Last class:

Message Authentication Codes

Security against Chosen Message Attacks

Construction: Using PRF F (or AES): MAC(k,m)=F(k,m)

For messages of specific length (128 bits for AES)

Today:

MAC for arbitrarily long messages

Authenticated encryption and CCA secure encryption

In practice: AES-GCM

Going from small MAC to big MAC:

One approach: Use a collision resistant hash function (CRHF)

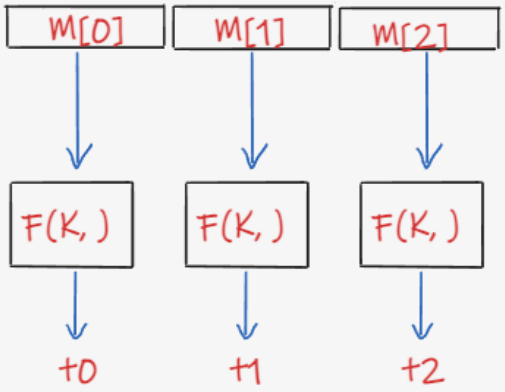
We will talk about this next week, when we talk about hash functions

Two MAC constructions standardized by NIST:

One based on AES (CMAC) and one based on SHA2 (HMAC).

AES-based MAC

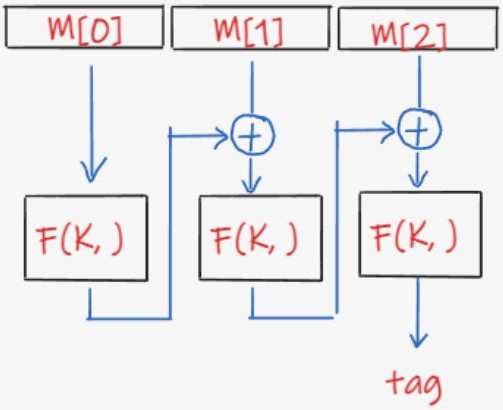
Try 1:



Insecure! Mix and match attack

Adversary can use the tag for message (M[0],M[1]) to tag the message (M[1],M[0]).

Try 2:



Insecure! Extension attack

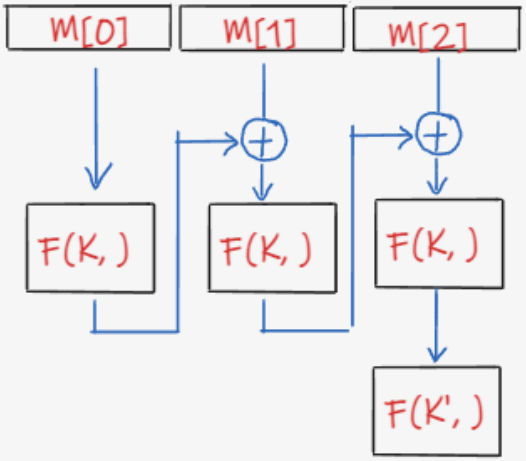
Adversary can use the tag for message (M[0],M[1],m[2]) and tag' for message m'[0],

to tag the message (M[0].M[1],M[2],tag xor M'[0]).

Final try:

CBC MAC:

Cipher Block Chaning



The secret key is (K,K')

This additional secret key prohibits these "extension attacks"

Note: Need to pad the message so that its length will be a multiple of 128.

The standardized version of CBC MAC is called CMAC

So far we talked about encryption and authentication.

What we really want is an authenticated encryption!

Authentication is also important for confidentiality: If an adv can change Enc(k,m) to Enc(k,m') this can help him find m!

For example, m' can append to m, the ending that if the answer is yes then send a long response. Recall that the length of the msg is not hidden by the encryption!

Goal: Construct an authenticated CPA encryption scheme

CPA encryption + MAC

$Enc'((k,k'),m) = (ct, MAC(k', ct))$ where $ct=Enc(k,m)$

Note:
1. We are using different keys for encryption and authentication
2. We are fist encrypting and then authenticating

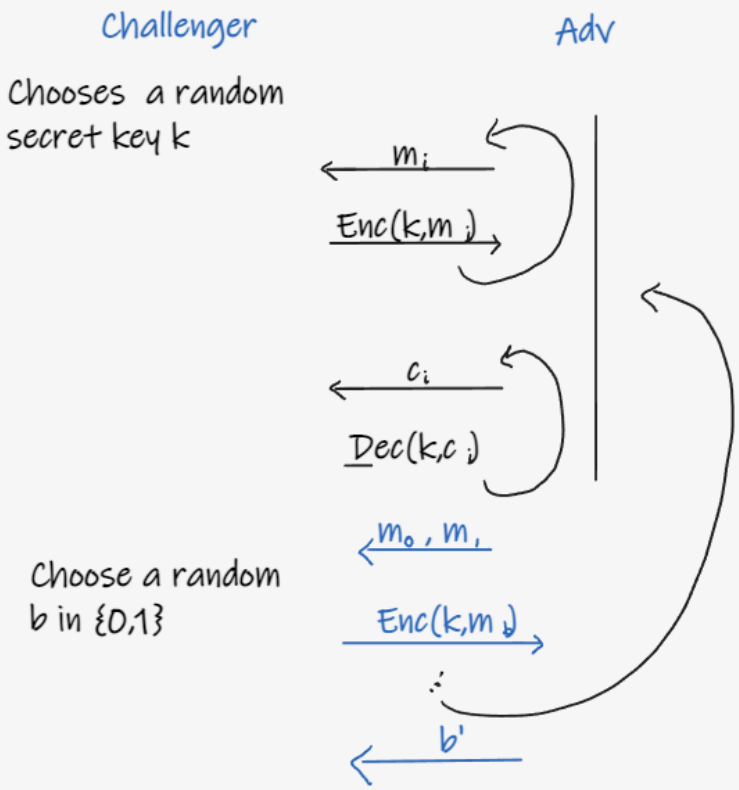
   (and not the other way around)

Both these choices are important for security!

