POIS ASSIGNMENT 1

TASK 4

USE THE PRF TO BUILD A SECURE MAC

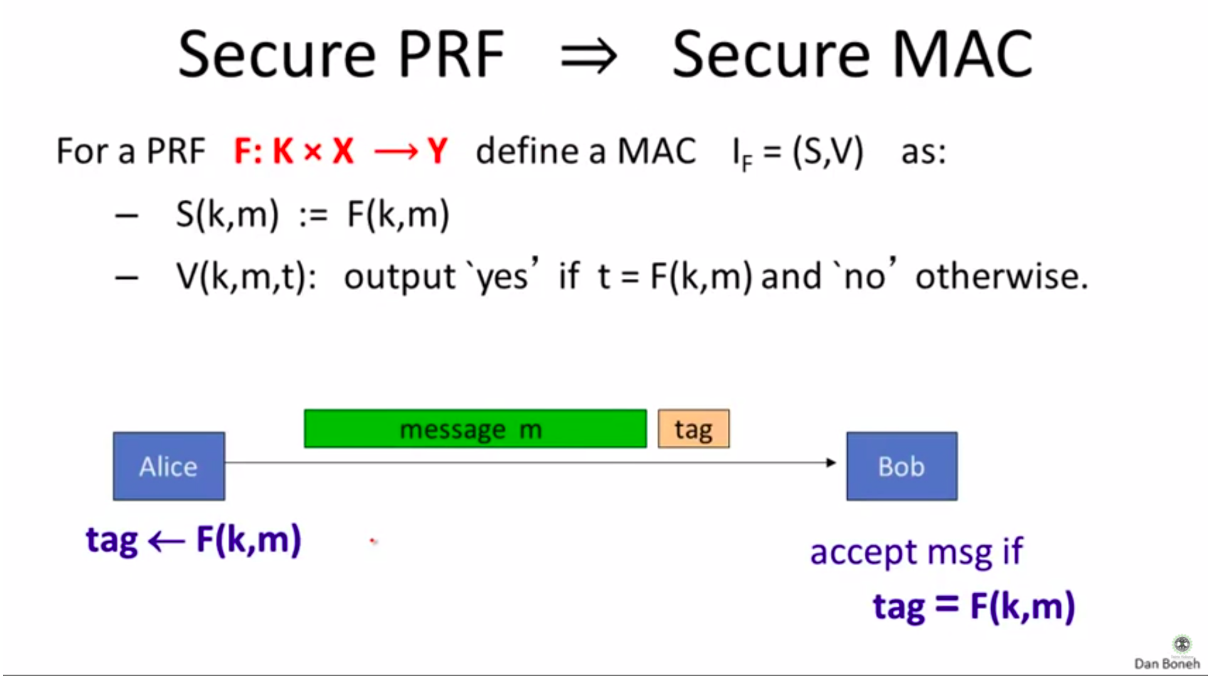
THEORY

Encryption without integrity is useless! Encryption provides security against eavesdropping, but does not protect from the attacker modifying or corrupting the ciphertext. Examples where integrity is important include the delivery of ads or software patch.

In particular, for a given message m, the sender will include a tag S(k, m) computed as a function of the message m and some secret key k. The verifier is then a function V (k, m, t) that outputs “yes” or “no” to indicate whether the message is properly signed or not. To achieve integrity, we always need a secret key! An example of a keyless integrity check is CRC.
The problem is that the attacker can compute the tag for a given message and thus, alter the message and compute a new tag for the message. Such signatures are suitable for checking for random errors, not malicious errors

A message authentication code (MAC) consists of two algorithms (S, V) defined over (K,M, T) where K is the key space, M is the message space, and T is the tag space.
Then, S(k, m) = t ∈ T and V (k, m, t) outputs “yes” or “no.”

A MAC must satisfy the consistency requirement:
∀k, m : V (k, m, S(k, m)) = ”yes”.

