RÉPUBLIQUE FRANÇAISE

Liberté
Égalité
Fraternité

Ultrablue

User-friendly Lightweight TPM Remote Attestation over BLUEtooth

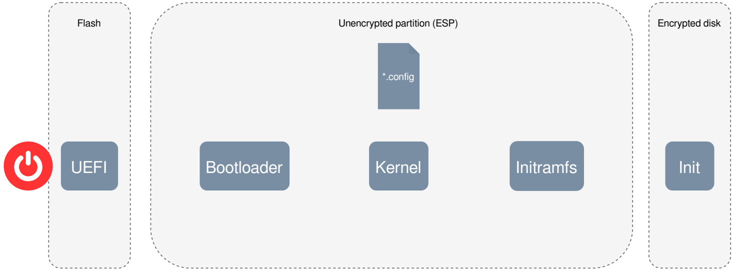
Nicolas Bouchinet, Gabriel Kerneis, Loïc Buckwell

Agence nationale de la sécurité des systèmes d’information

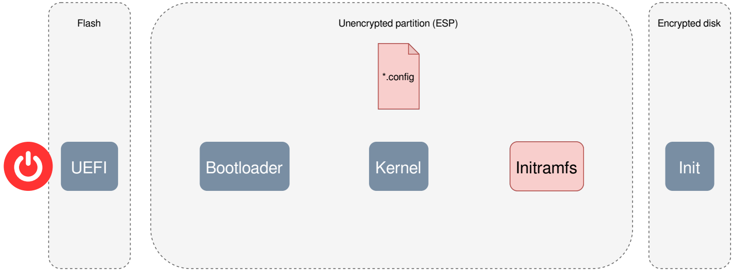
Jully 3-5 – Pass The Salt 2023

Attack Scenario

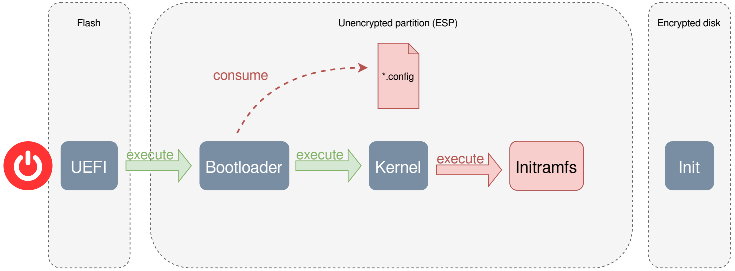
Bootchain


Flash
Unencrypted partition (ESP)
Encrypted disk
*.config
UEFI
Bootloader
Kernel
Initramfs
Init

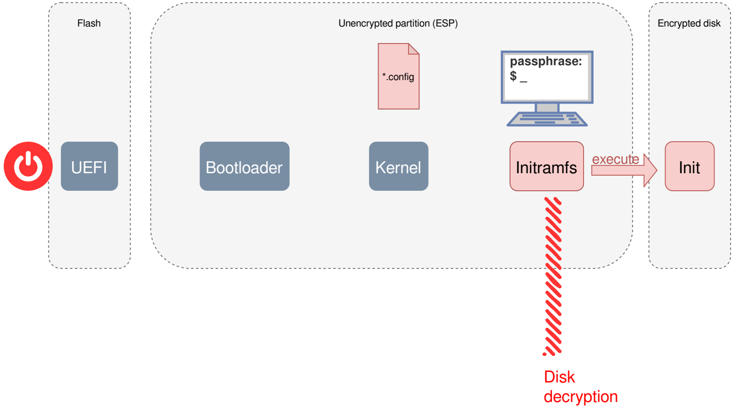
Bootchain


Flash
Unencrypted partition (ESP)
Encrypted disk
*.config
UEFI
Bootloader
Kernel
Initramfs
Init

Bootchain


Flash
Unencrypted partition (ESP)
Encrypted disk
consume
*.config
UEFI
execute
Bootloader
execute
Kernel
execute
Initramfs
Init

Bootchain


Flash
Unencrypted partition (ESP)
Encrypted disk
*.config
passphrase:
$ _
UEFI
Bootloader
Kernel
Initramfs
execute
Init
Disk decryption

What is a TPM (Trusted Platform Module)?

