Inside the World of Cryptographic Algorithm Validation Testing

Sharon Keller

CAVP Program Manager

NIST

ICMC, May 2016

Mission

- To provide federal agencies—in the United States and Canada—with assurance that a cryptographic algorithm has been implemented according to the specifications in the applicable standard.

History

- Established in 2003 by NIST and the Communications Security Establishment (CSE) of Canada as a separate program from the CMVP.
- Cryptographic algorithm validations are a prerequisite to cryptographic module validations.

Algorithm Validation Testing
Determining Algorithms to be validated

- NIST's Cryptographic Technology Group's Role
  - Determine which cryptographic algorithms become recommendations and standards.
  - Analyze the algorithm and finalize the specifications for the algorithm

Algorithm Validation Testing
Determining Algorithms to be validated

- CAVP develops validation test suites for approved cryptographic algorithms and their complete functions defined in applicable FIPS or NIST Recommendations (published as NIST Special Publication).

Algorithm Validation Testing

Testing Philosophy

- Validation test suites are designed to test all specifications (addressable at the algorithm level) within a complete standard or a complete function within that standard
- The complete algorithm implementation must be contained within a single cryptographic algorithm boundary

Algorithm Validation Testing
Philosophy "Speedbump"

- But situations exist where a complete implementation or function (as specified in the standard) isn't implemented within one boundary.

- Example: PIV cards
  - Limited processing
  - Limited memory

- Could not fit a complete algorithm or function (as specified in the standard) on the card (or in a single boundary)
- Some of algorithm on PIV card, some on reader, some on PC

Algorithm Validation Testing
Philosophy update: Component testing

- "Better to test something instead of nothing at all."
- Introduced in 2011
- Provides assurance of the individual components of an algorithm when the complete algorithm or function isn't contained in a single boundary.
- Testing components of a standard - a function or part in a standard – not a mathematical function.
- If there is a need for a component test, contact me to get it added.

Algorithm Validation Testing
Component testing

- Example:
  - ECDSA Signature Generation function
    - Steps in function:
      - Hash message
      - Sign hash
  - On a PIV card:
    - Hashing of message done off card
    - Signature of hash done on card
  - Created ECDSA Signature Generation Component that takes a hash-length input and signs it.

CAVP Validation Testing

Complete Algorithms and Components Currently Available

- Symmetric Algorithms
  - AES (FIPS 197)
  - Triple DES (ANSI X9.52-1998)

- Asymmetric algorithms
  - Both FIPS 186-4 and Legacy functions of 186-2
    - DSA
    - ECDSA
    - RSA

- Modes of Operation
  - CMAC (SP 800-38B)
  - CCM (SP 800-38C)
  - GCM/GMAC(SP 800-38D)
  - XTS-AES(SP 800-38E)
  - Methods for Key Wrapping (SP800-38F)

- SHS (FIPS 180-4)
- SHA-3 (FIPS 202)
- DRBG (SP 800-90A)
- Key Agreement Schemes (SP 800-56A)
- HMAC (FIPS 198)
- SP800-108 KDF

- Testing of Components
  - All SP 800-56A except KDF
  - SP800-56A ECCCDH Primitive
  - RSASP1 - Mod Exp for Sig Gen (PKCS1.5 and PKCS-PSS)
  - RSADP - Basic Decrypt Operation
  - KDFs in SP800-135-IKE, TLS, SSH, SNMP
  - ECDSA Sign Gen Component (excludes hash)

Algorithm Validation Testing
How Component testing is used

- **CMVP requires all applicable components to be tested by CAVP for a specific algorithm**
- **Example 1:**
  - **Module includes implementation of SP800-56A. This implementation uses the TLS KDF from SP800-135.**
  - **The vendor would need to supply 2 validated component tests (CVL) :**
    - **All SP 800-56A except KDF**
    - **TLS KDF in SP800-135**
- **Example 2:**
  - **Module includes ECDSA Signature Generation but the implementation is not contained within one algorithm boundary**
  - **The vendor would need to supply 2 algorithm validation tests to validate:**
    - **SHA validation**
    - **Component test for ECDSA Signature Generation Component**

