Validation Testing for Block Cipher Modes

Third NIST Workshop on Block Cipher Modes 2023
10/4/2023
Chris Celi, CAVP Program Manager, NIST
christopher.celi@nist.gov

National Institute of Standards and Technology
U.S. Department of Commerce

FIPS 140-3

- 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) in order 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

- **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

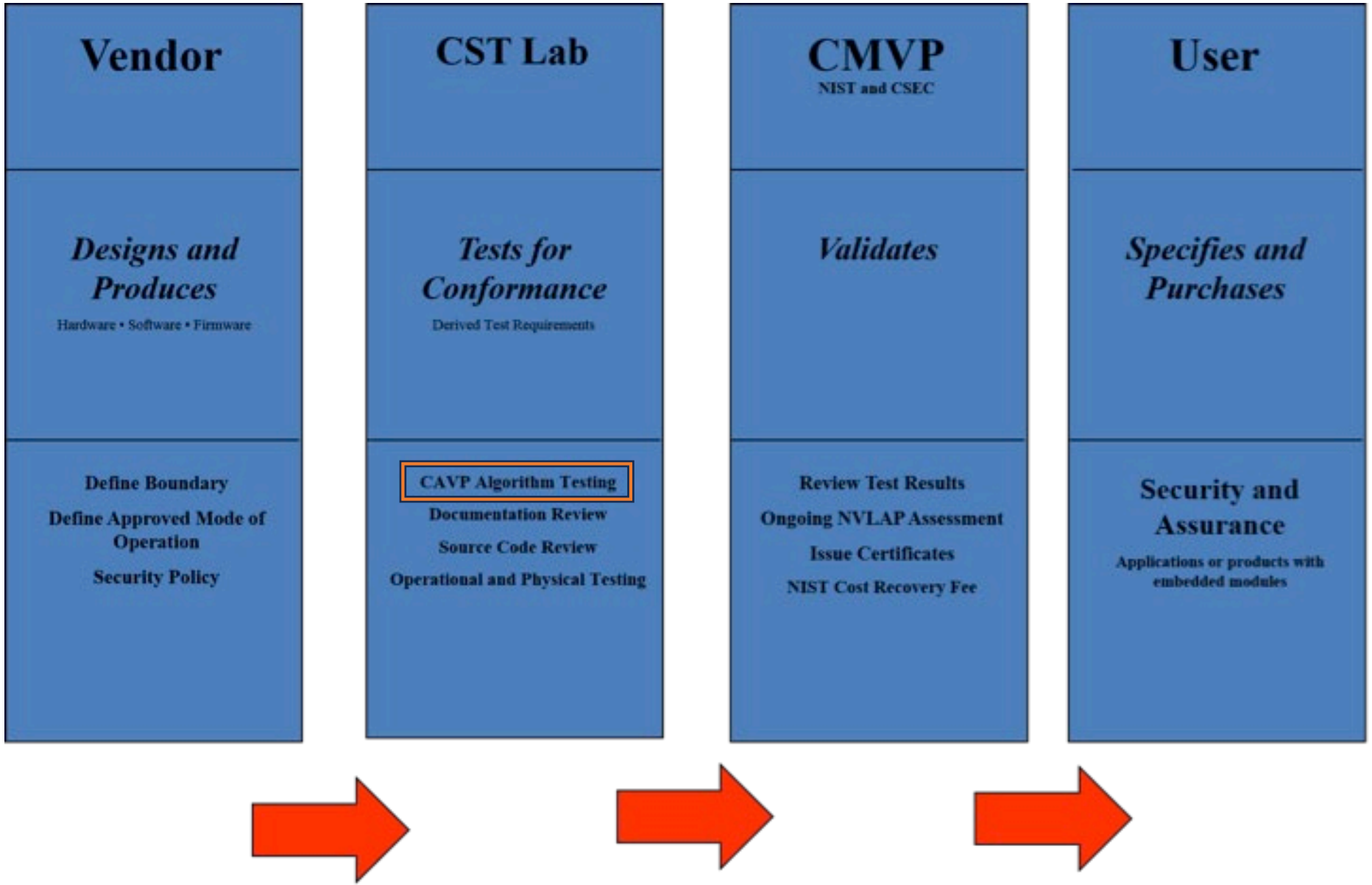
A cryptographic module is any software, hardware, hybrid, system, etc. that has at least one approved security function (cryptographic algorithm), such as encryption, authentication, digital signatures, key exchange…

Vendors, Labs, and CAVP

- Vendors of cryptographic modules use **NVLAP-accredited 17ACVT laboratories** to test their algorithms.
- First-party labs may also be **NVLAP-accredited** to **17ACVT**
- All testing happens on the NIST-hosted Automated Cryptographic Validation Test System (ACVTS)