- Passive component responding to commands
- Cryptographic operations : signature, encryption, **hash**
- Data storage : **24 PCR registers** and NVRAM

PCR Extension

PCR(n)$_{t+1}$ = **sha256**(PCR(n)$_t$ | **sha256**(blob))

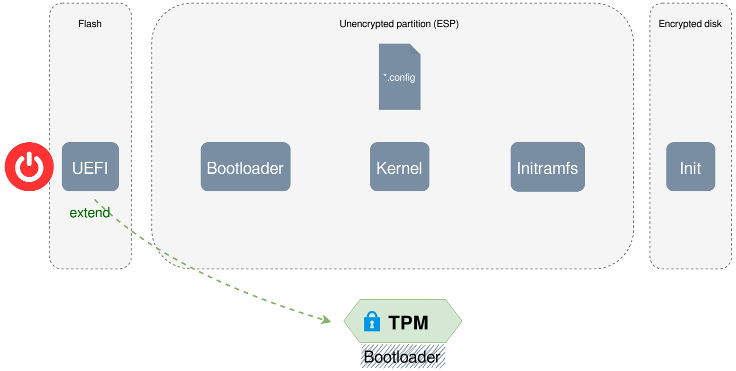
Sealing and Unsealing

Data encryption (resp. decryption) gated by **authorization values** based on an Internal TPM Key and PCR hashes.

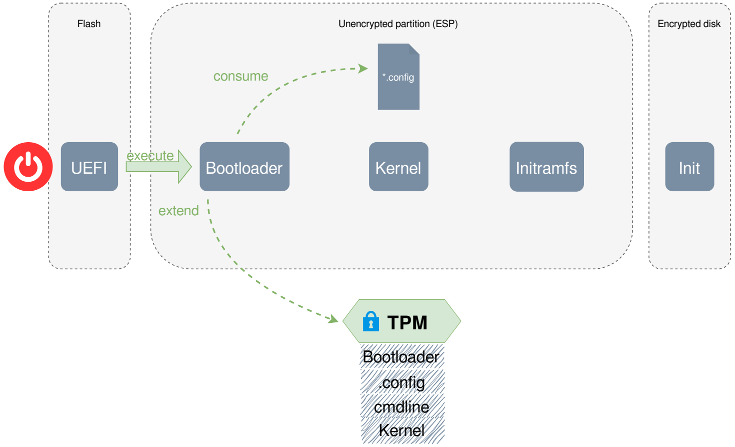
How can we use a TPM to protect our bootchain ?

- Use the **TPM extension operations** before any step of the bootchain.
- Protect our **LUKS encryption key** using the **TPM sealing operation** based on a specific PCR policy.