Algorithm Validation Testing Development Process

Review the requirements and algorithmic specifications in the cryptographic algorithm documents (SP, FIPS)

- **Identify the algorithm's:**
  - **Components**
  - **Functionality**
  - **Mathematical formulas**

Algorithm Validation Testing Development Process

Develop and implement the algorithm validation test suite

- Identify the requirements addressable at the CAVP level
- Develop the test metrics for testing the algorithm
- Exercise all mathematical elements of the algorithm
- Assure the specifications in the standard have been implemented correctly
- If deviates, validation test will fail indicating implementation flaw

Algorithm Validation Testing Development Process

Develop User Documentation and Guidance

- Validation System Document (VS)
  - Documents test suite
  - Provides instructions on implementing validation tests
- CAVP Frequently Asked Questions (FAQ) document
  - Provide guidance on
  - General algorithm issues
  - Specific algorithm questions
    - Algorithm interpretation
    - Validation tests
    - Special cases
    - Etc.

Algorithm Validation Testing Development Process

Manage Validation Information

- CAVP Validation Database development
- Automation of data entry
- Automation of webpage production

- CAVP Validation Database features

  - Table for every algorithm
  - Report production

Algorithm Validation Testing
Types of validation tests

Known Answer Tests (KAT)

- Designed to verify the components of algorithms (Sboxes, permutation tables, etc)

Algorithm Validation Testing
Types of validation tests

Multi-block Message Test (MMT)

- Designed to test the ability of the implementation to process multi-block messages, which may require chaining of information from one block to the next

- The test supplies the IUT with messages that are integral numbers of blocks in length

Algorithm Validation Testing
Types of validation tests

Monte Carlo Test (MCT)

- Designed to exercise entire implementation.

- Purpose: to detect the presence of flaws in the IUT that were not detected with controlled input of KATs

Algorithm Validation Testing
Types of validation tests

Positive testing

- The testing process where the implementation is validated against valid input data. In this testing, only valid sets of values are supplied and the results are checked to see if the expected results are generated.
- Goal: to prove that a given implementation has adhered to the specifications and requirements in the standard
- Used for PQG generation, signature generation, etc. where some values are given and the IUT has to compute the answer.

Algorithm Validation Testing
Types of validation tests

Negative testing

- The testing process where the implementation is validated against invalid input data. A negative test checks if an implementation behaves as expected with its negative inputs.
- Goal: to test the ability of the IUT to recognize valid and invalid values.
- Usually used for PQG Verification, Signature Verification, etc. where all values are provided and the IUT is verifying the correct result is achieved. Errors are introduced into the different parameters to assure the IUT can recognize the errors

Algorithm Validation Testing
Example of test suites and tests used

AES

- Known Answer Tests
  - GFSbox
  - KeySbox
  - Variable Key
  - Variable Text
- Multi-block Message Test
- Monte Carlo Test

Algorithm Validation Testing
Example of test suites and tests used

RSA

- Key Generation
  - Positive testing
- Signature Generation
  - Positive testing
- Signature Verification
  - Negative testing

