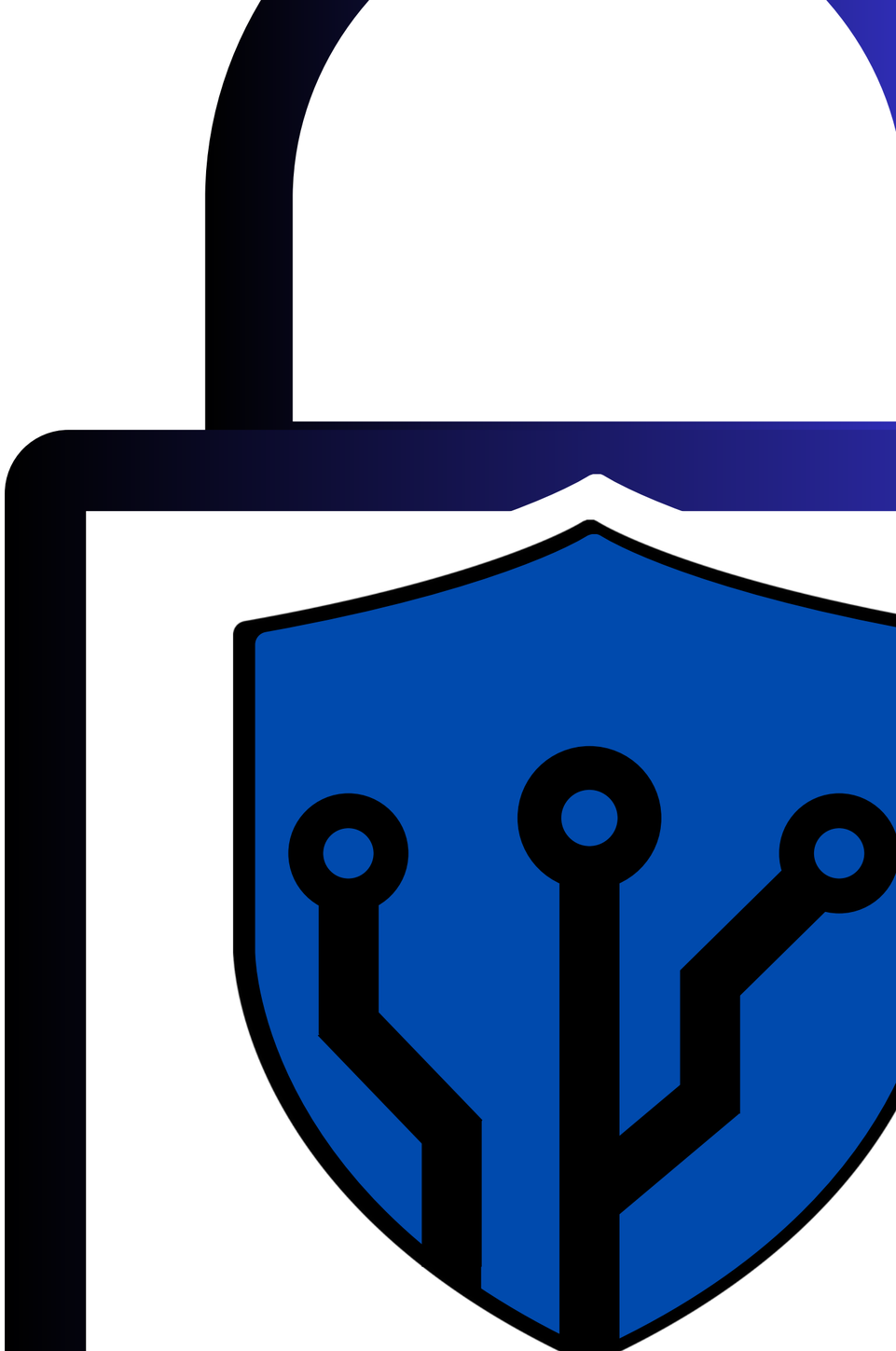
Reconfigurable HSMs: Future-Proofing Hardware Security Against Evolving Threats

Pablo Trujillo

CONTROLPATHS ENGINEERING



Pablo Trujillo

FPGA & Embedded systems consultant

I write on internet

Bringing FPGA where they are unknown.

@controlpaths
pablo@controlpaths.com



Privacy protectors

How they work?



HASH algorithms

MD5
SHA1/2/3



Keys

RSA
Kyber
AES

Privacy protectors

Hash algorithms



“Hello world”

08688e9f8d99dad67ed...

