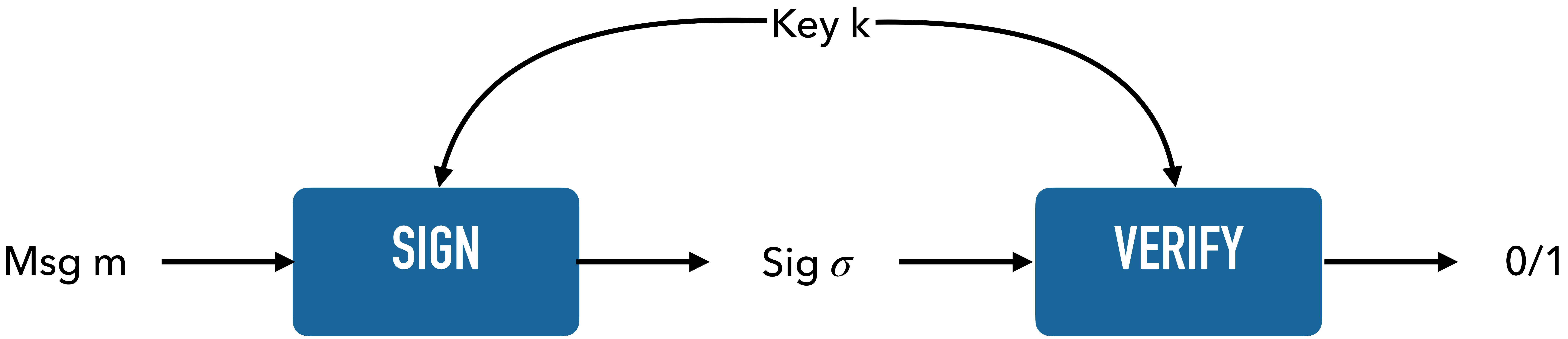
COL759:

CRYPTOGRAPHY AND COMPUTER SECURITY

2022-23 (SEMESTER 1)

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

REVIEW: MESSAGE AUTH. CODES



Weak Unforgeability

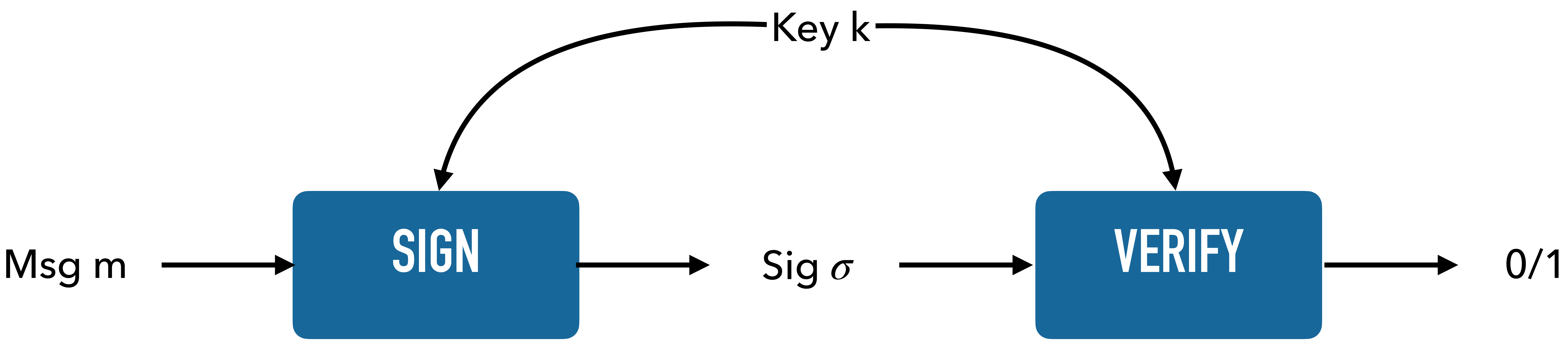
Adversary cannot produce sig. on **new** message, even after seeing many signtures.