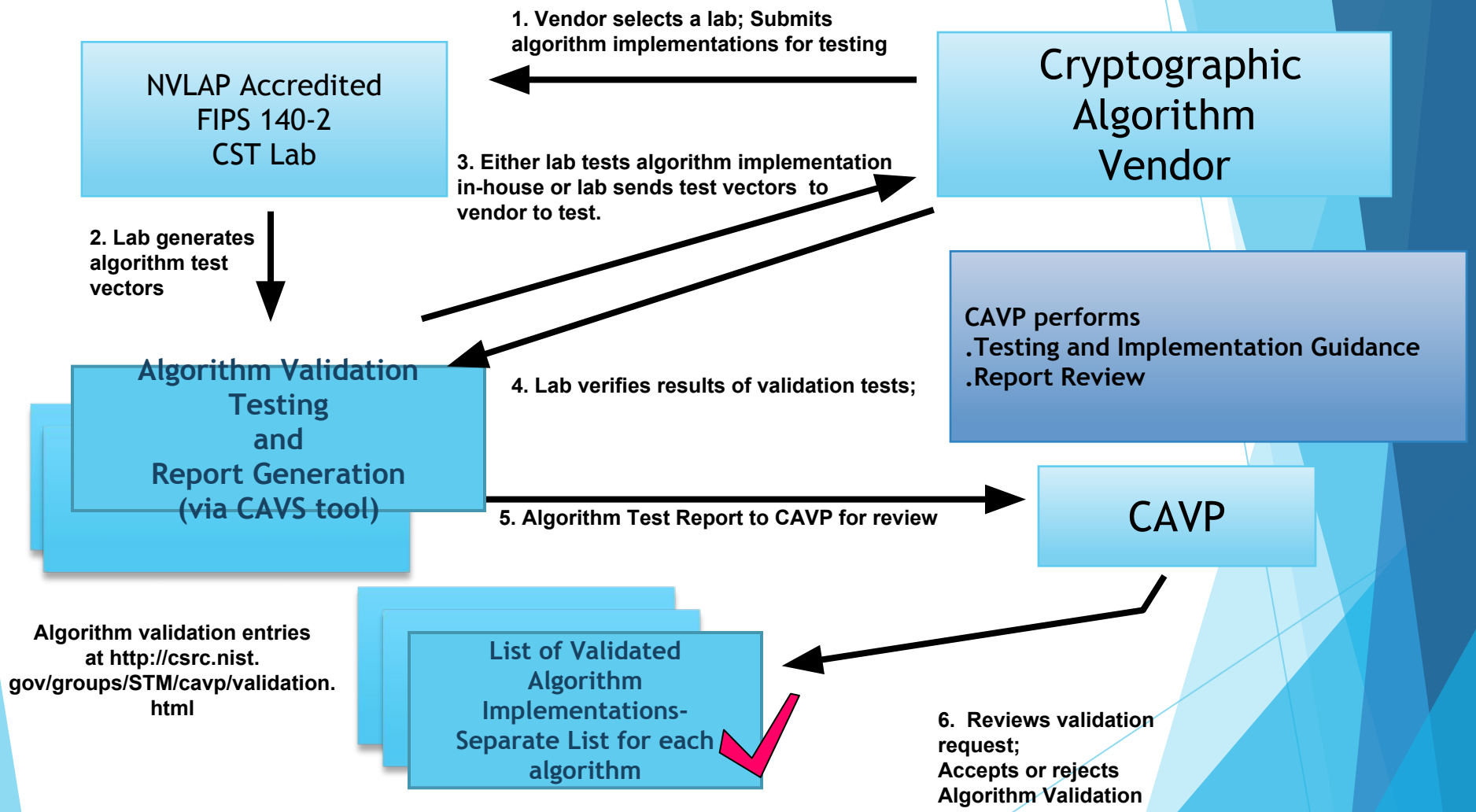
Algorithm Validation Testing
Example of test suites and tests used

SHA-3

- Options tested
  - SHA3-224, 256, 384, 512
  - Bit or Byte Orientation
  - Support zero length messages or not

- Test suite:
  - Short Messages Test
  - Selected Long Messages Test
  - Monte Carlo Test

CAVP Test and Validation



Algorithm Validation Testing
Test Suites in Progress

- SP 800-56C Key Derivation through Extraction-then-Expansion
- SP 800-132 Password-Based Key Derivation Part 1: Storage Applications
- Adding SHA-3 to HMAC, DSA, ECDSA, RSA, KAS

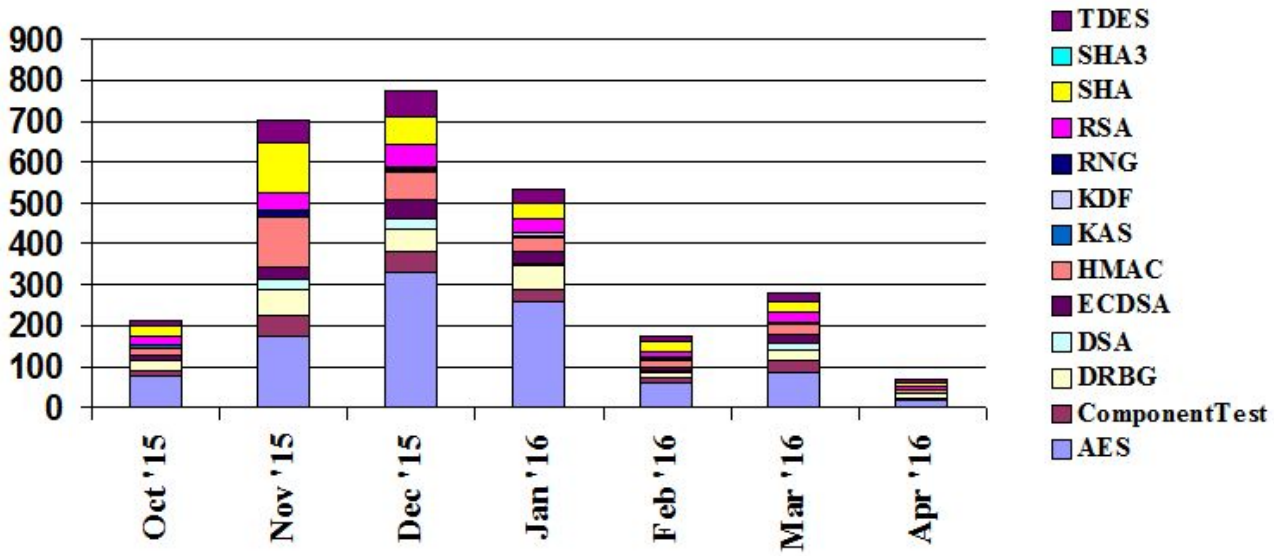
Algorithm Validation Testing
Future Algorithm Testing

- SP800-38G Methods for Format-Preserving Encryption
- SP 800-56B (Rev 1): Key Agreement Schemes with RSA – June 2014
- SP 800-106 Randomized Hashing for Digital Signatures
- SP800-38A Addendum Block Cipher Mode 3 variants of CT Stealing for CBC Mode
- SP800-56A (Rev 2) Key Agreement Schemes with DSA and ECDSA
- SP800-90A (Rev 1)DRBG

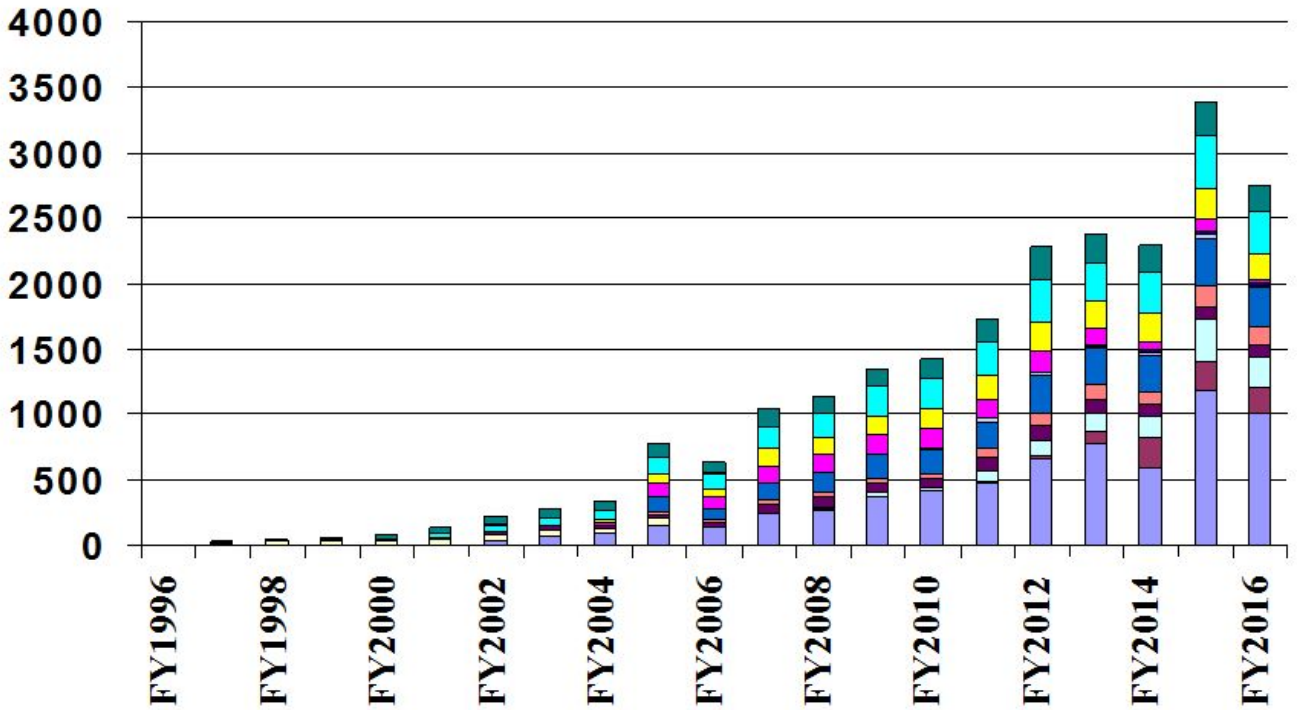
Algorithm Validation Testing
Future Algorithm Testing (In Draft)

