Authenticated Encryption with AES-CBC and HMAC-SHA for Information Security - Theory Vs. Reality 0368-4474

Ohad Amon

1 Encryption

The encryption algorithm takes as input five octet strings: an authentication secret key MAC_KEY, an encryption secret key ENC_KEY, a data DATA, additional associated data AAD, and a nonce N. An authenticated ciphertext value is provided as output. The data in the plaintext are encrypted and authenticated, and the associated data are authenticated, but not encrypted.

The keys and nonce are 16 bytes each, and must be generated in a way that is uniformly random or pseudorandom.

In this implementation, AES-CBC refers to the CBC mode of AES-128.

The CBC-HMAC encryption process is as follows:

1. A message authentication tag TAG is generated by applying HMAC (Section 3) to AAD||DATA, the concatenation of the associated data and the data.
2. The concatenation of the data and the authentication tag, DATA||TAG, is then applied a padding according to PAD (Section 4), giving a plaintext P.
3. P is then encrypted using AES-CBC with ENC_KEY and N, giving a ciphertext C.
4. C is returned as a result of the authenticated encryption operation.

In summary:

TAG=HMAC(MAC_KEY, AAD||DATA)

P=PAD(DATA||TAG)

C=AES-CBC(ENC_KEY, N, P)

2 Decryption

The decryption algorithm takes as input five octet strings - MAC_KEY, ENC_KEY, AAD, and N, as defined above, and the ciphertext C. The algorithm has a single output, either a plaintext value DATA or a special symbol FAIL that indicates that the inputs are not authentic. The authenticated decryption process is as follows:

1. The value C is decrypted, using ENC_KEY as the decryption key and N as the nonce, giving the plaintext P.
2. The padding string is *verified* and stripped from the resulting plaintext. The length of the padding is given by the value padding_length, which is stored in the final octet of P.
3. Of the result, the final 16 octets are denoted as TAG, while the rest are denoted as DATA.
4. The authenticity of AAD and DATA are checked by computing HMAC as described in step 2 of Section 1. If the result is equal to TAG, the plaintext value DATA is returned. Else, FAIL is returned.

3 HMAC

HMAC is performed as follows:

1. ipad is defined to be the byte 0x36 repeated 16 times.
2. opad is defined to be the byte 0x5C repeated 16 times.
3. To compute HMAC over the input DATA we perform

   SHA-256(MAC_KEY⊕opad ‖ SHA-256(MAC_KEY⊕ipad ‖ DATA))

   where MAC_KEY is the 16-octet-length MAC key.
4. The result of the above operation is truncated to 16 octets by stripping off the final 16 octets, then returned.

4 Padding

Padding is added to force the length of the plaintext to be an integral multiple of the block length. It is comprised of two fields: padding and padding_length.

- padding - A vector of length padding_length bytes where each byte is filled with the uint8 value padding_length.
- padding_length - A field of size 1 octet, specifying the length of the padding exclusive of the padding_length field itself. It should be such that the total size of the plaintext,

  namely |DATA| + padding_length + 1, is a multiple of 16. Legal values range from 0 to 255, inclusive.

The return value of PAD(DATA) is DATA‖padding‖padding_length.

Example: If the length of DATA is 25 bytes, then in order for |DATA| + padding_length + 1 to be a multiple of 16 padding_length needs to be 6. Therefore the final 8 octets of PAD(DATA) will be xx 06 06 06 06 06 06 06, where xx is the final octet of DATA.

The validity of the padding *must be verified* before it is striped. Padding is only valid, if all the bytes in the padding are equal to the padding length.