

CAVP / CMVP Perspective on KEMs

NIST Workshop on Guidance for KEMs 2025
2/25/2025

Chris Celi, CAVP Program Manager, NIST
christopher.celi@nist.gov

- Applies to all Federal agencies that use cryptography to protect sensitive information
- Requires that cryptographic modules undergo validation testing via the Cryptographic Module Validation Program (CMVP) to be used by the Federal government
- The Cryptographic Algorithm Validation Program (CAVP) exists as a branch of the CMVP to perform algorithm tests on cryptographic modules

Cryptographic Algorithm Validation Program **NIST**

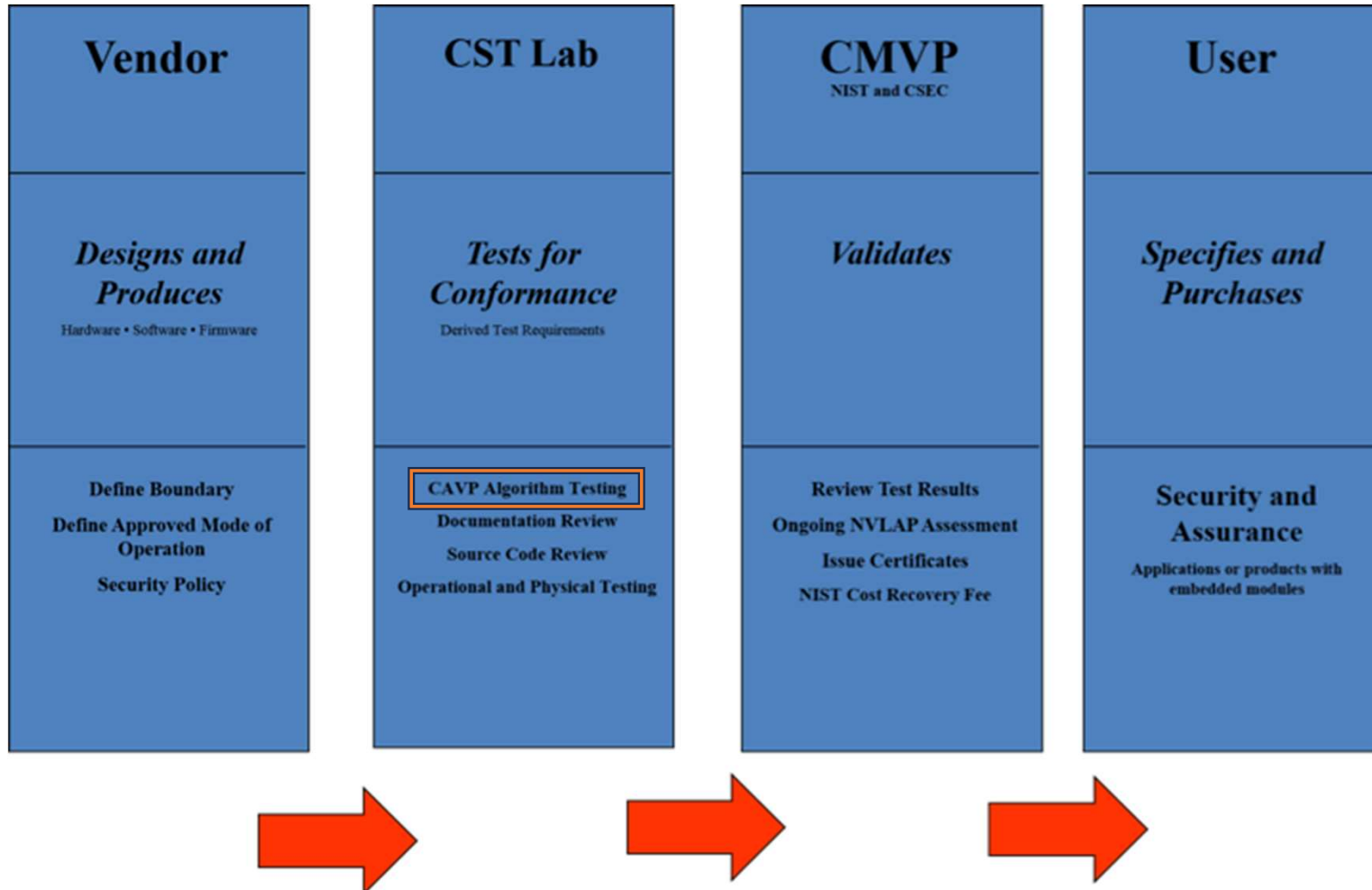
- **CAVP is a program within NIST**
- Validation consists of conformance testing to FIPS 140 “Security Requirements of Cryptographic Modules”
- Tested algorithms listed in SP 800-140 documents

A cryptographic module is any software, hardware, hybrid, system, etc. that has at least one approved security function (cryptographic algorithm), such as encryption, authentication, digital signatures, key exchange...

