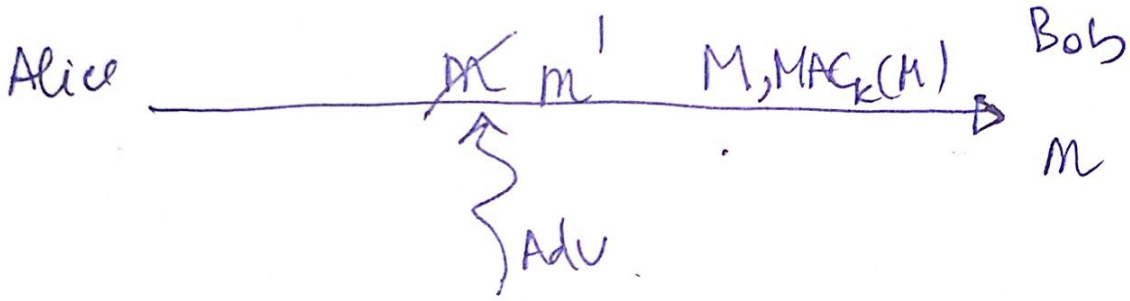
CS161 Announcements

HW2 Due Tomorrow

Midterm

Integrity & Authentication



Bob wants to check:

1. authentication : message comes from Alice
2. integrity : message is as sent by Alice (no attacker modified message)

ElGamal cph: $\underbrace{m \cdot (\ ) \bmod p}_{C}$

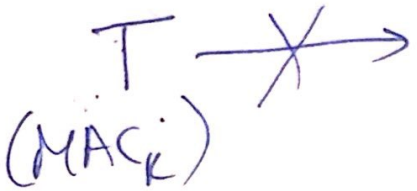
	Symmetric-key	Asymmetric-key
Confidentiality	AES-CBC, AES-CTR	El Gamal
	encryption schemes	
Integrity & authentication	MAC	Digital Signatures (RSA)
	TODAY	

MAC (Message Authentication Code)

"keyed checksum"