Privacy protectors

Hash algorithms

MD5

SHA0

SHA1

SHA2

SHA3

Privacy protectors

Hash algorithms

~~MD5~~

2004

Collision

~~SHA0~~

1998

Collision

~~SHA1~~

2011

Collision

SHA2



SHA3



Privacy protectors

Hash algorithms

MD5	SHA0	SHA1	SHA2	SHA3
2004	1998	2011		
Colisión	Colisión	Colisión		

Privacy protectors

Keys

Use a key to encrypt data

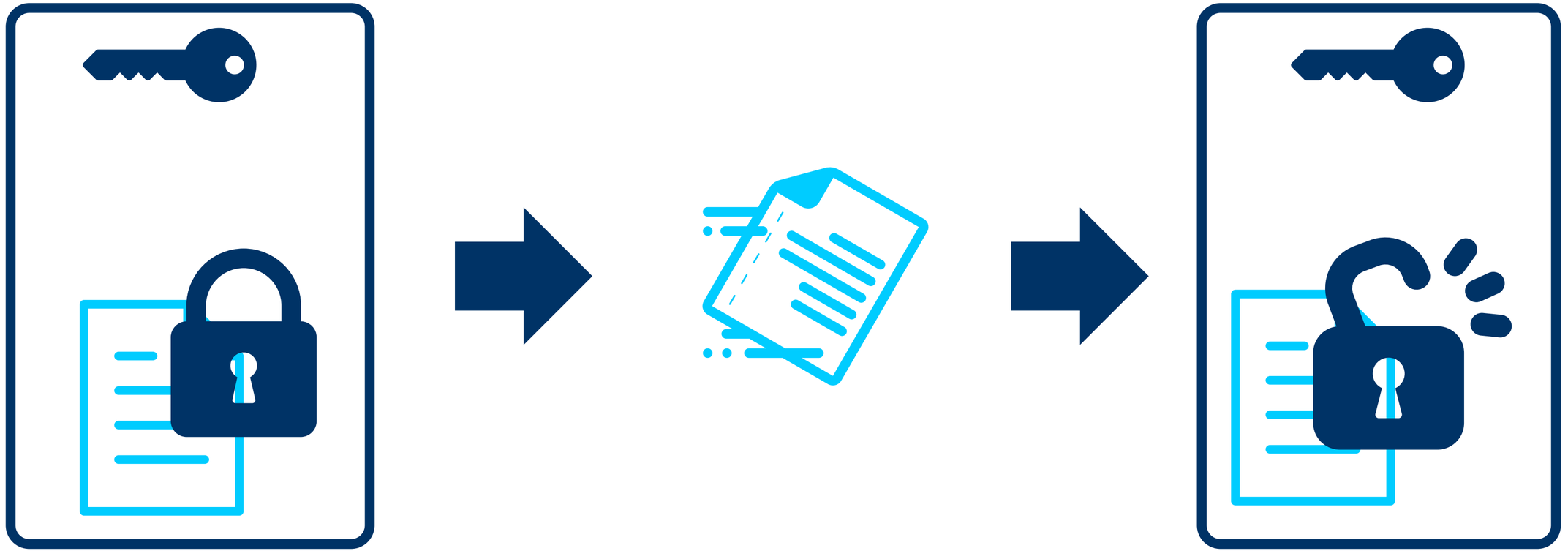
Only user who have the key can decrypt the data

Data can be a stored file, or a real-time communications (TLS, SSH..)

Two main types:
- _symmetric
- _assymmetric

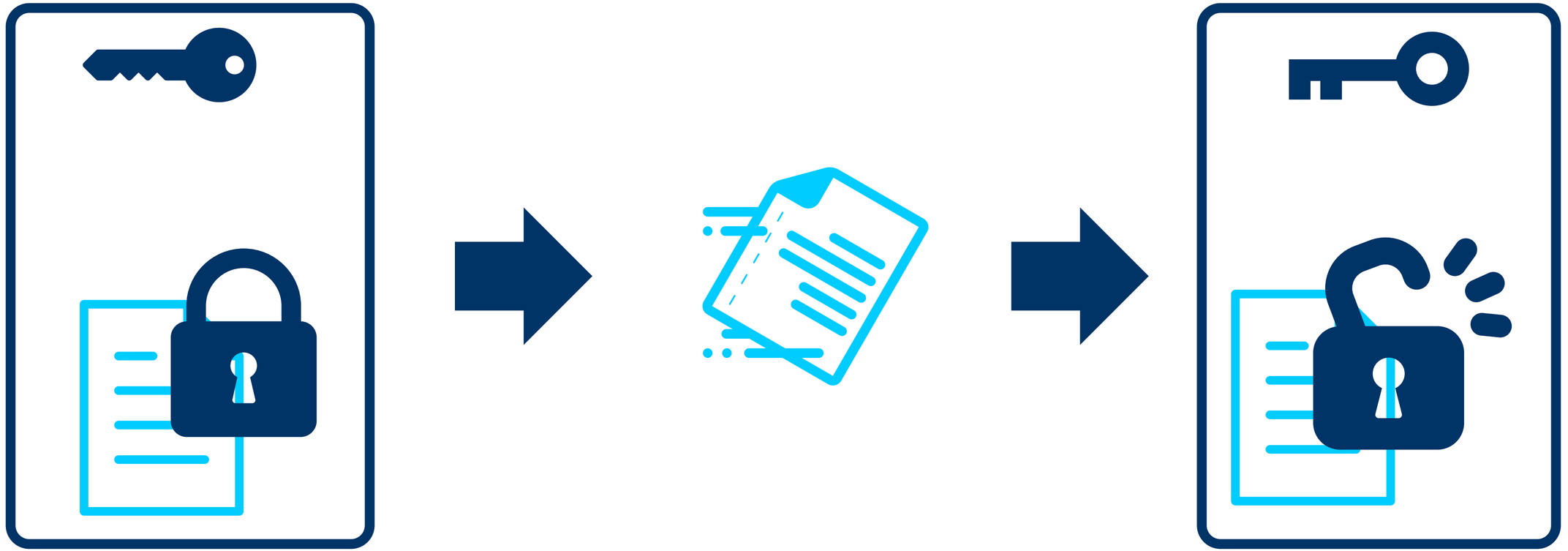
Privacy protectors

Keys



Privacy protectors

Keys



Ana

Bart

Privacy protectors

It is not magic, it’s just maths (and logic)

Rijndael, Diffie-Hellman, ChaCha20, Salsa20, …

Based on mathematical and logic operations

Performance is related to how fast they can be calculated

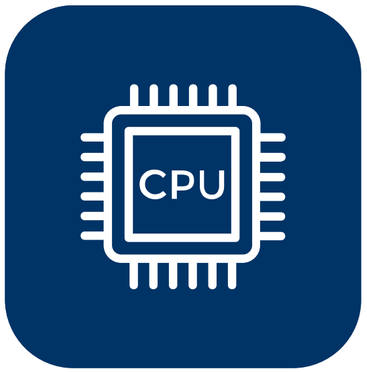
Not just algorithms, but also operating modes

Critical in real-time ciphers (SSH, TLS …)

