KeyNub

DATASHEET
Rev. 1.1 · August 2026

KeyNub USB-C License Dongle

Hardware license protection for professional software — cryptographic authentication in a secure element, 1 MB of encrypted license storage, and an SDK with bindings for more than twenty languages.

KeyNub is a USB-C dongle your application talks to in order to prove that a genuine license is present. Each unit holds an ECC P-256 key pair generated inside a hardware secure element, from which it can never be read, so the license cannot be copied. Verification is local — no activation server, no license server, no cloud account — so licensing keeps working offline, on air-gapped machines and on the factory floor.

No drivers, any OS

A vendor-defined USB HID device: nothing to install and no kernel module to maintain, on all three platforms.

Keys that cannot be copied

Every check is a live challenge-response against a private key that cannot leave its Common Criteria EAL6+ certified secure element.

Integrate in hours

One core C library with a stable ABI, plus thin bindings for over twenty languages, each with a runnable sample.

TECHNICAL SPECIFICATIONS

INTERFACE AND HOST SUPPORT

Connector	USB-C, USB 2.0 Full Speed, bus powered
Device class	Vendor-defined USB HID, usage page 0xFF6C — driverless
USB identifiers	VID 0x2E8A / PID 0x113B, an allocated pair
Transport	One 64-byte IN and one 64-byte OUT interrupt report every millisecond; approx. 64 KB/s payload
Serial number	Unique 7-byte secure-element serial (14 hex), in the USB descriptor and in the device certificate
Operating systems	Windows (x64, x86, ARM64), Linux (x86_64, ARM64) and macOS as one universal binary

SECURITY

Device identity	ECC NIST P-256 key pair generated inside the secure element; non-extractable
Attestation	X.509 device certificate issued by the KeyNub root CA, which the SDK verifies against out of the box
Genuineness check	Live ECDSA P-256 challenge-response, fresh every check
Encrypted session	Ephemeral P-256 ECDH → HKDF-SHA256 → AES-256-GCM, device-signed and replay-protected
Data at rest	Every record AES-256-GCM encrypted under a key derived in the secure element
Write protection	Writes, erases and counter increments need the developer's own master key
Bus protection	AES-256 authenticated, replay-protected secure messaging
Attack surface	The USB and parsing code the host can reach runs isolated in an unprivileged Arm TrustZone world; it cannot address keys, the secure element or license storage

LICENSE DATA STORAGE

User area	1 MB (1024 KB), encrypted and power-fail safe
Storage model	Named records: 1–32 character names, up to 2048 bytes each, replaced atomically
Counters	Hardware monotonic 32-bit counters (anti-rollback)
App encryption	The dongle wraps and unwraps your application keys, scoped to one dongle or to any dongle you have issued

HARDWARE AND FIRMWARE

Controller	32-bit Arm Cortex-M33 microcontroller, 2 MB flash
Secure element	Hardware key storage, TRNG and counters; Common Criteria EAL6+ AVA_VAN.5 certified, certificate reference on request
Firmware	A/B update slots with automatic rollback
Indicators	White status LED and red fault LED
Enclosure	Compact USB-C housing; mechanical drawing on request
Origin	Developed and supported in Germany

HOW AN INTEGRATION WORKS

1. **Verify the device**
   verify_genuine() checks the certificate chain and a fresh signature. A substituted device fails.
2. **Open a session**
   session_open() derives the AES-256-GCM keys. Everything after this is encrypted and replay-protected.
3. **Put it on the critical path**
   Read license records, increment counters, and decrypt values your software genuinely needs.

> *A dongle cannot stop somebody patching your executable, so do not hand them a single boolean to patch. Encrypt what your software depends on: then there is nothing to run without it.*

LANGUAGE BINDINGS

C · C++ · C# and .NET (VB.NET, F#) · Python · Java · Delphi and Free Pascal · Rust · Go · Node.js and Electron · MATLAB and Simulink · LabVIEW · VBA and Excel · Visual Basic 6 · twinBASIC · VBScript · Fortran · COBOL · Ruby · PHP · Perl · Lua · Julia · Nim · Zig

Shipped as NuGet, a Python wheel, a Maven artefact and C/Delphi archives. Every binding is tested against an in-process software dongle, so no hardware is needed to develop.

LICENSING AND ORDERING

Order code
AB-KN-DGL

Availability
Contact us for lead time and quantity pricing

SDK license
Bindings, samples and the C ABI header are Apache-2.0 at github.com/AB-KeyNub/KeyNub-SDK. The prebuilt native libraries ship under separate binary license terms.

OEM platform
Your own enclosure, logo, USB VID/PID and custom firmware, under a separate agreement.

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