Rethinking the Definition of the Operational Environment in FIPS 140-2

Kelvin Desplanque

Compliance Engineer

17 May, 2017

“*Delays have dangerous ends.*” ?

- The definition of the Operational Environment in FIPS 140-2 is somewhat vague, and even with the help of various IGs, and FAQs is still something which is far too often left up to the individual interpretation of a vendor, its FIPS lab, or the CMVP evaluators.
- This inconsistency has its consequences:
  - Evaluations delays resulting from disagreements between the parties.
  - As a result of being overly cautious, too many OEs are potentially identified and tested wasting both vendor, lab and CAVP/CMVP resources.
  - Vendors may be discouraged from performing FIPS validations as a result of this uncertainty and the potential accompanying additional expenses.

Where is the OE best defined?

- See the CAVP FAQ document at http://csrc.nist.gov/groups/STM/cavp/documents/CAVPFAQ.pdf
- GEN.12 - *What information is required in the Operational Environment field?*

Where is the OE best defined?

- ***Processor*** – This field shall identify the vendor and processor family.

  No further specificity is required unless the vendor or the lab knows that the software implementation executes differently on different processors within the same family.

- ***Operating System*** - This field shall identify the vendor and operating system family, or major version number where more appropriate.

  No further specificity is required unless the vendor or lab knows that the software implementation executes differently on different OSes within the same OS family or major version number.

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

	Level 1	Level 2	Level 3	Level 4
Software			■	■
Firmware				
Hardware				
Hybrid				

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

	Level 1	Level 2	Level 3	Level 4
Software			■	■
Firmware				
Hardware				

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

	Level 1	Level 2	Level 3	Level 4
Software			■	■
Firmware				

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

	Level 1	Level 2	Level 3	Level 4
Software			■	■

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

	Level 1	Level 2
Software		

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

	Level 1
Software	

IG 1.16 Software Module - A software module is a cryptographic module implemented entirely in executable or linked code executing in a ***modifiable*** operational environment.

A ***modifiable*** operational environment refers to an operating environment that may be reconfigured to add/delete/modify functionality, and/or may include general purpose operating system capabilities (e.g., use of a computer O/S, configurable smart card O/S, or programmable firmware). Operating systems are considered to be modifiable operational environments if software/firmware components can be modified by the operator and/or the operator can load and execute software or firmware (e.g., a word processor) that was not included as part of the validation of the module.

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

- Let’s only concern ourselves with the validation of Software Modules in a Level 1 Operational Environment (when operating on any general purpose computer).
- This will then be a porting exercise, that is there are no changes, additions or deletions of source code. Pure and simple recompilation of the source code may be required to run on another OE.

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

- In the IG, the following text is most important:

  > *The CMVP allows vendor porting and re-compilation of a validated software, firmware or hybrid cryptographic module from the operational environment specified on the validation certificate to an operational environment which was not included as part of the validation testing as long as the porting rules are followed. Vendors may affirm that the module works correctly in the new operational environment.*

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

- But there is a very important caveat:

> *However, the CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.*

IG G.5 Maintaining validation compliance of software or firmware cryptographic modules

- So what happens if you add a number of OEs but actually have a customer that requires that all of these OEs are listed on the CMVP validation certificate.
  - Additional algorithm testing to cover all of the new OEs.
  - Functional testing performed by a FIPS lab for all of the new OEs.
  - Submission of a revised Security Policy, VE, and Test Report to the CMVP in the form of a 1SUB (See IG G.8 – Revalidation Requirements).
  - Wait …

So let’s now consider the following hypothetical scenario …

- Your product development team has created a secure communications application and would like it to be FIPS certified.
- They say they will create both desktop and mobile versions.
- The desktop versions can run on 3 flavors of MS-Windows and 2 of macOS.
- The mobile version has to run on 6 versions of Android, 3 versions of Apple mobile iOS, and 2 versions of Windows mobile O/S.
- The desktop version should cover 90% of that CPU market.
- The mobile version should cover 95% of that CPU market.

So what does the math tell us?

