10/25/26

COMP 435: *Computer Security Concepts*



Lecture 25: Diffie Hellman Key Exchange, Collision Resistant Hashing



The University of North Carolina at Chapel Hill

https://tinyurl.com/comp435-fa26

Logistics



- Quiz is not moving! Thank you for participating in democracy.
  - Remember to vote
- Lab 3 is out
  - Due 11/5

Plan for Class Today

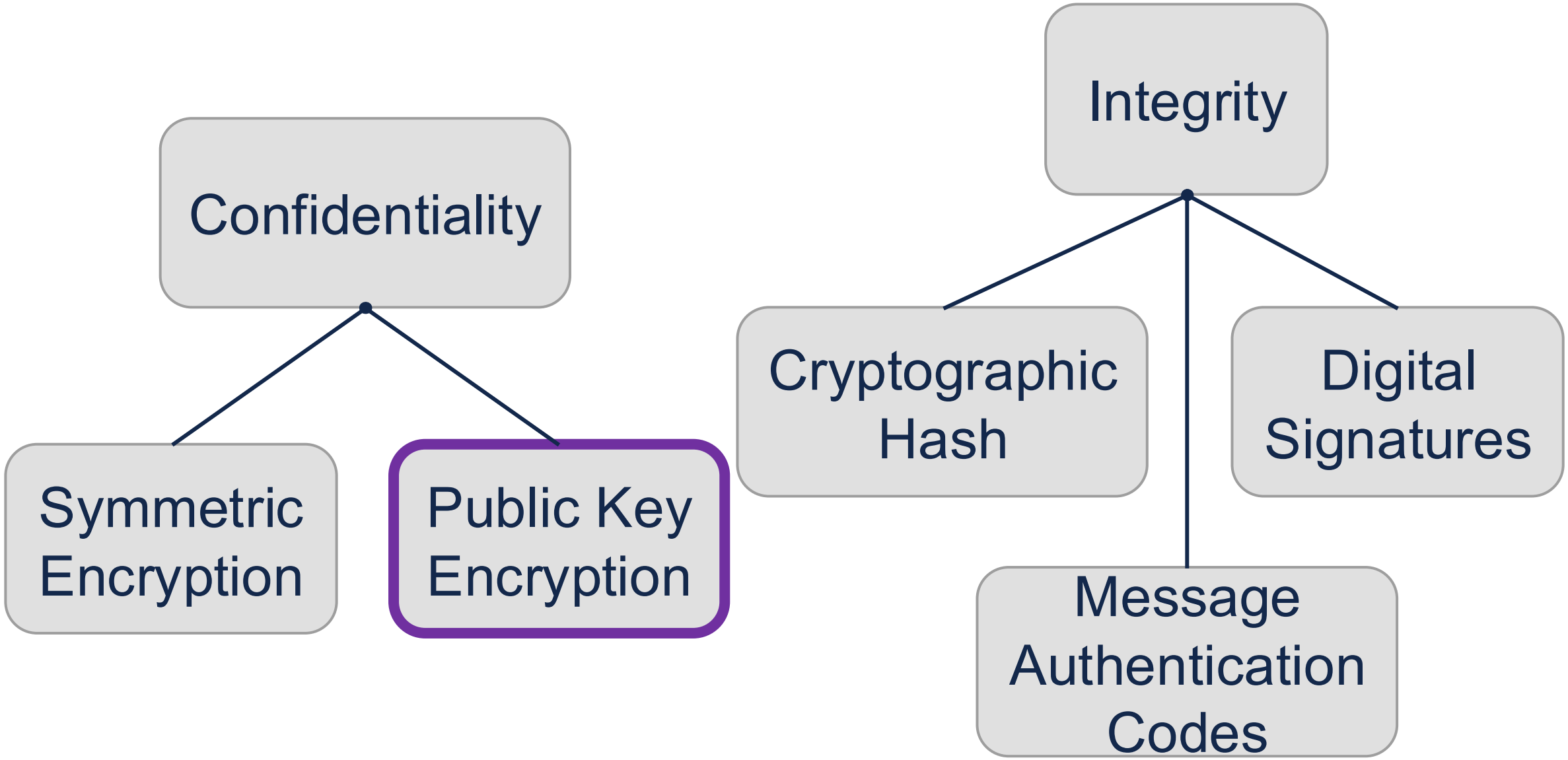


- 10:10-10:35 – Diffie Hellman Key Exchange
- 10:35- 11:00 – Collision Resistant Hashing

Guiding Question ☺

How can we establish shared secrets and detect changes to data over an insecure network?

Roadmap


Confidentiality
Symmetric Encryption
Public Key Encryption
Integrity
Cryptographic Hash
Digital Signatures
Message Authentication Codes