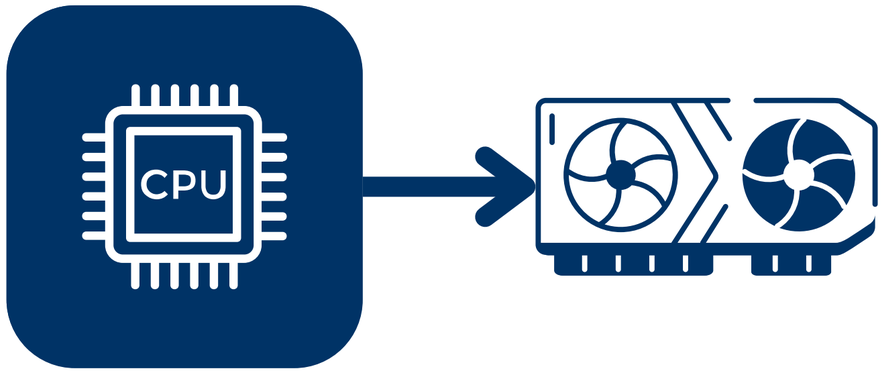
Privacy protectors

Implementation

Software



Hardware



Privacy protectors

SW Implementation

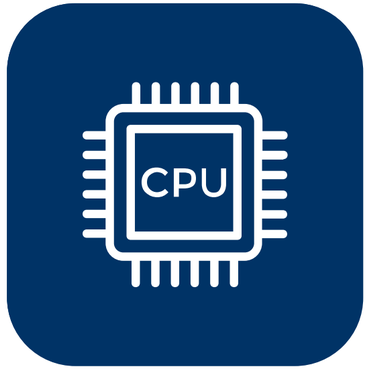
The implementation of the algorithm is made by code.

Can be accelerated with processor extensions.

Keys are stored in the hard disk.

Easy to update.

Software



Privacy protectors

SW Implementation

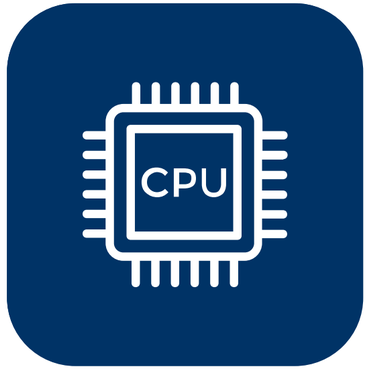
The implementation of the algorithm is made by code. **(Poor performance)**

Can be accelerated with processor extensions. **(Depends on the CPU used)**

Keys are stored in the hard disk. **(Not secure)**

Easy to update. **(Not secure)**

Software



Privacy protectors

HW Implementation

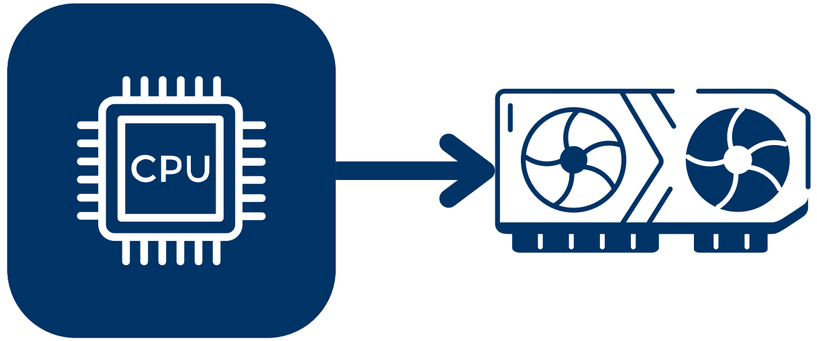
Uses external device to execute the algorithms.

Allow secure storage for keys.

Standarize the algorithms used.

Eliminate the dependency on the CPU.

Hardware



Privacy protectors

HW Implementation

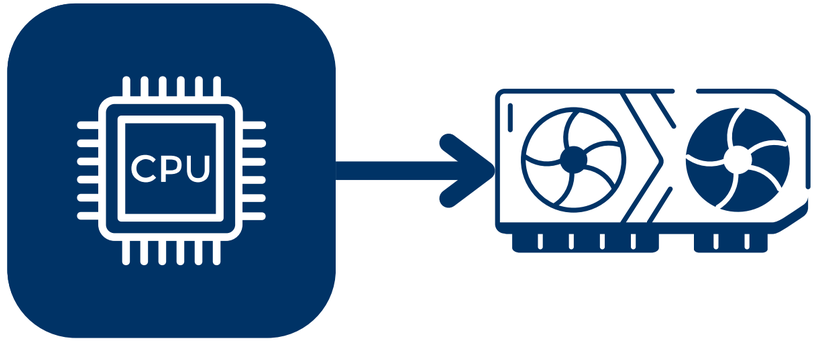
Uses external device to execute the algorithms **(Other processor)**.

Allow secure storage for keys. **(Good)**

Standarize the algorithms used. **(Fixed algorithms)**

Eliminate the dependency on the CPU. **(Good)**

Hardware



Privacy protectors

Trusted Platform Modules (TPM)

Cryptographic coprocessor usually integrated in the mother board.

Add secure storage for keys.

Includes a set of cryptographic functions.

Two important versions: TPM1.2 and TPM2.0

Privacy protectors

Trusted Platform Modules (TPM)

The TPM 2.0 specification allows for a number of different types of hash algorithms—SHA-1, SHA-256, and SHA-384 are just some of them. Typically, TPMs implement only a few of the allowed hashes. One problem that vexed the developers for a long time was how to integrate multiple hash algorithms (which are used to maintain integrity) if one of those hash algorithms was later broken. This is harder than it sounds, because usually hash algorithms themselves are used to provide integrity to reports, and if the hash algorithm can't be trusted, how can you trust a report of which hash algorithm is being used? The design that was chosen managed to avoid this problem: tags are used throughout the design in data elements that identify the hash algorithms used.

https://trustedcomputinggroup.org/resource/a-practical-guide-to-tpm-2-0/

The quantum threat

What?

New computational model based on quantum physics.

