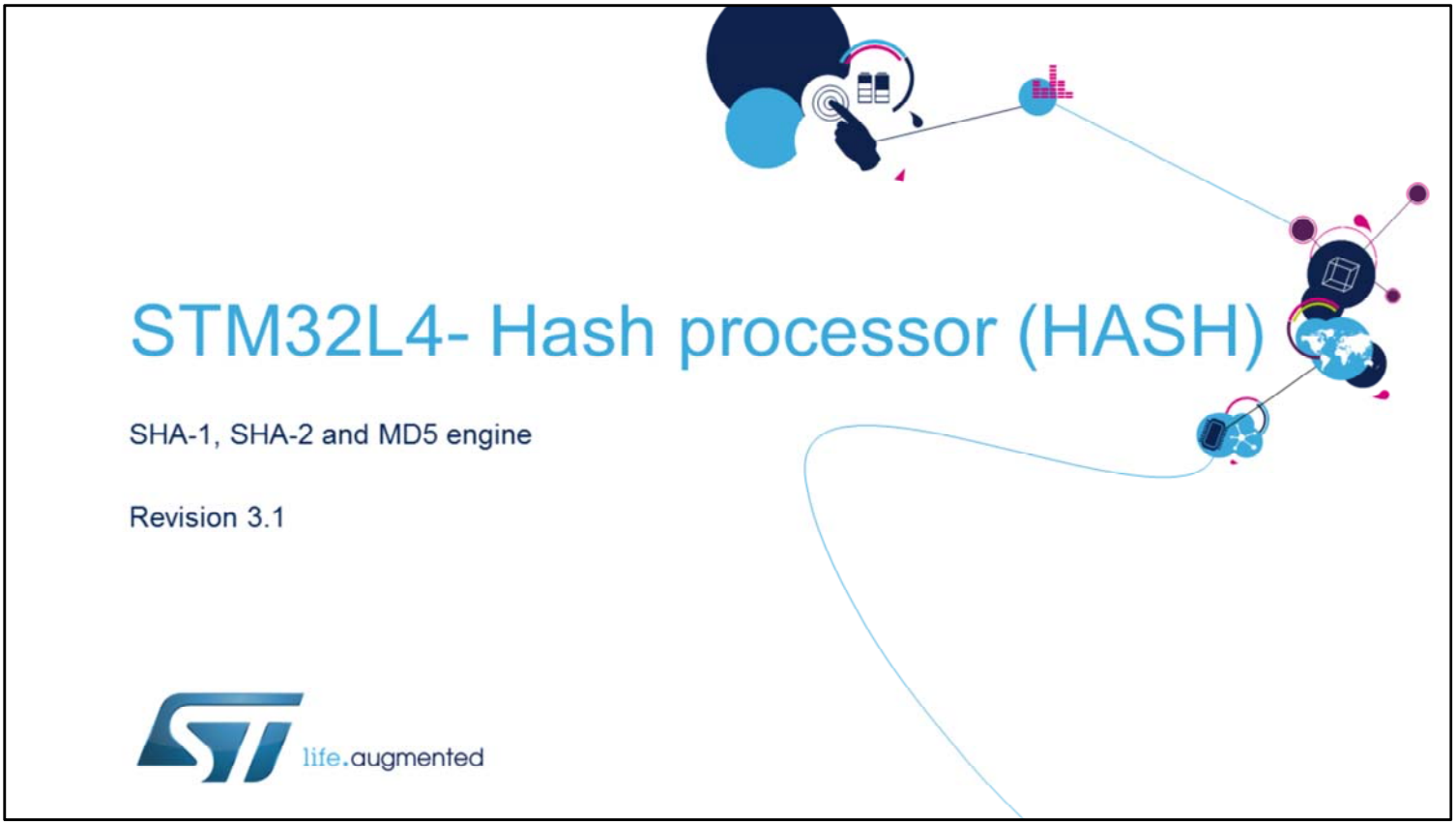
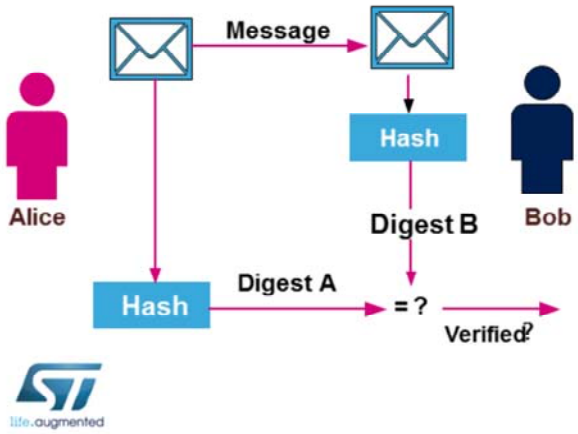


Hello, and welcome to this presentation of the Hash processor.