https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program

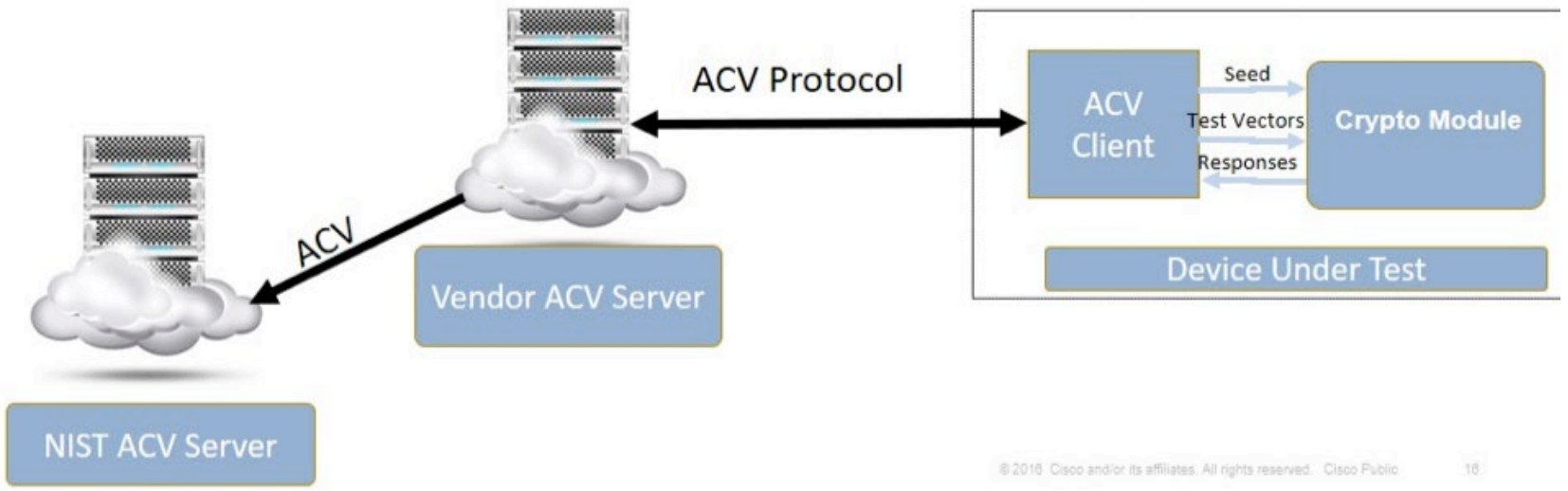
Validation Process



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 approach

ACVTS

- 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 1,850,000 vector sets served!

ACVTS – Block Cipher Modes

- Covers all modes in SP 800-38 series
- Apply Known Answer Tests (KATs)
  - Cycle through key bits, plaintext bits, ciphertext bits, S-boxes to cover everything
- Apply randomly generated tests
  - Generate random key, plaintext compute ciphertext
- Apply Monte Carlo Tests (MCTs)
  - Chain encrypt/decrypt calls hundreds of thousands of times

ACVTS – Block Cipher Modes – CTR

- These tests are good, but don't always cover interesting parts of the algorithm…
- Until 2017, the CAVP would ask "how the counter mechanism was set up" for CTR modes
  - Most common response was an empty text box
- CAVP developed "Counter tests" to provide assurances of the requirements


```
{
  "direction": "encrypt",
  "keyLen": 128,
  "testType": "CTR",
  "tests": [{
    "tcId": 829,
    "pt": "CE8E4B6F7C68DE5FDE3...",
    "iv": "00000000000000000000000000000039",
    "key": "3A9A8485E1B7BA1987F88F8C095257C4"
  }]
}
```

ACVTS – Block Cipher Modes – CTR

- This has lots of issues due to SP 800-38A
- "incrementing function **can** ensure [uniqueness]", Appendix B
  - Incrementing function definition?
  - Linear function? Wrap-around with random start? What bits?
- Uniqueness over 2^m messages is not testable


```
{
  "direction": "encrypt",
  "keyLen": 128,
  "testType": "CTR",
  "tests": [{
    "tcId": 829,
    "pt": "CE8E4B6F7C68DE5FDE3...(long)",
    "iv": "00000000000000000000000000000039",
    "key": "3A9A8485E1B7BA1987F88F8C095257C4"
  }]
}
```

ACVTS – Block Cipher Modes – CTR

- Would be helpful to…
- Define standard incrementing functions
  - Over full 128-bits, over 32-bits, etc.
- Could still allow other incrementing functions, but require a standardized one for testing
  - Similar to RSA key sizes


```
{
  "direction": "encrypt",
  "keyLen": 128,
  "testType": "CTR",
  "tests": [{
    "tcId": 829,
    "pt": "CE8E4B6F7C68DE5FDE3...(long)",
    "iv": "00000000000000000000000000000039",
    "key": "3A9A8485E1B7BA1987F88F8C095257C4"
  }]
}
```

ACVTS – Block Cipher Modes – GCM

- Initialization vectors uniqueness requirements are difficult to test
  - Require special acknowledgement in the CMVP submission

- Implementations often don't allow an IV to be provided
  - Around 40%
  - Leads to difficulty in testing the algorithm

"Normal" Test Cases

Test system generates all values up-front, and sends IV, plaintext, and key to implementation

"Deferred" Test Cases

Test system provides plaintext and key

Need the implementation to generate an IV and provide it back to the test system

ACVTS – Block Cipher Modes – XTS

- Payload length versus data unit length…

- Payload length – total length of the data to be encrypted
- Data unit length – length of the “chunks” to be encrypted at a time
- Block length – 128 bits, AES

- What if…
  - Payload length is not be divisible by data unit length?

ACVTS – Block Cipher Modes – XTS

- XTS is defined as a stream cipher

- Payload length = 576 bits
- Data unit length = 512 bits

- Data units are processed, but the last payload is only 64 bits
- Encryption of a data unit containing less than one AES block is not defined!

Conclusion

Questions?

See our GitHub
https://github.com/usnistgov/ACVP-Server

CAVP Program Manager

Chris Celi

christopher.celi@nist.gov