→ deterministic

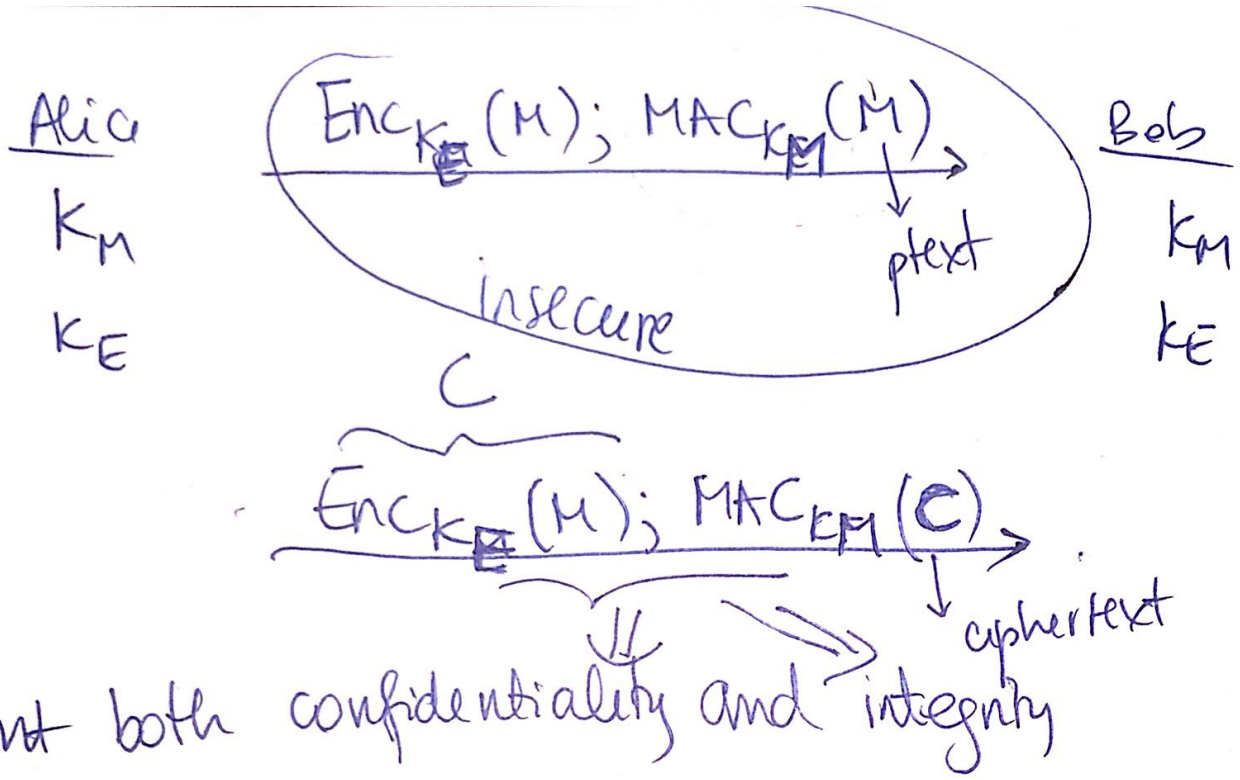
not IND-CPA



Cannot find M' & T' s.t. $MAC(k, M') = T'$

Existentially unforgeable: Adv cannot find any pair (M', T') s.t $MAC(k, M') = T$ unless it saw M', T' from a legitimate sender

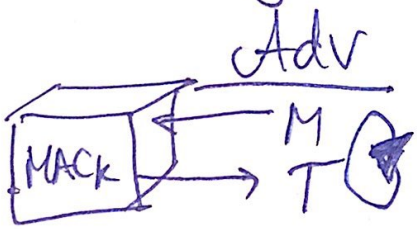
* cannot compute MAC for a new message



want both confidentiality and integrity

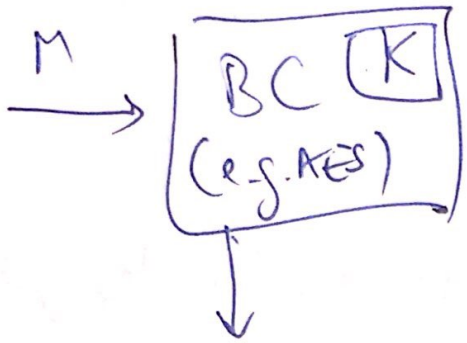
MAC security: existentially unforgeable

Ch
K



Adv knows MAC algorithm but not K

$Pr[Adv(\,) \to M', T' \text{ s.t } T' = MAC_K(M') \;\&\; \text{Adv did not ask for } M' \text{ to MAC Box}] = negl$



$\rightarrow BC(K, M)$

cannot forge without key K

fixed block length

(e.g. AES has 128 bits)

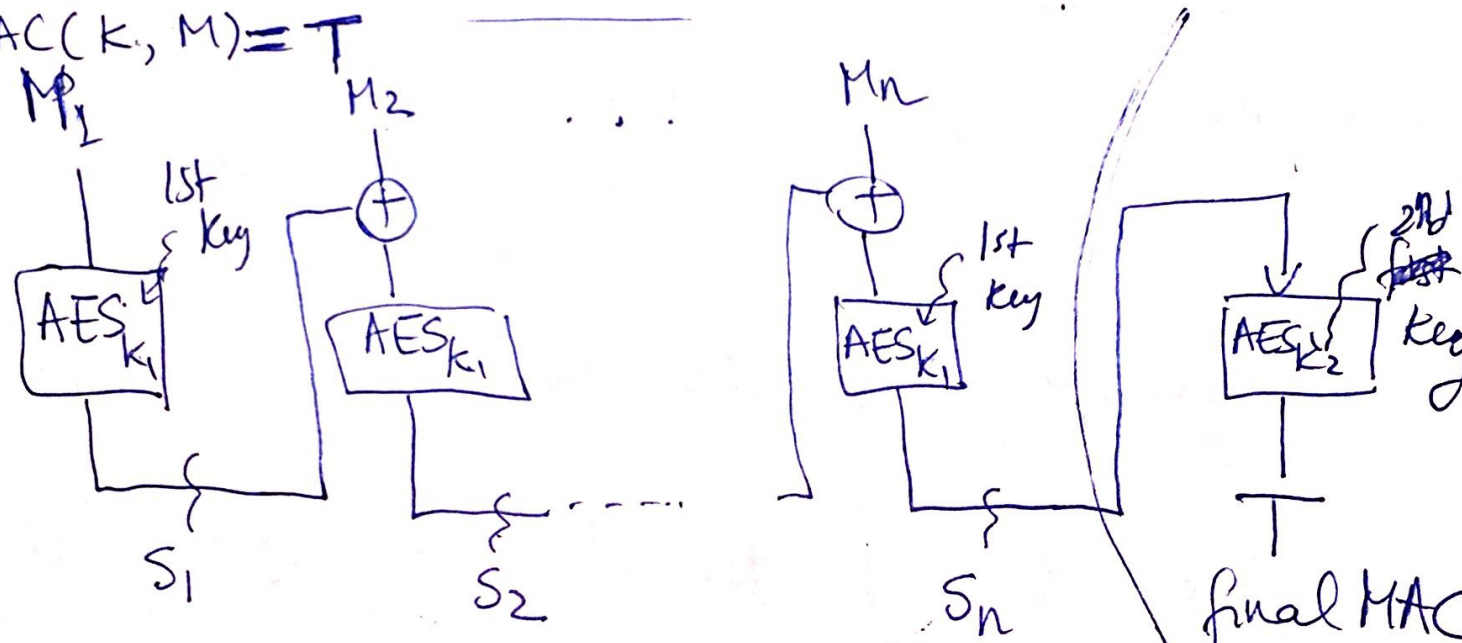
AES-EMAC: chaining mode that gives an existentially unforgeable MAC

