Automated Crypto Validation Protocol (ACVP)

Apostol Vassilev (NIST), David McGrew (Cisco)

IETF 102 SAAG
July, 2018 (Montreal, Canada)

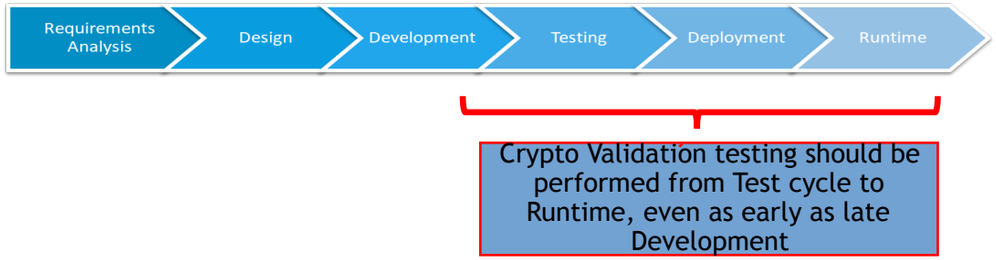
Background

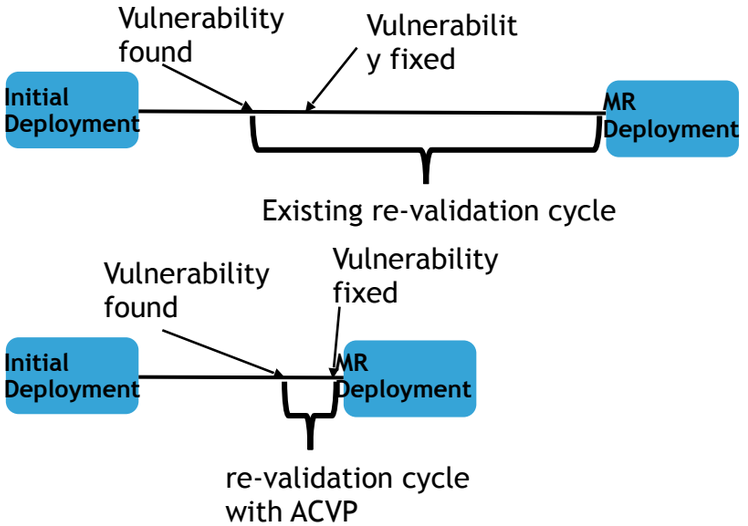
- **Cryptographic Module Validation**
  - What algorithms, key sizes, modes, and parameters are supported by the module?
  - How can I provide/gain assurance of correctness?
  - How can I re-validate a module after a software update?

- **FIPS-140 Cryptographic Module Validation Program (CMVP)**
  - Implementer provides description of algorithms, key sizes, parameters, modes to CMVP (as PDF)
  - CMVP provides test cases (as text file)
  - Implementer provides outputs to CMVP (as text file)
  - CMVP notifies implementer if there are inconsistencies – repeat until none
  - CMVP posts validation certificate (see NIST CMVP Certificate 3016 for example)

What problems does ACVP solve?

- **Typical crypto validation programs are not meeting implementer needs**
  - long review cycles, well-beyond industrial product development cycles
  - rigid procedures that prevent rapid updates





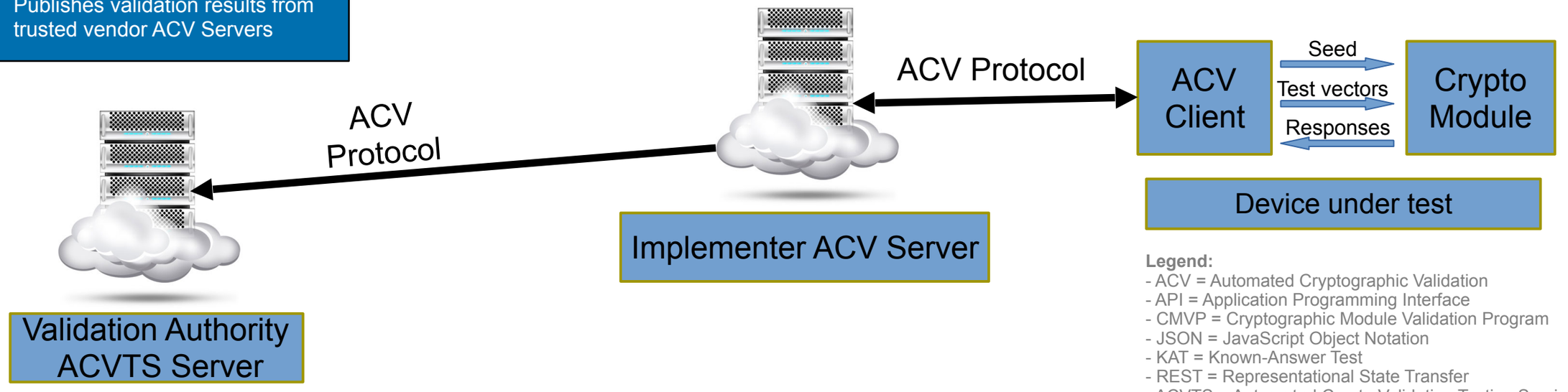
- **Crypto implementers face multiple validation authorities**
  - each with different rules and procedures
  - … but still slow and rigid