Strong Unforgeability

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

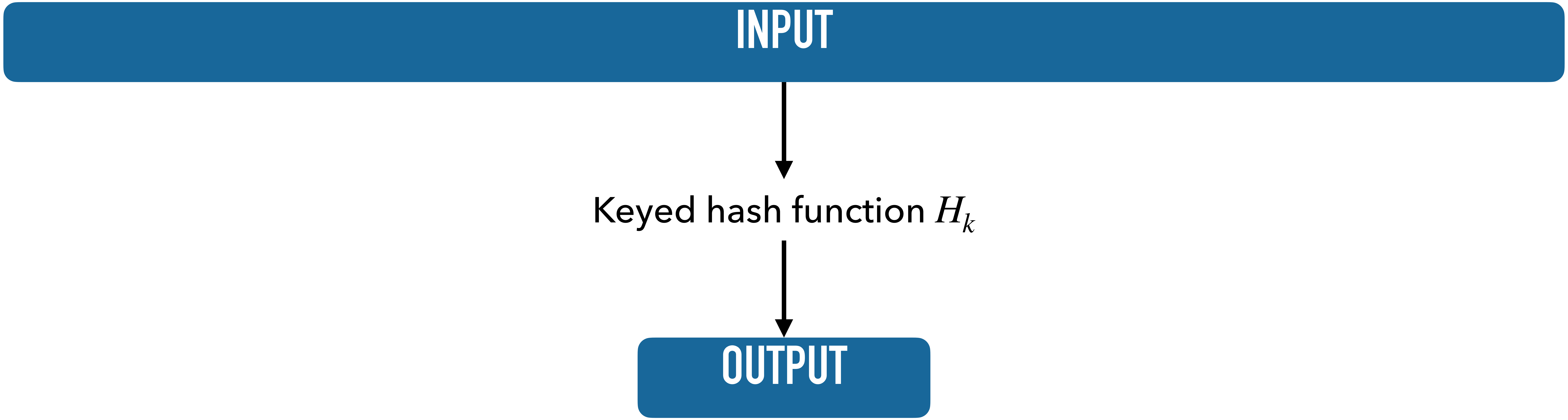
Verification queries useless

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 $y^2 = 1$

Many collisions

$x^q = 1,\ y^q \neq 1$

(q, b) and $(2q, b)$

$x^q \neq 1,\ y^q \neq 1$

(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

Encrypt m$_i$

$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{m}_i, k)$

ct$_i$

Decrypt ct′$_i$

$\mathsf{y}_i = \mathsf{Dec}(\mathsf{ct}'_i, k)$

y$_i$

Pre-chall.
queries

$m_0^* \quad m_1^*$

$\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{m}_b^* , k)$

ct*

Chall.

Encrypt m$_i$

$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{m}_i , k)$

ct$_i$

Decrypt ct′$_i$ ($\neq$ ct*)

$\mathsf{y}_i = \mathsf{Dec}(\mathsf{ct}'_i, k)$

y$_i$

Post-chall.
queries

b′

Guess

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall.

Adv.

Chooses key k, bit b

Setup

Encrypt m_i

$ct_i \leftarrow Enc(m_i, k)$

ct_i

Decrypt ct'_i

$y_i = Dec(ct'_i, k)$

y_i

Pre-chall. queries

m_0^* m_1^*

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

ct^*

Chall.

weaker than CCA security

Post-chall. queries

b′

Guess

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall.

Adv.

Setup

Chooses key k, bit b

Pre-chall. queries

weaker than CCA security

Chall.

m_0^* m_1^*

$\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{m}_b^*, k)$

ct^*

Post-chall. queries

Encrypt m_i

$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{m}_i, k)$

ct_i

Decrypt ct'_i ($\neq \mathsf{ct}^*$)

$\mathsf{y}_i = \mathsf{Dec}(\mathsf{ct}'_i, k)$

y_i

Guess

b′

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall.

Adv.

Chooses key k, bit b

Setup

Encrypt m_i

$ct_i \leftarrow Enc(m_i, k)$

ct_i

Decrypt ct'_i

$y_i = Dec(ct'_i, k)$

y_i

Pre-chall. queries

m_0^* m_1^*

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

ct^*

Chall.

Equivalent to CCA security

Decrypt ct'_i ($\neq$ ct*)

$y_i = Dec(ct'_i, k)$

y_i

Post-chall. queries

b′

Guess

WHICH OF THESE QUERIES ARE USELESS?

Not part of syllabus

Chall.

Adv.

Chooses key k, bit b

Setup

Encrypt m_i

$ct_i \leftarrow Enc(m_i, k)$

ct_i

Pre-chall. queries

weaker than CCA security

m_0^* m_1^*

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

ct^*

Chall.

Encrypt m_i

$ct_i \leftarrow Enc(m_i, k)$

ct_i

Decrypt ct'_i ($\neq ct^*$)

$y_i = Dec(ct'_i, k)$

y_i

Post-chall. queries

b′

Guess

AUTHENTICATED ENCRYPTION: SEMANTIC SECURITY + CIPHERTEXT INTEGRITY

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

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

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

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

Not secure. Given an encryption of $(m_1, m_2, \ldots, m_n)$, we can construct encryption of $(m_2, m_1, \ldots, m_n)$
Use decryption oracle to break distinguish between challenge messages.

What if we use different key for each position?

Not secure. Will need to make one encryption query, receive ct, followed by a challenge query, receive ct*.
Then mix ct and ct* to create a decryption query. Use this to learn whether ct* is encryption of m_0 or m_1

How to make this CCA secure?

Compute signature on the entire ciphertext. You can show that this construction is semantically secure.
Therefore, by computing signature on the entire ciphertext, we are using Encrypt-and-MAC approach, which is CCA secure.

AUTHENTICATED ENCRYPTION + KEY EXCHANGE

Alice and Bob can securely communicate!

Time to celebrate :) Let me know if you didn't get a brownie.
End of lecture…