- SP800-90B Draft : Entropy Sources
- SP800-90C Draft: construction of RBGS

CAVP Validation Status For FY16



CAVP Validation Status By FYs



CAVP Validated Implementation Actual Numbers

Updated As: Tuesday, April 12, 2016

FiscalYear	AES	Comp.	DES	DSA	DRBG	ECDSA	HMAC	KAS	KDF	RNG	RSA	SHA	SJ	TDES	Total
FY1996	0	0	2	0	0	0	0	0	0	0	0	0	0	0	**2**
FY1997	0	0	11	6	0	0	0	0	0	0	0	7	2	0	**26**
FY1998	0	0	27	9	0	0	0	0	0	0	0	6	0	0	**42**
FY1999	0	0	30	14	0	0	0	0	0	0	0	12	1	0	**57**
FY2000	0	0	29	7	0	0	0	0	0	0	0	12	1	28	**77**
FY2001	0	0	41	15	0	0	0	0	0	0	0	28	0	51	**135**
FY2002	30	0	44	21	0	0	0	0	0	0	0	59	6	58	**218**
FY2003	66	0	49	24	0	0	0	0	0	0	0	63	3	73	**278**
FY2004	82	0	41	17	0	0	0	0	0	28	22	77	0	70	**337**
FY2005	145	1	54	31	0	14	115	0	0	108	80	122	2	102	**774**
FY2006	131	1	3	33	0	19	87	0	0	91	63	120	1	83	**632**
FY2007	238	5	0	63	0	35	127	0	0	137	130	171	1	136	**1043**
FY2008	271	7	0	77	4	41	158	0	0	137	129	191	0	122	**1137**
FY2009	373	2	0	71	23	33	193	6	0	142	143	224	1	138	**1349**
FY2010	406	2	0	70	31	39	179	12	0	150	155	239	0	142	**1425**
FY2011	474	11	0	102	79	68	201	34	0	148	183	255	0	177	**1732**
FY2012	654	24	0	121	122	92	283	20	3	157	231	323	1	248	**2279**
FY2013	778	88	0	106	145	113	276	12	9	132	208	293	0	217	**2377**
FY2014	595	223	0	95	167	96	276	14	23	63	225	314	0	196	**2287**
FY2015	1179	226	0	99	320	164	355	32	35	80	243	396	0	258	**3387**
FY2016	1008	191	0	83	247	142	294	24	18	23	192	324	0	200	**2749**
Total	**6430**	**781**	**331**	**1064**	**1138**	**856**	**2544**	**154**	**88**	**1396**	**2004**	**3236**	**19**	**2299**	**22343**

Questions??

Sharon Keller
CAVP Program Manager
NIST

sharon.keller@nist.gov

(301)975-2910