- ACVTS Prod (2019) used by accredited labs to conduct validation testing.
- ACVTS Demo (2017) is a sandbox-style environment for anyone to request access and test.
- Nearly 3 million vector sets served between Demo and Prod.
- 17ACVT scope open to first-party test labs, see NIST Handbook 150-17.
- Source code at <https://github.com/usnistgov/ACVP-Server>

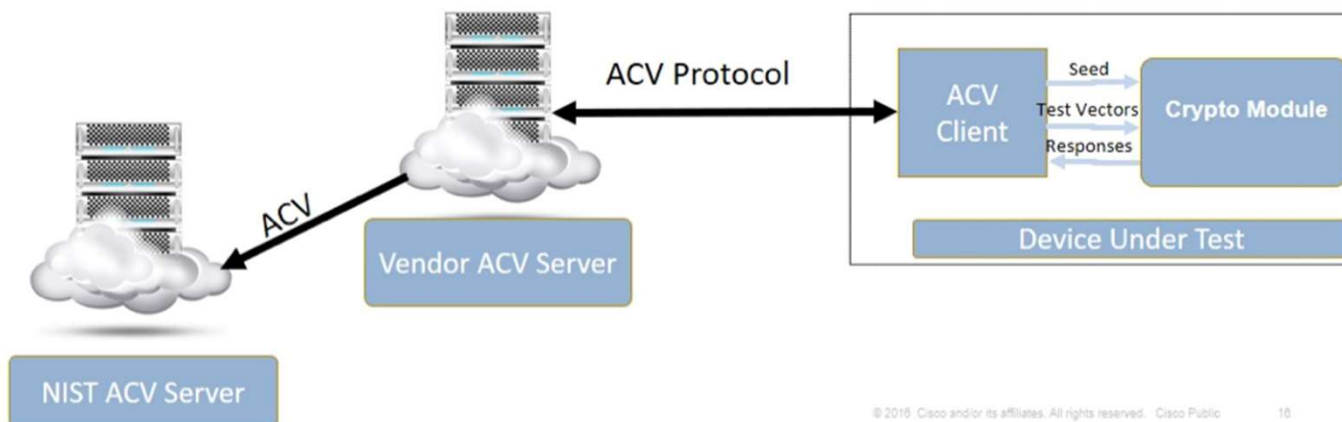


Validation Process



Algorithm Validation Process

Proxy/Validation Authority Architecture Automated Cryptographic Validation System



Algorithm Validation Process



- NIST-hosted server called Automated Cryptographic Validation Test System (ACVTS) provides algorithm test vectors
- JSON-based communication over an API
- Tests (almost) all NIST-approved cryptographic algorithms
- Server provides inputs to a client that returns the outputs for verification, black-box testing

- Production Server active since 2019
 - Access limited to NVLAP-accredited 17ACVT labs
 - Pay per vector set (or unlimited for one year)
- Demo Server active since 2017
 - Access open to those who request
 - No costs
 - See <https://github.com/usnistgov/ACVP> for more information
- Over 3,000,000 vector sets served!

- Uses the internal interfaces, Algorithms 16, 17, and 18
- ML-KEM KeyGen tests, 45 validations
- ML-KEM Encap/Decap tests, 52 validations

As of 2/21/25

ML-KEM Testing – KeyGen

- Key pair generation tests are very simple
- Provide the random seeds, z and d , and expect the implementation to generate the correct (ek, dk) pair
- 25 key pairs for each ML-KEM parameter set
- Prerequisite testing required on SHA3-256, SHA3-512, SHAKE-128 and SHAKE-256

ML-KEM Testing – Encapsulation

- Encapsulation tests check for exact equality
- Provide a key pair, random seed m , and expect the implementation to generate the correct ciphertext, shared secret pair
- 25 encapsulations for each ML-KEM parameter set
- Prerequisite testing required on SHA3-256, SHA3-512, SHAKE-128 and SHAKE-256 for encapsulation

ML-KEM Testing – Decapsulation

- Decapsulation tests check for the correct shared secret
- Provide a key pair, and ciphertext, and expect the implementation to generate the shared secret
- 25 decapsulations for each ML-KEM parameter set
- Basic failure condition introduced to trigger the implicit rejection
 - Plan to add more soon
- Prerequisite testing required on SHA3-512, SHAKE-128 and SHAKE-256 for decapsulation

- Looking at potential component tests for K-PKE.Encrypt() where an implementation only performs decapsulation
- Normally inputs to K-PKE.Encrypt() can be tested via encapsulation, but if the implementation does not run the encapsulation tests, some additional testing may be helpful
- Inputs to K-PKE.Encrypt() from decapsulation are the results of other functions so the inputs can not be easily controlled for testing unless K-PKE.Encrypt() is tested directly

Cryptographic Module Validation Program



- **CMVP is joint program between NIST and the Canadian Centre for Cyber Security (CCCS)**
- Validation consists of conformance testing to FIPS 140 “Security Requirements of Cryptographic Modules”

