Cryptography

Message Authentication Codes

Prof. Dr. Heiko Knospe

TH Köln – University of Applied Sciences

May 25, 2020

Message Authentication Codes

A message authentication code (MAC) is a cryptographic tag which protects the *integrity* and the *origin* of a message. A correct tag shows that the data has not been tampered with by an adversary, and it also protects against accidental errors. MACs are widely used to protect mass data, for example in the network security protocols TLS and IPsec.

Message authentication codes use a *symmetric secret key* for tag generation and verification. This constitutes a major difference to *signatures*, where messages are signed with a private key and verification is performed using a public key.

Definition of a MAC

Definition

A *message authentication code* (MAC) is given by the following spaces and polynomial-time algorithms:

- A message space $\mathcal{M}$,
- A key space $\mathcal{K}$,
- A key generation algorithm $Gen(1^n)$ that takes a security parameter 1^n as input and outputs a key k,
- A tag generation algorithm, which may be randomized. It takes a message m and a key k as input and outputs a tag $\mathrm{MAC}_k(m)$,
- A deterministic verification algorithm that takes a key k, a message m and a tag t and outputs 1 if the tag is valid, or otherwise 0. *Canonical verification* means to re-compute $\mathrm{MAC}_k(m)$ and to compare the result with the given tag.

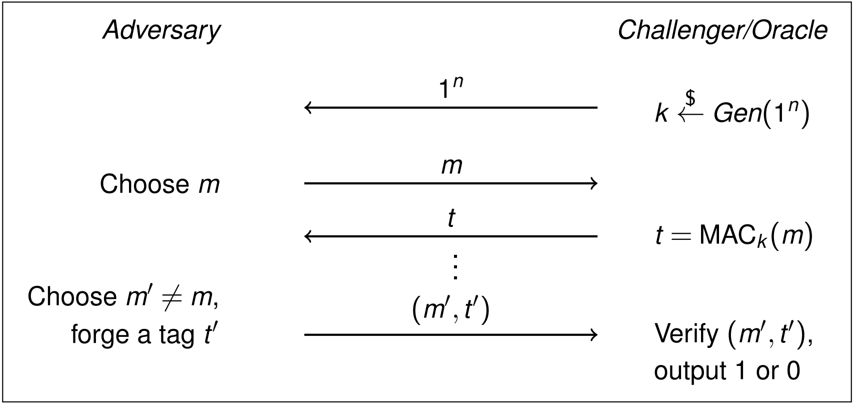
Security of a MAC

A message authentication tag is usually short (similar to a hash value) and does not include the message. Verification therefore requires the message, the tag and the key.

The security of message authentication codes is determined by the difficulty to *forge a valid tag* of a message without knowing the key.

We assume that an adversary can choose messages and obtains a valid MAC (*chosen message attack*). A MAC is *EUF-CMA secure* (existentially unforgeable under an adaptive chosen message attack), if an adversary cannot generate a message, which they did not query previously, and a valid tag in polynomial time.

MAC Forgery Experiment



Adversary | *Challenger/Oracle*

$\xleftarrow{\quad 1^n \quad}$ | $k \overset{\$}{\leftarrow} Gen(1^n)$

Choose m | $\xrightarrow{\quad m \quad}$

$\xleftarrow{\quad t \quad}$ | $t = \mathrm{MAC}_k(m)$

$\vdots$

Choose $m' \neq m$,
forge a tag t' | $\xrightarrow{\quad (m', t') \quad}$ | Verify (m', t'),
output 1 or 0

CBC MAC

We want to construct a MAC for messages of arbitrary length based on a *block cipher*.

A popular construction is the *CBC MAC* (Cipher Block Chaining Message Authentication Code). The message is encrypted in CBC mode (with zero IV) and the *last ciphertext block* is used as authentication tag.

Unfortunately, the basic CBC MAC is not secure for messages of variable length. This problem can be fixed with the CMAC construction (see below).

CBC MAC

Definition

Let $E : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^l$ be a block cipher. Fix a number N of input blocks and set $\mathcal{M} = \{0,1\}^{N \cdot l}$. Let $k \overset{\$}{\leftarrow} \{0,1\}^n$ a key. The *basic CBC MAC* of a message $m = m_1 \| m_2 \| \ldots \| m_N$ of fixed length $N \cdot l$ is defined by encrypting m in CBC mode and outputting the last ciphertext block:

$$\begin{aligned} c_0 &= 0^l \\ c_i &= E_k(m_i \oplus c_{i-1}) \text{ for } i = 1, 2, \ldots, N \\ \mathrm{MAC}_k(m) &= c_N \end{aligned}$$

The CBC MAC is deterministic and verification is canonical.

CMAC

Definition

Let E be a block cipher with 128-bit block length, k a secret key and $m = m_1 \| m_2 \| \ldots \| m_N$ a sequence of 128-bit message blocks, where the last block m_N may be shorter. Then the CMAC of m is defined as the CBC MAC of $m' = m_1 \| m_2 \| \ldots \| m_{N-1} \| m'_N$, where the last block is tweaked in the following way: first, two 128-bit subkeys k_1 and k_2 are derived from k. Then set

$$m'_N = \begin{cases} m_N \oplus k_1 & \text{if } |m_N| = 128 \\ (m_N \| 10\ldots0) \oplus k_2 & \text{if } |m_N| < 128 \end{cases}$$

$$\text{CMAC}_k(m) = \text{CBC-MAC}(m_1 \| m_2 \| \ldots \| m_{N-1} \| m'_N) = c_N$$

Verification of the CMAC is canonical.

