OpenSSL CONFERENCE

Jostle, OpenSSL's new Java Cryptography Provider for JVMs 8 to 25+, with FIPS support

Megan Woods · OpenSSL Conference 2026, Prague

About Me

By the numbers

20.10.2026

- 6809
- 68000
- MCS-51 aka 8051*
- AT90xx*
- ATMegaXX*
- ESP32
- x86_64*
- Aarch64*

- CB400
- R1200GS
- DRZ650



* Paid to program

What is Jostle?

OpenSSL's libcrypto, served through the standard JCA Provider API.

No new API to learn.

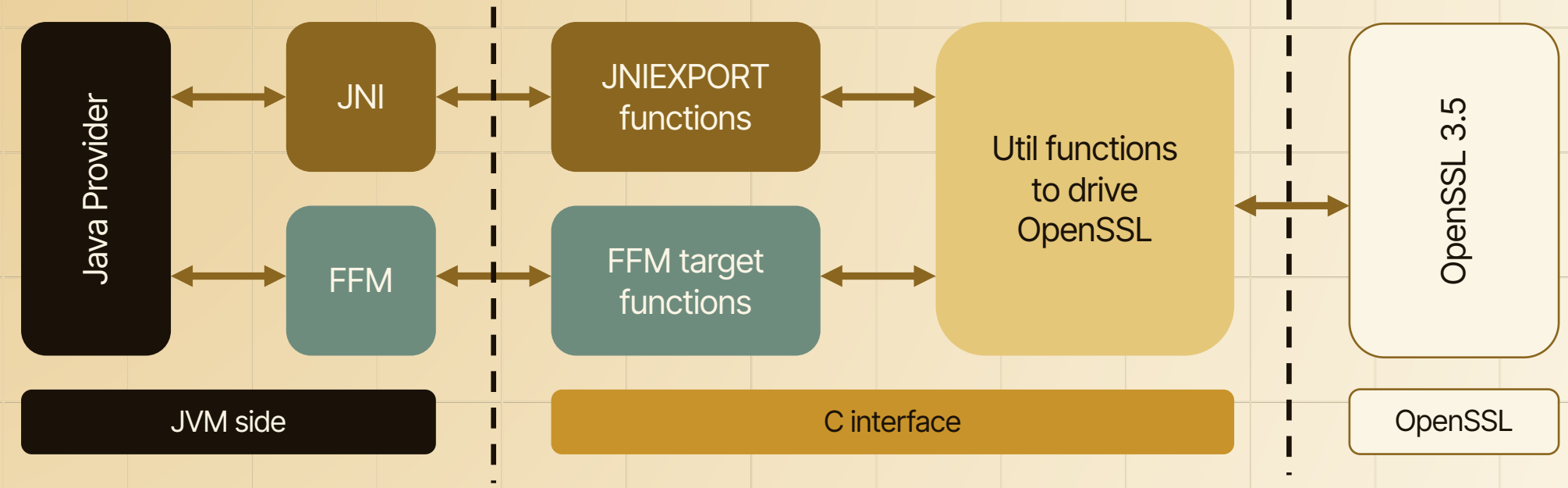
2026

OpenSSL CONFERENCE

- A collaboration between:
  - OpenSSL Corporation
  - Legion of the Bouncy Castle Inc.
- Apache 2.0, the same license as the OpenSSL Library
- Supports on Java 8 to 25, builds on Java 25
- Two providers in one jar:
  - “JSL" drives OpenSSL 3.5.x,
  - “JSLFIPS" drives OpenSSL FIPS modules (3.1.2, 3.5.x) user to supply
- Repository opened:
  - September 2025,
  - First provider code October 2025

Design overview

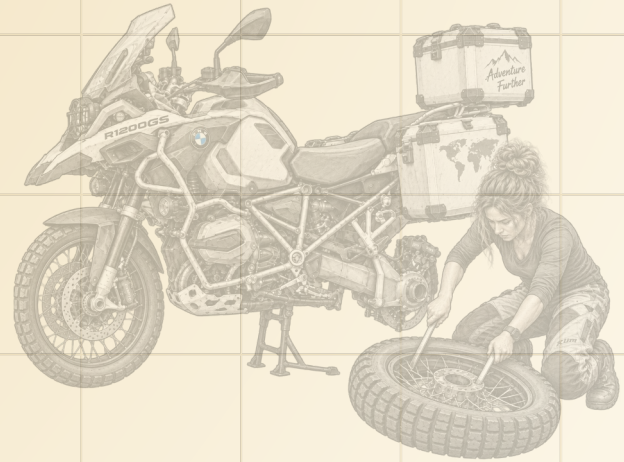
OpenSSL CONFERENCE



Three layers:

1. The Java provider drives the C interface through either JNI or the FFM bridges
2. The C interface drives OpenSSL through one set of utility functions.

