What shall we do with two PKCS#11 providers?

OpenSSL Conference 2026

Jakub Jelen
Software Engineer
Red Hat

What is a provider and PKCS#11

OpenSSL Providers

- Replaced OpenSSL engines
- Since OpenSSL 3.0
- dynamically loadable module
- implementing cryptographic operations for OpenSSL and its users

PKCS#11

- Standard API for cryptographic token operations
- smart cards, tokens and hardware security modules

Why

- Keep cryptographic material secure
- Allow applications to do operations transparently

PKCS#11 provider

- https://github.com/openssl-projects/pkcs11-provider
- Started in 2022
- Replacement for engine_pkcs11
- Official OpenSSL project since 2025
- Focus on reliability, test coverage, compatibility
-

libp11 pkcs11prov provider

- Provider interface wrapping general purpose libp11 library
  - `libp11` dating back to 2005
  - and `engine_pkcs11` (merged into codebase in 2016: v0.4.0)
  - provider introduced in 2025 (0.4.14)
- Initially considered for extension in 2022
  - But instead we started a new project
-

Difference

Name:		
	pkcs11	pkcs11prov
Architecture:		
	native	wrapper
Configuration:		
Integration testing		
Algorithm support		

Configuration

pkcs11-provider

[pkcs11_sect]

activate = 1

pkcs11-module-path = /path/to/pkcs11.so

pkcs11-module-login-behavior = always

pkcs11-module-token-pin = file:/etc/ssl/pinfile.txt

Environment:

PKCS11_PROVIDER_DEBUG=file:/dev/stderr,level:2

libp11 pkcs11prov

[pkcs11_sect]

activate = 1

identity = pkcs11prov

module = /usr/lib64/ossl-modules/pkcs11prov.so

pkcs11_module = /path/to/pkcs11.so

debug_level = 7

force_login = 1

pin = 123456

Integration testing

pkcs11-provider

- 3 different software tokens (softokn, softhsm, kryoptic)
- Distributions: Ubuntu, Fedora, Debian, CentOS, Alma
- OpenSSL versions: master
- 68% lines code coverage
- integration testing with bind, httpd, libssh

libp11 pkcs11prov

- softhsm only
- Ubuntu + MacOS + Windows only
- OpenSSL 3 and 4
- unknown code coverage

Architecture

pkcs11-provider

- written from scratch around the OpenSSL provider API

libp11 pkcs11prov

- Wrapper around general purpose pkcs11 wrapper

Hardware compatibility

pkcs11-provider

- Configuration quirks: `pkcs11-module-quirks`
- Handle missing functionality: Operation State, Allowed Mechanisms
- Thales Luna HSM specific login (`luna-non-standard-cku-user`)
- Thales Luna specific Ed25519 OIDs

libp11 pkcs11prov

- Support for non-standard Falcon on Thales Luna
- Workaround for CKA_WRAP attributes not accepted by Thales Luna and Entrust nShield HSMs for PQC keys

Algorithms

pkcs11-provider:

- Asymmetric: RSA, ECDSA, EDDSA, ML-DSA, SLH-DSA, ML-KEM
- Symmetric: AES, ChaCha20, Generic Secrets
- Digests
- Derivation: HKDF, ECDH, XEDDSA, TLS-KDF
- Whole TLS session crypto tested

libp11 pkcs11prov

- Same asymmetric algorithms set
- Falcon (Thales proprietary)
- Missing symmetric algorithms and KDFs
-

What to do with 2 pkcs11 providers?

- PKCS#11 URI (RFC 7512) complexity
  - Conformance testing
  - Fuzzing
  - Reuse corpus from OpenSC
- Configuration compatibility.
  - How much would it take to support each others configuration or define some standard subset?
- Benchmark operation
  - Signature, search, fork, threading
- Can I run their tests against each other?

Future work

- PKCS#11 Operations Fuzzing
  - Driver
  - Mock PKCS#11 module
  - How the providers handle unexpected responses from underlying PKCS#11 module