Hello and welcome to this presentation of the STM32H5 public key accelerator, widely used for asymmetric cryptography applications.

PKA features

- Acceleration of asymmetric cryptography, up to 4160 bits for RSA/DH and 640 bits for elliptic curves
  - Used in NIST FIPS186-4, RSA PKCS#1, ANSI X9.62, IETF RFC5639 (Brainpool), Chinese SM2 and SEC2 curves
- Side channel protection for operations manipulating secrets
  - RSA / DSA private modular exponentiation
  - ECC scalar multiplication, signature generation
- Operations not manipulating secrets are also supported
  - RSA / DSA public modular exponentiation and its faster CRT (Chinese Remainder Theorem) version
  - ECDSA signature verification
  - ECC point on curve check, complete addition, double base ladder & projective to affine
  - Arithmetic and modular operations like addition, subtraction, multiplication, comparison, reduction…

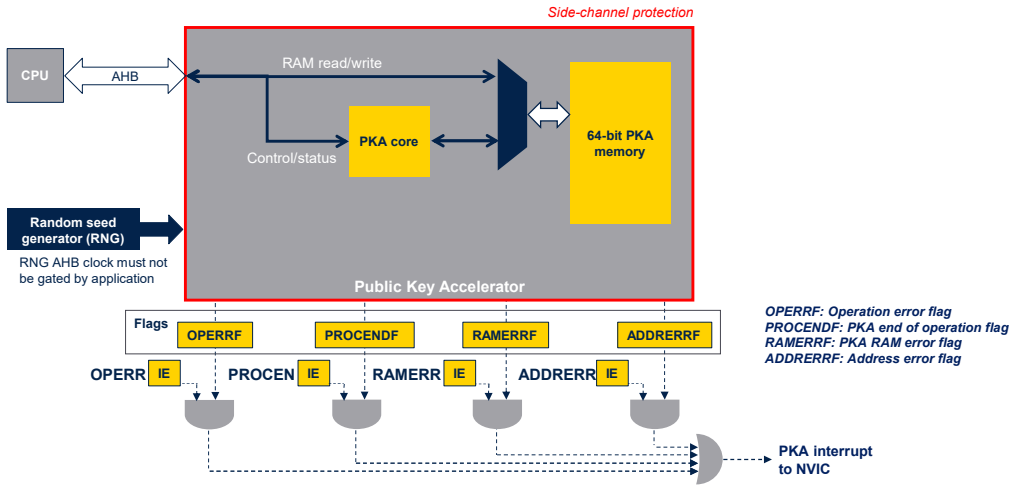
life.augmented

Public Key cryptography is part of many security standards and is widely used to establish secure communication channels across unsecure open networks like the Internet or to provide authentication via electronic signatures. Software-only solutions can be too slow for real-time applications, impacting the system's overall performance.

The PKA module is an efficient hardware accelerator that speeds up the public key cryptography operations performed by the CPU. It accelerates Rivest, Shamir and Adleman (RSA), Diffie-Hellman (DH) as well as Elliptic Curve Cryptography (ECC) over prime field operations. Supported operand sizes are up to 4160 bits for RSA and DH, and up to 640 bits for ECC. Binary curves, Edwards curves and Curve25519 are not supported by the PKA.

The list of supported operations are described here. Modular exponentiation for RSA decryption, scalar multiplication and signature for ECC are protected against side-channel attacks. Those operations are used when manipulating secret keys.

PKA features (2/2)



life.augmented

PKA lightens the CPU workload by performing key operations in the PKA core, using dedicated PKA memory. First the CPU loads initial data into the PKA internal RAM, which is located at address offset 0x400. Then in the PKA control register, the CPU specifies the operation which is to be executed and finally asserts the START bit. Once the PKA reports the end of operation (PROCENDF), the CPU reads the resulting data from the PKA RAM, then clears the PROCENDF flag.
Software can abort a PKA operation at any time by clearing the EN bit in the PKA_CR register. In this case, the content of the PKA memory is not guaranteed, except for side-channel protected operation for which PKA memory is guaranteed erased.

The PKA has three error flags: the Operation error flag

(OPERRF), the Address Error flag (ADDRERRF) and the RAM Error Flag (RAMERRF). All flags can generate an interrupt if the corresponding Interrupt Enable bit is set (OPERRIE, PROCENDIE, ADDRERRIE or RAMERRIE). When the PKA peripheral reset signal is released, the PKA RAM is cleared automatically, taking 667 clock cycles. During this time the setting of the EN bit in PKA_CR is ignored.

PKA side-channel protected operations (modular exponentiation for RSA decryption, scalar multiplication and signature for ECC) manages secrets that are automatically erased from PKA RAM at the end of the operation.

PKA processing time @160MHz

- Modular exponentiation operation (in milliseconds) (side-channel attacks protected)

Exponent length (in bits)		Operand length (in bits)		
		1024	2048	3072
Public	3	0.8	3	4.3
Private	1024	60, 36 or 11 (CRT)	-	-
	2048	-	400, 260 or 73 (CRT)	-
	3072	-	-	1246, 852 or 230 (CRT)

Note: CRT is Chinese Remainder Theorem optimization

- Other operations in ms (DPA resistant)

	Modulus length (in bits)			
	256	384	512	521
ECC scalar multiplication	19	53	111	131
ECDSA signature	17	45	90	104
ECDSA verification	18	50	105	120

life.augmented

4

Here are the modular exponentiation processing times using different exponent and operand sizes. Other important operations like ECC scalar multiplication, and ECDSA signature/ verification are also mentioned. Values are computed with the clock frequency defined on the slide. Operations protected against side-channel attacks are also highlighted.

Thank you

life.augmented

Thank you for attending this presentation.