Unlike traditional bits, that can take values of 1 and 0, quantum bits (qubits) can take 1, 0 or any other value between them (superposition principle)

Allow to process **multiple choices** at the same time

Allow the execution of quantum algorithms

The quantum threat

Who

Grover's algorithm

Uses the quantum superposition principle

Used for finding elements within an unordered list

$$O(N) \rightarrow O(\sqrt{N})$$

The quantum threat

Who

Shor's algorithm

Used for finding the prime factors of an integer N

$$O(N) \rightarrow O(logN^3)$$

The quantum threat

	Old school	Q school	
SHA256	2^{256}	2^{128}	*Grover*
SHA512	2^{512}	2^{256}	*Grover*
SHA3-256	2^{256}	2^{256}	
AES 128 bits	2^{128}	2^{64}	*Grover*
AES 256 bits	2^{256}	2^{128}	*Grover*
RSA	2^{2048}	$\log(n)^3$	*Shor*

The quantum threat

	Old school	Q school	
SHA256	2^{256}	2^{128} ⚠	*Grover*
SHA512	2^{512}	2^{256} ✓	*Grover*
SHA3-256	2^{256}	2^{256} ✓	
AES 128 bits	2^{128}	2^{64} ✗	*Grover*
AES 256 bits	2^{256}	2^{128} ✓	*Grover*
RSA	2^{2048}	$\log(n)^3$ ✗	*Shor*

Privacy protectors

Trusted Platform Modules (TPM)

Parametrics	OPTIGA TPM SLB 9673 FW26
Additional Information	Version 2.0 Rev. 01.59
Ambient Temperature min max	-40 °C 85 °C ; -40 °C 105 °C
Applications	trusted computing
Asymmetric Cryptography	RSA1024 ; RSA2048 ; RSA3072 ; RSA4096 ; ECC NIST P256 ; ECC BN256 ; ECC NIST P384
CPU	32-bit
Certifications	CC EAL4+ ; FIPS 140-2 level 2
Delivery Forms	UQFN-32
Interfaces	I2C
NVM / 凌捷掩膜	51 kByte
Package	UQFN-32
Product Description	I2C TPM for embedded systems ; FW26.xx
Symmetric Cryptography	HMAC ; SHA-1 ; SHA2-256 ; SHA2-384 ; AES-128 ; AES-192 ; AES-256
Use Cases	Embedded TPM



