DCI Digital Cinema Initiatives, LLC

Approved 16 March 2017
Digital Cinema Initiatives, LLC, Member Representatives Committee

DCI Memorandum on Digital Cinema Compliance with NIST SP800-56Br1

By Anthony Wechselberger and FIPS expert Travis Spann[1] consulting on behalf of DCI

Executive Summary: On September 1, 2015, DCI provided to SMPTE a memorandum regarding its investigation into the implications of SP800-56Br1 compliance, should NIST decide to enforce it. In May 2016, NIST announced that it would enforce SP800-56Br1. This document is an update to DCI's earlier FIPS memoranda regarding impact to Media Block (MB) designs and the Key Delivery Message (KDM).

In order to continue to rely upon NIST and FIPS standards and FIPS 140-2 validations, DCI compliant MBs must be compliant to NIST SP800-56Br1 as of January 2018. Contrary to DCI's September 2015 observations, a detailed examination has concluded that the existing structure of SMPTE KDMs as currently used is compliant and needs no changes. However, MB designs will be modestly impacted by other SP800-56Br1 requirements, as discussed below.

PART I – Detailed examination of SP800-56Br1

Problem Statement: In the eyes of NIST, the KDM is an "RSA key transport" mechanism. NIST issued SP800-131Ar1[2] in November 2015 proposing to disallow non-SP800-56Br1 compliant implementations of RSA key transport after December 31, 2017. Now approved, SP800-56Br1 compliance will be mandatory as of January 1, 2018, per current NIST guidelines.

Background: The existing use of RSA key encapsulation (key wrapping) by the media block (MB) via Key Delivery Messages (KDM) is "Allowed" in FIPS modules, but not "Approved." It means that the RSA key encapsulation in the KDM can be used in "FIPS-mode," but such usage is not Approved within a FIPS standard or NIST Special Publication (SP). The reason that it has been Allowed to be used in FIPS-mode is the clause in FIPS 140-2 IG D.9 which states:

"'Allowed' methods for key transport: Any key encapsulation scheme employing an RSA-based key methodology that uses key lengths specified in SP800-131A as acceptable or deprecated."[3]

Impact: The following are required to meet SP800-56Br1 mandates.[4]

A) The MB must use one of the "Approved" RSA key encapsulation methods from SP800-56Br1 for processing the KDM. DCI's recommendation is that Section 9.2.3 "KTS-OAEP-basic" be used. This scheme is recommended over the other methods because:

> i) "Key confirmation" as required by the other options is not needed in the D-Cinema use-case (*i.e.*, the KDM is already signed by the content author with RSA 2048).

[1] President and Laboratory Director, Aegisolve Inc.: http://www.aegisolve.com/
[2] See SP800-131Ar1: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf
[3] See FIPS 140-2 IG D.9: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf
[4] See SP800-56Br1: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Br1.pdf

ii) KDM ingest is an out of band operation without any direct interaction between the MB and the content author (and the key confirmation messages and infrastructure do not exist).

iii) The KDM already supports RSA OAEP, with the "Additional Input A" from data items in the CipherData element in ST430-1, required by KTS-OAEP-basic (see ST430-1 at section 6.1.2).

Regarding item (iii), the current ST 430-1 KDM specification language at section 6.1.2 does not state that the CipherData field elements "shall" be as shown in the table; it only seems to suggest so. As detailed below under "SP800-56Br1 Section 9.2.1 KTS-OAEP Assumptions" item #3, carriage of the parameters is mandatory to satisfy the requirements of "Additional Input A." Current implementations (both KDM creators and MB designs) that are not so compliant must be changed; implementations that are already so compliant will not need to change.[5]

It is recommended that SHA-256 be considered as the hash function in the Mask Generation Function of RSA-OAEP. However, this will break the current use of the W3C XML Encryption Syntax "EncryptionMethod" which currently mandates the use of SHA-1 per ST430-3 (ETM):[6]

<enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
*<ds:DigestMethodAlgorithm="http://www.w3.org/2000/09/**xmldsig#sha1**"/></enc:EncryptionMethod>*

However, the W3C specification also states:[7]

"EncryptionMethod is an optional element that describes the encryption algorithm applied to the cipher data. If the element is absent, the encryption algorithm must be known by the recipient or the decryption will fail."

