KeyNub

DATASHEET
Rev. 1.0 · September 2026

KeyNub USB-C Hardware Security Module (HSM)

A driverless PKCS#11 security module: every key is generated and kept inside a smart card on an EAL6+ certified platform, in a device that verifies its own firmware at boot.

The KeyNub HSM protects keys that must never be copyable. Every private key is generated inside the fitted smart card and reported non-extractable, so no export path exists. To the host the device is an ordinary USB smart-card reader driven by existing PKCS#11 tooling, and around the card it adds verified boot, an isolated USB stack, signed updates and attestation.

Nothing to install

A standard USB smart-card reader class: the host binds its own inbox driver. No INF, no kernel module.

Keys born on the card

RSA to 4096 and ECC to P-521, generated in hardware inside a Common Criteria EAL6+ certified card.

You hold every credential

The card ships uninitialised: you set the SO-PIN, the user PIN and the backup shares. We keep nothing.

TECHNICAL SPECIFICATIONS

INTERFACE AND HOST SUPPORT

Connector	USB-C, USB 2.0 Full Speed, bus powered
Device class	Standard USB CCID smart-card reader (class 0x0B) plus a vendor HID interface, usage page 0xFF6C
USB identifiers	VID 0x2E8A / PID 0x114B, an allocated pair
Card link	T=1 with extended APDUs to 3072 bytes; the reader owns rate negotiation, chaining and time extension
Serial number	Unique device serial (14 hex), in the USB descriptor and the device certificate
Operating systems	Windows, Linux and macOS, driverless on all three

KEYS AND CRYPTOGRAPHY

Key types	On the card, in hardware: RSA 1024–4096, ECDSA and ECDH 192–521 bits
Mechanisms	RSA PKCS#1 v1.5, PSS, OAEP and X.509; ECDSA with SHA-1 and SHA-224/256/384/512; ECDH with and without cofactor
Key protection	Created inside the card and reported non-extractable; the firmware has no path to key material
Secure element	A Common Criteria EAL6+ certified smart card; certificate reference on request

PERFORMANCE

Signing	ECDSA P-256 39 ms, RSA-2048 113 ms, RSA-4096 1.04 s
Key generation	EC P-256 1.1 s, RSA-2048 2.6 s, RSA-4096 40–100 s (prime search)

OWNERSHIP, PINS AND BACKUP

Ownership	Ships uninitialised: you set the SO-PIN, user PIN and backup domain. We hold no credential and cannot recover your card
PIN protection	User PIN with a card-enforced retry counter; a separate SO-PIN authorises administration
Key backup	Encrypted key export and import under a key encryption key you split into n-of-n shares at initialisation

FIRMWARE AND DEVICE INTEGRITY

Secure boot	The controller ROM verifies an ECDSA signature over the firmware before it runs, enforced by one-time fuses
Firmware update	Signed A/B images with try-before-buy: an update that fails its self-test reverts unaided
Attack surface	The USB stack and every parser the host can reach run isolated in a separate Arm TrustZone world, unable to reach the card, keys or storage. SWD is permanently disabled
Device identity	ECC NIST P-256 key pair in the controller; X.509 certificate from the KeyNub HSM root CA
Attestation	A fresh challenge returns a signed report of firmware version, flags and the card, verified genuine at manufacture

HOW YOU PUT IT TO WORK

1. **Initialise it yourself**
   sc-hsm-tool --initialize sets the SO-PIN, the user PIN and the backup shares. Nobody else holds them.
2. **Generate keys on the card**
   pkcs11-tool --keypairgen, or your own CA tooling. The key never leaves the card.
3. **Use it as your PKCS#11 token**
   Sign, decrypt and derive from your CA, a signing pipeline, OpenSSL or Java.

> *A key you can export is a key you must protect everywhere it has ever been. Generate it on the card and there is only one copy — which is why the backup domain is worth setting up on day one.*

HARDWARE

A 32-bit Arm Cortex-M33 controller with 2 MB of flash and white status and red fault LEDs, in a sealed USB-C housing with no user-serviceable parts. Developed and supported in Germany.

HOST SOFTWARE

OpenSC has supported this card for years: opensc-pkcs11 for PKCS#11, sc-hsm-tool for initialisation and key backup, and the minidriver to put certificates in the Windows store. XCA, EJBCA, OpenSSL and Java ride on that, from github.com/OpenSC/OpenSC.

LICENSING AND ORDERING

Order code
AB-KN-HSM

Availability
Contact us for lead time and quantity pricing

Software licensing
OpenSC is open source. khsm-tool and the management library ship under separate binary license terms.

OEM platform
Your own enclosure, logo, USB VID/PID and firmware by agreement; mechanical drawing on request.

AB-Tools GmbH Marsstraße 78, 80335 Munich, Germany · +49 89 38898588 · info@keynub.com **www.keynub.com**

Subject to change without notice; for information only and not an offer. © 2026 AB-Tools GmbH. All rights reserved.