Overview

2

- Hash processing
  - Computes fixed-length digest from a message
  - Message cannot be retrieved from its digest
  - It is virtually impossible to find two messages with the same digest (collision)



Application benefits

- Used to secure transaction by ensuring
  - Message Integrity (HASH)
  - Message Authentication (HMAC)
- Reduces CPU processing time

Hash peripheral is in charge of efficient computing of message digest.
A digest is a fixed-length value computed from an input message. A digest is unique - it is virtually impossible to find two messages with the same digest. The original message cannot be retrieved from its digest.
Hash digests and Hash-based Message Authentication Code (HMAC) are widely used in communication since they are used to guarantee the integrity and authentication of a transfer.

Key features

3

- The hash processor supports:
  - Fast computation of the following hash functions:
    - MD5
    - SHA-1
    - SHA-224 and SHA-256
  - Computation of a simple hash digest or a hash-based message authentication code
  - Automatic byte swapping to comply with big and little endianness
  - Automatic padding to complete the input bit string to fit modulo 512 (16 × 32 bits) message digest computing
  - Automatic data flow control with support for direct memory access (DMA)

life.augmented

The hash processor supports widely used hash functions including Message Digest 5 (MD5), Secure Hash Algorithm SHA-1 and the more recent SHA-2 with its 224- and 256-bit digest length versions.
A hash can also be generated with a secrete-key to produce a message authentication code (MAC).
The processor supports bit, byte and half-word swapping. It supports also automatic padding of input data for block alignment.
The processor can be used in conjunction with the DMA for automatic processor feeding.

Hash functions

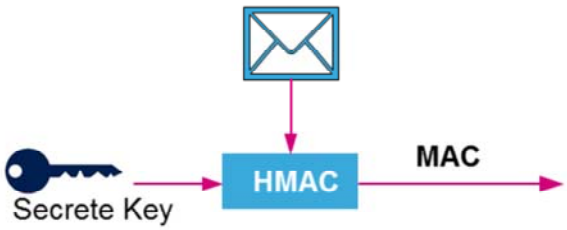
- Block-based algorithms
  - Supported hash functions work on 512-bit blocks of data. The original message is split into subsequent blocks of 512 bits after padding if needed.
  - Internal computation is done on 32-bit words
  - Collision robustness increases with digest length

Hash function		Digest (bits)	Strength (collision)
MD5		128	2^64
SHA-1		160	2^80
SHA-2	SHA- 224	224	2^112
	SHA- 256	256	2^128

All supported hash functions work on 512-bit blocks of data. The input message is split as many times as needed to feed the hash processor. Subsequent blocks are computed sequentially.
MD5 is the less robust function with only a 128-bit digest.
The SHA standard has two versions SHA-1 and the more recent SHA-2 with its 224- and 256-bit digest length versions.

HMAC Hash-based message authentication code

- H-MAC
  - Ensures authentication of messages in addition to Integrity
  - Involves a secret key shared by both sender and receiver
- The algorithm consists of two nested hash operations:
  - HMAC(message) = Hash[((key | pad) ⊕ 0x5C) | Hash(((key | pad) ⊕ 0x36) | message)]
  - Hash function is any of the ones supported by the peripheral