Public Key Encryption

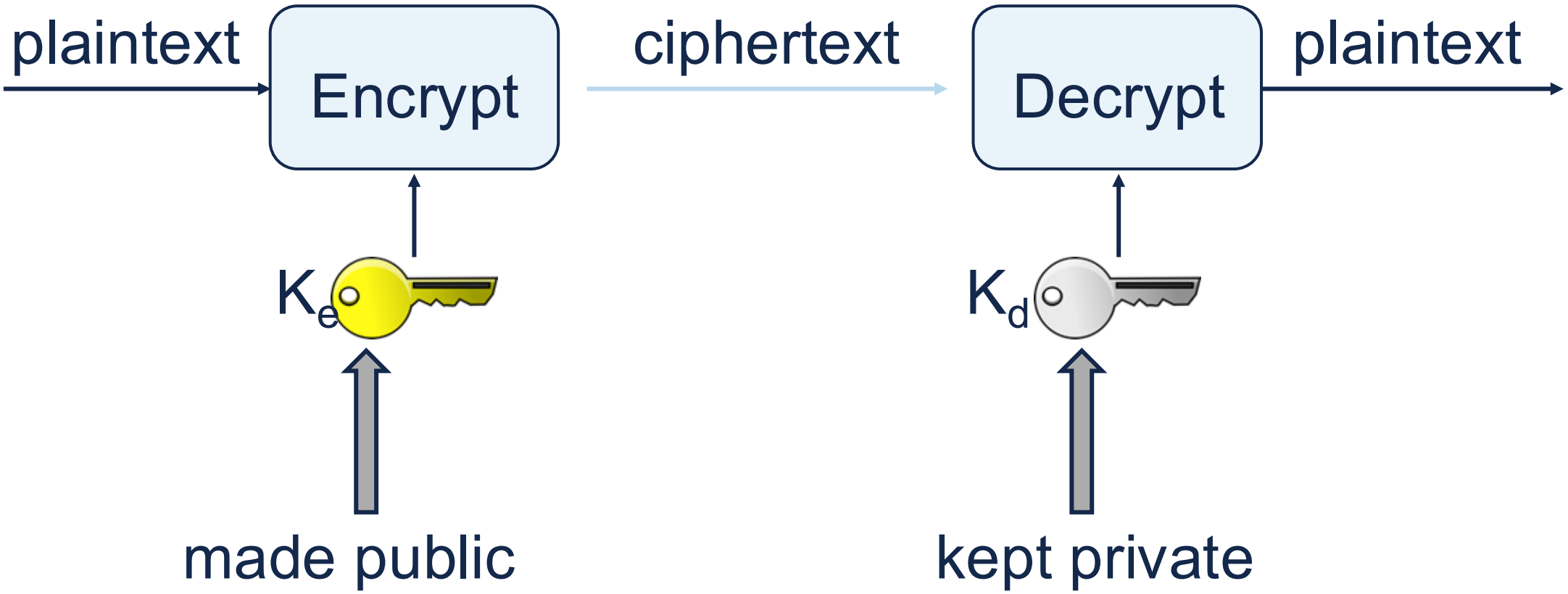


Public Key Encryption



Public Key Encryption

$$c = Enc_{Ke}(msg)$$
$$msg = Dec_{Kd}(c)$$





Establishing a shared secret key! ☺

Key Establishment



Def’n: a means of agreeing on a shared secret

E.g., Two communicating parties establish the symmetric key to be used for AES encryption



Hybrid Encryption for Key Establishment



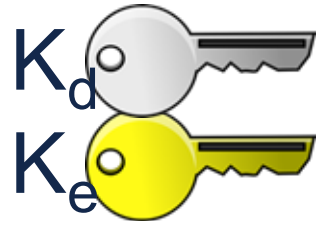
k

msg

Dear Bob,
The secret word is cookie.
Sincerely,
Alice







$c_1 = PK\text{-}Enc_{Ke}(k)$

$c_2 = S\text{-}Enc_k(msg)$

c_1, c_2

$k = PK\text{-}Dec_{Kd}(c_1)$

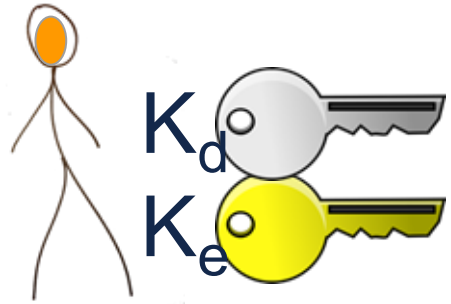
$msg = S\text{-}Dec_k(c_2)$

Purple encrypts session key S with Orange's public key! Then, they use S for symmetric encryption.

Hybrid Encryption



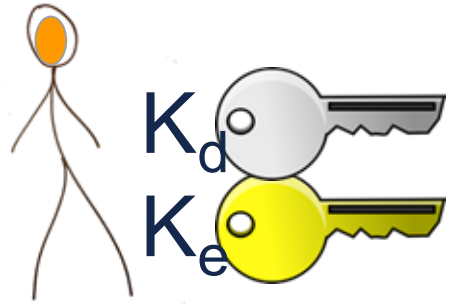
I'm Purple. Let's use: RSA-$\mathtt{Enc}_{K_e}(S)$



Hybrid Encryption: Replay Attack



I'm Purple. Let's use: $Enc_{Ke}(S)$

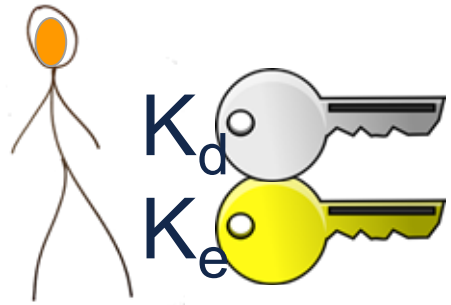


The attacker records the setup message and sends it again later, claiming to be purple.

Orange may accept old key as a new session key. Encryption alone doesn't establish the sender's identity or message freshness.