Building a New Crypto Validation Program

ACV Proxy/Server:
- Web hosted service
- Interacts with NIST ACV Server to obtain JSON KAT data
- Optionally generates JSON test vectors
- Optionally performs results verification
- Reports JSON KAT results to NIST ACV Server

ACV Client:
- Integrated into Device under test
- May convert JSON test vectors to format acceptable by crypto module under test
- Returns KAT answers to ACV server in JSON format

Validation Authority Server:
- Web hosted service w/ REST API
- Registers ACV Servers
- Generates JSON KAT vectors
- Validates JSON KAT results
- Publishes validation results from trusted vendor ACV Servers



Legend:
- ACV = Automated Cryptographic Validation
- API = Application Programming Interface
- CMVP = Cryptographic Module Validation Program
- JSON = JavaScript Object Notation
- KAT = Known-Answer Test
- REST = Representational State Transfer
- ACVTS = Automated Crypto Validation Testing Service

Computer-based testing and validation

Where are we today?

- **ACVP development collaboration - https://github.com/usnistgov/ACVP**
  - idea and first prototype by David McGrew (Cisco)
  - working group led by Barry Fussell (Cisco)
  - currently working on v0.5, last deployed v0.4
  - demo server at https://demo.acvts.nist.gov/acvp/home
    - ReadMe provides a ton of useful information, e.g. currently covers **90+** algorithms/modes
  - targeting v1.0 in Q3, 2018
  - specs already in RFC format
  - open client implementation libacvp by Cisco

- **Future enhancements – contributions from industry and academia**
  - project Wycheproof (Google) – deeper testing of crypto libraries against known attacks
  - HACL* project (Prosecco team @ INRIA Paris & Microsoft Research) – formal verification of crypto implementations
  - extended test coverage of SP 800-56B key-agreement schemes (IPA, Japan)

See also a high-level public project plan at http://csrc.nist.gov/projects/acvt/ for further details

ACVP Features

- **Defines**
  - a transport
    - based on HTTP or HTTPS
  - an encoding and message format
    - which is negotiated
  - a set of message exchanges
- **Works over the Internet where the testing system is remote from the cryptographic module**
  - e.g. running as a process on a separate device and enables automated cryptographic algorithm testing.
- **Enables the discovery of the capabilities of the module being tested**
- **Generates corresponding tests**
  - enables also the request/response exchanges between the testing server and the tested module

ACVP Features (continued)

- **Provides a standard communication method**
  - implementers of cryptographic technology can potentially utilize the same testing service for validating algorithms in multiple validation programs
    - operated by different governments
    - or private sector organizations.
- **Provides extensibility that can be used to introduce:**
  - tests for new algorithms
  - new tests for existing algorithms
  - new protocol features w/o changing algorithm tests

Why ACVP with IETF?

- **Openness and transparency of crypto standards and validation methodologies are necessary for acceptance**

- **Global open standards facilitate international adoption**
  - very important for the industry as the need for crypto validations spreads around the world
  - other nations can host own validation servers
  -
    - using common protocols and testing methodologies for same algorithms
    - this does not mean all nations use the same algorithms
    - however, if an algorithm is used by more than one nation, e.g. AES, the testing methodology should be the same
    - based on state-of-the-art crypto testing

ACVP Next Steps

- **Consider publication of version 1 of ACVP with NIST algorithm tests as an Informational RFC**
  - goal is submission before IETF 103

- **NIST's Cryptographic Algorithm Validation Program will transition to ACVP**
  - available in Q4, 2018 (tentative)
  - required in Q3, 2019 (tentative)

- **Several non-US validation programs are considering adopting ACVP**

- **After gaining initial experience, NIST intends to transition change control for ACVP to an appropriate standards body**
  - we believe the IETF would be one appropriate SDO

Our Asks

- **Please join us at a side meeting (Thursday 19:30-21:30, Van Horne room) for:**
  - presentation of in-depth details of ACVP
  - demonstration of ACVP
  - additional discussion of future collaboration opportunities regarding ACVP
- **Join the new email list for discussion of ACVP-related topics (acvp@ietf.org)**
  - how to position ACVP within the IETF for future standardization work?
- **Our long term Asks:**
  - if you are planning for future crypto validations consider incorporating ACVP in your plan
  - when standardizing algorithms (e.g., new ECC curves) consider developing ACVP extensions

Questions?