Authenticated CPA encryption guarantees the following desirable security guarantee:

Security against chosen ciphertext attacks (CCA security)

Def: An encryption scheme (Enc,Dec) is CCA secure if any efficient adv wins in the following game with probability at most 1/2 + negl.



Adv wins iff b'=b.

Thm: Let (Enc,Dec) be CPA secure encryption, and let MAC be a secure MAC,

then (Enc',Dec') is CCA secure encryption, where

Enc'((k,k'), m) = (ct, MAC(k', ct)), where ct = Enc(k,m)

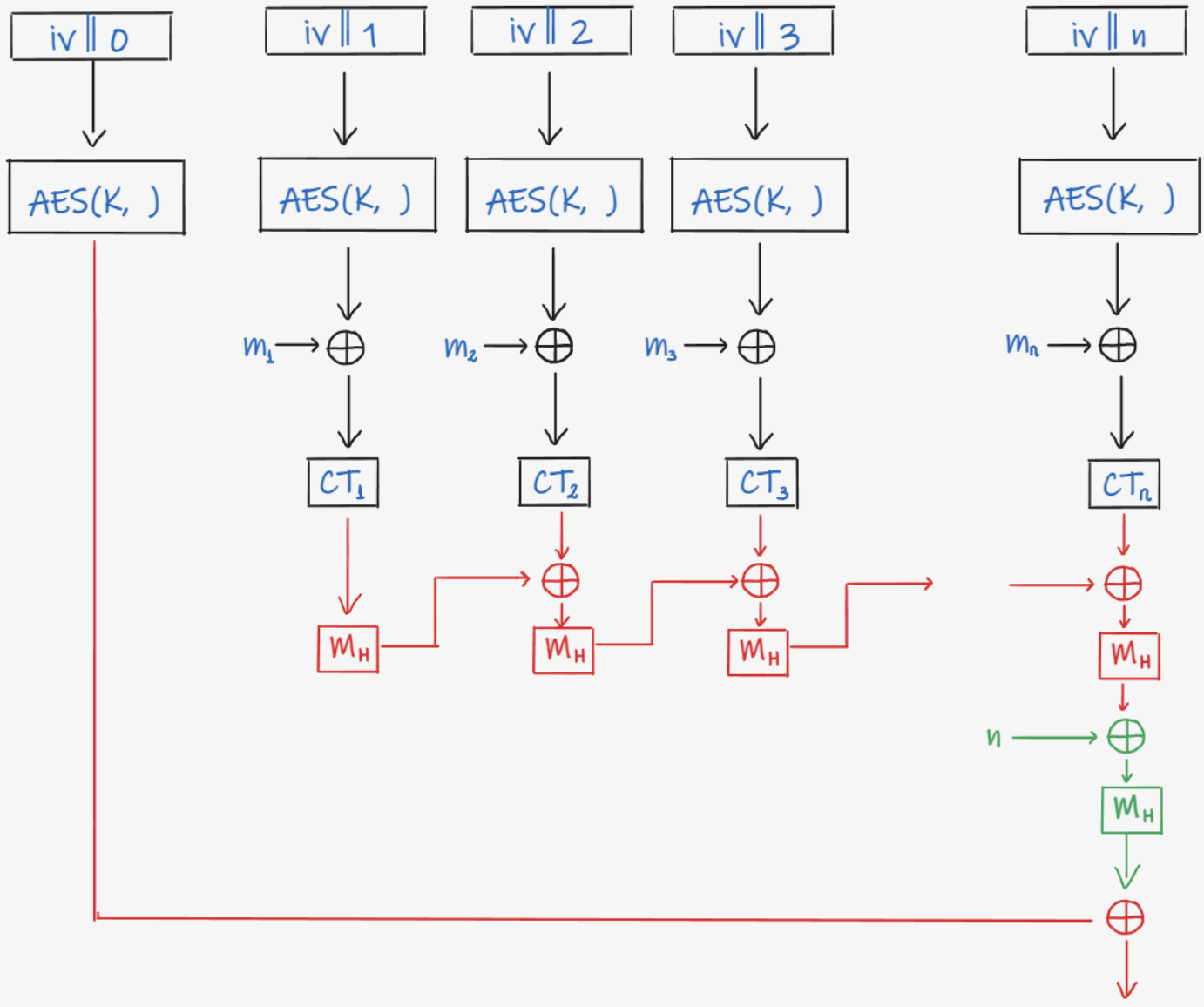
Dec'((k,k'), (ct,σ)): If MAC(k',σ)=0 then output fail. Otherwise, output Dec(K,ct)

HW: Show that if we first MAC then Enc, the resulting scheme may not be CCA secure!

AES GCM: authenticated encryption scheme used in practice.

CPA secure encryption, using AES in counter mode (as eplained above),

together with GMAC (Galois MAC).



M_H is not a secure MAC!

$$M_H(x) = H\,x$$

where $H = AES(k,0)$ is a string of length 128,

$M_H : \{0,1\}^{128} \longrightarrow \{0,1\}^{128}$ is multiplication by H,

where multiplication is in a finite field of size 2^{128}

known as Galois Field ($GF[2^{128}]$)

M_H is a one-time secure MAC!

where H is the secret key.

We cannot reveal even a single tag!

Also provides an effcient way to authenticate auxiliary unencrypted data, such as IP addresses...

(essentially by adding the data to the GMAC, need to authenticate the length of the auxiliary data).