Alternative implementations can be substituted at the dashed lines.



How it is built

- Layers
  - JCA SPI classes in Java, call native bridges that drive libcrypto.
  - Runtime Bridge Selection:
    - JNI for Java 8 to 24
    - FFM on Java 25
    - JNI on Java 25+ is optional
  - Multi-release source tree:
    - a feature enters at the lowest JDK level where the API exists
- Shipping
  - One multi-release jar per architecture:
    - amd64 and aarch64
  - Module descriptor at META-INF/versions/9
    - Module: org.openssl.jostle.prov
    - Pass *--enable-native-access* on JDK 24+
  - The FIPS module is not bundled, must be deployed on the host.
  - Size today is about 92k lines of Java, 76k of C, and 189k lines of tests

Algorithm coverage, measured from the jar

OpenSSL CONFERENCE

Service type	JSL	JSLFIPS (3.5.8 module)
Cipher	55	32
Signature	74	72
KeyPairGenerator	34	31
KeyFactory	35	31
KeyAgreement	25	11
KeyGenerator	17	12
Mac	20	15
MessageDigest	21	15
SecretKeyFactory (KDFs)	43	34
SecureRandom	18	14
AlgorithmParameters	21	17
KeyStore	5	1
CertPathValidator / Builder	1 / 1	0
Total registered services	**373 across 16 types**	**288 across 14 types (213 on the 3.1.2 module)**

Symmetric, PQC, Asymmetric

Symmetric

AES-128/192/256: ECB, CBC, CTR, CFB, OFB, CTS, XTS, GCM, CCM, OCB
Key wrap: AES KW (RFC 3394), KWP (RFC 5649) and inverse; RFC 3211 password wrap for AES, Camellia, Triple-DES,ChaCha20 and ChaCha20-Poly1305 ARIA, Camellia, SM4 in the block modes; CCM for ARIA and SM4
Triple-DES

Post-quantum

ML-KEM-512, 768, 1024 ML-DSA-44, 65, 87, with external-mu and calculate-mu.
SLH-DSA: all twelve SHA2 and SHAKE parameter sets, pure and deterministic
TLS hybrids: X25519MLKEM768, X448MLKEM1024, SecP256r1MLKEM768, SecP384r1MLKEM1024

Asymmetric

RSA: OAEP and PKCS#1 v1.5 encryption; KTS-KEM-KWS key transport; PKCS#1 v1.5, PSS and raw signatures with MD5, SHA-1, SHA-2, SHA-3
EC: ECDSA with SHA-1, SHA-2, SHA-3 and raw; ECDH and ECCDH with X9.63 and concatenation KDFs; ETSI/IEEE 1609.2 KEM
Ed25519 and Ed448: pure, prehash and context forms X25519, X448, XDH with concatenation and HKDF KDFs
DH on named groups with the RFC 2631 KDF;
DSA with SHA-1, SHA-2, SHA-3 and raw; DH and DSA parameter generation



Digest, MAC, KDF, DRBG & Miscellaneous

OpenSSL CONFERENCE

Digest, MAC, KDF, RNG

Digests: SHA-1, SHA-2 (224 to 512/256), SHA-3, SHAKE128/256, BLAKE2b/2s, SM3, RIPEMD-160, MD5, MD5-SHA1
MACs: HMAC over every digest, CMAC, GMAC, KMAC128/256, Poly1305
KDFs: PBKDF2 with every HMAC, HKDF, scrypt, Argon2, SSKDF, SSHKDF, KBKDF (HMAC and CMAC)
DRBGs: CTR with AES-128/192/256; Hash and HMAC with SHA-1 to SHA-512

Miscellaneous

Key stores: BCFKS; PKCS12 with 3DES, AES-256 and PBMAC1 variants
X.509 certificate factory: PKIX certificate path validation and building
AlgorithmParameters for AES, ARIA, Camellia, SM4, Triple-DES, GCM, CCM, ChaCha20-Poly1305, DH, DSA, EC and RSASSA-PSS



Working with an AI assistant

Started 3 May 2026 with a 155-line CLAUDE.md.

