Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	Dot16KDF (CMAC) Test Vector	
Date Submitted	**2006-07-07**	
Source(s)	Jicheol Lee Samsung Electronics	Voice: +82-31-279-3605 jicheol.lee@samsung.com
	Avishay Shraga Intel Corporation	voice: +972-3-920-5763 avishay.shraga@intel.com
Re:	IEEE 802.16e-2005	
Abstract	This contribution provides Dot16KDF (CMAC) test-vector	
Purpose	Clarification of the Dot16KDF (CMAC) by providing test vectors for it.	
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.	

Dot16KDF (CMAC) Test Vector

Samsung Electronics
Intel Corporation

Background

The current Dot16KDF described in section 7.5.4.6.1 of IEEE802.16e-2005 is not clear for some interpretation of parameter when it is implemented.

This contribution clarifies the Dot16KDF in case of CMAC mode by providing Test Vectors.

```
line 1: Dot16KDF(key, astring, keylength)
line 2: {
line 3:     result = null;
line 4:     Kin = Truncate (key, 128);
line 5:     for (i = 0;i <= int((keylength-1)/128); i++) {
line 6:         result = result | CMAC(Kin, i | astring | keylength);
line 7:     }
line 8:     return Truncate (result, keylength);
line 9: }
```

So, suggested remedy is to include the test vector of this contribution into the Annex F.

The clarification of these test vectors provides the following interpretations of the Dot16KDF algorithm

1) In the line 6, the size of the variables ‘i’ and ‘keylength’ field when it is used as inputs of CMAC algorithm.

   Our interpretation of the size is 4 octets (32bits) in most-significant-bit first order.

   For example, if astring is “test”, the astring as follows: (it doesn’t include null-termination)

   0x74657374

2) In the line 8, the specification says at the end of 7.5.4.6.1 in page 309 of IEEE802.16e-2005.

   “Truncate(*x, y*) is the rightmost *y* bits of a value *x* only if *y* *x*.”
   The Test Vectors from Example 6 to 10 are fully compliant to this definition of ‘Truncate’ function.

Proposed Text Changes

[Add the following text at line17 in the section 7.5.4.6.1 at page 308 of 802.16e-2005]

When CMAC algorithm is used, the size of the variables 'i' is 4 octets (32bits) in most-significant-bit first order.
'astring' is a character string. For example, if 'astring' is "test", then 'astring' is: 0x74657374 (no null-termination)
The size of 'keylength' field is 4 octets (32bits) in most-significant-bit first order.

[Add the following section at the end of Annex F]
[Editor Note: Please use a fixed-size font when incorporating the following test vectors]

F.3. Test Vectors for Dot16KDF (CMAC mode)