400 kbps

No SHA2-512 or SHA3

Reconfigurability as a solution

Devices able to **reconfigure its hardware** in real-time

Allow partial reconfigurability

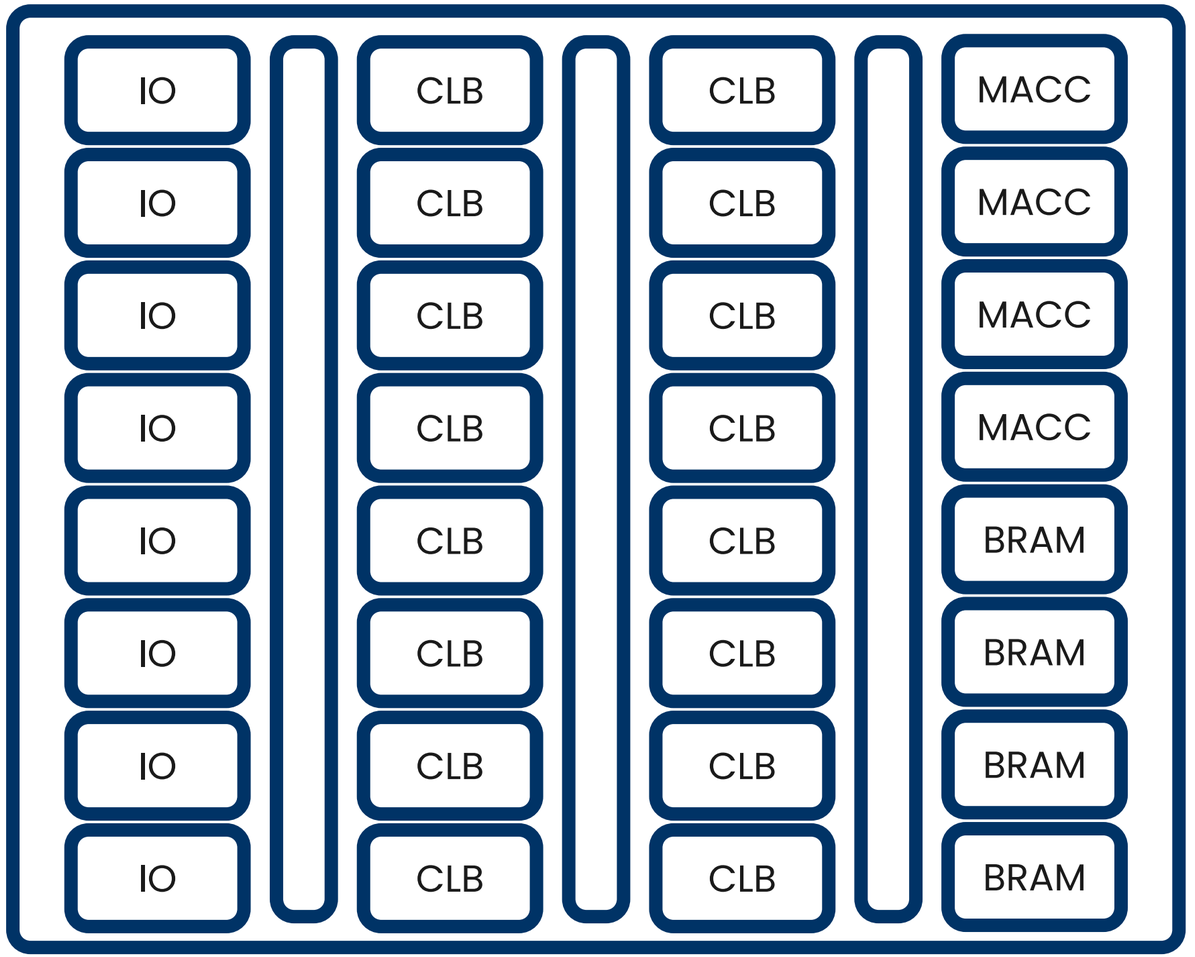
Able to be updated once deployed

Keep all the benefits of HW implementations

Based on FPGA devices

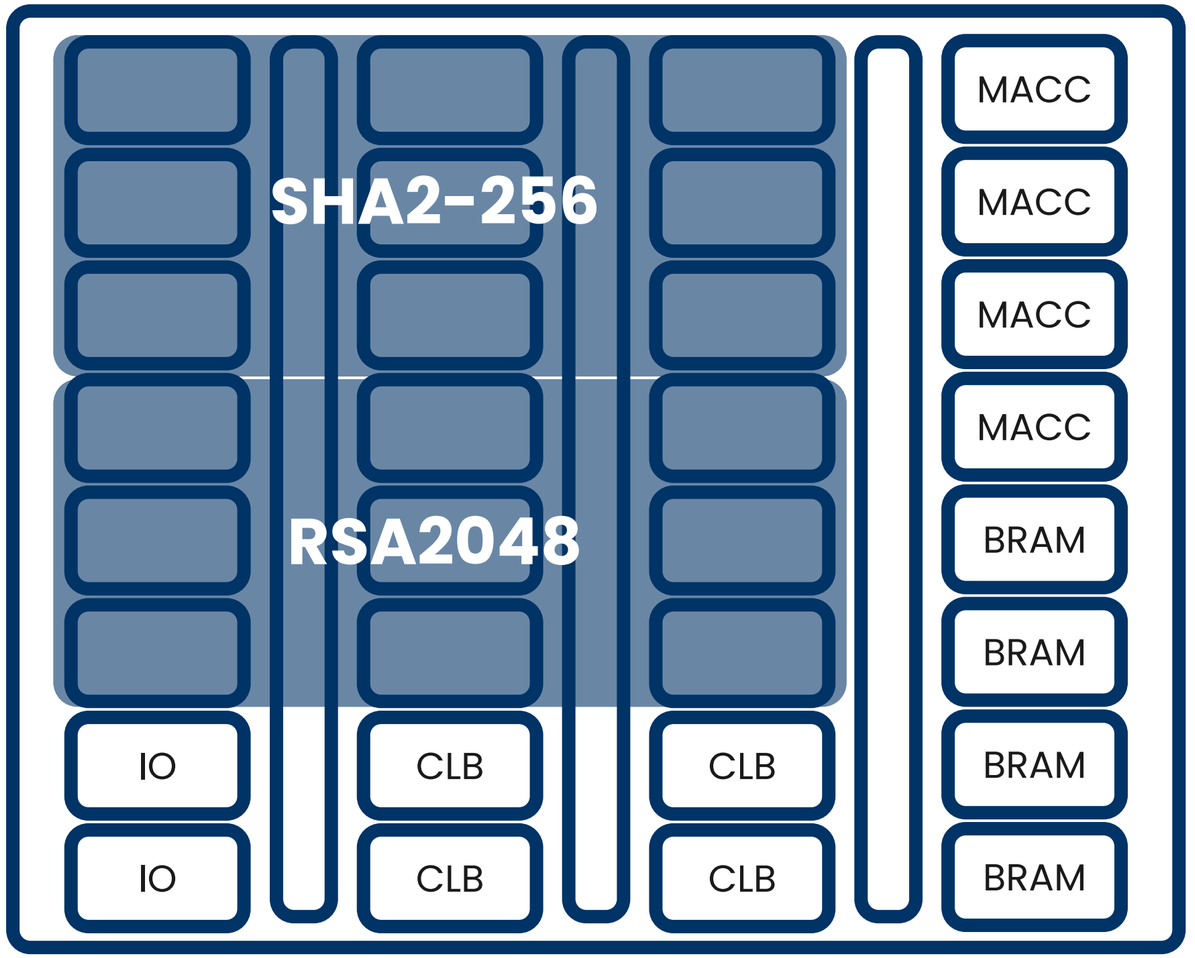
Reconfigurable devices

FPGA



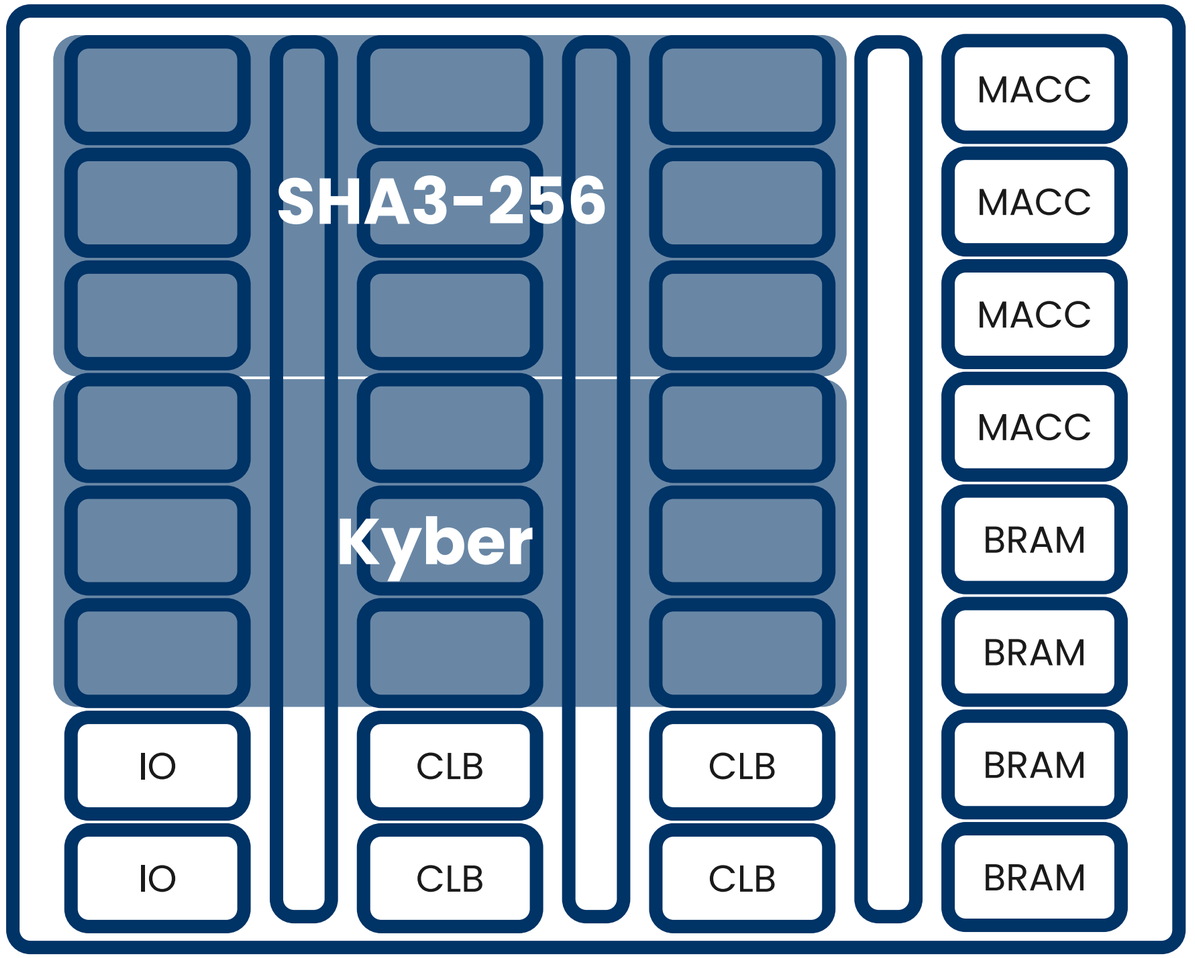
Reconfigurable devices

FPGA