The rules are as much the product as the code is.

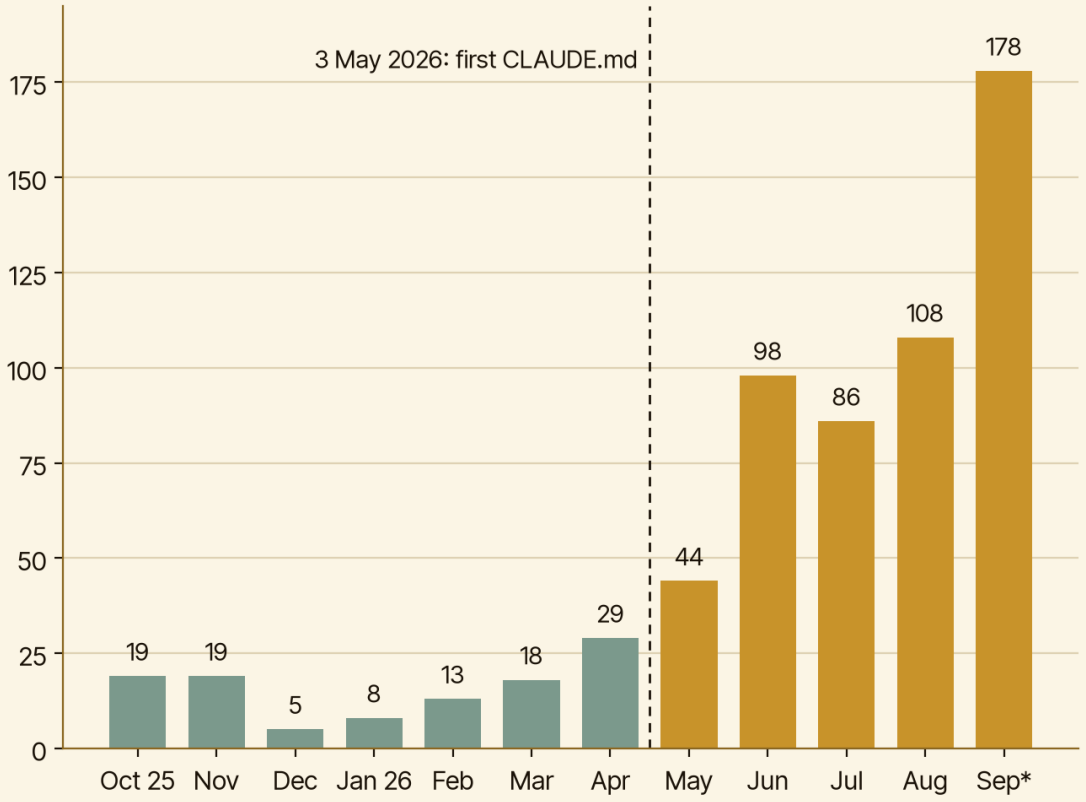
2026

OpenSSL CONFERENCE

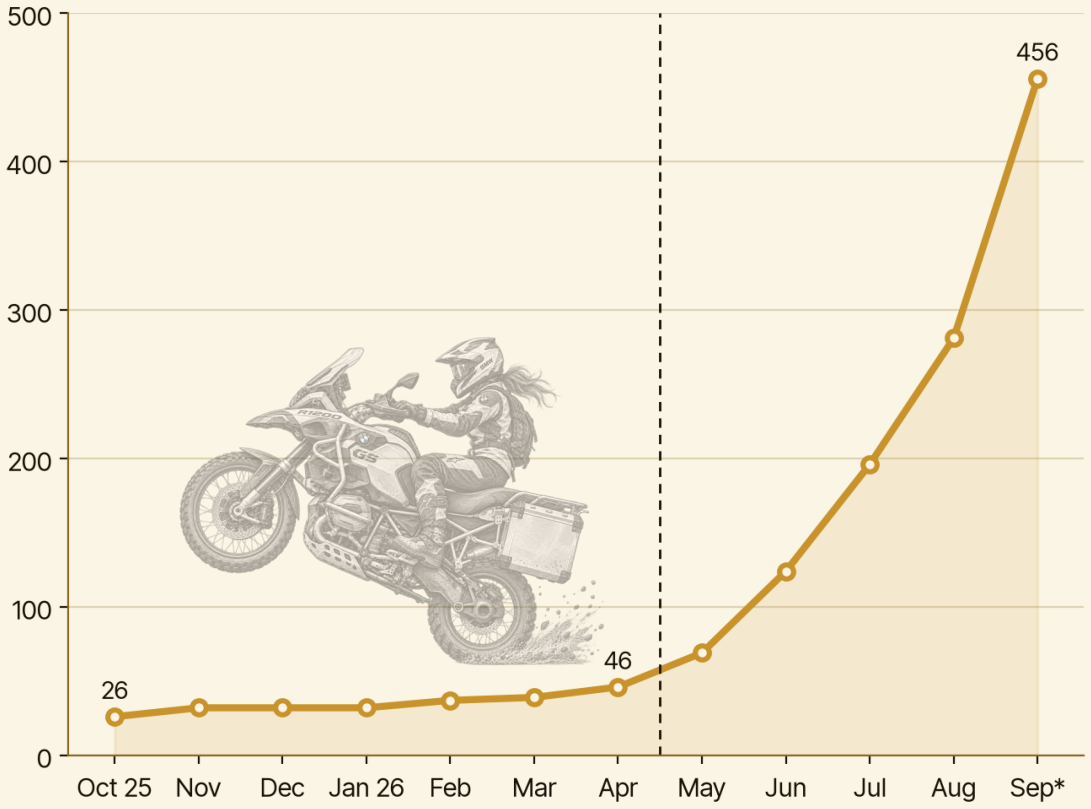
- Before May 2026, all hand-written API at about 16 commits a month
- May 2026:
  - First CLAUDE.md with the assistant working inside the repository under written rules.
- June 2026:
  - The rules split into three guides, now ~2,862 lines: Java SPI, native code, test discipline
- August: (Dual agent workflow)
  - A worker session that codes and stages (Opus) and a review session
  - Reviewer reads from the git index and never codes anything, generates plans for the worker
  - Human reviews and triggers commits, pushes and gives overall instruction
  - Worker and reviewer communicate over a local socket
- Twelve repository skills capturing frequent tasks and making them deterministic
- Planning board with numbered design decisions in a register, each answered by a person, each cited by the code review that followed



Commits per month on main



Test classes in the tree at month end