Since EncryptionMethod element is an optional field, it should be able to be removed from the KDM without causing problems, because a MB designed to use SHA-256 can be hardcoded to do so.

As of this writing, NIST has been silent as to whether the use of SHA-1 in EncryptionMethod will remain Allowed. SHA-1 is being increasingly revoked for similar applications.[8] DCI's FIPS expert's advice is that SHA-1 should be explicitly disallowed in the Mask Generation Function of SP800-56Br1 for security reasons.[9] In other words, the use of KTS-OAEP-basic with SHA-1 "may" still be an option, however the question of long term FIPS-mode Allowed use is uncertain.

Should SMPTE decide to future-proof the ETM/KDM by adopting SHA-256 and remove the current EncryptionMethod, the absence of this field can identify for the MB that this is a new KDM type. Once such new KDMs are being used, however, the KDM creator must know which hash function the targeted MB will use, and create the KDM accordingly.[10] More on this below.

---

[5] It is believed that all current implementations are compliant for reasons of interoperability.
[6] Readers are reminded that the KDM is an instance of Extra-Theater Message (ETM) (ST430-3).
[7] See EncryptionMethod Element: https://www.w3.org/TR/2002/REC-xmlenc-core-20021210/Overview.html#rsa-oaep-mgf1p
[8] See SP800-131Ar1 Section 9 which disallows SHA-1 for digital signature generation because it does not provide the necessary minimum 112-bits of strength: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf
[9] See Section B.1 of RFC 3447 as referenced in SP800-56Br1 which shows that SHA-1 used in the Mask Generation Function of RSA-OAEP does not provide the minimum 112-bit of strength: "For the signature schemes in this document, a collision attack is easily translated into a signature forgery. Therefore, the value L / 2 should be at least equal to the desired security level in bits of the signature scheme (a security level of B bits means that the best attack has complexity 2^B). The same rule of thumb can be applied to RSAES-OAEP; it is recommended that the bit length of the seed (which is equal to the bit length of the hash output) be twice the desired security level in bits."
[10] Sending the MB both types of KDM works technically, but is believed to be commercially unacceptable.

(Preliminary tests were performed using multiple XML schema validation tools; the removal of the EncryptionMethod element from otherwise valid KDMs did not result in a violation of the current schemas defined by SMPTE ST430-1 and SMPTE ST430-3.[11] Preliminary tests have not yet been performed using the KTS-OAEP-basic encryption/decryption operations with functional KDMs, however there are no known incompatibilities related to the same.)

B) The MB must support a SP800-90Ar1 DRBG[12] for RSA key generation and all other operations that require an Approved random bit generator (such as random number generation for TLS, symmetric key generation, etc.). This requires design changes to the MB and additional test evidence as follows:

1) Implement one of the random bit generators specified in SP800-90Ar1 DRBG.
2) SP800-90Ar1 Section 11.1 specifies minimal documentation requirements for the DRBG.[13]
3) Implement the DRBG health-tests specified in SP800-90Ar1 Section 11.3.[14]
4) Implement the continuous RNG test on the output of the entropy source(s) and the output of the DRBG as per FIPS 140-2.[15]
5) Implement a strong entropy source to seed the DRBG that provides a minimum of 128-bits of unpredictability (the DCI DCSS already requires 128-bits strength).[16]
6) Provide evidence on the entropy source(s) per FIPS 140-2 Implementation Guidance 7.15.[17]
7) Implement RSA key generation as per FIPS 186-4.[18]
8) Perform the following self-tests at MB power-up:
    i. RSA decryption known answer test.[19]
    ii. Critical functions test: confirm the RSA key(s) used for KDM function correctly.[20]
9) Perform RSA pairwise consistency tests for keys generated per FIPS 186-4.[21]
10) Perform algorithm validation testing on the SP800-90Ar1 DRBG[22] and RSA per FIPS 186-4.[23]
(Algorithm validation testing and self-tests on the SP800-56Br1 is further discussed below.)