- $K = 256 \text{ bits} = (K_1, K_2)$ = two AES keys

MAC(K, M): - split M into 128 bit blocks:

$M_1, M_2, \ldots, M_n$

MAC(K, M) = T



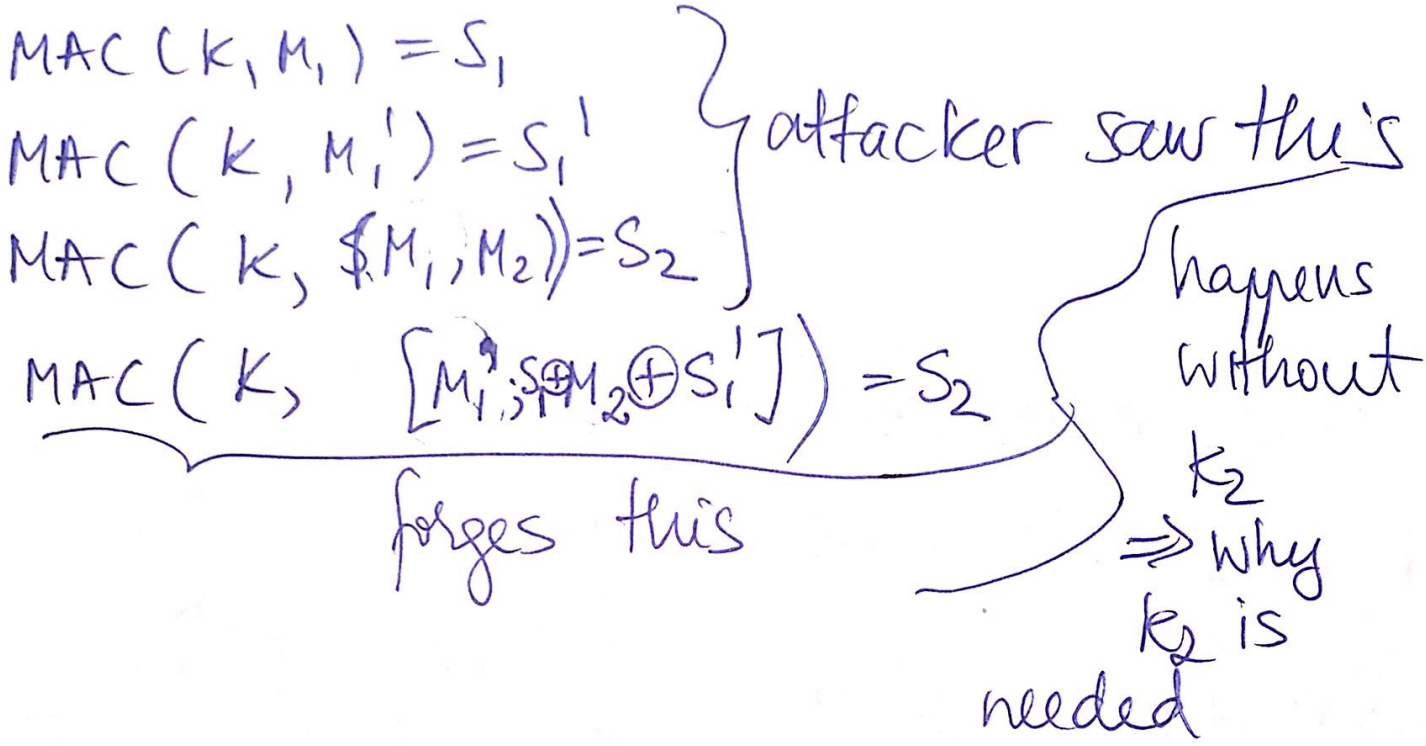
MAC(K,M) ≠ S1, S2, ..., Sn, T

not in MAC ↓ only this is MAC

because no need to decrypt

& because receiver can compute T without S1, ..., Sn

MAC(K,M) = Sn ⇒ forgeable



Forgeable $MAC(K, M) = S_n$ forgeable.