* September measured to the 24th. Source: git log / git ls-tree on main at 73234b0d, 24 Sep 2026.

main, Oct 2025 to 24 Sep 2026

sage before, gold with the assistant

OpenSSL CONFERENCE

Front loading rules

Every human found issue became a testing rule.

This also included experiences from implementing BC's native code and testing BC in general.

2026

- Project defines testing rules, "guides" that defines, for example:
  - "Signatures / MACs — after signing, flip a byte in the message and assert verification returns `false` (or throws). For algorithms with key consistency checks (RSA, EC, key-validating PQC schemes), also test that a corrupted public/private key is rejected at parse time or causes verification to fail."
- Human Review happens on what is in the git stage



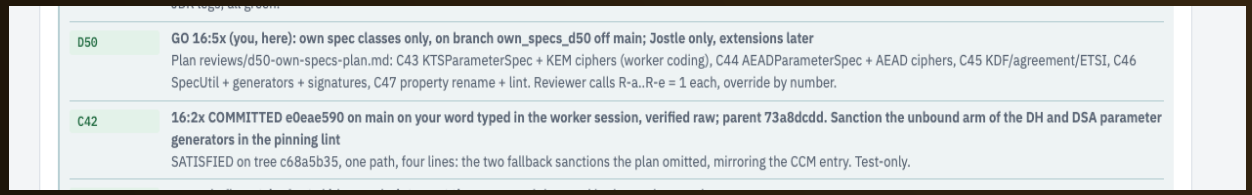
OpenSSL CONFERENCE

Workflow

Workflow is important

2026

- I had the reviewer agent maintain a dashboard on disk locally that could be loaded into to a browser, this lists decision points, commit points and possible options.



