Device Boot-up Security Restoration with Post-Quantum Split-key KEM

Joonas Ahola, Sampo Sovio, Jan-Erik Ekberg

*__Akman et al.__, “Split keys for station-to-station (STS) protocols”, Journal of surveillance security and safety, vol. 4, no.3, 2023.

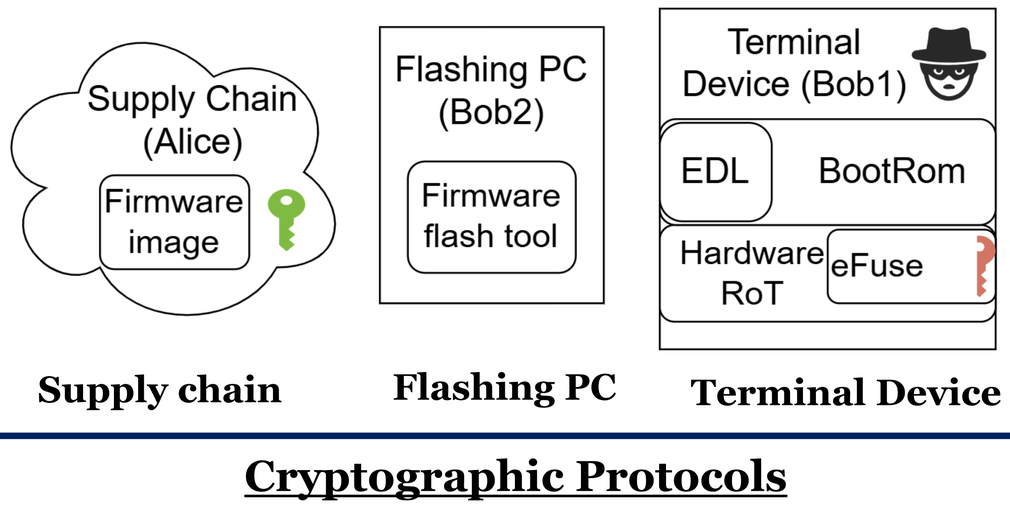
Premises of the Work

- Embedded devices boot-up security relies on non-PQC cryptographic primitives, so it cannot be expect to be secure in the future.
- A naïve way of employing non-PQC protocols along with PQC protocols is not a secure solution (Mosca’s Theorem).
- A solution is a protocol for **SecureBoot** security restoration by a firmware update.

Attacker Model and Protocol Requirements

- The **attacker** can (i) break legacy crypto, (ii) infect the system with malicious code and (iii) read secret keys already present on the device.
- The **protocol requirements** are then (i) the protocol must be based on PQC primitives, (ii) it must be run in **BootROM** in Emergency Download Mode (**EDL**), and (iii) the protocol must not rely only on the keys/secrets on the compromised device

Protocol Devices



Supply chain **Flashing PC** **Terminal Device**

Cryptographic Protocols

- **Split-key KEM (MLS-KEM)**: A modification of existing ML-KEM protocol stack to allow for multiple secret keys in the key derivation process.
- **STS-KDF with split-key KEM*:** A protocol between three parties, two of which have **split keys,** and the result is a **session key**. Assuming the split-key KEM is secure, STS-KDF is formally proven to provide (i) **confidentiality** of the session key, (ii) **mutual authentication**, (iii) **forward security**.
- If split-key KEM is QC secure ⇒ STS-KDF is PQC.

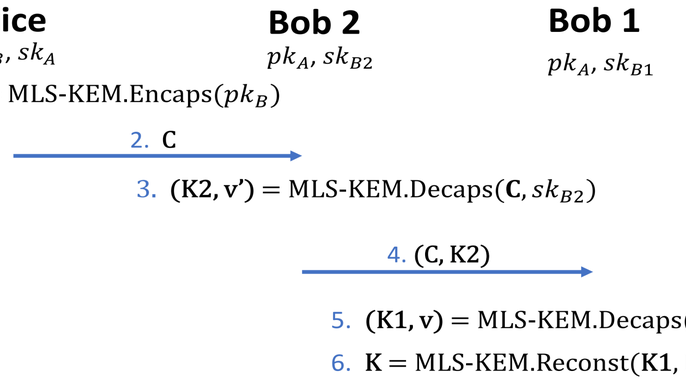
The Use Case

- The protocol is between a **user** (with a split-key), a **supply chain** (with a public key) and the infected **terminal device** (with a split-key).
- Basic assumptions, (i) user’s key half has not leaked, (ii) the terminal device is compromised (key half has leaked), (iii) BootRom is not compromised

SecureBoot

- **SecureBoot** is a verification mechanism for the software run during boot. It checks software signatures and checksums, which ensures that they are trusted and executed in correct order.