Plaintext blocks:

M_1, M_2, M_i

$MAC(k, M_1) = S_1$

$MAC(K, M_1') = S_1'$

~~$MAC(K, (M_1', M_2)) \equiv$~~

$MAC(k, (M_1, M_2)) = S_2$

$MAC(K, [M_1' \| M_2 \oplus S_1']) = S_2$?

new message

$AES_{K_1}(M_1') = S_1'$

~~$S_1' \oplus M_2 \oplus S_1'$~~

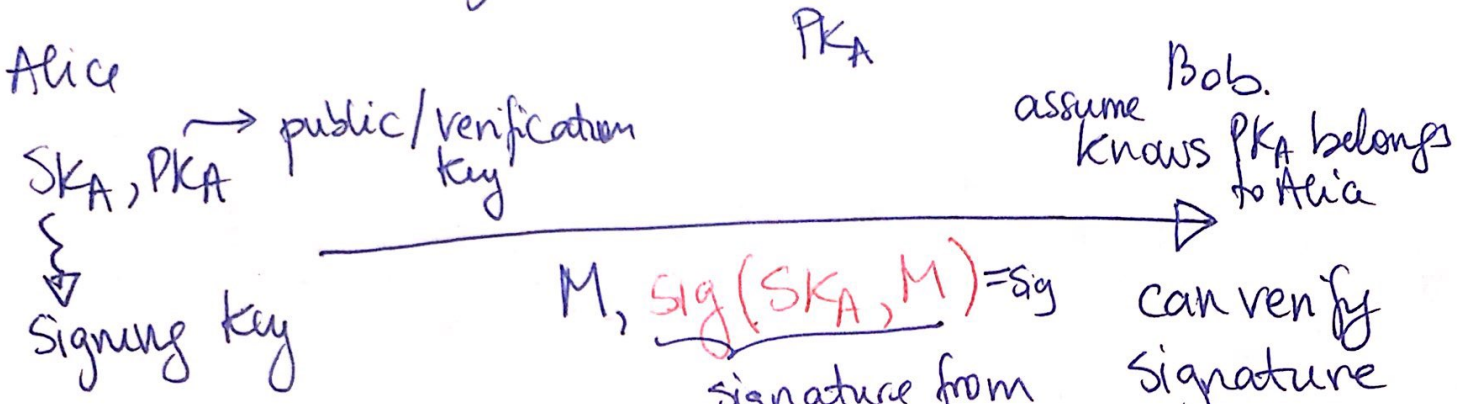
HMAC (K, M) = MAC that is also a hash. function
(collision resistant & one way)

construct from hash function

$$H(K \oplus opad \,||\, H(K \oplus ipad \,||\, M))$$

padding 0x5C repeated

Digital signatures



Syntax:

1) $KeyGen() \rightarrow (SK, PK)$
2) $Sign(SK, M) \rightarrow Sig$.
3) $Verify(PK, M, Sig) \rightarrow$?

$Verify(PK_A, M, sig) = 1$ if correctly signed by Alice

No attacker can forge a signature

Existentially unforgeable (same as for MAC) = attacker cannot produce M', sig' s.t. $Verify(PK, M', sig') = 1$ (if the attacker did not see M', sig' bef

RSA Signature.

Keygen(): pick random p, 2 primes 2048 bits = Keep secret

$p, 2 \equiv 2 \bmod 3$

$n = p \cdot 2$

Euler's totient function $\phi(n) = (p-1)(2-1)$

$x^{\phi(n)} \equiv 1 \bmod n$ (Fermat's Little Thm)

$Pk = n$

$SK = d$ = secret

computed s.t. $3d \equiv 1 \bmod \phi(n) \Rightarrow 3d = \phi(n) \cdot 2 +$

$Sign(SK, M) = \underbrace{H(M)^d}_{sig} \bmod n$, where H is crypto hash

$Verify(Pk, M, Sig) = sig^3 \bmod n \stackrel{?}{=} H(M)$

$(H(M)^d)^3 \equiv H(M) \bmod n$

Assumes factoring is hard

Cannot factor n into p & q

Crypto assumption RSA