- Most decisions are clear and can be easily answered with a simple option selection. Anything else the reviewer was prompted or I looked at the code.
- Human absolutely in loop before commits but after first Fable review.

OpenSSL CONFERENCE

Overnight mode

Overnight mode, was an experiment I tried a couple of times, productive but expensive in terms of token spend.

Filled me with dread, a lot, to be honest.

2026

- Overnight mode was a time boxed special set of rules:
  - Accept recommended option
  - Worker can commit, define shape of commit message
  - Worker and Reviewer to trash out issues
  - Run test gates between logical pieces of work
  - Stop committing at 8:30am

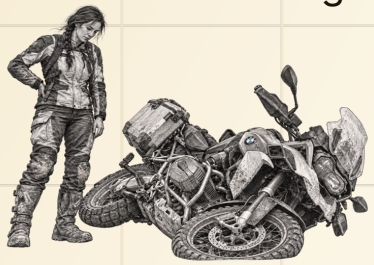


- I disabled any ability for my machine to push.
- Ran without issue and commits were sound.
- Context grew without regular compaction.
- Couldn't shake the fear of it running unsupervised!
- **Only tried it twice!**

Testing overview

- **Agreement tests**, two or a three-way comparison between BC, JSL and JSLFIPS
- **Limit tests**, assert the native layer's handling of invalid values.
- **Miscellaneous**, tests that don't fall into this category for example an abort probe test or one designed to verify some aspect of the source code.
- Every family
  - Tests against BouncyCastle with random inputs: JSL vs BC, and JSLFIPS vs JSL vs BC
  - Exception type parity with BouncyCastle, the messages stay ours.

- Generally
  - Exercises the negative path, vandalised messages, keys etc.
  - Randomised messages, keys & other inputs.
  - Integers:
    - ± 1 off known ranges
    - -1, MIN & MAX into unsigned receiver
  - Read / Write into same array
  - Combinations of byte by byte & globing
  - Input / output start location jitter to force handling of unaligned reads and writes.



Testing, OPS testing

- Operations testing, a concept borrowed from CMVP testing.
- A special build that allows testing of code paths that would only be followed if there was an error in either the JVM or OpenSSL.
- Paths that are impossible to trigger normally.
- Requires special build.

```
if (OPS_OPENSSL_ERROR_3 1 != EVP_EncryptUpdate(ctx->evp,
output, &written, input, (int) in_len)) {
    ctx->poisoned = 1;
    return JO_OPENSSL_ERROR;
}
```

```
@Test
public void testUpdateFailure() throws Exception
{
   Assumptions.assumeTrue(operationsTestNI.opsTestAvailable(), "Ops Test only");

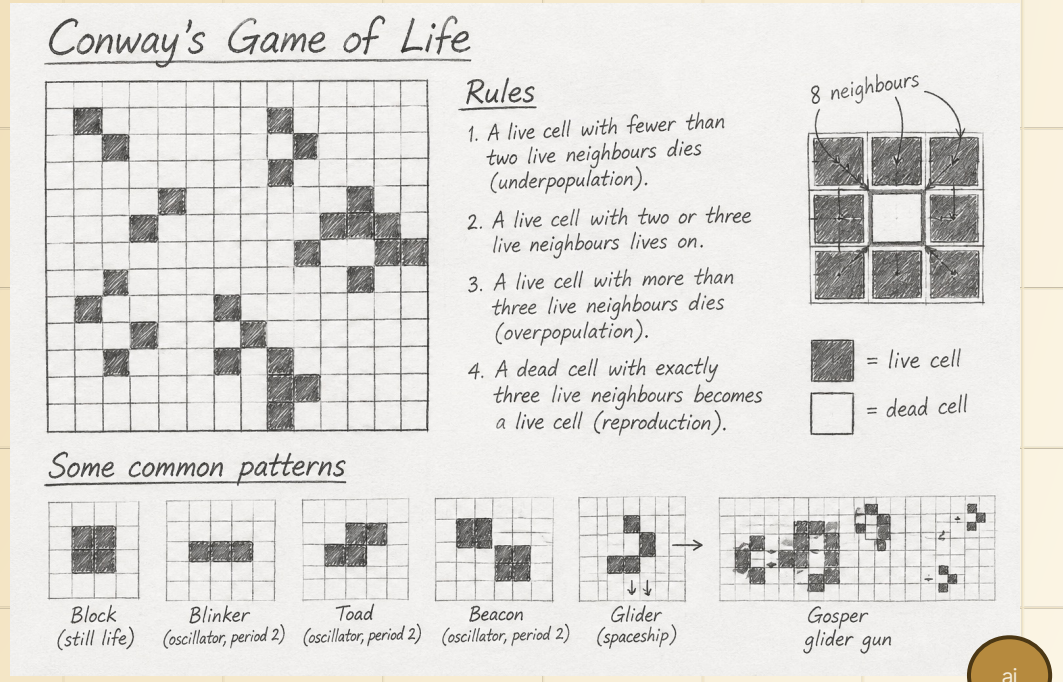
  long ref = 0;
  try
  {
     ref = blockCipherNI.makeInstance(8, 1, 0); // AES128, CBC, NO_PADDING
     Assertions.assertEquals(0, blockCipherNI.init(ref, Cipher.ENCRYPT_MODE,
sequentialKey(16), sequentialIv(16), 0));

     // Exercises interface/nonfips/util/block_cipher_ctx.c:1141

operationsTestNI.setFlag(OperationsTestNI.OpsTestFlag.OPS_OPENSSL_ERROR_3);
.. etc
```

