Network Device Interpretation # 3, Revision 2

Requirements on destruction of cryptographic keys

Status: ☒ *Active* ☐ *Inactive*

Date: *14-Nov-2016*

Type of Document: ☒ *Technical Decision* ☐ *Technical Recommendation*

Approved by: ☒ *Network iTC Interpretations Team* ☐ *Network iTC*

Affected Document(s): *NDcPP V1.0, FWcPP V1.0, ND SD V1.0*

Affected Section(s): *FCS_CKM.4*

Superseded Interpretation(s): *RfI#3 (first Revision), February 22nd 2016.*

Issue:

When using SSDs in a device that shall be evaluated according to NDcPP,

FCS_CKM.4 requires:

" For non-volatile flash memory, the destruction shall be executed by

[selection: a single, direct overwrite consisting of zeroes, a block erase] followed by a read-verify.

If the read-verification of the overwritten data fails, the process shall be repeated again."

With current technology, for a number of SSDs a block erase command would clear large parts or the entire disk which is not desirable when deleting a file containing a key. In addition, numerous SSDs do not allow bypassing the wear-leveling to allow a single direct overwrite of blocks.

Thus, both allowed mechanisms in the PP are unusable. However, some vendors of SSDs encrypt all data stored on the SSD -- as provided with the attached reference, Intel does that for some of their enterprise SSDs. Now, is it permissible that for zeroization of key data stored in a file that the file is overwritten with zeros by the operating system followed by a deletion operation? The following rationale would apply:

* When overwriting the file contents with zeros, a disk dump via the operating system disk driver will only return zeros for the blocks that used to be the file.

* It is known that the wear-leveling may not physically overwrite the data on disk when the operating system overwrites the file contents. When overwriting a file, the wear leveling mechanism will not allow

a user accessing the SSD via the controller to access the old data on the disk as the blocks holding the old data are not mapped by the wear-leveling. Therefore, a user accessing the SSD via its regular interface will not be able to get hold of it.

* Considering that the SSDs in question have an encryption schema implemented, an attacker cannot de-solder the flash chips and thus physically disconnect the chips from the controller implementing the wear leveling logic, because the controller has the encryption key. If an attacker would de-solder and access the flash chips, he can only access encrypted data. Therefore, the data that may still reside on the physical flash chips after the zeroization operation are not accessible.

Can we assume that interpretation TD0057 applies to this protection profile as well, so the read-verify is removed?

Resolution:

NIT sees the need to keep the wording for FCS_CKM.4 technology neutral. The wording of TD0057 is regarded as not sufficiently technology neutral for an SFR. NIT sees the risk of excluding some types of technologies due to the technology specific requirements. Therefore NIT proposes to change the wording for FCS_CKM.4as follows.

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- *For plaintext keys in volatile storage, the destruction shall be executed by a [selection: single overwrite consisting of [selection: a pseudo-random pattern using the TSF's RBG, zeroes, ones, a new value of the key, [assignment: a value that does not contain any CSP]], destruction of reference to the key directly followed by a request for garbage collection];*

- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [selection:*

    - *logically addresses the storage location of the key and performs a [selection: single, [assignment: number of passes]-pass] overwrite consisting of [selection: a pseudo-random pattern using the TSF's RBG, zeroes, ones, a new value of the key, [assignment: a value that does not contain any CSP]];*
    - *instructs a part of the TSF to destroy the abstraction that represents the key]]*

that meets the following: *No Standard*.

The NIT recommends adding the following Application Note to FCS_CKM.4.

In parts of the selections where keys are identified as being destroyed by "a part of the TSF", the TSS identifies the relevant part and the interface involved. The interface referenced in the requirement could take different forms for different TOEs, the most likely of which is an

application programming interface to an OS kernel. There may be various levels of abstraction visible. For instance, in a given implementation the application may have access to the file system details and may be able to logically address specific memory locations. In another implementation the application may simply have a handle to a resource and can only ask another part of the TSF such as the interpreter or OS to delete the resource.

Where different key destruction methods are used for different keys and/or different destruction situations then the different methods and the keys/situations they apply to are described in the TSS (and the ST may use separate iterations of the SFR to aid clarity). The TSS describes all relevant keys used in the implementation of SFRs, including cases where the keys are stored in a non-plaintext form. In the case of non-plaintext storage, the encryption method and relevant key-encrypting-key are identified in the TSS.

Some selections allow assignment of "a value that does not contain any CSP". This means that the TOE uses some specified data not drawn from an RBG meeting FCS_RBG_EXT requirements, and not being any of the particular values listed as other selection options. The point of the phrase "does not contain any CSP" is to ensure that the overwritten data is carefully selected, and not taken from a general pool that might contain current or residual data that itself requires confidentiality protection.

Key destruction does not apply to the public component of asymmetric key pairs.

Rationale:

N/A

Further Action:

- Approval by Network iTC required

- If approved, change FCS_CKM.4 and the corresponding sections in the ND Supporting Document accordingly.

Action by Network iTC: