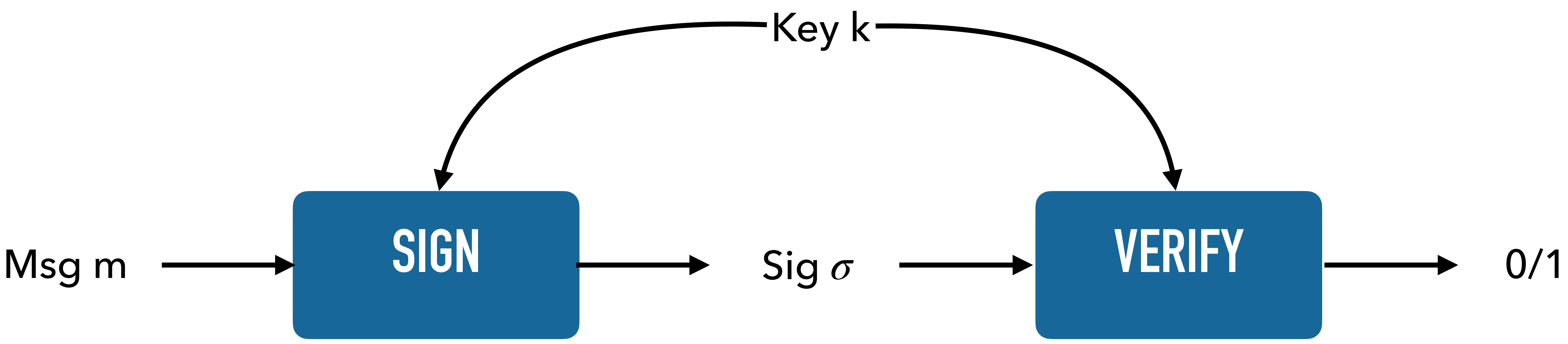
COL759:
CRYPTOGRAPHY AND COMPUTER SECURITY

2022-23 (SEMESTER 1)

LECTURE 28 PART 1: REVIEW (MAC, UHF, CRHF, AUTH. ENC)

REVIEW: MESSAGE AUTH. CODES



Weak Unforgeability

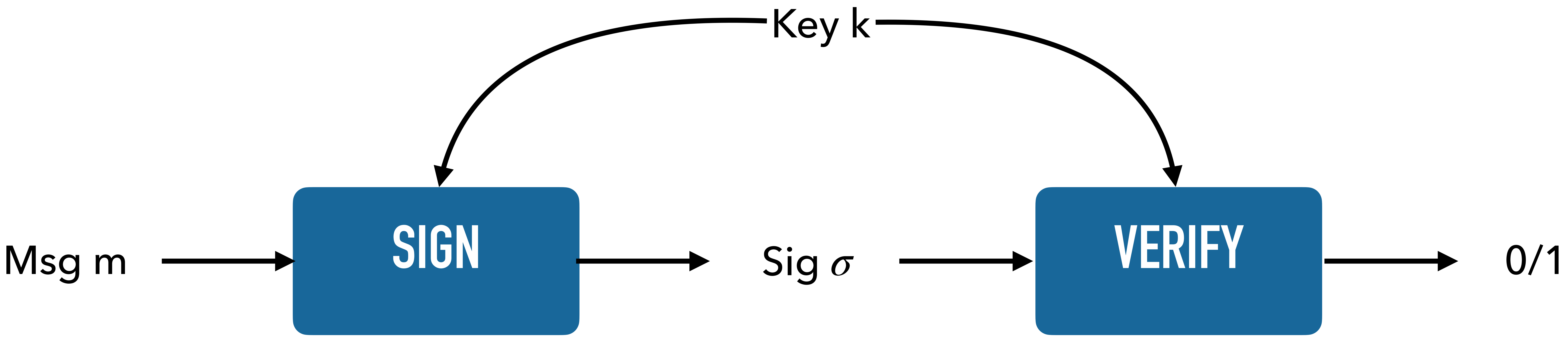
Adversary cannot produce sig. on **new** message, even after seeing many signtures.

Strong Unforgeability

Adversary cannot produce **new** sig, even after seeing many signtures.

Ver. queries useless

REVIEW: MESSAGE AUTH. CODES



- PRF based construction: bounded message space
- To support unbounded message space: ECBC-MAC, randomised counter-based MAC. Both based on PRF security
- Hash-and-sign : based on security of hash function

REVIEW: HASH FUNCTIONS

INPUT

↓

Keyed hash function H_k

↓

OUTPUT

Hard to find two different inputs that map to same output (a.k.a. 'collision')

REVIEW: HASH FUNCTIONS

Hard to find two different inputs that map to same output (a.k.a. 'collision')

Universal Hash Functions

Adversary cannot produce collision, does not receive any information about hash key

Constructions:
- polynomial based inf. theoretic construction
- PRF/MAC based construction

Collision Resistant Hash Functions

Adversary cannot produce collision, even after seeing hash key

Constructions ??
- Practical hash functions: SHA

CRHF CONSTRUCTION: ATTEMPT

$p = 2q + 1$: safe prime

g : generator of $\mathbb{Z}_p^*$

Hash key: $x, y \in \mathbb{Z}_p^*$

$H_k : \mathbb{Z}_p^* \times \mathbb{Z}_p^* \to \mathbb{Z}_p^*$

$$H_{(x,y)}(a, b) = x^a \cdot y^b \mod p$$

$x^2 = 1$ or $y^2 = 1$	$x^q = 1, y^q \neq 1$	$x^q \neq 1, y^q \neq 1$
Many collisions	(q, b) and $(2q, b)$	(q, q) and $(2q, 2q)$

AUTHENTICATED ENCRYPTION: SEMANTIC SECURITY + CIPHERTEXT INTEGRITY

After seeing many ct, adversary should not be able to produce a new ciphertext that decrypts to valid msg.