The following test vectors clarify the Dot16KDF with CMAC algorithm.
[Note: The size of variables 'i' and 'keylength' field is 4 octets (32bits) in most-significant-bit first order.]

```
-----------------------------------------------------------
Example  1: Dot16KDF(key,"test",64)

Input
  Input key (128bits): 00010203 04050607 08090a0b 0c0d0e0f
  Astring            : test
  Keylength (in bits): 64

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000000 74657374 00000040
  size of input in bytes: 12

Output
  Result Key(064bits): 5d7dbdcb ff17fa36
-----------------------------------------------------------
Example  2: Dot16KDF(key,"test",128)

Input
  Input key (128bits): 00010203 04050607 08090a0b 0c0d0e0f
  Astring            : test
```

```
  Keylength (in bits): 128

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000000 74657374 00000080
  size of input in bytes: 12

Output
  Result Key(128bits): 79f7ef91 eaeb6ccf 1a9d7ffb e8594881
-------------------------------------------------------
Example  3: Dot16KDF(key,"test",160)

Input
  Input key (128bits): 00010203 04050607 08090a0b 0c0d0e0f
  Astring            : test
  Keylength (in bits): 160

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000000 74657374 000000a0
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000001 74657374 000000a0
  size of input in bytes: 12

Output
  Result Key(160bits): 462fe5a6 62ad3885 5355cf59 a6e18941
                       8320cf9c
-------------------------------------------------------
Example  4: Dot16KDF(key,"test",320)

Input
  Input key (128bits): 00010203 04050607 08090a0b 0c0d0e0f
  Astring            : test
  Keylength (in bits): 320

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
```

```
  00000000 74657374 00000140
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000001 74657374 00000140
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000002 74657374 00000140
  size of input in bytes: 12

Output
  Result Key(320bits): 35858f1c b41d6dca b8ad1532 87ee4a88
                       8882dc4f 84d08923 b7a098ab 47dca41b
                       6aaf8414 524dc319
------------------------------------------------------------
Example  5: Dot16KDF(key,"test",384)

Input
  Input key (128bits): 00010203 04050607 08090a0b 0c0d0e0f
  Astring            : test
  Keylength (in bits): 384

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000000 74657374 00000180
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000001 74657374 00000180
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000002 74657374 00000180
  size of input in bytes: 12
```

```
Output
  Result Key(384bits): 5d098bac bd816395 d5d0d378 07ae0b44
                       973e0c05 4a54ce01 d5bdbf0f 4c9c9dab
                       1585ad02 526ecca4 de91c4ea 3769e5fa
------------------------------------------------------------
Example  6: Dot16KDF(key,"test",64)

Input
  Input key (160bits): 00010203 04050607 08090a0b 0c0d0e0f
                       10111213

  Astring            : test
  Keylength (in bits): 64

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000000 74657374 00000040
  size of input in bytes: 12

Output
  Result Key(064bits): 1665c444 5858d763
------------------------------------------------------------
Example  7: Dot16KDF(key,"test",128)

Input
  Input key (160bits): 00010203 04050607 08090a0b 0c0d0e0f
                       10111213

  Astring            : test
  Keylength (in bits): 128

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength is:
  00000000 74657374 00000080
  size of input in bytes: 12

Output
  Result Key(128bits): 4941fa8c cdc842f9 1fa61288 e820084c
------------------------------------------------------------
Example  8: Dot16KDF(key,"test",160)
```

```
Input
  Input key (160bits): 00010203 04050607 08090a0b 0c0d0e0f
                       10111213

  Astring            : test
  Keylength (in bits): 160

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000000 74657374 000000a0
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000001 74657374 000000a0
  size of input in bytes: 12

Output
  Result Key(160bits): df82c141 88ff0c9d 988e40a5 c1a1cd92
                       a0da080b
-------------------------------------------------------
Example  9: Dot16KDF(key,"test",320)

Input
  Input key (160bits): 00010203 04050607 08090a0b 0c0d0e0f
                       10111213

  Astring            : test
  Keylength (in bits): 320

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000000 74657374 00000140
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000001 74657374 00000140
  size of input in bytes: 12
```

```
  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000002 74657374 00000140
  size of input in bytes: 12

Output
  Result Key(320bits): c330eba9 139eb0a3 a4727b7f b76581ac
                       16f3c110 8b53a459 99cf84ef 959446cc
                       3fcba53f 51cb87cc
-------------------------------------------------------
Example  10: Dot16KDF(key,"test",384)

Input
  Input key (160bits): 00010203 04050607 08090a0b 0c0d0e0f
                       10111213

  Astring            : test
  Keylength (in bits): 384

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000000 74657374 00000180
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000001 74657374 00000180
  size of input in bytes: 12

  Input string of CMAC algorithm
  Concatenation of 'i' || astring || keylength as follows:
  00000002 74657374 00000180
  size of input in bytes: 12

Output
  Result Key(384bits): 41bbbb3d 13be30ee 34f51f12 815caa46
                       71e0cca2 2b25e08b 0a04b92a 51d0c847
                       bd8a9d99 a4e94940 a9bf150e 8f10c2d5
-------------------------------------------------------
```