package Diagrama de classes

CRIPTOGRAFIA

«abstract»
Crypto

+shaGen(buffer_in: uint8_t, in_len: size_t, out buffer_out: uint8_t *, out_len: size_t, sha_type: SHA_t): STATUS_t*
*+shaCheck(buffer_expected: uint8_t *, buffer_digest: uint8_t *, sha_type: SHA_t): STATUS_t*
*+sigCheck(buffer_expected: uint8_t *, buffer_in: uint8_t *, buffer_in_len: size_t, pubkey: key_t *, algo: ALGORITHM_t): STATUS_t*

«enumeration»
SHA_t

SHA1
SHA224
SHA256
SHA512

«enumeration»
ALGORITHM_t

RSA_PKC1_V1_5
RSA_PSS
ECDSA

«struct»
key_t

modulus: uint8_t *
exponent: uint8_t *
modulus_len: size_t
exponent_len: size_t

BOOTLOADER

«enumeration»
COMMAND_t

RESET
STATE
BOOTLOADER_START
BOOTLOADER_WRITE
BOOTLOADER_DONE
ACK
ERROR

«enumeration»
STATE_t

VERIFICATION
CONFIRMATION
UPDATE_PREPARATION
UPDATE_RECEPTION
UPDATE_VALIDATION
UPDATE_INSTALLATION

«abstract»
Bootloader

-state: STATE_t
-is_ready_to_boot: STATUS_t

+checkFirmwareIntegrity(): STATUS_t
+checkNewFirmwareSignature(): STATUS_t
+checkNewFirmwareVersion(): STATUS_t
+receiveNewFirmware(): STATUS_t
+hasToUpdate(): STATUS_t
+getState(): STATE_t
+setState(state: STATE_t): STATUS_t
+getIsReadyToBoot(): STATUS_t
+setIsReadyToBoot(status: STATUS_t): STATUS_t
+launchFirmware(): void

«enumeration»
STATUS_t

FAIL
SUCCESS

ARMAZENAMENTO SEGURO

«enumeration»
ASSET_t

+PUBLIC_EXPONENT
+MODULUS
+FIRMWARE_HASH
+FIRMWARE_SIGNATURE
+FIRMWARE_VERSION
+NEW_FIRMWARE_HASH
+NEW_FIRMWARE_SIGNATURE
+NEW_FIRMWARE_VERSION
+FIRMWARE_SIZE
+NEW_FIRMWARE_SIZE
+UPDATE_REQUEST

«abstract»
SecureStorage

*+readData(address: DATA_t, out out: uint8_t *, asset: ASSET_t): STATUS_t*
+writeData(address: DATA_t, data: DATA_t): STATUS_t
+eraseData(address: DATA_t): STATUS_t

GERÊNCIA DE MEMÓRIA

«union»
WORD_t

word_8: uint8_t
word_16: uint16_t
word_32: uint32_t
word_64: uint64_t

«struct»
DATA_t

word: WORD_t

«abstract»
FlashManager

+flashWrite(address: DATA_t, word: DATA_t): STATUS_t
+flashErase(address: DATA_t): STATUS_t
+flashLock(): STATUS_t
+flashUnlock(): STATUS_t
+flashReadAll(address: DATA_t, out: uint8_t *, out_len: size_t): STATUS_t
+isFlashLocked(): bool
+setFlashLocked(flash_locked: bool): void

COMUNICAÇÃO

«abstract»
Comm

-buffer: uint8_t *
-has_data: bool

*+sendData(buffer_out: uint8_t *, buffer_len: size_t): size_t*
*+getData(out bufer_in: uint8_t *, buffer_len: size_t): size_t*
+hasData(): bool

TROCA DE CONTEXTO

«abstract»
Switcher

+goToSecure()
+goToNonSecure()