Security of CMAC

An adversary cannot produce a valid CMAC without knowing k_1 or k_2, and these subkeys depend on the secret MAC key k. Furthermore, k_1 or k_2 cannot be recovered from valid tags, since m'_N is protected by encryption.

Theorem

If E is a pseudorandom permutation or function, then the CMAC construction defines an EUF-CMA secure MAC for messages of variable length.

Hash-based MAC

Another widely used MAC construction is based on *hash functions*.

However, the obvious *prefix* construction $H_k(m) = H(k, m)$ (with $k = IV$) or $H_k(m) = H(k \| m)$ (for an unkeyed hash function with fixed IV) is *insecure* for messages of variable length if H is a Merkle-Damgård hash function (*length extension attack*). Note that the SHA-3 family is not vulnerable to this attack.

The *Hash-based Message Authentication Code* (HMAC) uses *two nested hashing* operations; this protects against length extension attacks. HMAC is widely used in practice, not only as message authentication code, but also as a pseudorandom function (*prf*) and as a building block in key derivation functions.

HMAC

Definition

Let H be a Merkle-Damgård hash function. Let b be the input block length in bytes of the underlying compression function, e.g., $b = 64$. The message space is $\mathcal{M} = \{0,1\}^*$ and HMAC keys $k \overset{\$}{\leftarrow} \{0,1\}^n$ are chosen uniformly at random. Define `ipad` and `opad` strings by repeating the bytes `36` and `5C`, respectively, b times. The key k is padded by zeros such that the byte length of $\overline{k} = (k \,\|\, 0 \ldots 0)$ is b. Then the *HMAC* message authentication tag of a message m is defined as

$$\mathrm{HMAC}(k,m) = H(\overline{k} \oplus \mathtt{opad} \,\|\, H(\overline{k} \oplus \mathtt{ipad} \,\|\, m)\ ).$$

The verification of an HMAC tag is canonical.

HMAC Security

The following Theorem reduces the security of HMAC to the pseudorandomness of the compression function f and its dual function $\overline{f}$.

Theorem

Let $f : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^n$ be a compression function and let $\overline{f} : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^n$ be the dual function with the same values as f, but keyed via the second component. Let H be the Merkle-Damgård hash function associated with f.

If f is a prf and $\overline{f}$ is a prf under restricted related-key attacks, then HMAC is a pseudorandom function and an EUF-CMA secure MAC for messages of arbitrary length.

Unforgeable Encryption

In practice, encryption and message authentication are often combined in order to provide *confidentiality* and *integrity*. *Tampering with encrypted messages* should be impossible and the scheme should be secure under a *chosen ciphertext attack*.

Definition

An encryption scheme is called *unforgeable* if an adversary, who has access to an encryption oracle, cannot produce a new and valid ciphertext in polynomial time.

An encryption scheme is called an *authenticated encryption scheme* if it is CCA2-secure and unforgeable.

Authenticated Encryption

An obvious construction is to combine an encryption scheme and a message authentication code.

Theorem

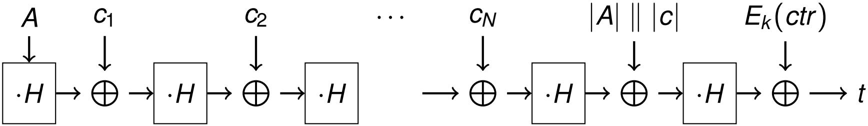
Consider an encryption scheme and a message authentication code. Suppose the encryption scheme is CPA-secure and the message authentication code is strongly secure, for example a secure MAC with canonical verification. Then the encrypt-then-authenticate construction defines an authenticated encryption scheme.

Note: the *encrypt-then-authenticate* construction *first* encrypts the plaintext and *then* computes the MAC of the ciphertext.

GCM Mode

The *Galois Counter Mode* (GCM) extends the CTR mode and provides encryption and message authentication in one pass for ciphers with a block length of 128 bits. GCM can also authenticate *additional data A*.

First, the ciphertext $c = IV \| c_1 \| \ldots \| c_N$ is computed in CTR mode, where $IV \overset{\$}{\leftarrow} \{0,1\}^{96}$ and $ctr = IV \| 0^{31} \| 1$ is the initial counter value. The GCM tag t is derived from the ciphertext c. Encryption and message authentication use the same key k.



$H = E_k(0^{128})$ is called *hash key* and multiplication $(\cdot H)$ is defined over the field $GF(2^{128}) = GF(2)[x]/(x^{128} + x^7 + x^2 + x + 1)$.

Authenticated Encryption using GCM

Let m be a plaintext and A additional data. First, the CTR mode ciphertext c is computed from m using an initial counter of the form $ctr = IV \| 0^{31} \| 1$. Then the GCM tag t is computed from c and A as explained above.

The authenticated ciphertext in GCM mode is given by

$$(c, t, A).$$

For decryption, the tag is re-computed using c and A. If t is valid, then the plaintext m is computed by decrypting c in counter mode. Otherwise, an error symbol $\perp$ is output.