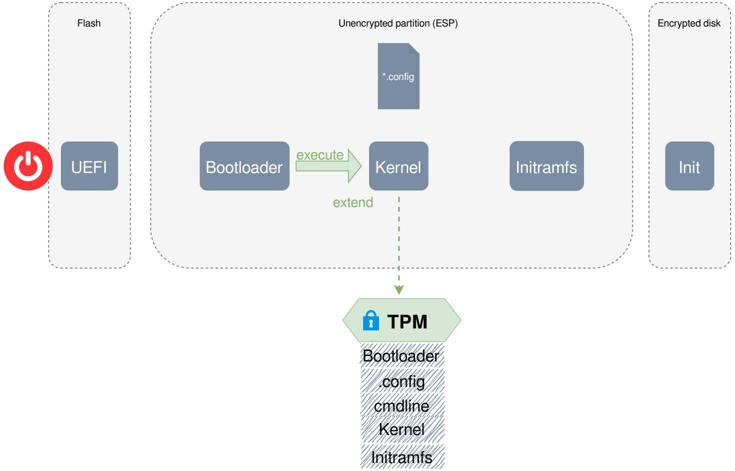
Bootchain with a TPM



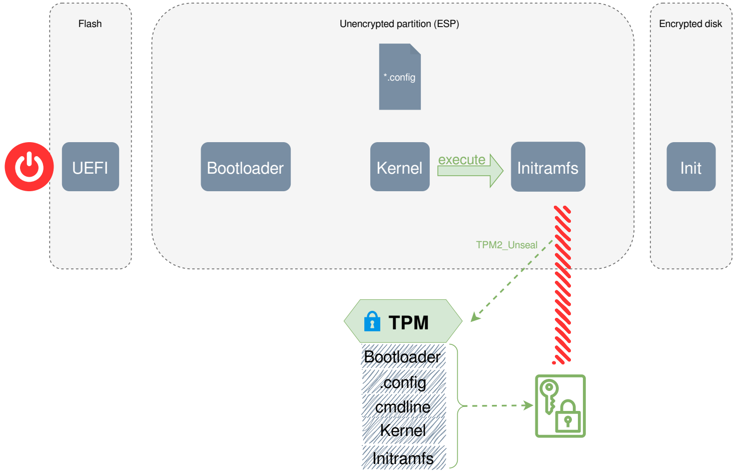
Bootchain with a TPM



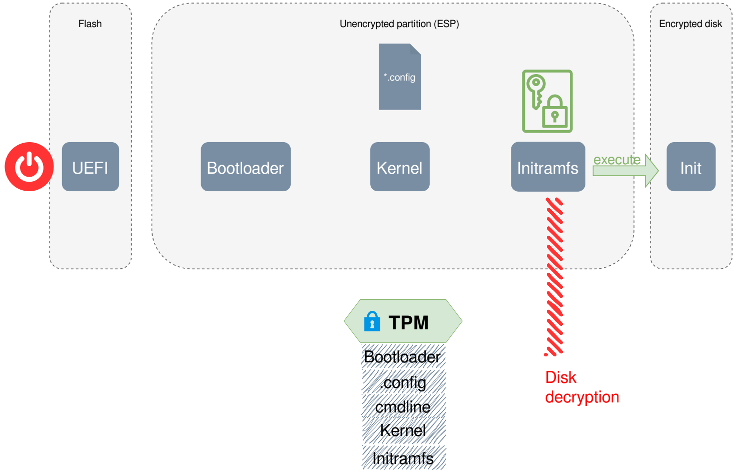
Bootchain with a TPM



Bootchain with a TPM



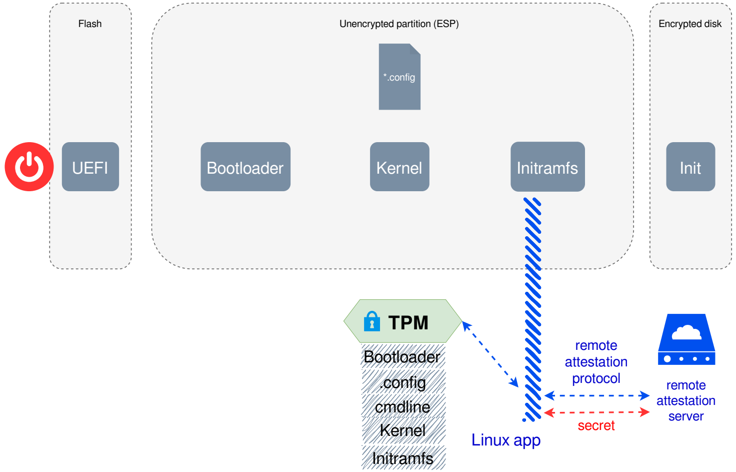
Bootchain with a TPM



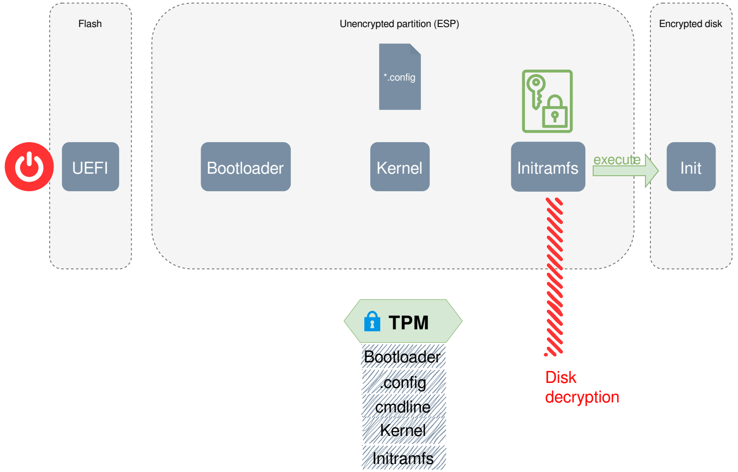
What’s wrong with PCR-based disk encryption?

