UConn, School of Computing
Fall 2024
CSE 3400/CSE 5850: Introduction to Computer and Network Security / Introduction to Cybersecurity

Practice Problems - MACs

Instructor: Prof. Ghada Almashaqbeh

The following are two MAC examples to help you better understand proving MAC security (or insecurity). A scheme works in two phases:

- Alice generates message m and passed it to the MAC to generate tag t, such that $t = \text{MAC}_k(m)$
  - Alice sends *(m, t)* to Bob
- Bob runs the verify function (call it Vrfy) to determine if the tag is authentic for the message Bob was given:

$$\text{Vrfy}(m, t) = \begin{cases} 1 & \text{if } t = \text{MAC}_k(m) \\ 0 & \text{otherwise} \end{cases}$$

It is important to remember that **MAC schemes do not guarantee confidentiality**. Rather, secure MACs guarantee **authenticity**, which is to say MAC schemes guarantee that the message was not tampered with or altered from the time it was sent to the time it was received. So, MAC security is determined not by an attacker's distinguishing probability, but by its ability to forge a MAC for a new message of his choice. Security of MAC schemes means that an attacker can only forge tags with negligible probability.

For the following constructions state whether they are secure MAC schemes:

1. Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a secure PRF. Let m $= m_1, ..., m_l$, where $m_i \in \{0,1\}^n$ for $i = 1, \ldots, l$. Suppose we have the following MAC scheme:

$$\text{MAC}_k(m) = F_k(m_1 \parallel 1) \oplus F_k(m_2 \parallel 2) \oplus ... \oplus F_k(m_l \parallel l)$$

   Solution:
   This scheme applies F_k to every block of the message and XORs all of them together. So, how can the attacker abuse this forge new tags? The answer lies in the properties of PRFs and the XOR operation. PRFs are deterministic and the XOR operation is commutative. The attacker can utilize this fact to obtain a forgery using three queries to the MAC oracle.

   Concretely, the attacker asks the MAC oracle for a tag over $m = m_1$ (one block message), gets tag $t = F_k(m_1 \parallel 1)$ back. Then he asks the MAC oracle for a tag over message $m' = m_1 \parallel m'_2$ and gets tag $t' = F_k(m_1 \parallel 1) \oplus F_k(m'_2 \parallel 2)$ back. then he asks for tag over message $m'' = m''_1$ and gets tag $t'' = F_k(m''_1 \parallel 1)$ back.

Then the attacker casts the following forgery: $m''' = m_1'' \parallel m_2'$, which is a different message from all the queried messages. And the tag over this new message can be computed as: $t''' = t \oplus t' \oplus t'' = F_k(m_1'' \parallel 1) \oplus F_k(m_2' \parallel 2)$ which is a a correct tag over m'''.

The success probability in making a correct forgery is 1, and the advantage of the attacker is $1 - \frac{1}{2^n}$ which is non-negligible. So, this MAC scheme is not secure.

2. Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a secure PRF, and suppose we have a MAC scheme with the following construction (where $k = k_1 \parallel k_2$):

$$\text{MAC}_k(m) = F_{k_2}(F_{k_1}(m))$$

Solution:
This is a secure MAC due to the use of a secure PRF and that it is deterministic. The inner PRF invocation will return a pseudorandom string, and so the outer PRF call will return a pseudorandom string. Thus, for a new message, the attacker will succeed in producing a correct forged tag with probability $\frac{1}{2^n}$. Thus, the attacker's advantage in the MAC security game will be 0, which is negligible.

3. We know that CBC-MAC is secure for fixed length messages. Consider this modified version: we output all intermediate ciphertexts as part of the tag, rather than just the last one. So, for a message $m = m_1 \| \cdots \| m_\ell$, we have CBC-MAC'$(m) = tag = c_1 \| \cdots \| c_\ell$ (recall that in CBC-MAC the IV $= 0^n$, so no need to send it as part of the tag; it is known to everyone).

Is this a secure MAC? Justify your answer.

Solution:
This is not a secure MAC. Consider the following attack against the above CBC-MAC' for $\ell = 2$:
Set $m = 0^n \parallel 0^n$ and obtain the tag from the oracle as $t = t_1 \parallel t_2 = E_k(0^n) \parallel E_k(E_k(0^n))$.

Output the forgery as $m' = t_1 \parallel t_2$ with the tag $t' = t_2 \parallel t_1$ (i.e., flipping the order of the tag obtained from the oracle for the previous query).

To see how this is a valid forgery:
$CBC - MAC'(m') = CBC - MAC'(t_1 \parallel t_2) = E_k(E_k(0^n)) \parallel \Big(E_k\big(E_k(E_k(0^n)) \oplus E_k(E_k(0^n))\big)\Big)$
$= E_k(E_k(0^n)) \parallel E_k(0^n)$
$= t_2 \parallel t_1 = t'$

Thus, the attacker's success probability in making a forgery is 1, and its advantage is $1 - \frac{1}{2^{2n}}$, which is non-negligible.