I'm Purple. Let's use: $Enc_{Ke}(S)$



Nonce

Def'n: a number used only once as part of a cryptographic protocol

- random number used to provide freshness, message continuity
- sequence number to demonstrate uniqueness
- timestamp to demonstrate timeliness

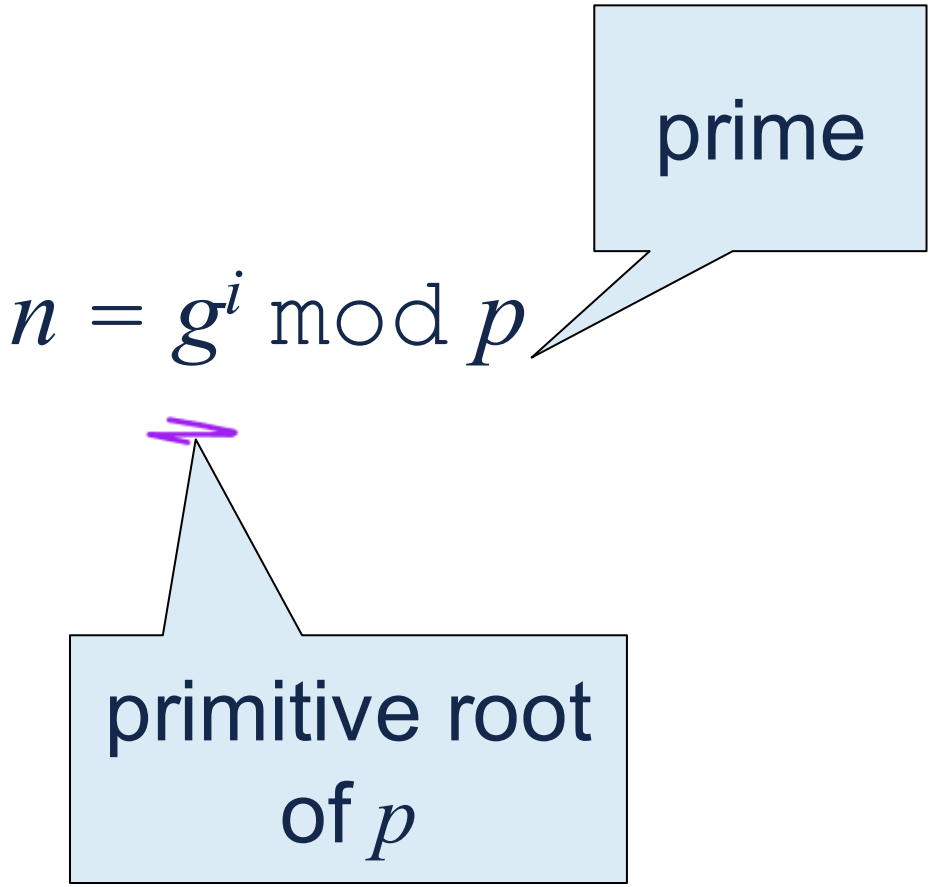
E.g., IV used in CBC, OFB, Ctr modes of block ciphers