- You do a bit of market research and find out to cover 90% of the CPU market for desktop for Windows implies 20 different CPUs and for Apple 6 different CPUs.
- For mobile platforms you need 10 CPUs for Android, 6 for macOS, and 4 for Windows mobile.
- This results in the following equation:

$$[(20 \times 3) + (6 \times 2)] + [(10 \times 6) + (6 \times 3) + (4 \times 2)] = \mathbf{158}$$

- That’s a considerable amount of algorithm and functional testing.

What else might you have to consider?

- The application software would be required to query its host platforms O/S and CPU information and prevent it from executing if it determined that the OE combination did not falls within the bounds of the tested and validated list of OEs (for strict FIPS validation compliance).

What might be a solution?

- Let’s assume that the five different builds of this software will operate equally well (are fully binary compatible) on all CPUs without the need for any code modification.
- The original equation:

  [(20 x 3) + (6 x 2)] + [(10 x 6) + (6 x 3) + (4 x 2)] = **158**

- … now becomes:

  [(1 x 3) + (1 x 2)] + [(2 x 6) + (1 x 3) + (1 x 2)] = **22**

- That’s 80% less testing !!!!

Then what is the problem with this solution?

- Current CMVP policy on OEs does not allow for this approach.
- In recent months, the CMVP has been ‘cracking’ down on the way in which some vendors have been identifying their OEs.
- Late last year, the CMVP Program Manager, sent e-mails to the labs (which were subsequently forwarded to the vendors) clarifying the CMVP’s position on identification and naming of OEs. The Program Manager has been quoted as saying the following:

  “Intel Xeon” – many examples – there are over 30 Xeon families on ark.intel.com”

  “IBM PowerPC” – there are several different IBM PowerPC families”

But why make the distinction anyway?

- Let’s take a Level 1 Software Cryptographic module that was compiled for a current 64-bit Intel CPU (or AMD for that matter).
- The binary code, resulting from the compilation, would most likely execute correctly on every one of these Intel and AMD 64-bit CPUs.
- There is no question that different families, architectures, and microarchitectures of these CPUs have been implemented differently but the opcodes which execute the cryptographic code are essentially like black boxes, providing the same output given the same input.

Not implying that different CPUs are all the same

- Intel and AMD have added various extension instruction sets to the core x86 and x64 instructions (AVX, SSE, etc.)
- Depending on the compiler (and how it configured) some of these extended instructions sets might very well be used in the computation of certain cryptographic operations.
- The AES-NI instruction set extension, which was specifically designed for cryptographic acceleration, has long been recognized by the CMVP and binaries which employ it must test this separately [see IG 1.21 *Processor Algorithm Accelerators (PAA) and Processor Algorithm Implementation (PAI)*].

So what should the CMVP recognize?

- That certain cryptographic module binaries designed for a specific processor will run equally well (without modification or recompilation) on a rather wide range of CPUs.
- The same can be said for the underlying operating systems.

How could the CMVP accommodate this?

- Develop a mapping of the more common CPUs that are functionally identical (those which possess the same core and extended instruction sets).
- This could be done as a cooperative venture with leading FIPS vendors and the major CPU manufacturers.
- Share this data on the CMVP website.

What would the PROs and CONs of this be?

PRO	CON
Reduce amount of time and resources tied up in testing unnecessary OE combinations.	Slight possibility that reducing the number of tested OEs in a submission might allow a rogue OE to be validated when it fact it should not have been.
Free-up resources within the CMVP to do other more security relevant evaluations.	

Now this is odd …

- Take a look at a somewhat old, yet still valid IG (2011):
  IG 1.4 Binding of Cryptographic Algorithm Validation Certificates
- Under the ***Additional Comments*** section, part 2, it states:

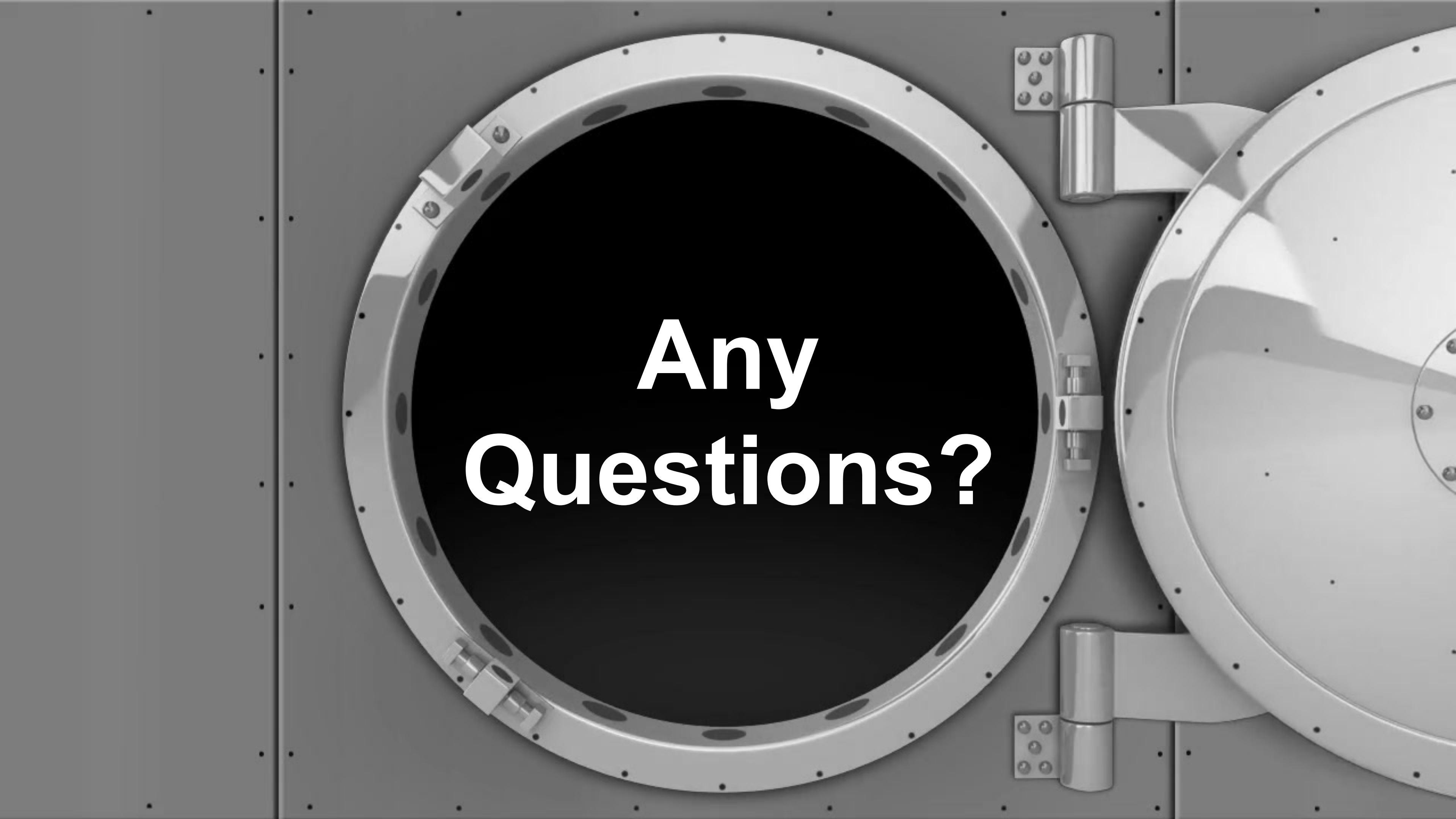
  If an implementation has been tested on one processor, can a claim be made that the implementation also runs on a different processor when it is submitted for module testing?

  *The answer to this question is dependent on the security assurance Level of the module validation and on whether or not the two processors are **architecturally compatible** or not.*

  If the module is being validated as a Level 1 validation and the two processors are architecturally compatible platforms, the answer is Yes.

And on a final note…

- It is possible that the introduction of the *Automated Cryptographic Validation Program* (ACVP) will radically change the way and the speed by which algorithms are tested. (http://csrc.nist.gov/projects/acvt/)



Any Questions?