Alin Tomescu, CSE408
Tuesday, March 1st, Lecture #9

Message authentication codes (MACs)

The birthday problem

Application: You are given a bunch of people in a room, and you want to pick a number of them such that the probability of selecting two people with the same birthday is at least 50%.

Theorem:

Fix a positive integer N, and say q elements $y_1, y_2 \dots y_q$ are chose uniformly and independently at random from a set of size N. Then the probability that there exist distinct i, j with $y_i = y_j$ is at most $\frac{q^2}{2N}$, that is: $Coll(q, N) \leq \frac{q^2}{2N}$

Proof:

$$\Pr[Coll_{i,j}] = \frac{1}{N}$$

$$\Pr[Coll(q,N)] = \Pr\left[\bigvee_{i \neq j} Coll_{i,j}\right] \leq \sum_{i \neq j} \Pr[Coll_{i,j}] = \binom{q}{2}\frac{1}{N} \leq \frac{q^2}{2N}$$

Message authentication codes (MACs)

MACs are used to verify that messages have not been modified by adversaries.

MAC structure

A MAC is a tuple $(Gen, Mac, Vrfy)$, where:

- Gen – generate a key of length n
- $Mac_k(m)$ – generates a tag (or a MAC) on the message m using the key k generated by Gen as input
- $Vrfy_k(m, t)$ – used to verify the integrity of the message m that was tagged with the tag t and key k

```
t' = Mac_k(m)
if t' = t then output 1 else output ⊥
```

  - returns 1 when the message is authentic
  - returns $\perp$ when the message has been forged

Security goal

No poly-time adversaries should be able to generate a correct (m, t)-pair such that $Vrfy_k(m, t) = 1$

Adversary model

Adversary can query $Mac_k(m), \forall m \in M_q$ and get a list of tags. The goal is for the adversary not to be able to build a message tag pair (m, t) with $m \notin M_q$ that will be correctly validated by $Vrfy$.

Examples of MAC

PRFs like AES

Alin Tomescu, CSE408
Tuesday, March 1st, Lecture #9

Replay attack against MACs

Eve will see (m, t) pairs between Alice and Bob and she can later send the same pair if she wants too. You can use sequence numbers or timestamps to protect against such attacks.

Building secure MACs using PRFs

Construction:

- $Mac_k(m) = F_k(m)$
- $Vrfy_k(m, t)$ is defined as follows:

```
if F_k(m) = t then output 1 else output ⊥
```

Theorem: If F_k is a PRF then this construction is an EU-MAC.

Proof: We will show that if you can break the MAC construction then you can also break the PRF.

If ∃ a PPT adversary that can forge a tag t for some message m the we can build a PPT distinguisher D to guess correctly with a high probability if a function is the PRF or truly random.

Our distinguisher would forge $t = Mac_k(x)$ and check if $blackbox(x) = t$. If it is, then the blackbox is a PRF, otherwise it's an RF.

MACs for variable-length messages

$$m = m_1 \parallel m_2 \parallel \cdots \parallel m_x$$

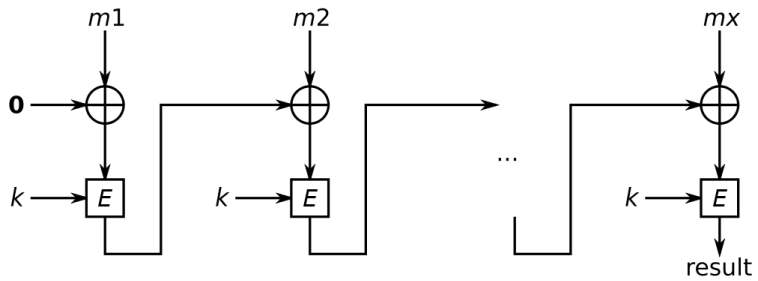
$$|m_1| = |m_2| = |m_i| = n$$

Example of constructions:

1. $t = Mac_k(m_1 \otimes m_2 \otimes \ldots \otimes m_x)$, not secure because you could have a message pair $m_i = m_j = 0$
2. Authenticate each block separately: $t_i = Mac_k(m_i)$, $t = \langle t_1, t_2, \ldots, t_x \rangle$, bad because you could swap m_i, m_j and t_i, t_j creating a new valid message-tag-pair
3. Use sequence # along with the messages: $t_i = Mac_k(seq + m_i)$, bad because you could drop the last blocks and create a new valid message-tag-pair

If you have sequence numbers, and the size of the message (maybe as the first tag t_1), you should be able to build a secure MAC. (Hint: plan to use a CBC MAC for the HW)

CBC-MAC for fixed length messages



Attack against CBC-MAC for variable length messages

Given (m, t) and (m', t') we can create m'' with tag t'

$$m = 00, m' = 11$$

$$t = Mac_k(m) = F_k(F_k(0^{IV} \otimes 0) \otimes 0) = F_k\big(F_k(0)\big) \stackrel{\text{def}}{=} F_k^2(0)$$

$$t' = Mac_k(m') = F_k(F_k(0^{IV} \otimes 1) \otimes 1) = F_k(F_k(1) \otimes 1)$$

$$m'' = 00\left(1 \otimes F_k^2(0)\right)1$$

If you compute t'' you will get t'.

$$t'' = Mac_k(m'') = Mac_k\left(00\left(1 \otimes F_k^2(0)\right)1\right) = F_k\left(F_k\left(F_k(F_k(0^{IV} \otimes 0) \otimes 0) \otimes \left(1 \otimes F_k^2(0)\right)\right) \otimes 1\right)$$
$$= F_k\left(F_k\left(F_k^2(0) \otimes \left(1 \otimes F_k^2(0)\right)\right) \otimes 1\right) = F_k(F_k(1) \otimes 1) = t'$$

This attack can actually be generalized.

CBC-MAC for variable length messages

You want to feed the length of the message through F_k and then XOR the result with m_1 and then just do the same thing as in the fixed length one.