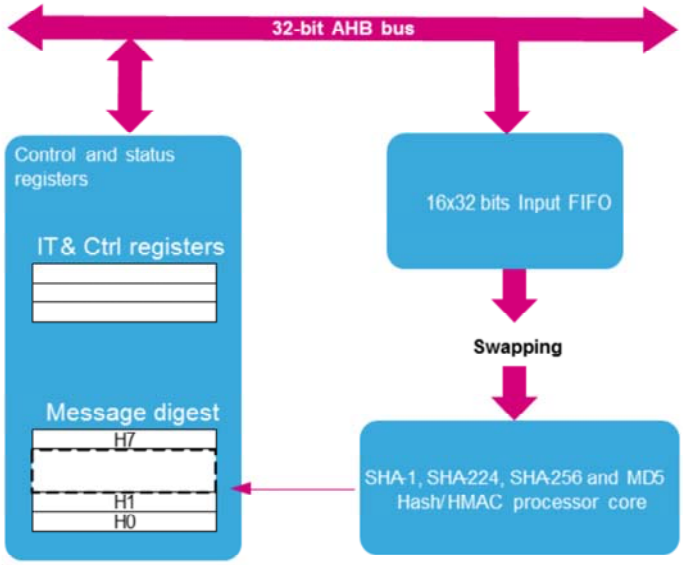
The hash-based message authentication code (HMAC) is used to authenticate messages and verify their integrity.
The HMAC function consists of two nested Hash function with a secrete key that is shared by the sender and the receiver.
The hash function involved in the HMAC computation can be any one supported by the peripheral: MD5, SHA-1 or SHA-2

Standard compliance

- This hash processor , suitable for data authentication applications, is compliant with the following standards:
  - FIPS PUB 180-2 (Federal Information Processing Standards Publication 180-2) for Secure Hash Standard specifications (SHA-1, SHA-224 and SHA-256)
  - IETF RFC 1321 (Internet Engineering Task Force Request For Comments number 1321) specifications for Message-Digest algorithm (MD5)
  - FIPS PUB 198-1 for The Keyed-Hash Message Authentication Code (HMAC)

The hash processor complies with the international standards for Secure Hash Algorithms (SHA), Message Digest algorithms (MD5) and for Message Authentication Code (MAC).

Block diagram



This simplified block diagram of the hash processor shows the basic data flow and control modules.
The hash processor processes 512-bit data blocks and generates digests of up to 256 bits depending on the algorithm.
Input data may be swapped before entering the core unit where they will be processed to generate a simple hash or a message authentication code (MAC).

Interrupts and DMA

Interrupt event	Description
Hash digest calculation completion	Set when a digest becomes ready (the whole message has been processed).
Hash data input ready	Set when the input buffer is ready to get a new block of 512 bits (16 locations are free).

- DMA capability: Only one channel for input data from memory
  - The hash processor initiates a DMA request (HASH_IN) to load data of one full block of 512 bits.
  - DMA complete transfer interrupt can be used for data flow control.

life.augmented

An interrupt in the nested vectored interrupt controller (NVIC) is triggered when a hash digest has been successfully calculated or when the hash processor is ready to accept a new block of data.

In Direct memory access (DMA) mode, requests are generated internally for incoming data. The DMA channel must be configured in Memory-to-peripheral mode with a data size equal to 512 bits.

Performance

- Hash performance
  - The computation of an intermediate block (512 bits) of a message takes:
    - 66 HCLK clock cycles in SHA-1
    - 50 HCLK clock cycles in SHA-224
    - 50 HCLK clock cycles in SHA-256
    - 50 HCLK clock cycles in MD5
    - The time needed to load the 16 words of the block into the processor is considered too (at least 16 clock cycles for a 512-bit block)
  - This peripheral increases speed and saves more power compared to the software version of the algorithms.

life.augmented

These are the times it takes top process a single block of data depending on the chosen algorithms.
HCLK is the CPU clock and can go as high as 80MHz.
Note that main benefit of using a hardware accelerator is to increase speed and save power compared to a full software implementation of the hash functions.

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.

life.augmented

Here is an overview of the status of the hash processor in each of the low-power modes.
Hash operations are not possible when the device is in Stop mode.

Related peripherals

11

- Refer to these peripherals trainings linked to this peripheral
  - AES
  - DMA

This is a list of peripherals related to the hash processor. Please refer to AES peripheral trainings if you want to know more about cryptographic functions.
Refer to training on the DMA peripheral for information on how to configure the hash channel.

References

- For more details and additional information, refer to following:
  - UM0586: STM32 Cryptographic Library

For more details, please refer to this user manual available on our website.