Reconfigurable devices

FPGA



CryptoLite project

FPGA-based HSM

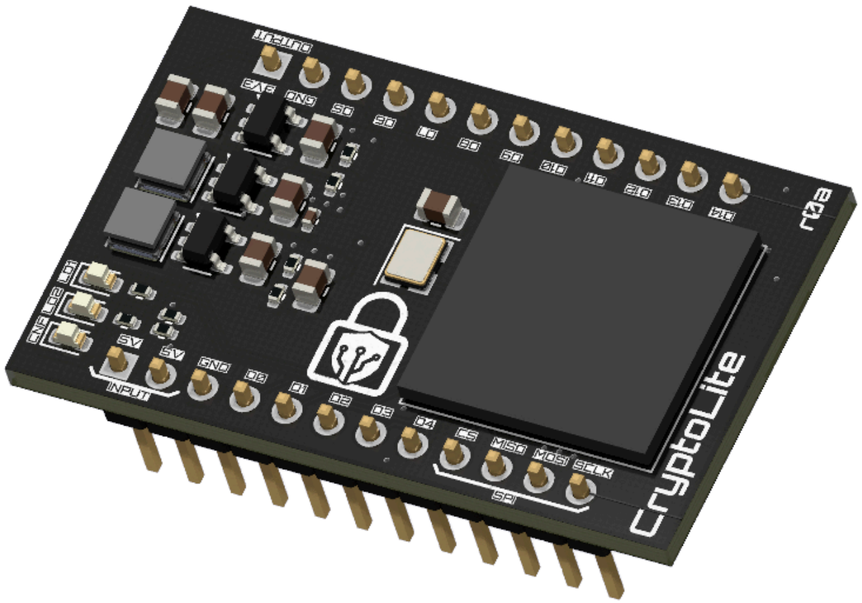
Hardware Secure Module (HSM)

Based on AMD Spartan7 FPGA

Open-Source

SPI interface (50Mbps)

Remote update



/controlpaths/cryptolite

FPGA-based HSM

Advantages

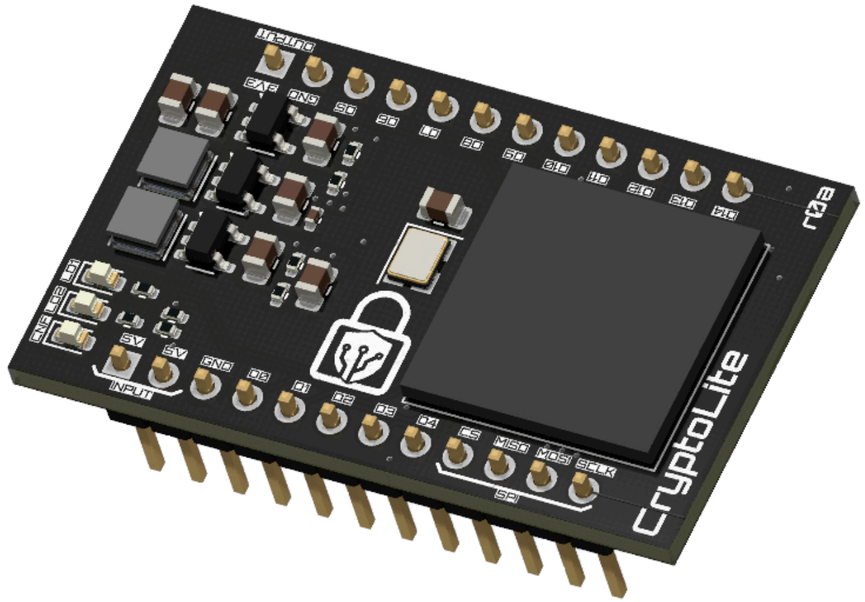
Reconfigurability that allows to update the algorithms

Avoid/confuse side-channel and timing attacks

Able to generate True-Randon numbers.