Rules driving testing behaviour

- Examples:
  - Agent sabotaging code to verify a test fires.
  - Rigging some native code so that a destination buffer is up against the page boundary to detect overrun.
  - Building tests to verify source code parity.
  - Building exploratory code to see what kinds of errors OpenSSL will emit.
  - Building own test harness to verify that “aborts” are occurring as expected.



FIPS: JSLFIPS

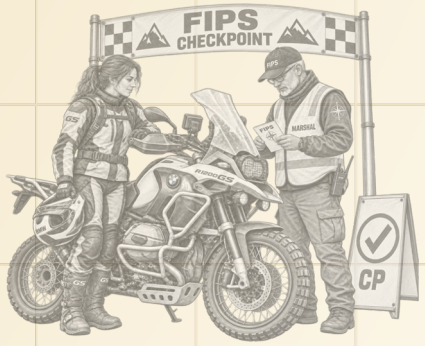
The OpenSSL module decides what is served.

Jostle drives it and surfaces the module's answers as typed Java exceptions, never as silent fallbacks.

2026

OpenSSL CONFERENCE

- One build serves two modules:
  - OpenSSL FIPS 3.1.2, the CMVP-validated one (certificate #4985),
  - and 3.5.x, the current LTS
  - It's a wrapper, Jostle doesn't need to be certified.
- No PKCS#1 v1.5 encryption at all: OAEP only,
  - No PKCS1Padding refused at lookup.
  - NoneWithRSA (raw PKCS#1 v1.5 signing) is available.
- DH parameter generation refused, use named groups, reached through:
  - Use KeyPairGenerator.getInstance("DH","JSLFIPS")
  - Set the bit size to RFC7919 bit size (eg. ffdhe2048)
- A caller-supplied SecureRandom is ignored inside FIPS module
- A Key belong to the provider instance that instantiated it:
  - a JSLFIPS Signature rejects a JSL key, this also includes public keys.



The two supported modules disagree

OpenSSL CONFERENCE

Family	3.1.2 module	3.5.8 module
X25519 / X448	available	unavailable
Ed25519 / Ed448	unavailable	available
ML-KEM, ML-DSA, SLH-DSA	unavailable	available (70 services)
Triple-DES	unavailable	available
GMAC Mac.clone()	refused by the module	works



Approved mode

JSLFIPS exposes the modules capability.

Users are required to use it in a way that is FIPS compliant.

2026

- The module does not enforce its own validated envelope
- OpenSSL 3.1.2 exposes no runtime approved-mode indicator
- Deployments that need an enforced restriction use the JVM's own jdk.security.providers.filter
- What JSLFIPS registers is what the module serves, not a list filtered against the policy

Where to find it

Questions?

2026

- https://github.com/openssl-projects/openssl-jostle
- See SERVICES.md for a list of services
- Point AI coding tools at: docs/jostle-ai-guide.md

AI Credit:
All motorcycle images generated by ChatGPT, graphs generated by Claude based on real data.