Network Security and Cryptography
Symmetric-key cryptography

Lecture 9: Message authentication codes (MACs)

Mark Ryan

Message authentication codes

A hash function can be used to guarantee the *integrity* of messages (e.g., the integrity of downloaded software).

However, a hash function alone is insufficient to guarantee the *authenticity* of messages (i.e., the fact that a message came from a particular source). If you merely use a hash function, the attacker can modify message and recompute hash.

To guarantee authenticity, we include a secret key inside the message being hashed. Then we know only the authentic party that holds the key is capable of computing the hash. Including a secret key in the hash is called producing a “message authentication code”. Cryptographers have studied the best way of doing that, as we see in this lecture.

Message authentication codes

A MAC is a function which takes a key k and a message m, and produces a short piece of data (called a "tag") which authenticates m using k. Sometimes, a MAC is called a keyed hash function.

Assumption: Alice and Bob share key k

Alice sends to Bob: m, $\mathrm{MAC}_k(m)$.

When Bob receives this message, say m, x, he computes $\mathrm{MAC}_k(m)$ and then checks if $x = \mathrm{MAC}_k(m)$.

How to define a MAC function from a hash function?

How to define MAC from a hash function?

- $\mathrm{MAC}_k(m)$ could be defined as $h(k||m)$. However, if h is vulnerable to a "length extension attack", then so is this MAC. Given m and $h(k||m)$, one can construct m' and $h(k||m')$ (for example, let m' be $m||padding||length(m)||m''$).

  Thus, if Alice sent the message m with $\mathrm{MAC}_k(m)$ using this definition, the attacker could modify the message to m' with $\mathrm{MAC}_k(m')$.

- The constructions $\mathrm{MAC}_k(m) = h(m||k)$ and $\mathrm{MAC}_k(m) = h(k||m||k)$ have also been found to have weaknesses.

HMAC

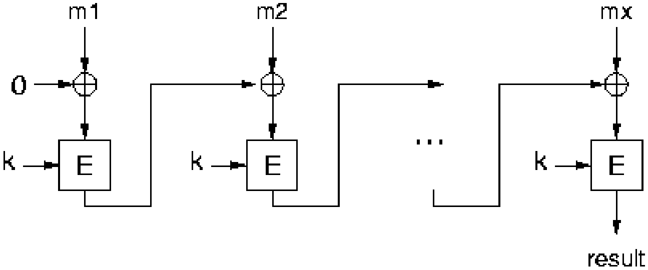
$\mathrm{HMAC}_k(m)$ defined as:

$$\mathrm{HMAC}_k(m) = h\Bigg((k \oplus opad) \| h\Big((k \oplus ipad) \| m\Big)\Bigg),$$

Here, the key k is padded with zeros to the blocksize of the hash function, and *ipad* and *opad* are constants of that blocksize. The values of ipad and opad are not critical to the security of the algorithm, but were defined in such a way to have a large Hamming distance from each other and so the inner and outer keys will have fewer bits in common.
This definition can be shown to have some good security properties: if you can break HMAC, then you can break the underlying hash function.

CBC-MAC

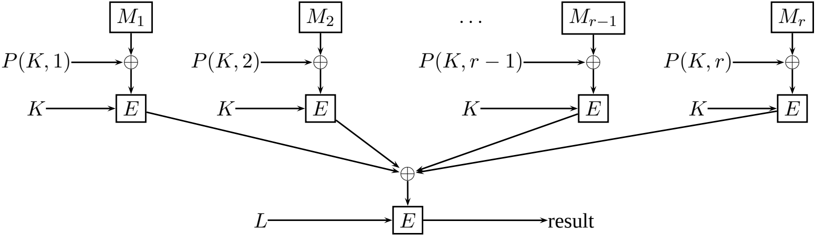
CBC-MAC uses CBC mode of operation for block cipher



Source: Wikipedia

PMAC

Hash functions, HMAC and CBC-MAC are not parallelisable
PMAC addresses this issue
Have two keys K and L
Have function $P(K, i) = K * x^i$ in $\mathbb{F}_{2^n}$



Security of MAC

Let m be a message. Then $\mathrm{MAC}_k(n)$ is sometimes called the *tag* for m.

A MAC function is *secure* if an attacker (not having the key) cannot produce a valid (message, tag)-pair which s/he hasn't seen before.

This is called *secure against existential forgery,*

Definition

The MAC-game between challenger and attacker is defined as follows:

- The attacker does some computations and may in the process supply messages $m_1, \ldots, m_n$ to the challenger
- The challenger returns $t_1, \ldots, t_n$ to the attacker , which are the result of creating the MAC for the messages $m_1, \ldots, m_n$.
- The attacker does some more computations and then supplies to the challenger a pair (m, t), which is not equal to any of the pairs $(m_1, t_1), \ldots, (m_n, t_n)$.
- The challenger outputs 1 if t is obtained by creating the MAC for m, otherwise he returns 0.

The attacker wins the MAC-game if the challenger outputs 1.

Definition

We call a MAC *secure* if no attacker can win the MAC-game with non-negligible probability.

Here, as before, the probability is a function of the key length.

Example

CBC-MAC is not secure (unless you add restrictions).

Suppose the attacker possesses (m, t) and (m', t'). Then he can forge a third pair, (m'', t''):

We assume that m' is more than one block long; say $m' = m'_1||m'_2||\ldots||m'_p$.
Set $m'' = m||(m'_1 \oplus t)||m'_2||\ldots||m'_p$, and $t'' = t'$.
Check that (m'', t'') is a valid message-tag pair.

CBC-MAC result

Theorem

Assume CBC-MAC is used only on messages of a fixed length. If the block cipher used is a secure block cipher, then CBC-MAC is a secure MAC.

Another way to achieve this is to prepend the length in the message.

HMAC and PMAC results

Theorem

If the hash function used is secure, then HMAC is a secure MAC.

Theorem

If the block cipher used is secure, then PMAC is a secure MAC.