Diffie-hellman Key Agreement

Diffie-Hellman Key Agreement

group ~ set



For all n, $1 \leq n < p$,

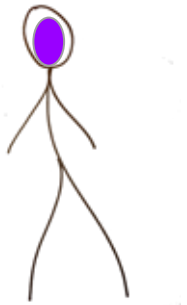
there exists an integer i such that

$$g^i \bmod p = n$$

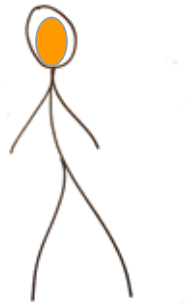
$$n = g^i \bmod p$$

primitive root of p

Diffie-Hellman Key Agreement



g, p



$a < p$

$A := g^a \bmod p$

$b < p$

$B := g^b \bmod p$

A

B

$K := B^a \bmod p$

$= (g^b)^a \bmod p$

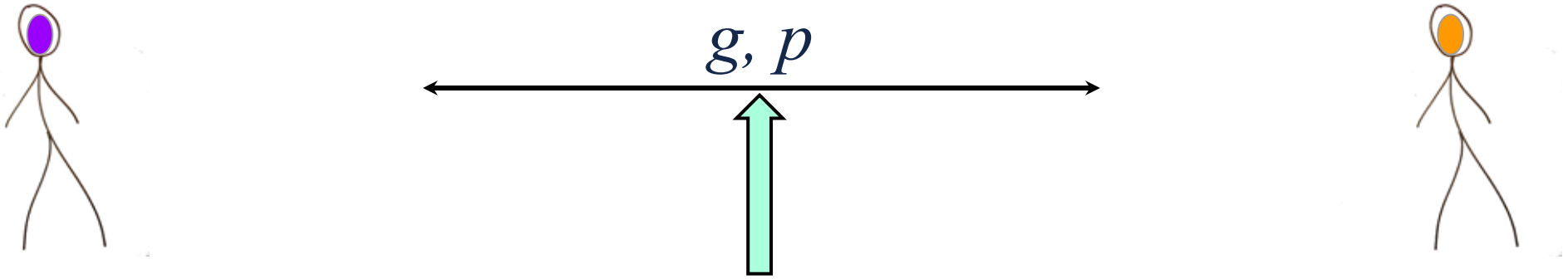
$= g^{ba} \bmod p$

$K := A^b \bmod p$

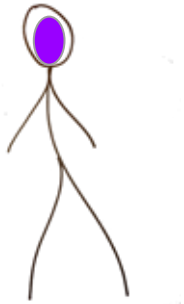
$= (g^a)^b \bmod p$

$= g^{ab} \bmod p$

Diffie-Hellman Key Agreement



Diffie-Hellman Key Agreement





g, p

$a < p$

$A := g^a \bmod p$

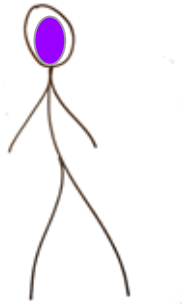
$b < p$

$B := g^b \bmod p$

A

B

Diffie-Hellman Key Agreement





g, p

$a < p$

$A := g^a \bmod p$

$b < p$

$B := g^b \bmod p$

A

B

$K := B^a \bmod p$

$= (g^b)^a \bmod p$

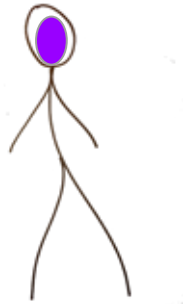
$= g^{ba} \bmod p$

$K := A^b \bmod p$

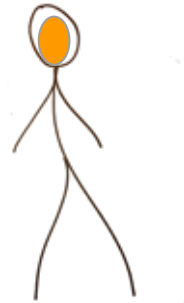
$= (g^a)^b \bmod p$

$= g^{ab} \bmod p$

Diffie-Hellman Key Agreement



g, p



$a < p$

$A := g^a \bmod p$

$b < p$

$B := g^b \bmod p$

A

B

$K := B^a \bmod p$

$= (g^b)^a \bmod p$

$= g^{ba} \bmod p$

$K := A^b \bmod p$

$= (g^a)^b \bmod p$

$= g^{ab} \bmod p$

DH Correctness



- $K = A^b = (g^a)^b = B^a = (g^b)^a \bmod p$
- Both sides have computed the same value for key K

DH Security

- $K = A^b = (g^a)^b = (g^b)^a = B^a \bmod p$
- The attacker's challenge:
  - given g, p, A, B,
  - find K
- If the attacker can calculate the discrete log of A or B, they win

One-Way Function

- a function F: X → Y is one-way if F is easy to compute but hard to invert
- modular exponentiation is one-way
  - computing $g^a \bmod p = A$ is easy
  - computing $\text{dlog}_{g,p}(A) = a$,
    the discrete log of A mod p with base g, is hard

Adversary-in-the-Middle attack on DH





a

$A := g^a \bmod p$

c, d

$C := g^c \bmod p$

$D := g^d \bmod p$

b

$B := g^b \bmod p$

A →

← D

C →

← B

$K_{ad} := D^a \bmod p$

K_{ad}

K_{cb}

$K_{cb} := C^b \bmod p$

Adversary-in-the-Middle attack on DH







No authentication ☹

a

$A := g^a \bmod p$

b

$B := g^b \bmod p$

D

B

$K_{ad} := D^a \bmod p$

K_{ad}

K_{cb}

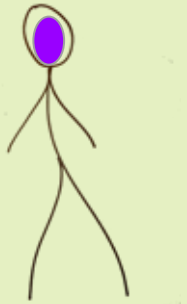
$K_{cb} := C^b \bmod p$

Machine in the Middle Attack (MitM) (or Adversary in the Middle (AitM))



Def’n: two parties, believing that they are communicating with each other, are in fact communicating with an attacker

Diffie-Hellman Key Agreement (Q3 from last time)