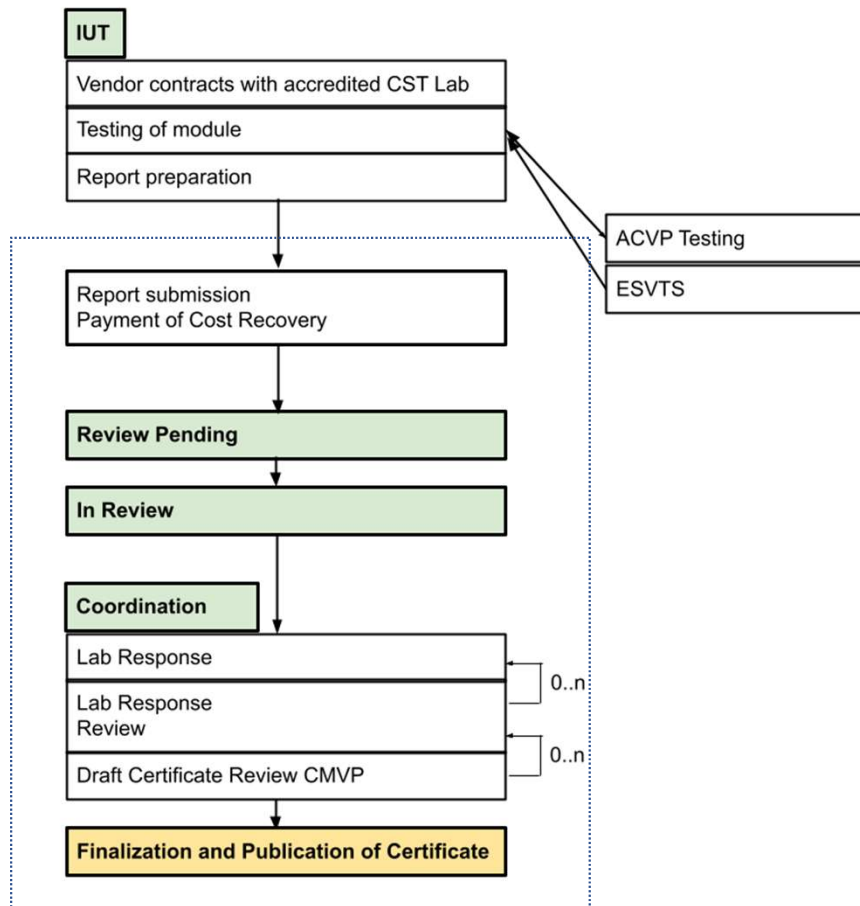
A cryptographic module is any software, hardware, hybrid, system, etc. that has at least one approved security function (cryptographic algorithm), such as encryption, authentication, digital signatures, key exchange

Vendors, Labs, and CMVP

- Vendors of cryptographic modules use independent, **NVLAP-accredited Cryptographic and Security Testing (CST) laboratories** to test their modules. Over 20 labs worldwide.
- CST laboratories use the Derived Test Requirements (DTR), Implementation Guidance (IG) and other CMVP programmatic guidance to test conformance against FIPS 140.
- FIPS 140-3 and NIST SP 800-140 modify ISO/IEC 19790 and ISO/IEC 24759.



Module Validation Process Flowchart



- Green/Gold boxes correspond to Status on CSRC (IUT List or “Status” column on MIP List)
- CAVP and ESV processes managed separately, “out of band”; certs are pre-requisites
- A few states (status) not shown, e.g., “Hold”
- During “In Review” a module is checked to meet the requirements of FIPS 140-3, the appropriate algorithm standards, and the FIPS 140-3 Implementation Guidance

FIPS 140-3 Self-Test Requirements



- Key Generation has two separate self-tests required:
 - Pairwise consistency test (PCT) for each key-pair generated or imported, apply the encapsulation key ek to encapsulate a shared secret K leading to ciphertext c , and then apply decapsulation key dk to retrieve the same shared secret K .
 - Known answer test (KAT) typically run at start-up but must be before the key generation function is used in practice. ML-KEM is the first time this is a requirement for Key Generation functions.

More information found in [IG 10.3.A](#), Resolution #14 and Additional Comment #1

FIPS 140-3 Self-Test Requirements



- Separate Encapsulation and Decapsulation KAT self-tests are required:
 - Use one of: ML-KEM-512, ML-KEM-768, or ML-KEM-1024
 - Must be run prior to first usage but does not need to be on power up
 - For decapsulation, must cover *both* the implicit rejection and non-rejection paths

More information found in [IG 10.3.A](#), Resolution #14

Conclusion



Questions?

See our GitHubs

<https://github.com/usnistgov/ACVP-Server>

<https://github.com/usnistgov/ACVP>

CAVP Program Manager

Chris Celi

christopher.celi@nist.gov