- **“PCR brittleness”** makes updates hard.
- **Missing debug information** for the end user.
- All the secrets are **stored locally**.

Remote attestation



Remote attestation



Remote attestation protocol

Goals

What we want from **remote attestation**:

- A **log** of every significant boot event
- An attestation that this log was **not tampered with**
- An attestation that it is **signed by the TPM** associated with our machine
- Bonus: an **encrypted communication** channel

TPM event log

A log of **significant boot events**, stored in RAM:

```
- PCRIndex:   0
  EventType:  EV_EFI_PLATFORM_FIRMWARE_BLOB
  Event:      BlobBase: 0xff970000    BlobLength: 0x3c0000
  Digest: "4b22c4bc249046bf9c08fd4a443ee858080e3588e6bff5374f215ba42c387c3d"
- PCRIndex:   0
  EventType:  EV_POST_CODE
  Event:      ACPI DATA
  Digest: "bfc46da9ef25182f848dd38b96728eaa41409bb3c7c8db4b6b2e4019dbd1a107"
- PCRIndex:   7
  EventType:  EV_EFI_VARIABLE_DRIVER_CONFIG
  Event:      VariableName: 8be4df61-93ca-11d2-aa0d-00e098032b8c
              UnicodeName: SecureBoot         Enabled: 'Yes'
  Digest: "ccfc4bb32888a345bc8aeadaba552b627d99348c767681ab3141f5b01e40a40e"
```

Goals

What we want from **remote attestation**:

- ✓A **log** of every significant boot event
- An attestation that this log was **not tampered with**
- An attestation that it is **signed by the TPM** associated with our machine
- Bonus: an **encrypted communication** channel

TPM event log integrity

- The event log is **stored in RAM**, which can be tampered with.
- But every event is hashed to a digest, which is extended into a PCR.
- **Replaying** the event log, we get expected **PCR values**.

Signing PCR values: TPM Quote and Attestation Key

TPM Quote

The TPM command `TPM2_Quote(AK, nonce, PCRselect)`:

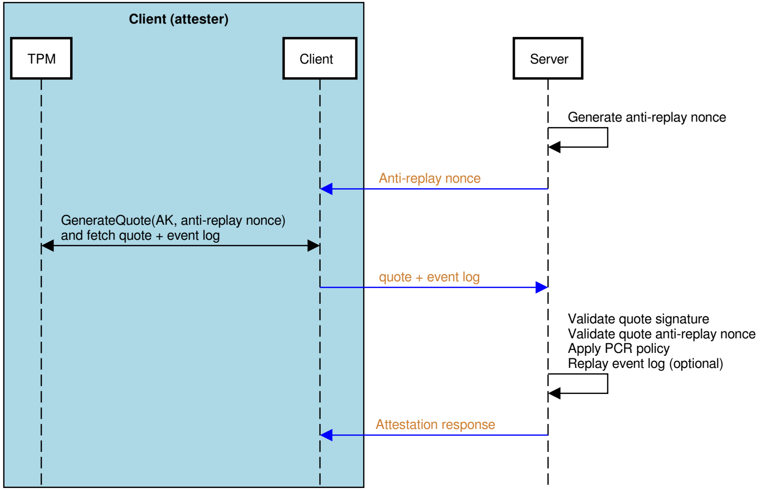
- reads selected PCR values, hashes them,
- appends the provided nonce (anti-replay),
- signs the result with an **Attestation Key**.

Attestation Key (AK)

An **attestation key** (AK) is:

- a **signing** key (cannot encrypt),
- a **fixed**, **sensitive** key (generated on TPM, cannot be exported),
- a **restricted** key (can only sign internal TPM data, eg. a quote).

Protocol excerpt - attestation


Client (attester)
TPM
Client
Server
Generate anti-replay nonce
Anti-replay nonce
GenerateQuote(AK, anti-replay nonce)
and fetch quote + event log
quote + event log
Validate quote signature
Validate quote anti-replay nonce
Apply PCR policy
Replay event log (optional)
Attestation response

Verifying a quote

Things to check to verify a given blob is a valid quote:

- the AK **signature**,
- the **magic value** indicating that the blob was generated on a TPM,
- the **type** indicating it is a quote,
- that the **nonce** matches the server-generated challenge,
- that the **PCR digest** matches expected PCR values from the event log replay.

Goals

What we want from **remote attestation**:

- ✓A **log** of every significant boot event
- ✓An attestation that this log was **not tampered with**
- An attestation that it is **signed by the TPM** associated with our machine
- Bonus: an **encrypted communication** channel

Endorsement Key (EK)

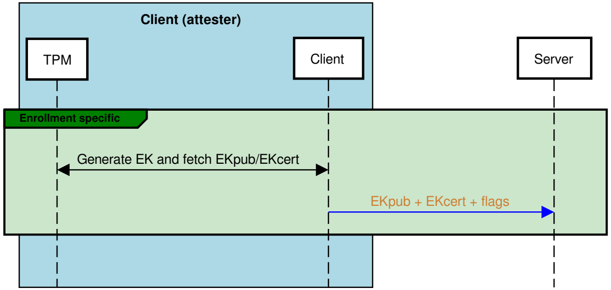
The **Endorsement Key** (EK) is:

- an **encrypting** key (cannot sign),
- **immutable**,
- generated by the **manufacturer**,
- provided with a **certificate** proving it comes from a genuine TPM.

TPM identity and privacy

The EK provides **TPM identity** but cannot be used to sign, to ensure user privacy when used online.

Protocol excerpt - EK enrollment



Problem

The EK provides **TPM identity** but cannot be used to **sign the AK**.

MakeCredential and ActivateCredential

`TPM2_MakeCredential(EKpub, credential, AKname) -> encryptedBlob`:

- takes a nonce (`credential`),
- encrypts `{credential, AKname}` with `EKpub`,
- does not need to run on a TPM.

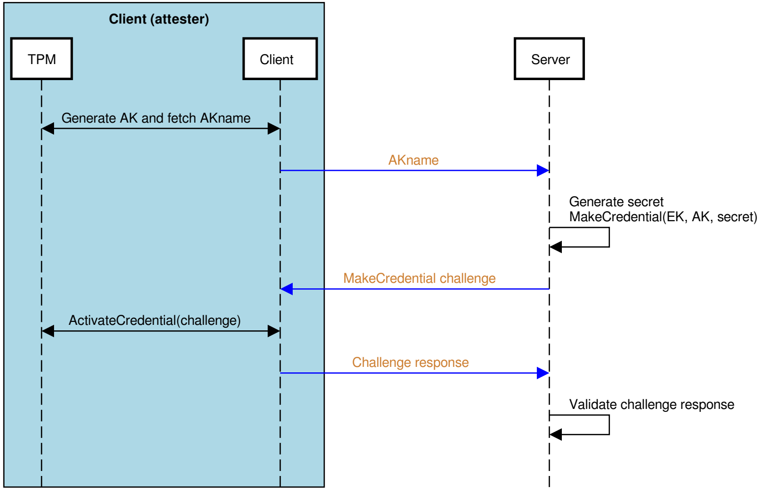
`TPM2_ActivateCredential(EKname, AKHandle, encryptedBlob) -> credential`:

- decrypts `encryptedBlob`,
- checks that the `AK` is present on the TPM and that the user is allowed to access it,
- returns the decrypted `credential`.

Linking the AK to the EK

Exhibiting `credential` proves that `AK` is stored on the same TPM as the `EK`; otherwise the TPM would refuse to return it.

Protocol excerpt – credential activation

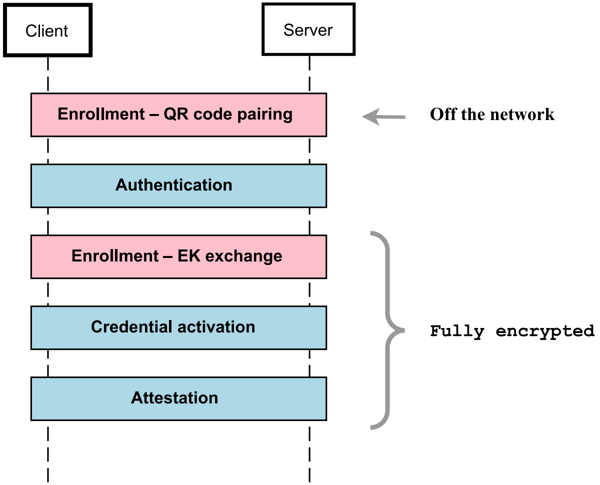

Client (attester)
TPM
Client
Server
Generate AK and fetch AKname
AKname
Generate secret
MakeCredential(EK, AK, secret)
MakeCredential challenge
ActivateCredential(challenge)
Challenge response
Validate challenge response

Goals

What we want from **remote attestation**:

- ✓A **log** of every significant boot event
- ✓An attestation that this log was **not tampered with**
- ✓An attestation that it is **signed by the TPM** associated with our machine
- Bonus: an **encrypted communication** channel

Protocol summary

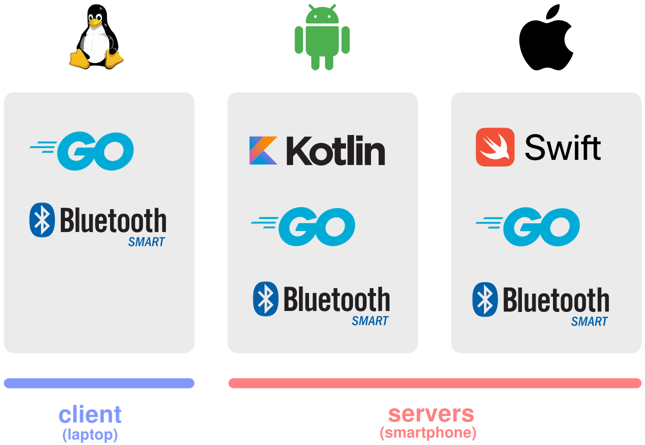


Ultrablue – User-friendly lightweight TPM remote attestation over Bluetooth

Ultrablue

- The **attestation server** is always in your pocket.
- Allows **short-range communication** - Ultrablue uses **Bluetooth**
- Provides **useful information** upon attestation failure.

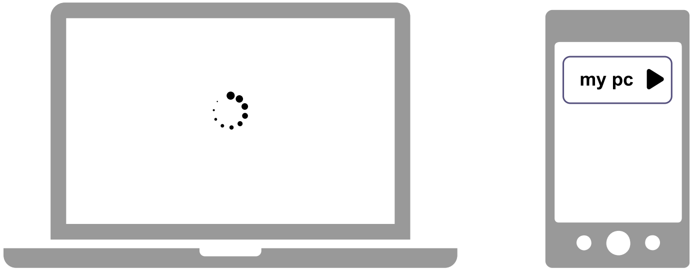
Ultrablue stack



Ultrablue workflow - Enrollment

- Use a QR code to share channel encryption key (AES-GCM).
- Get the reference state (EK, event log, PCR values).
- Boot state is **trusted on first use**.

Ultrablue workflow - Attestation



- Get new boot state.
- Inspect changes easily.
- Control attestation result.

Ultrablue demo

Ultrablue is versatile

- Ultrablue is easy to:
  - Embed in **initramfs**,
  - Use as a **second factor** for disk decryption,
  - Or as the primary factor (as in the demo).

- We provide mkosi scripts to **test it** easily.

What’s next?

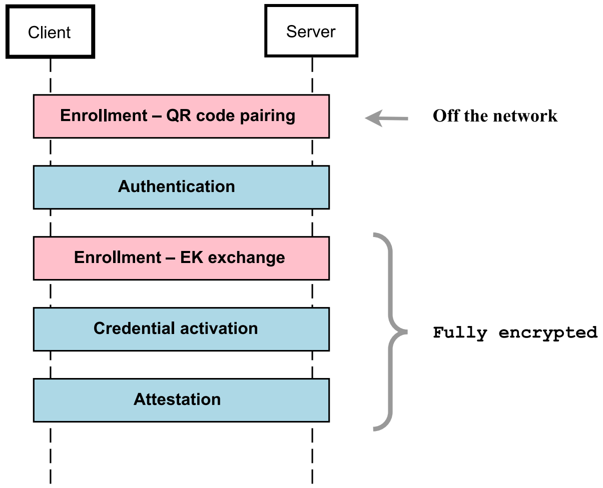
- Improve the Android app UI
- Implement a **windows client**
- Support **more protocols** (USB, IP, . . .)
- Add **mutual authentication** of both devices
- Allow remote logs and administration
- Perform **runtime integrity** checks using IMA/EVM