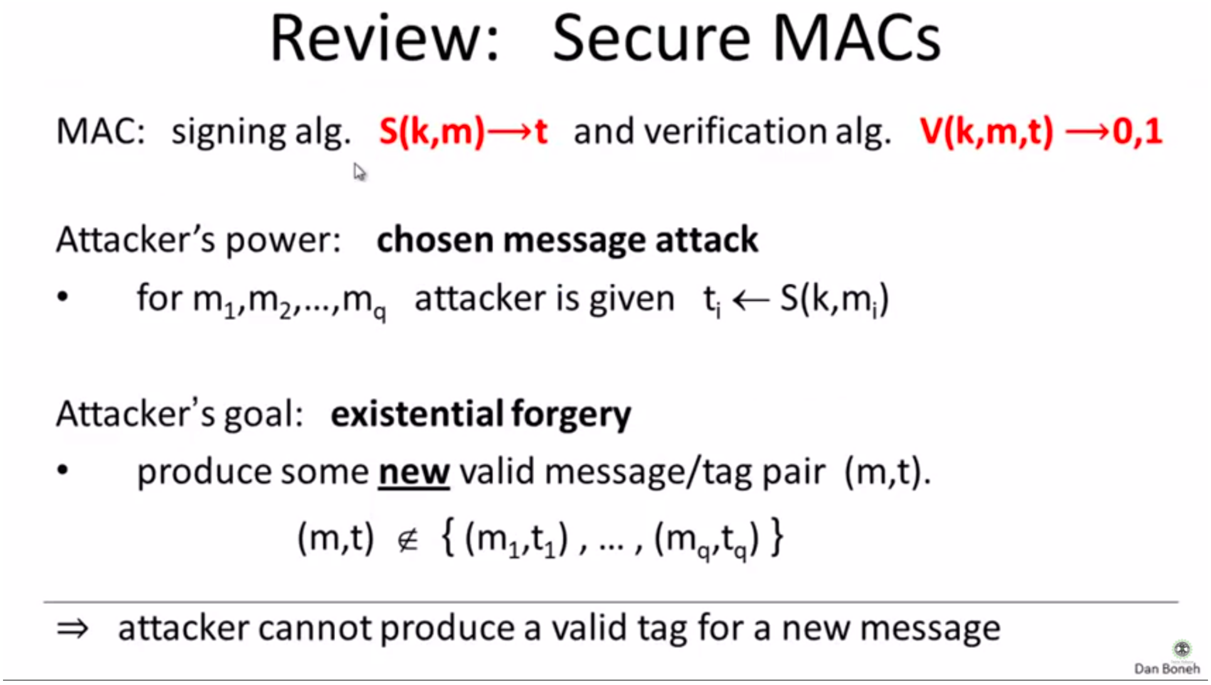


Secure PRF ⇒ Secure MAC

For a PRF **F: K × X ⟶ Y** define a MAC $I_F = (S,V)$ as:

- S(k,m) := F(k,m)
- V(k,m,t): output `yes’ if t = F(k,m) and `no’ otherwise.

Now, we define the notion of a secure MAC. We consider a chosen message attack in which the attacker can submit messages m1, . . . , mq and will receive ti ← S(k, mi).
The goal of the attacker is an existential forgery where he produces some valid (m, t) where (m, t) ∈ { / (m1, t1), . . . ,(mq, tq)}. In other words, the attacker should not be able to produce a valid tag for any message.



Review: Secure MACs

MAC: signing alg. **S(k,m)⟶t** and verification alg. **V(k,m,t) ⟶0,1**

Attacker's power: **chosen message attack**

- for $m_1,m_2,\ldots,m_q$ attacker is given $t_i \leftarrow S(k,m_i)$

Attacker's goal: **existential forgery**

- produce some **new** valid message/tag pair (m,t).

$$(m,t) \notin \{ (m_1,t_1), \ldots, (m_q,t_q) \}$$

⇒ attacker cannot produce a valid tag for a new message

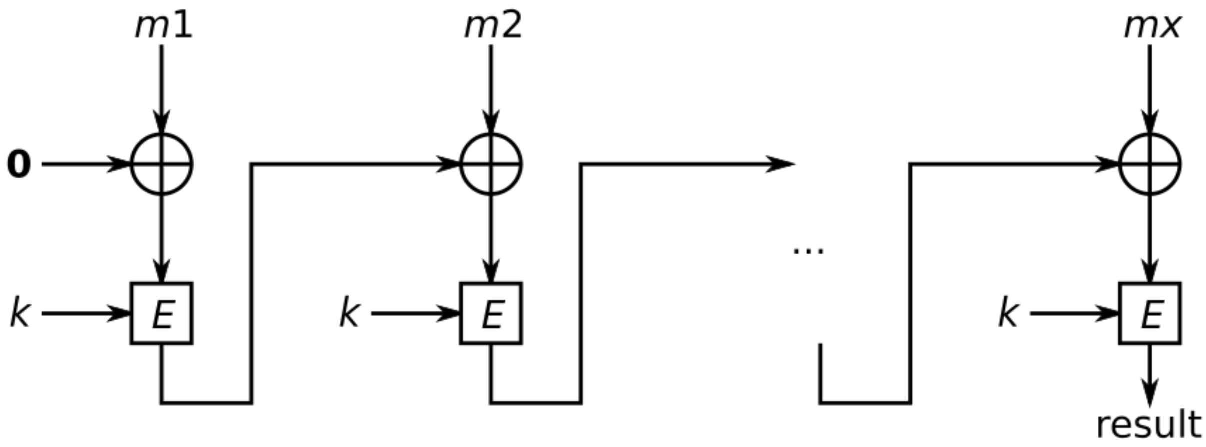
Dan Boneh

Consider a MAC I = (S, V) and an adversary A. Now, the challenger begins by generating a random key k. The adversary then sends messages m1, . . . , mq and receives t1 ← S(k, m1), . . . , tq ← S(k, mq). The adversary then submits a message (m, t). The challenger then outputs 1 if V (k, m, t) = 1 and (m, t) ∈ { / (mi , ti)} and 0 otherwise

What I have done is using the CBC MAC:

In cryptography, a **cipher block chaining message authentication code** (**CBC-MAC**) is a technique for constructing a message authentication code from a block cipher. The message is encrypted with some block cipher algorithm in CBC mode to create a chain of blocks such that each block depends on the proper encryption of the previous block. This interdependence ensures that a change to any of the plaintext bits will cause the final encrypted block to change in a way that cannot be predicted or counteracted without knowing the key to the block cipher.

To calculate the CBC-MAC of message m, one encrypts m in CBC mode with zero initialization vector and keeps the last block. The following figure sketches the computation of the CBC-MAC of a message comprising blocks using a secret key k and a block cipher E


m1
m2
mx
0
k
E
k
E
...
k
E
result

: