2 of 2

S

### 9.6.3 Derivation Function Using a Block Cipher Algorithm

Let **CBC_MAC** be the function specified in Section 9.6.4. Let **ECB_Encrypt** be an encryption operation in the ECB mode using the selected block cipher algorithm. Let *outlen* be its output block length, and let *keylen* be the key length.
The following or an equivalent process **shall** be used to derive the requested number of bits.

**Input:**

1. *input_string*: The string to be operated on.
2. *no_of_bits_to_return*: The number of bits to be returned by **Block_Cipher_df**. The maximum length (*max_number_of_bits*) is implementation dependent, but **shall** be  bits.

**Output:**

1. *status*: The status returned from **Block_Cipher_df**. The status will indicate **SUCCESS** or **ERROR**.
2. *requested_bits* : The result of performing the **Block_Cipher_df**.

**Process:**

1. If (*number_of_bits_to_return* > *max_number_of_bits*), then return an **ERROR**.
2. $L$ = **len** (*input_string*)/8. Comment: $L$ is the bit string represention of the integer resulting from **len** (*input_string*)/8.
3. $N$ = *number_of_bits_to_return*/8. Comment : $N$ is the bitsting represention of the integer resulting from *number_of_bits_to_return*/8.

   Comment: Prepend the string length and the requested length of the output to the *input_string*.
3. $S = L \| N \|$ *input_string* $\|$ 0x80.

   Comment : Pad $S$ with zeros, if necessary.
4. While (**len** ($S$) mod *outlen*) $\neq 0$, $S = S \|$ 0x00.

   Comment : Compute the starting value.
5. *temp* = the *Null* string.
6. $i = 0$.
7. $K$ = Leftmost *keylen* bits of 0x010203...1F.
8. While **len** (*temp*) < *keylen* + *outlen*, do

   8.1 $IV = i \| 0^{outlen - \text{len}(i)}$. Comment: The integer represenation of $i$ is padded with zeros to *outlen* bits.

   8.2 *temp* = *temp* $\|$ **CBC-MAC** ($K$, (IV $\|$ $S$)).

8.3 $i = i + 1$.

Comment: Compute the requested number of bits.

9. $K$ = Leftmost *keylen* bits of *temp*.
10. $X$ = Next *outlen* bits of *temp*.
11. *temp* = the *Null* string.
12. While **len** (*temp*) < *number_of_bits_to_return*, do
    12.1 $X$ = **ECB_Encrypt** ($K$, $X$).
    12.2 *temp* = *temp* || $X$.
13. *requested_bits* = Leftmost *number_of_bits_to_return* of *temp*.
14. Return **SUCCESS** and *requested_bits*.

### 9.6.4 CBC-MAC Function

The CBC-MAC function was an Approved method for computing a message authentication code. Let **ECB_Encrypt** be an encryption operation in the ECB mode using the selected block cipher algorithm. Let *outlen* be the length of the output block of the block cipher algorithm to be used.
The following or an equivalent process **shall** be used to derive the requested number of bits.

**Input:**

1. *Key*: The key to be used for the block cipher opeation.
2. *data_to_MAC*: The data to be operated upon.

**Output:**

1. *output_block*: The result to be returned from the CBC-MAC operation.

**Process:**

1. *chaining_value* = $0^{outlen}$.

   Comment: Set the first chaining value to *outlen* zeros.

2. $n$ = **len** (*data_to_MAC*)/*outlen*.
3. Split the *data_to_MAC* into $n$ blocks of *outlen* bits each forming $block_1$ to $block_n$.
4. For $i$ = 1 to $n$ do
   4.1 *input_block* = *chaining_value* $\oplus$ $block_i$ .
   4.2 *chaining_value* = **ECB_Encrypt** (*Key*, *input_block*).
5. *output_block* = chaining_value.
6. Return *output_block*.