C) MB vendors must obtain algorithm validation certificates for SP800-56Br1 via an accredited FIPS laboratory. The newest version of the NIST Cryptographic Algorithm Validation Systems (CAVS version 20.1) only supports the RSADP per SP800-56Br1 (RSA Decryption Primitive). This is the raw underlying decryption operation from KTS-OAEP-basic. FIPS laboratories cannot yet test the full KTS-OAEP-basic via the CAVS tool, however, FIPS laboratories can perform a "component validation" and the implementation under test can receive a "CVL" certificate.[24] FIPS Laboratories can augment the basic RSADP testing with design and source code review, and require vendors to provide signed attestations affirming correct implementation.

---

[11] XML schema validation tools used for preliminary tests: schema-check; NotePad++; oXygen; XMLSpy.
[12] See SP800-90Ar1: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
[13] See SP800-90Ar1 Section 11.1: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
[14] See SP800-90Ar1 Section 11.3: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
[15] See FIPS 140-2 DTR AS09.41 and AS09.42: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402DTR.pdf
[16] See DCI DCSS Section 9.7.6: http://www.dcimovies.com/errata/v1_2_with-errata-8-12/DCSS-replacement_Chapter_9--20140904.pdf
[17] See FIPS 140-2 IG 7.15: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf
[18] See FIPS 186-4 Section 5.1 and Appendix B.3.1: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
[19] See FIPS 140-2 DTR AS09.16: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402DTR.pdf
[20] See FIPS 140-2 DTR AS09.27: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402DTR.pdf
The test involves encrypting a plaintext message with the RSA public key. If the output is equal to the input message, the test fails. The encrypted message is then decrypted using the RSA private key and if the output is not equal to the original message, the test fails.
[21] See FIPS 140-2 DTR AS09.31: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402DTR.pdf
[22] See CAVP DRBG: http://csrc.nist.gov/groups/STM/cavp/random-number-generation.html#drbg
[23] See CAVP RSA: http://csrc.nist.gov/groups/STM/cavp/digital-signatures.html
[24] See CAVP CVL: http://csrc.nist.gov/groups/STM/cavp/documents/components/componentnewval.html

D) MB must perform a power-up known answer test on the SP800-56Br1 algorithm implementation each time the MB is powered-up. This requires design changes to the MB and additional test evidence. Refer to FIPS 140-2 Implementation Guidance (IG) D.4.

E) Since committing to FIPS 140-2 certification as an independent security assurance process for digital cinema over a dozen years ago, DCI has witnessed at least three instances of evolving NIST/FIPS requirements for which response options included the use of more than one RSA key pair.[25] In one instance (log signing), the industry was forced to implement a second key pair. DCI's current investigation into SP800-56Br1, again, has suggested that having two or more identity key pairs optionally available in the MB could provide insurance against future changes by NIST that might otherwise be cause for concern for the digital cinema industry.

Therefore, DCI will require that future MB designs include the capability to be securely commanded to self-generate (internally generate) and store at least two RSA key pairs that can be used for KDM targeting purposes. The feature will also permit an existing key pair to be replaced with a freshly generated key pair. DCI believes this capability enables the MB to have improved robustness against future NIST/FIPS changes, should the feature need to be called upon. No specific use cases have been identified by DCI at this time; DCI will publish information and/or errata as required in the future.

Summary:

In order to continue to rely upon FIPS 140-2 validation, DCI compliant MBs must be compliant to SP800-56Br1 as of January 2018. The KTS-OAEP-basic encryption "Additional Input A" requirement is mandatory (and satisfied by making the KDM's "CipherData" table normative). MB design changes and additional test evidence as identified above will be required. To provide additional flexibility to respond to future NIST changes, DCI will require MBs to carry at least two identity RSA key pairs.

DCI suggests that SMPTE consider whether to modify the KDM to eliminate EncryptionMethod now or at a later date, presumably when the use of KTS-OAEP-basic with SHA-1 becomes disallowed.[26]

PART II – Overview and Comments to KTS-OAEP "Assumptions"

For the benefit of MB vendors, the below provides additional implementation information.