One run of the MLS-KEM Protocol



MLS-KEM

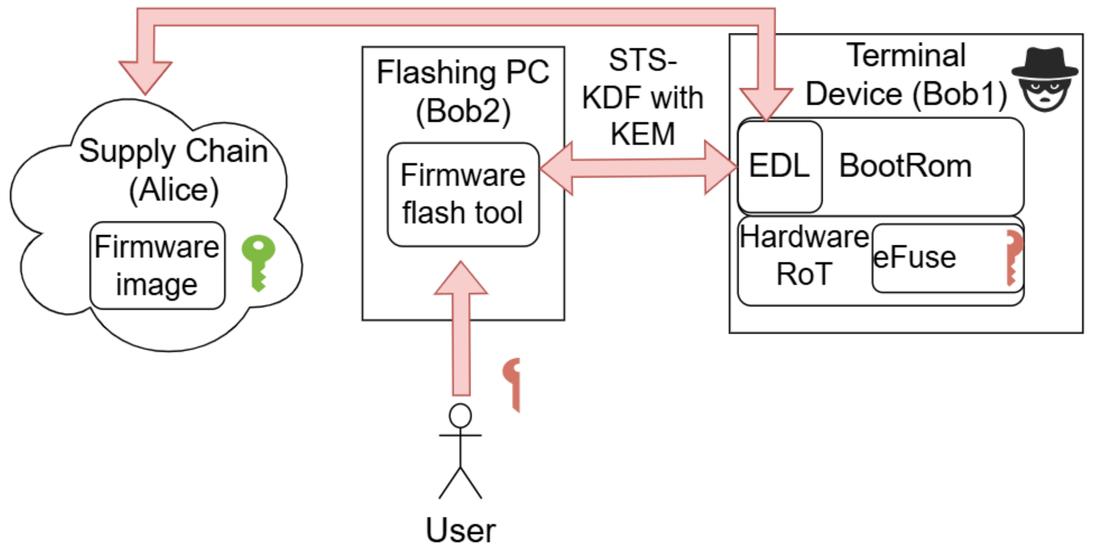
- **MLS-KEM** is a modified ML-KEM (NIST standardized version of CRYSTALS-Kyber) lattice-based KEM with split-keys.
- **MLS-KEM** performance is comparable to ML-KEM provided that similar parameters are used.

Protocol Architecture

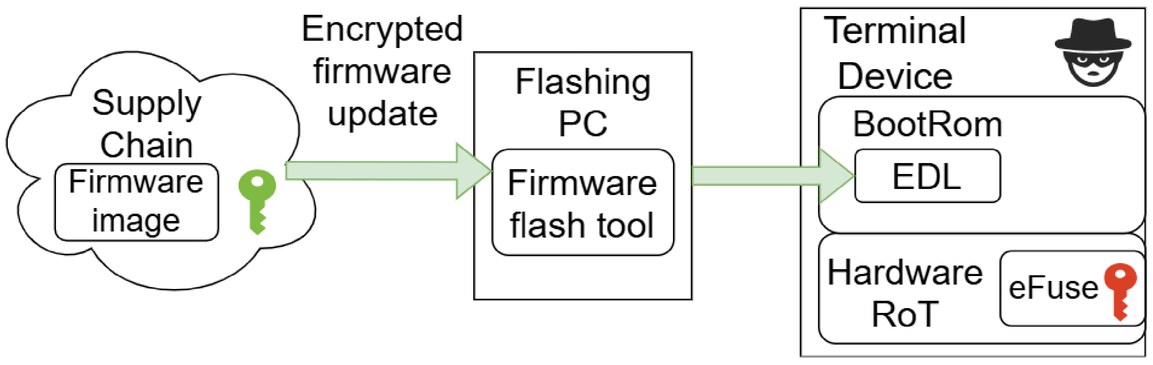
- The solution to boot-up security restoration: use two protocols for **secure provisioning**, and for **endpoint attestation.**

1. Endpoint Attestation Protocol

STS-KDF with KEM



2. Secure Provisioning Protocol



The Protocol Functionality

- To download the firmware update to the compromised device, the protocol executes the **endpoint attestation protocol** and then the **secure provisioning protocol**.

1. First, the **User** initialises the **Terminal Device** in **EDL** model. This ensures that no compromised software is run before the firmware update.
2. The **User** provides his keyshare to initialise the **endpoint attestation protocol** (STS-KDF with split-KEM). The result is the establishment of an encrypted connection between the **Terminal Device** and the **Supply Chain**.
3. The **secure provisioning protocol** is executed, and the result is a **firmware update** to the **Terminal Device**. The **Terminal Device** then installs the firmware update in **EDL** mode, restoring the boot-up security of the device.

Helsinki System Security Lab (HSSL)

HSSL drives renewal and mastery in the field of platform and device related security technologies, especially for Huawei consumer devices such as mobile phones, laptops, televisions and automotive. We do research in topics such as hardware-assisted isolation and integrity, as well as in operating system protection (hypervisor, TEE, secure enclaves and kernel hardening). We also carry expertise in cryptography and systems security functionality such as device key management (PKI), device attestation and key-store solutions.

HUAWEI