$g=2,\ p=11$



$a = 3$

$A = \ ?$

$b = 5$

$B = \ ?$

A

B

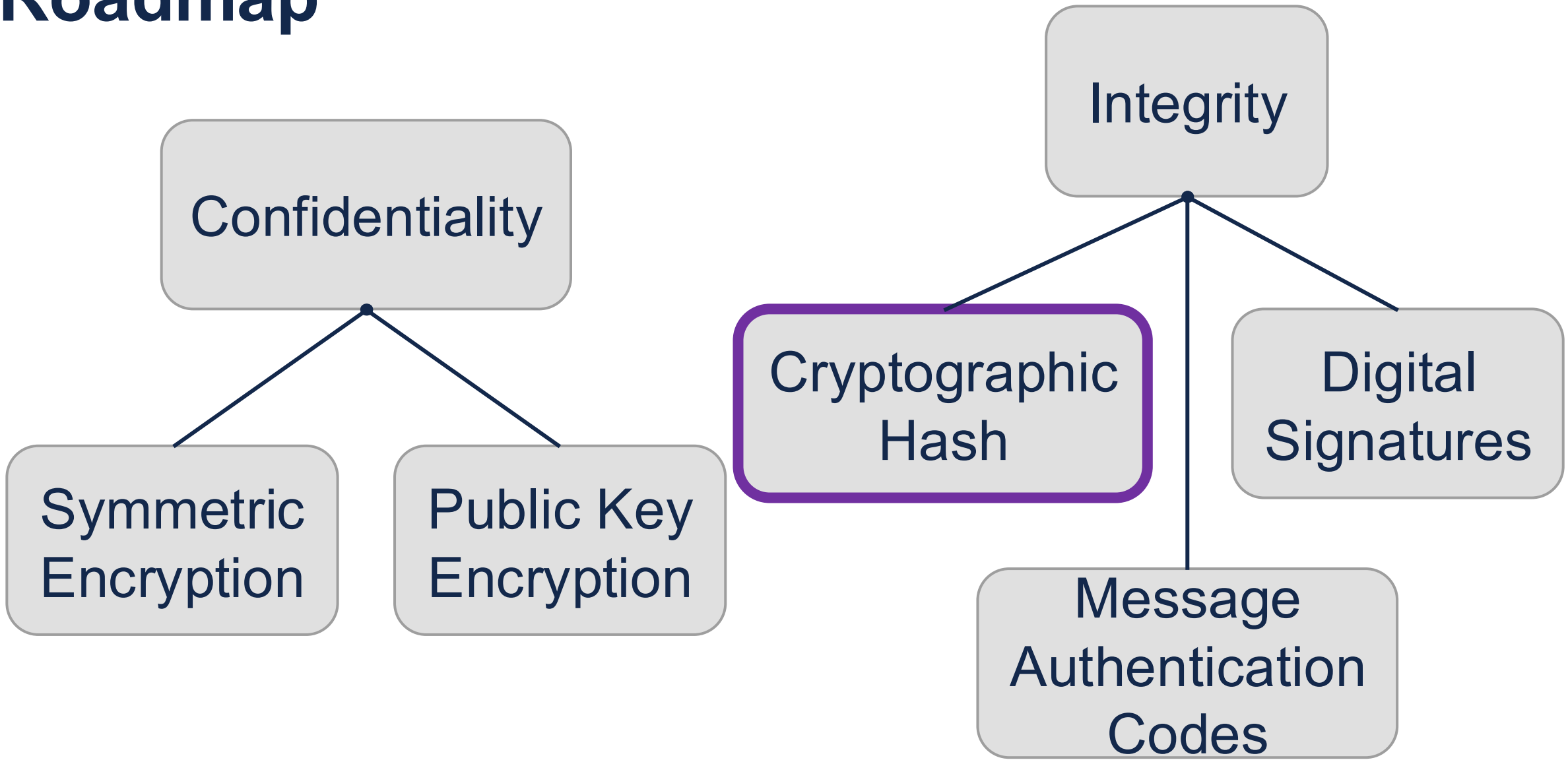
$K \ = \ ?$

$K \ = \ ?$



Cryptographic Hash Functions

Roadmap


Integrity
Confidentiality
Cryptographic Hash
Digital Signatures
Symmetric Encryption
Public Key Encryption
Message Authentication Codes

Message Integrity



Def’n: message has not been modified

Alice

Bob



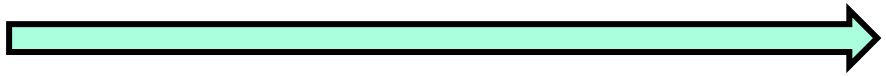




Dear Bob,
Do not trust Mallory!
Sincerely,
Alice

msg

msg



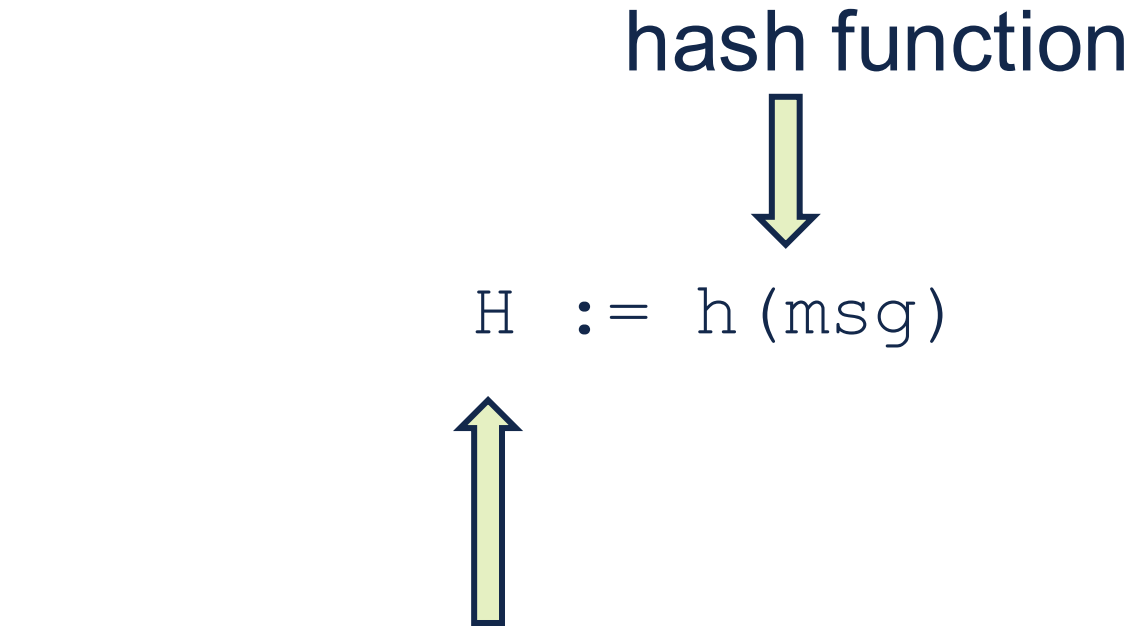
Dear Bob,
You can trust Mallory.
Sincerely,
Alice

msg

Goal: Bob would like to know that the message has been modified.

Tool: cryptographic hash functions!

Terminology



Hash Function

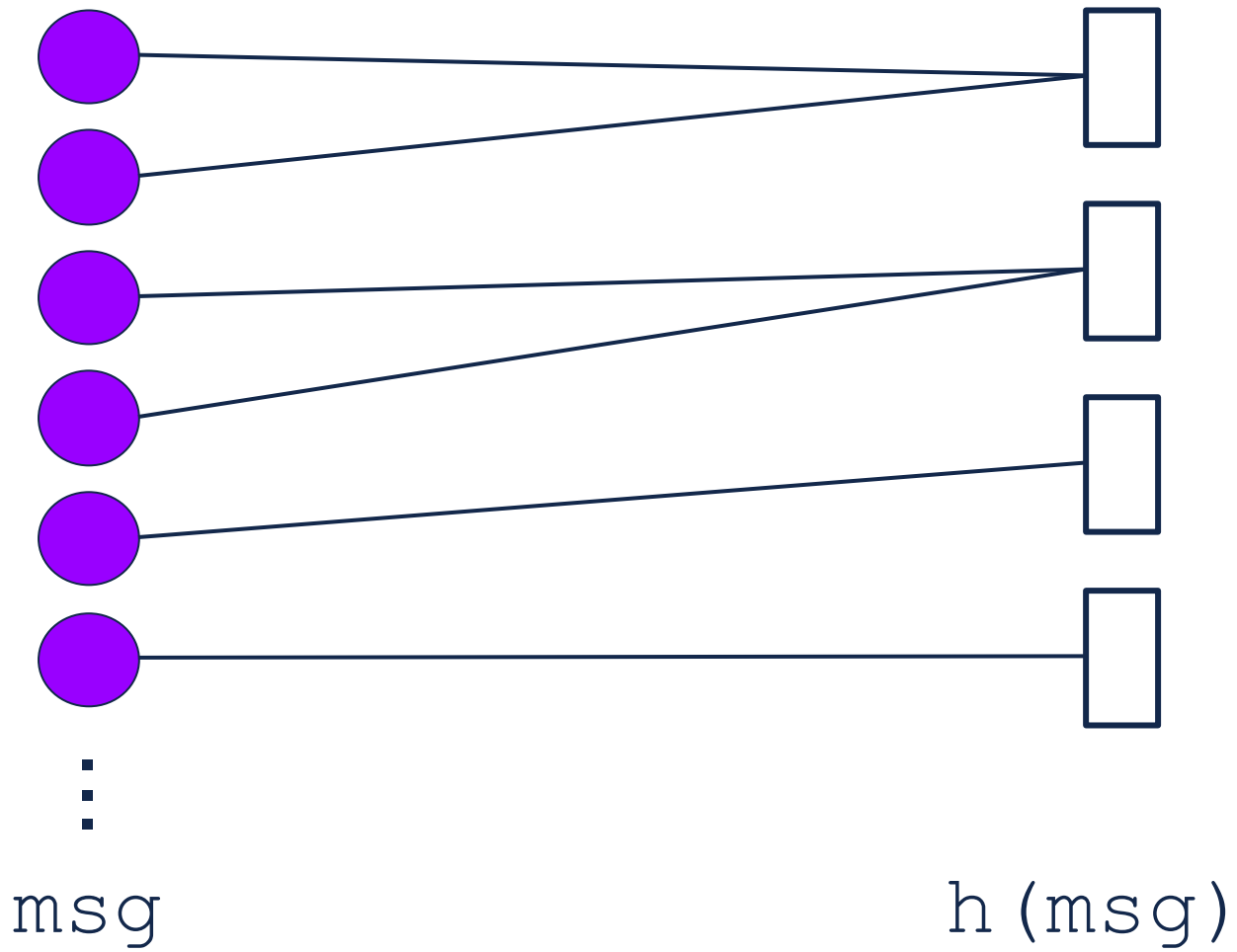


Def’n: a function that takes an arbitrary-length input and produces a fixed-length output

E.g., Modulo operation

3 % 10 = 3
3590224723940253 % 10 = 3

Pigeonhole Principle



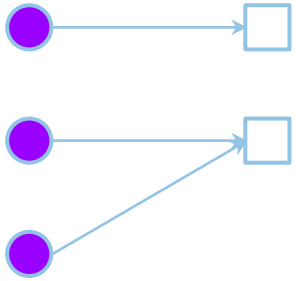
Functions

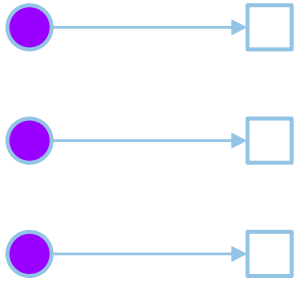
surjective

¬injective

injective

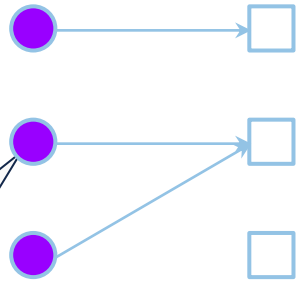
¬surjective

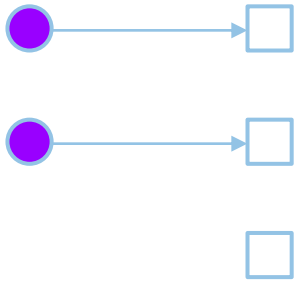




bijective

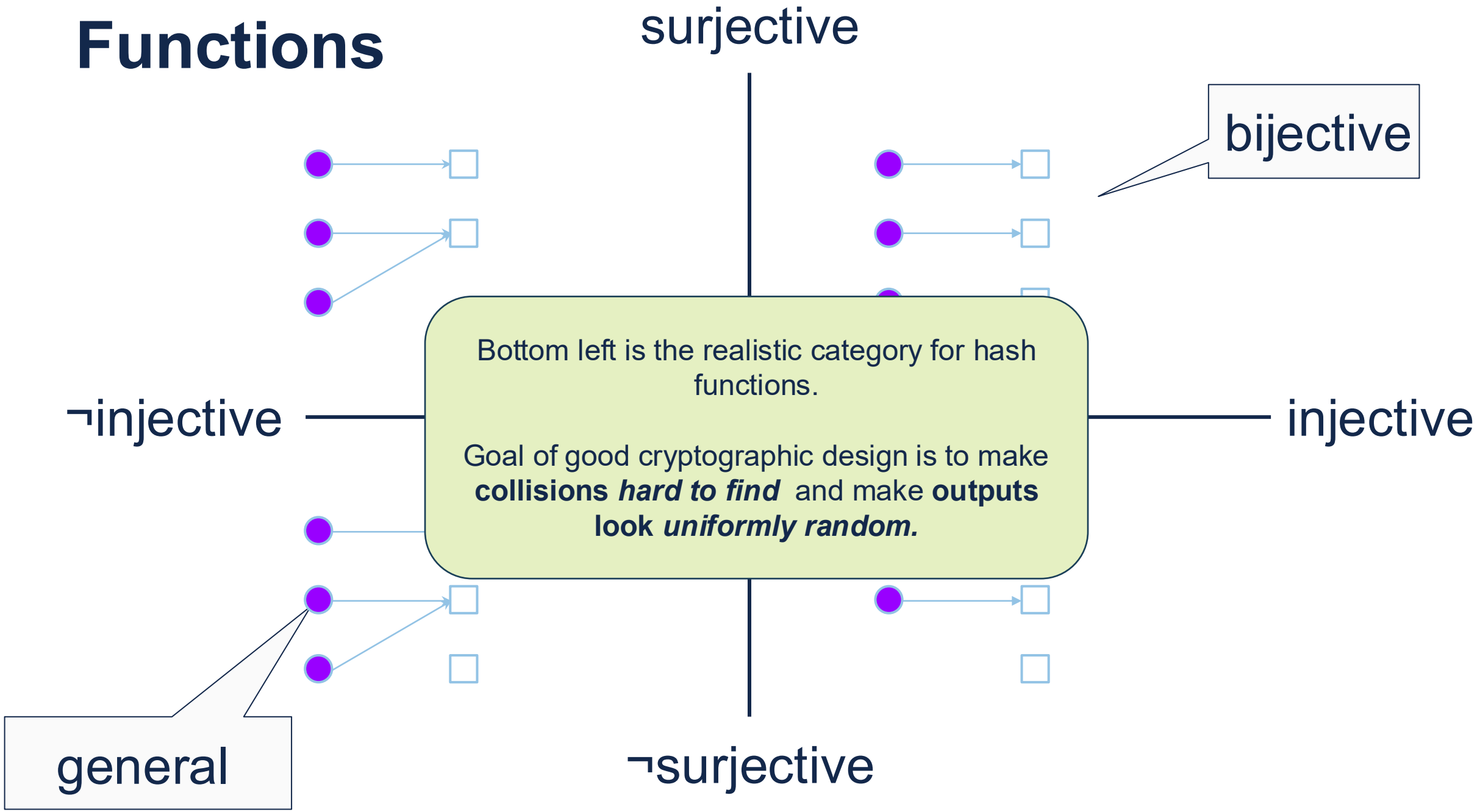
Recall:
*Bijective: every input has unique output AND
* Every output used
* AKA Injective + surjective



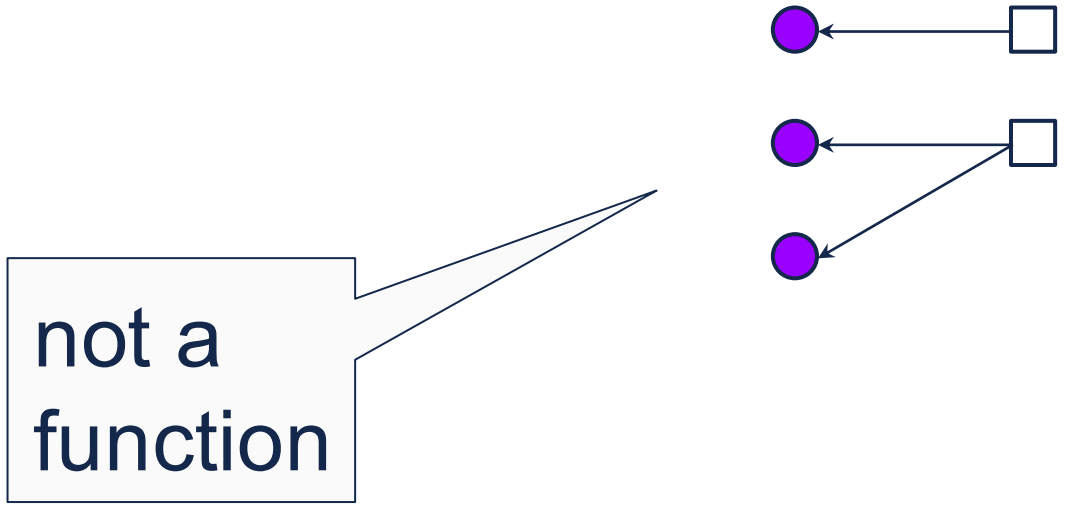


general

Functions

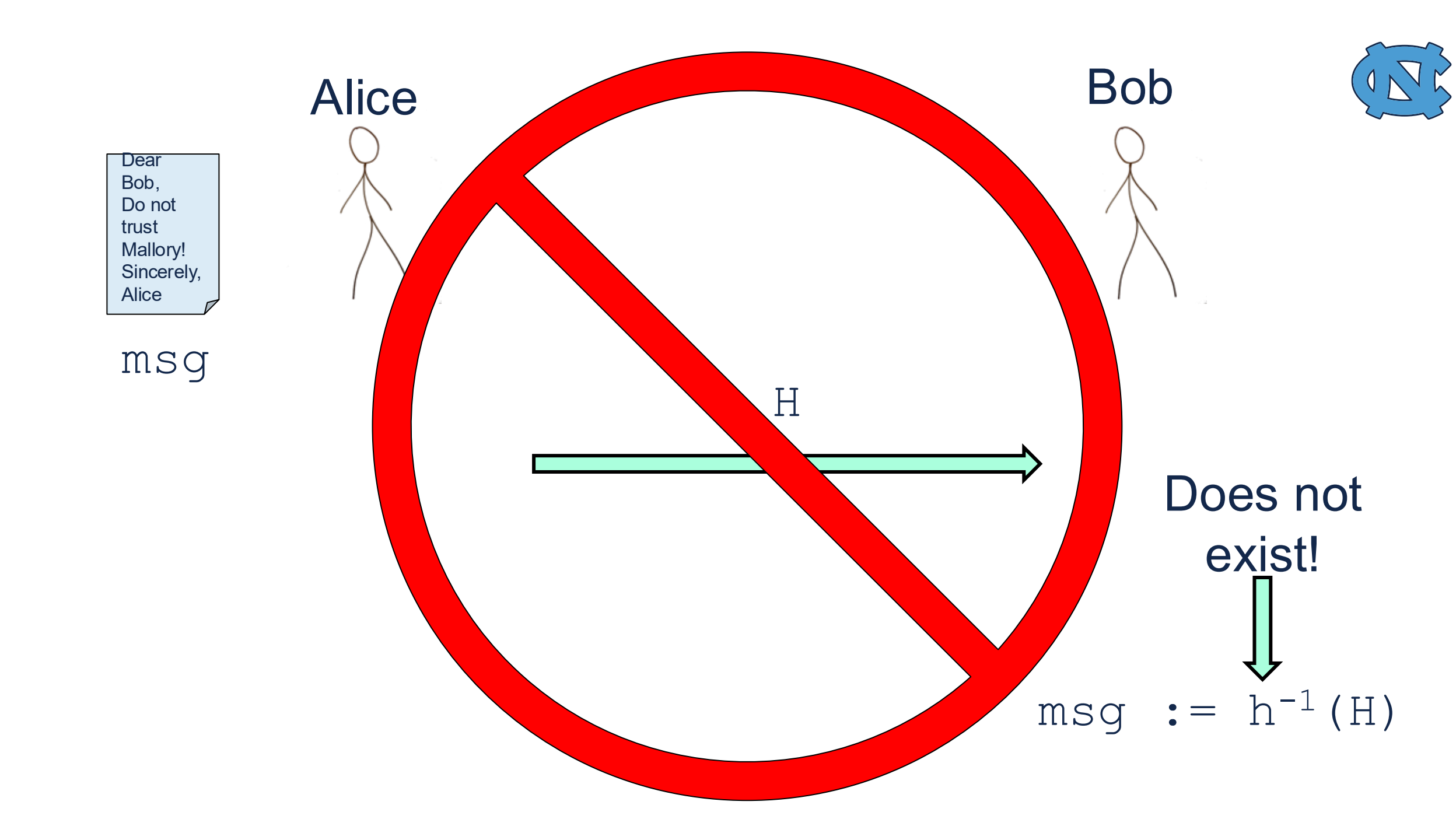

surjective
bijective
¬injective
injective
Bottom left is the realistic category for hash functions.
Goal of good cryptographic design is to make **collisions *hard to find*** and make **outputs look *uniformly random.***
general
¬surjective

Inverse of a hash function?



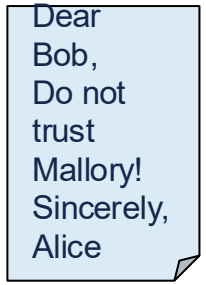
Hash functions not invertible!

One input mapping to two outputs.


Alice
Bob
Dear Bob, Do not trust Mallory! Sincerely, Alice
msg
H
Does not exist!
msg := h^-1(H)

Alice

Bob

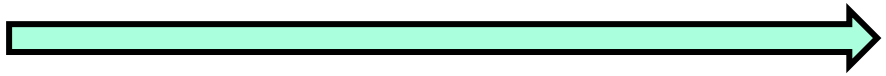






msg

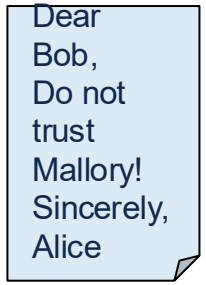
(msg, H)



$H \stackrel{?}{=} h(msg)$

Alice

Bob

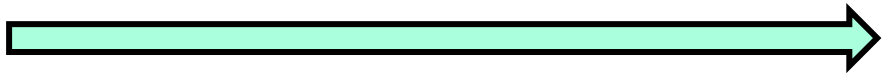






msg

(msg, H)



h ≜ msg % 256

Many collisons! ☹

H ?= h(msg)



***** A hash can reveal modifications, but cannot guarantee that no modifications exist! *****



Cryptographic Hash Functions

Cryptographic Hash Functions



Def'n: a hash function that is strong enough to resist collisions

- collision resistant
  - should be hard to find **any two different** inputs that hash to the same output
- second pre-image resistant
  - can't find a different input with same hash
- pre-image resistant:
  - can't go backwards

These are in decreasing order of strength!

Collision Resistant



Hard to find `msg`, `msg'` such that

`msg ≠ msg'` and

`h(msg) = h(msg')`

Second Pre-image Resistant



Given `msg`, hard to find `msg′` such that

`msg ≠ msg′` and

`h(msg) = h(msg′)`

Pre-image Resistant



Let `H := h(msg)`
Given `H`, hard to find `msg'` such that
`H = h(msg')`

Cryptographic Hash Functions



Def’n: a hash function that is strong enough to resist collisions

- collision resistant
- second pre-image resistant
- pre-image resistant



Collision resistant & the birthday paradox

Number of people	Probability of a birthday collision
5	3%
20	41%
50	97%
100	99.99%

Cryptographic Hash Functions

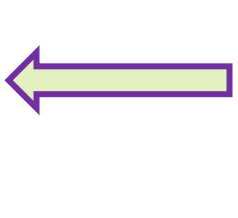


- MD5
- SHA-1
- SHA-2
- SHA-3

Cryptographic Hash Functions



- MD5
- SHA-1
- SHA-2
- SHA-3



deprecated

Cryptographic Hash Functions



- MD5
- SHA-1
- SHA-2
- SHA-3



SHA-256

Cryptographic Hash Functions



- MD5
- SHA-1
- SHA-2
- SHA-3



Multiple Variants

WS Questions!