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

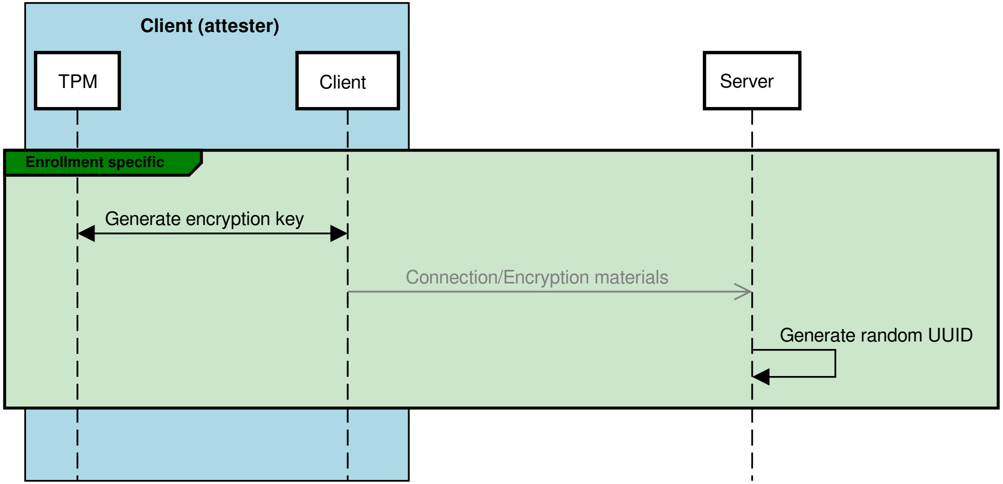
https://github.com/ANSSI-FR/ultrablue

Appendix

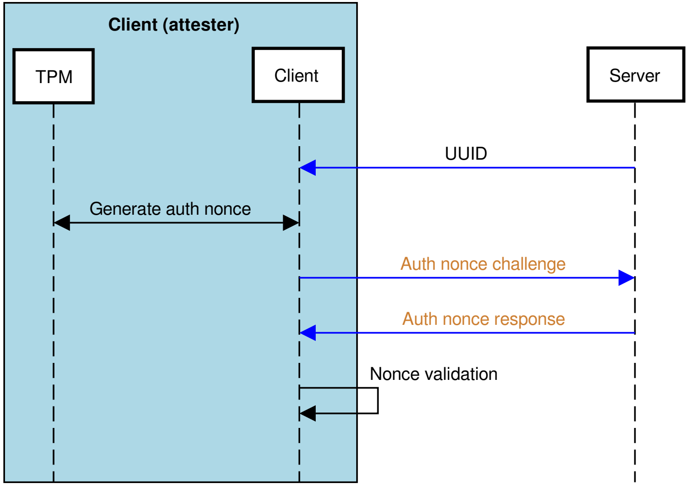
Protocol overview



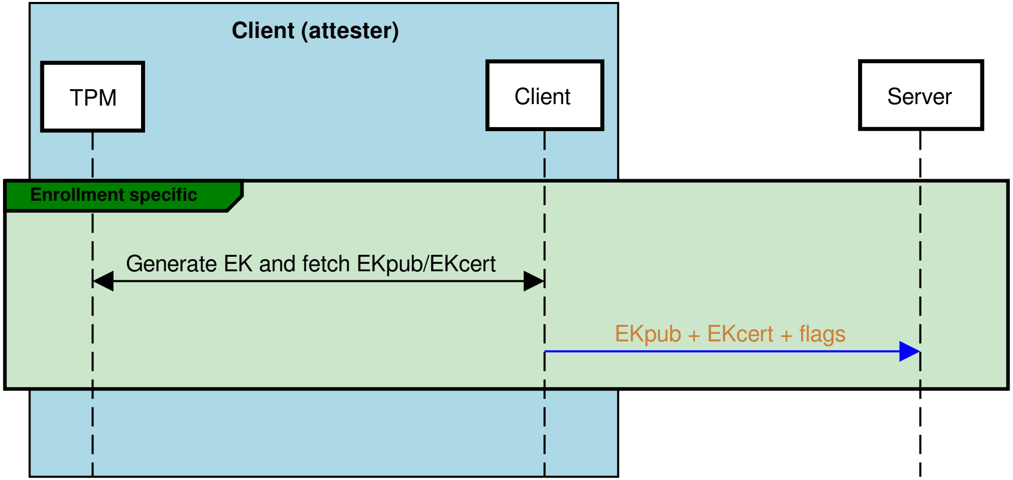
Enrollment – QR Code and UUID generation



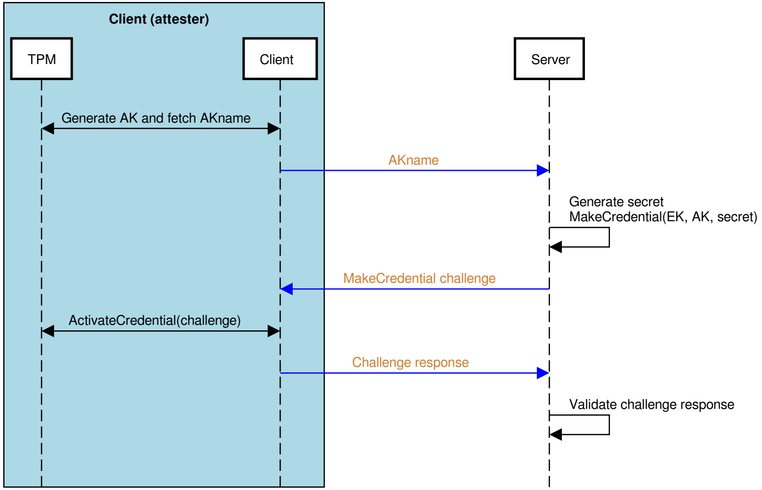
Authentication – Mitigate DoS attacks



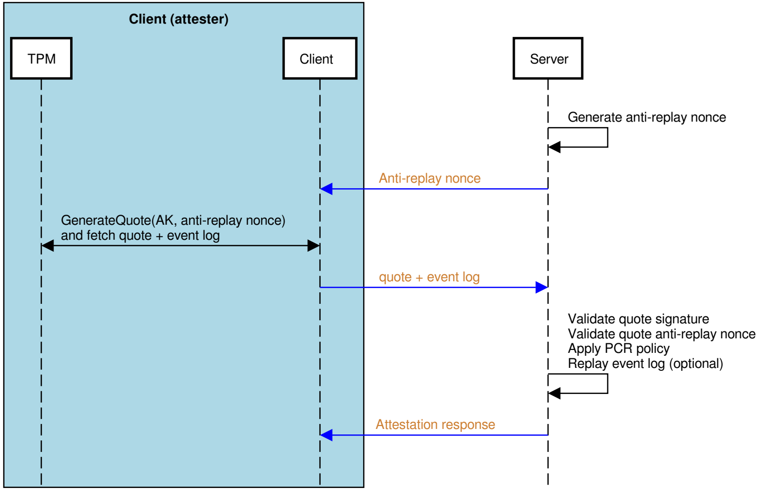
Enrollment – Endorsement key exchange



Credential activation – Generate an Attestation Key


Client (attester)
TPM
Client
Server
Generate AK and fetch AKname
AKname
Generate secret
MakeCredential(EK, AK, secret)
MakeCredential challenge
ActivateCredential(challenge)
Challenge response
Validate challenge response

Attestation – Validate boot state


Client (attester)
TPM
Client
Server
Generate anti-replay nonce
Anti-replay nonce
GenerateQuote(AK, anti-replay nonce)
and fetch quote + event log
quote + event log
Validate quote signature
Validate quote anti-replay nonce
Apply PCR policy
Replay event log (optional)
Attestation response