Ciphertext integrity is needed because msg. integrity does not prevent **'chosen ciphertext attacks'**

After seeing many ct, adversary should not be able to produce encryption of a new msg

SECURITY AGAINST CHOSEN CIPHERTEXT ATTACKS

Chall.		Adv.	
Chooses key k, bit b			Setup
	$\xleftarrow{\text{Encrypt } m_i}$		Pre-chall. queries
$ct_i \leftarrow Enc(m_i, k)$	$\xrightarrow{ct_i}$		
	$\xleftarrow{\text{Decrypt } ct'_i}$		
$y_i = Dec(ct'_i, k)$	$\xrightarrow{y_i}$		
	$\xleftarrow{m_0^* \quad m_1^*}$		Chall.
$ct^* \leftarrow Enc(m_b^*, k)$	$\xrightarrow{ct^*}$		
	$\xleftarrow{\text{Encrypt } m_i}$		Post-chall. queries
$ct_i \leftarrow Enc(m_i, k)$	$\xrightarrow{ct_i}$		
	$\xleftarrow{\text{Decrypt } ct'_i \ (\neq ct^*)}$		
$y_i = Dec(ct'_i, k)$	$\xrightarrow{y_i}$		
	$\xleftarrow{b'}$		Guess

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall. | Adv.

Setup: Chooses key k, bit b

Pre-chall. queries:

$\leftarrow$ Encrypt m_i

$ct_i \leftarrow Enc(m_i, k)$ — ct_i $\rightarrow$

$\leftarrow$ Decrypt ct'_i

$y_i = Dec(ct'_i, k)$ — y_i $\rightarrow$

Chall.:

$\leftarrow$ m_0^* m_1^*

$ct^* \leftarrow Enc(m_b^*, k)$ — ct^* $\rightarrow$

Post-chall. queries:

weaker than CCA security

Guess:

$\leftarrow$ b'

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall.		Adv.	
Chooses key k, bit b			Setup
	weaker than CCA security		Pre-chall. queries
	m_0^* m_1^* ←		Chall.
$ct^* \leftarrow Enc(m_b^*, k)$	ct^* →		
	Encrypt m_i ←		Post-chall. queries
$ct_i \leftarrow Enc(m_i, k)$	ct_i →		
	Decrypt ct'_i ($\neq ct^*$) ←		
$y_i = Dec(ct'_i, k)$	y_i →		
	b' ←		Guess

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall.		Adv.	
Chooses key k, bit b			Setup
	Encrypt m_i ←		Pre-chall. queries
$ct_i \leftarrow Enc(m_i, k)$	ct_i →		
	Decrypt ct'_i ←		
$y_i = Dec(ct'_i, k)$	y_i →		
	m_0^* m_1^* ←		Chall.
$ct^* \leftarrow Enc(m_b^*, k)$	ct^* →		
	Equivalent to CCA security		Post-chall. queries
	Decrypt ct'_i ($\neq ct^*$) ←		
$y_i = Dec(ct'_i, k)$	y_i →		
	b' ←		Guess

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall.		Adv.	
Chooses key k, bit b			Setup
	Encrypt m_i ←		Pre-chall. queries
$ct_i \leftarrow Enc(m_i, k)$	ct_i →		
	weaker than CCA security		
	m_0^* m_1^* ←		Chall.
$ct^* \leftarrow Enc(m_b^*, k)$	ct^* →		
	Encrypt m_i ←		Post-chall. queries
$ct_i \leftarrow Enc(m_i, k)$	ct_i →		
	Decrypt ct'_i ($\neq ct^*$) ←		
$y_i = Dec(ct'_i, k)$	y_i →		
	b' ←		Guess

AUTHENTICATED ENCRYPTION: SEMANTIC SECURITY + CIPHERTEXT INTEGRITY

After seeing many ct, adversary should not be able to produce a new ciphertext that decrypts to valid msg.

Semantic sec. + ciphertext integrity prevents
'chosen ciphertext attacks'

ENCRYPT-THEN-MAC

Semantic sec. + ciphertext integrity

AUTHENTICATED ENCRYPTION: PRACTICE QUESTION

(Enc, Dec): CCA secure encryption scheme with msg space $\{0,1\}$

Want: CCA secure encryption scheme with message space $\{0,1\}^n$

Candidate scheme

$$\mathsf{Enc}\big(m = (m_1, m_2, \ldots, m_n), k\big) = \Big(\mathsf{Enc}(m_1, k), \mathsf{Enc}(m_2, k), \ldots, \mathsf{Enc}(m_n, k)\Big)$$

What if we use different key for each position?

How to make this CCA secure?

AUTHENTICATED ENCRYPTION + KEY EXCHANGE

Alice and Bob can securely communicate!