Multiple algorithms being executed at the same time

Parallelizable operation modes can be used (counter mode CTR)



FPGA-based HSM

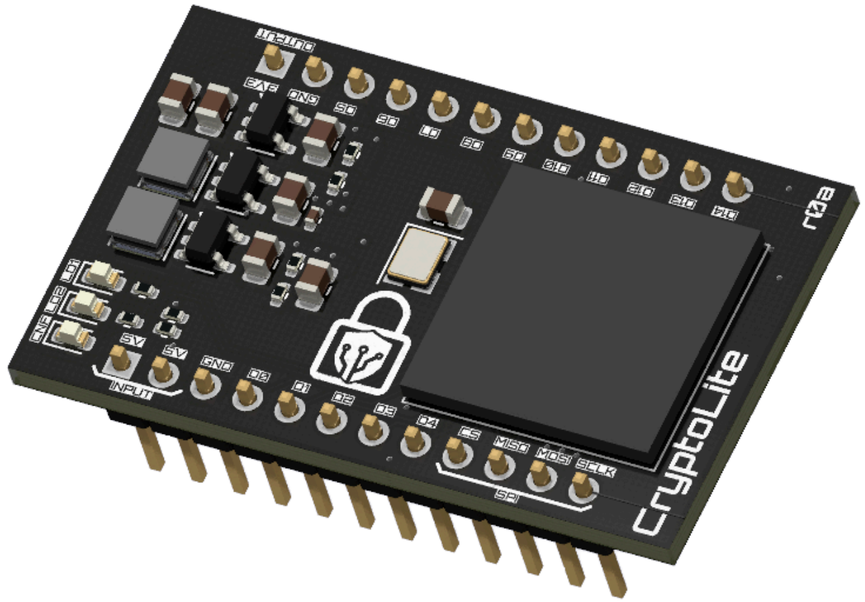
Possible disadvantages

FPGAs are primarily RAM-based.

Designs typically require external memory.

Usually do not incorporate non-volatile memory.

Tend to have higher power consumption compared to microcontrollers (MCUs).



FPGA-based HSM

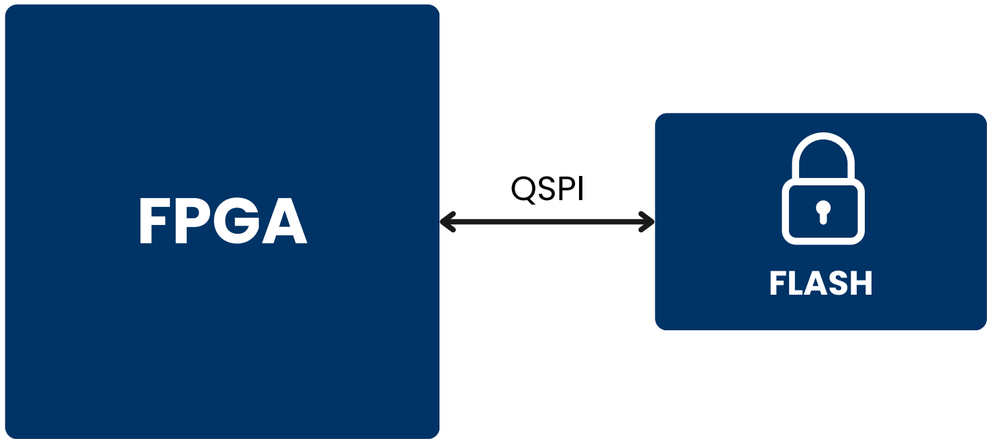
Bitstream encryption

Field Programmable Gate Array (FPGA) designs are stored in a bitstream format.

Bitstreams can be encrypted using AES-256 encryption.

JTAG interface is disabled to enhance security.

The encryption key is generated during the manufacturing process.



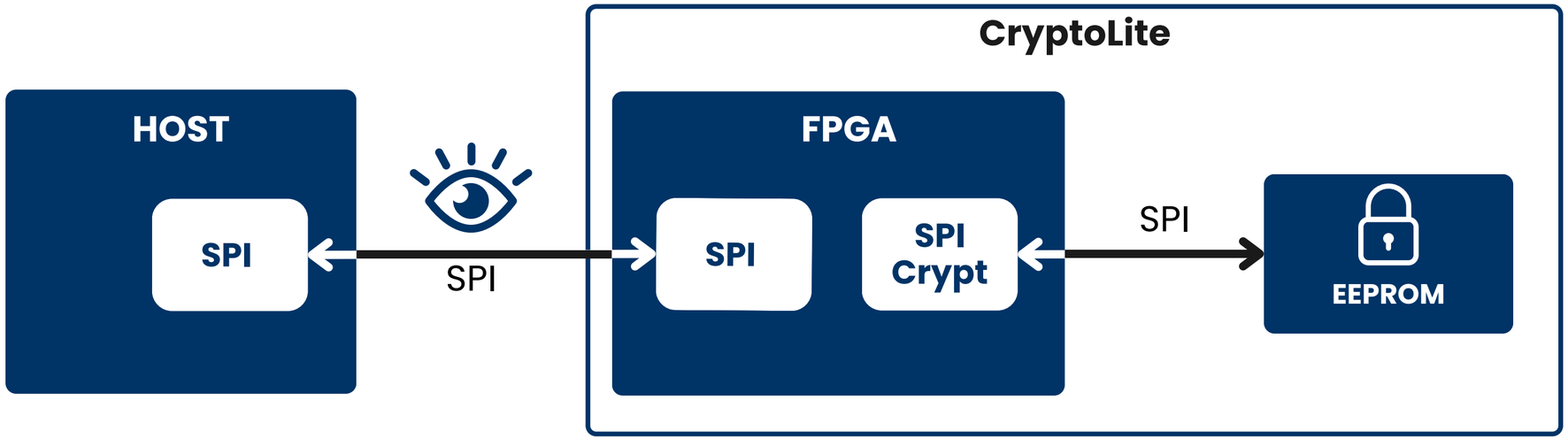
FPGA-based HSM

Key storage

Key storage in a 1Mb encrypted external memory.

Encryption key stored in fuses in the FPGA.

Encryption key cannot be read.



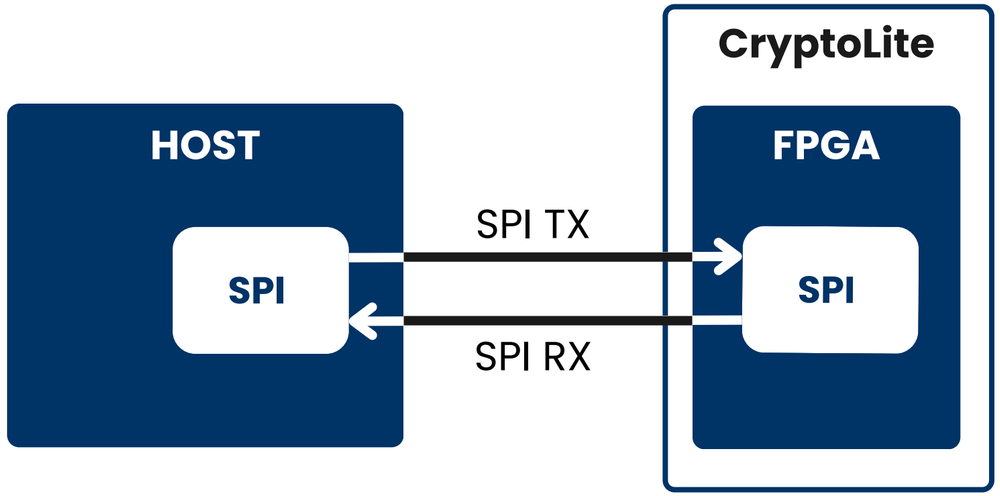
FPGA-based HSM

Real-time cypher

SPI is a full-duplex interface (unlike I2C)

Higher bitrates than I2C (50 Mbps)

Improved performance using processing modes like CTR

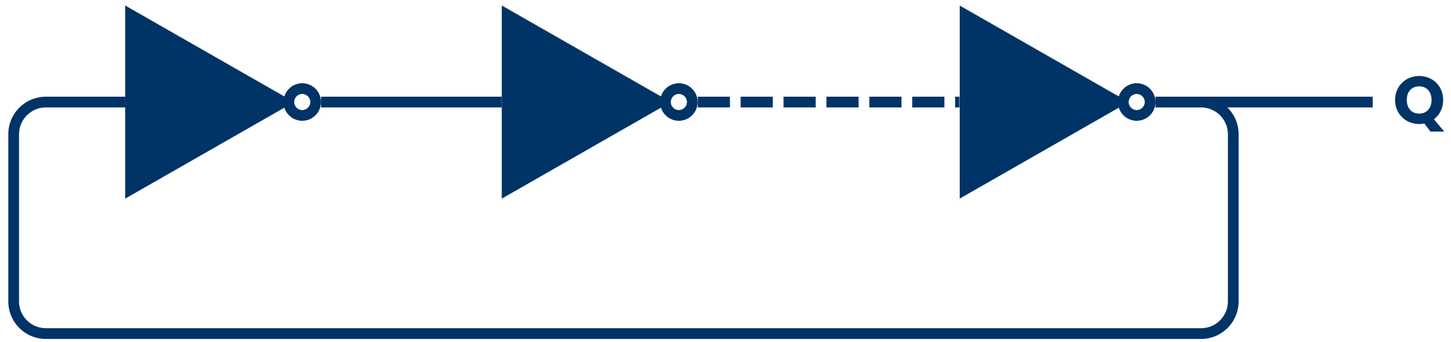


FPGA-based HSM

True randon numbers generation (TRNG)

FPGA designs works at bit-level

Asynchronous designs (latches) generate meta-stable states that takes random values according the silicon purity of the wafer, temperature, ...



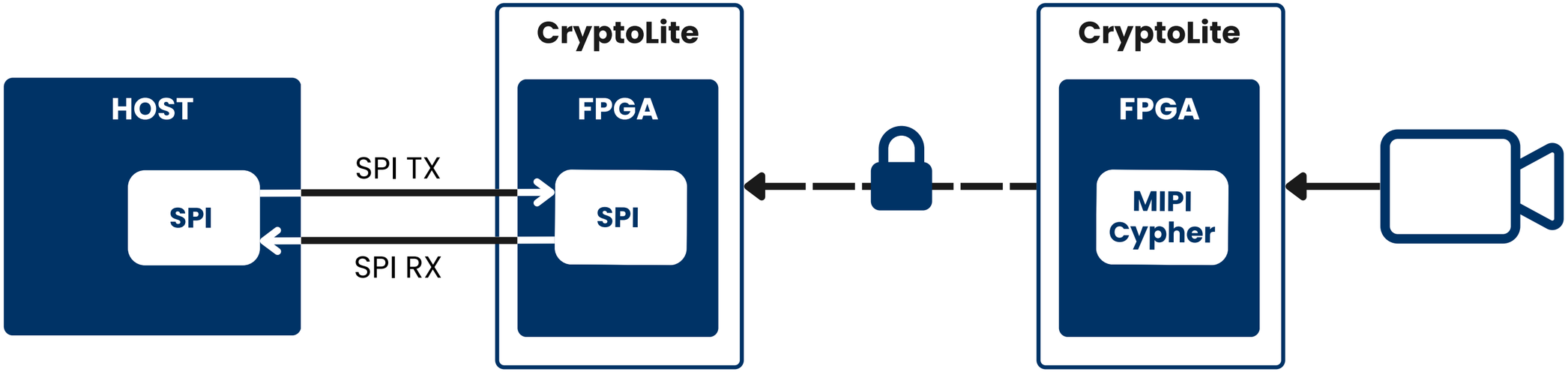
Ring oscillator

FPGA-based HSM

Custom solutions

Facilitate the integration of custom solutions within the HSM.

This includes the implementation of custom ciphering schemas.



Reconfigurable HSMs: Future-Proofing Hardware Security Against Evolving Threats

Pablo Trujillo

/controlpaths

@controlpaths

pablo-trujillo-juan

CONTROLPATHS ENGINEERING