SP800-56Br1 Section 9.2.1 KTS-OAEP Assumptions:

SP800-56Br1 lists a number of assumptions for implementing key transport using KTS-OAEP. Copied below, these are considered prerequisite requirements. The *[blue text]* is added to assist in understanding the assumptions:

1. Party V *[KDM receiver]* has been designated as the owner of a key-establishment key pair that was generated as specified in Section 6.3. *[The RSA key pair must be generated by SP800-90Ar1 DRBG and perform RSA key generation as per FIPS 186-4, as discussed in bullet "B" above.]*

Party V has obtained assurance of its possession of the correct value for its private key as specified in

[25] For example, one option identified for addressing the deprecated/disallowed RNG used in MIC processing was to separate FIPS-mode and non-FIPS mode MB operation using KDMs targeted at different identity key pairs. Having the KDM carry the MIC key designed around the issue, and avoided this dual key pair approach.
[26] The SMPTE FIPS Study Group has targeted SP800-56Br1 requirements for review.

Section 6.4.1.5. *[The MB can perform a pairwise consistency test as discussed in bullet "B" above.]*

2. The parties have agreed upon an Approved hash function appropriate for use with the mask-generation function used by RSA-OAEP (see Sections 5.1, and 7.2.2). *[This requires design changes to the MB if SHA-1 is replaced with SHA-256 (which is already supported by the MB).]*

3. Prior to or during the transport process, the sender and receiver have either agreed upon the form and content of the additional input *A* (a byte string to be cryptographically bound to the transported keying material so that the cipher is a cryptographic function of both values), or agreed that *A* will be an empty string (see Section 9.1 above). *[This requirement is met by mandating that the KDM's CipherData field carry the parameters as shown in the table at KDM section 6.1.2.]*

4. If key confirmation is used, the parties have agreed upon an approved MAC algorithm and associated parameters (see Section 5.2). *[Key confirmation is not used in KTS-OAEP-basic.]*

5. When an identifier is used to label either pduring the key-transport process, both parties are aware of the particular identifier employed for that purpose. In particular, the association of the identifier used to label party V *[KDM receiver]* with party V's public key is trusted by party U *[KDM creator]*. When an identifier is used to label party U during the key-transport process, it has been selected/assigned in accordance with the requirements of the protocol relying upon the use of the key-transport scheme. *[The certificate thumbprint of the D-Cinema certificate could be used.]*

6. Party U *[KDM creator]* has obtained assurance of the validity of party V's public key, as specified in Section 6.4.2. *[KDM creators will defer to out of band assurances. Refer to Plausibility tests specified in SP 800-89, Section 5.3.3 and item #7 below for additional information.]*

7. Prior to or during the key-transport process, party U has obtained (or will obtain) assurance that party V is (or was) in possession of the (correct) private key corresponding to the public key-establishment key used during the transaction, as specified in Section 6.4.2. *[KDM creators should obtain assurance that party V (the KDM receiver) has possession of its private key. SP800-56Br1 Section 6.4.2.3.1 states that "The methods used by a third party trusted by the recipient to obtain that assurance are beyond the scope of this Recommendation". As such, defer to the content authors how they obtain such assurance.]*

8. Prior to or during the key-transport process, the keying material to be transported has been/is determined and has a format as specified at the beginning of Section 9. *[This is defined by the KDM specification.*[27]*]*

[27] There is a restriction on size of material encrypted under a single encryption process. Per SP800-56Br1 Section 7.2.2: "RSA-OAEP can process up to nLen – 2HLen – 2 bytes of keying material, where nLen is the length of the recipient's RSA modulus in bytes *(i.e., 256 or 384, in this Recommendation), and HLen is the length (in bytes) of the values output by the underlying hash function."*

- nLen = 256 bytes (*i.e.*, RSA 2048-bits)
- HLen = 32 bytes (*i.e.*, SHA-256 has a 256-bit output)

With RSA 2048 and SHA-256: RSA-OAEP as used in the new KDM can process up to: 256-(2 x 32)-2 = 190 bytes (1520 bits) of keying material – for each encryption process. By encrypting each set of secret materials (*e.g.*, each essence key) separately instead of encrypting them all together, this is not an issue.