Network Device Interpretation # 201815

Fixing AES-CTR Mode Tests

Status: ☒ *Active* ☐ *Inactive*

Date: *23-Oct-2018*

End of proposed Transition Period (to be updated after TR2TD process): *23-Nov-2018*

Type of Change: ☐ Immediate application ☒ Minor change ☐ Major change

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

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

Affected Document(s): *ND SD v2.0e, ND SD v2.1*

Affected Section(s): *FCS_COP.1/DataEncryption*

Superseded Interpretation(s): *None*

Issue:

Issue:

NDcPPv2.0/2.1 adds the option to support AES-CTR for data encryption with corresponding test requirements; however, the specified test requirements do not correspond to AES-CTR or do not add assurance.

1) The tests reference an IV. CTR mode does not utilize an IV.

2) The tests do not specify a mode; however, we assume the AES-CTR Known Answer Tests would use AES-CTR unless specified differently. This should be clarified in the tests.

3) Decryption does not need to be tested, since CTR uses XOR to encrypt and decrypt. The chance of an error in the one of the XOR operations is negligible.

4) For the Multi-Block Message Test, AES-CTR does not specify how the counter is managed, so there cannot be a "known good implementation" without either associating the testing with a protocol or requiring the implementation to output the counter used to encrypt each block.

5) For the Monte-Carlo Test, an AES-CTR_Encrypt operation would require the input of a counter; however the test does not specify inputting a counter to the function.

Proposed Resolution:

To make this consistent with CAVP testing with the fewest edits necessary, it appears that:

1) the reference to IVs should be removed,

2) the test requirements for decrypt operations should be removed,

3) the tests should be updated to specify the use of AES ECB.

Resolution:

The NIT acknowledges the issue described in the 'Issue' section above. Therefore paragraphs 96 – 104 (AES-CTR subsection of section 2.2.4.1) of the SD shall be replaced with the following:

AES-CTR Known Answer Tests

The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Due to the fact that Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter is tested through FCS_SSH*_EXT.1.4. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AES-GCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm):

There are four Known Answer Tests (KATs) described below to test a basic AES encryption operation (AES-ECB mode). For all KATs, the plaintext~~, IV,~~ and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

KAT-1 To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros.

KAT-2 To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value.

KAT-3 To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N].

KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128].

AES-CTR Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key using a known good implementation. The evaluator shall perform this test using each selected keysize.

AES-CTR Monte-Carlo Test

The evaluator shall test the encrypt functionality using 100 plaintext/key pairs. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows:

```
# Input: PT, Key
for i = 1 to 1000:
CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i]
```

The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation. The evaluator shall perform this test using each selected keysize.

Rationale:

The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Since CTR mode only uses the forward cipher function and an XOR operation, references to an IV have been removed and the testing for AES-CTR mode has been restricted to the forward cipher function (i.e. ECB mode encryption). Not all implementations will support 128 and 256 bit keys, so the requirement for testing both keysizes has been replaced with a requirement to test the selected keysizes.

Ensuring that the same counter is never reused with the same key is a critical security consideration for AES-CTR mode; however, the generation and management of the counter is outside the scope of AES-CTR mode. Within the cPP generation and management of the counter is tested through FCS_SSH*_EXT.1.4.

Further Action:

None

Action by Network iTC:

None