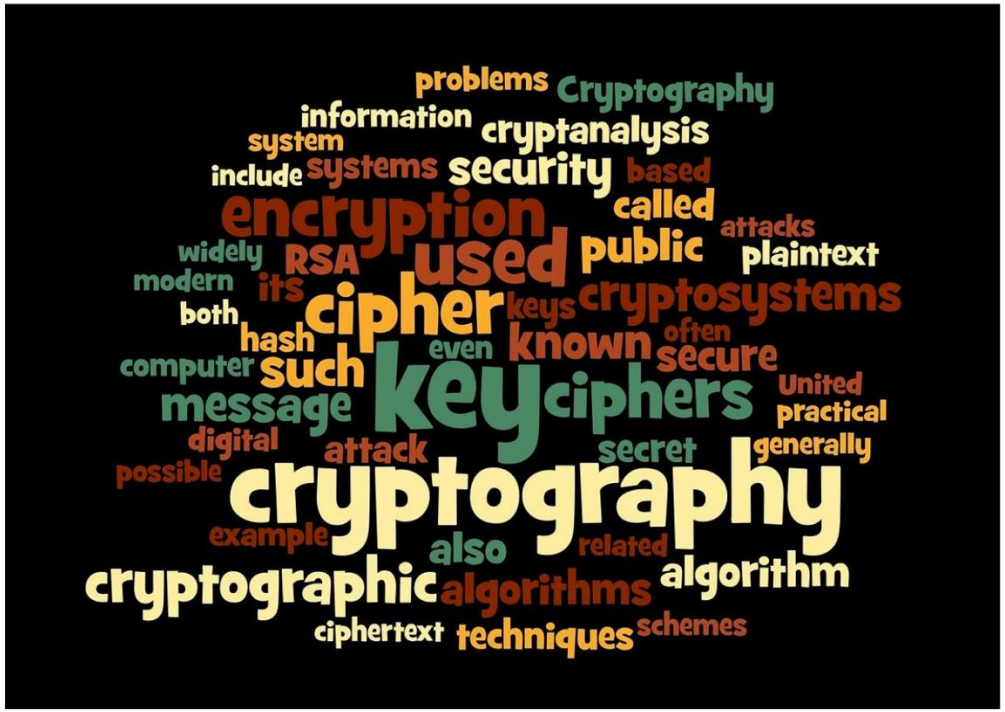
Applied Cryptography

Mike “Perk” Perkins
mperkin2@andrew.cmu.edu



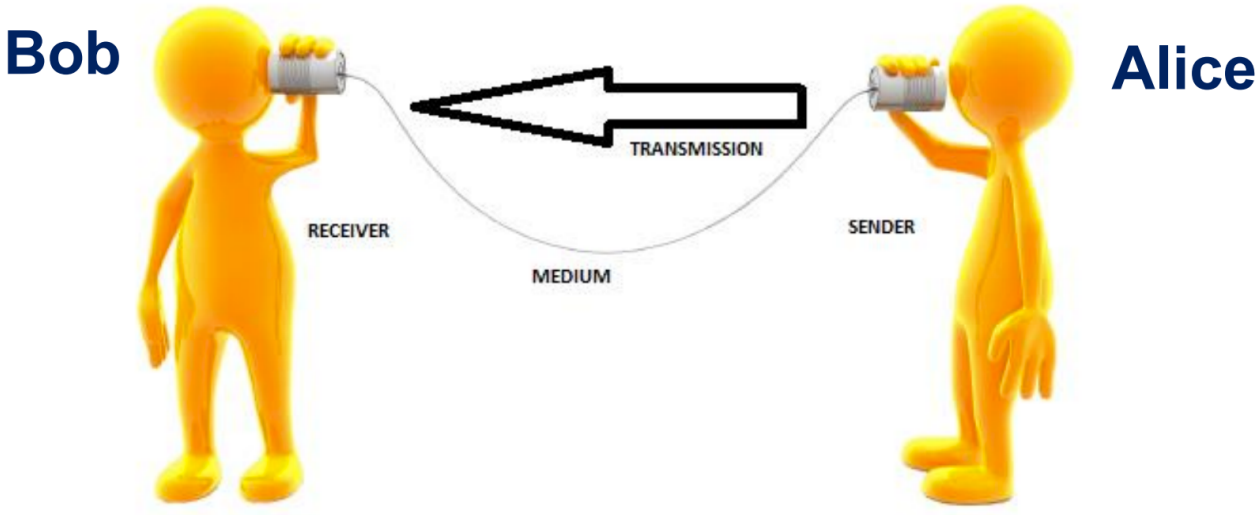
https://www.michalsons.com/wp-content/uploads/2012/05/cryptographic-1091257_1280.jpg

Lecture 1

Introduction and Course Logistics

Fundamental Objective of Cryptography

Allow Alice to securely send a message to Bob that no-one else can read



https://www.daytodaygk.com/wp-content/uploads/2015/12/communic.png

Eve the "Eavesdropper"

Classic evil actor

eaves·drop

/ˈēvzˌdräp/

verb

secretly listen to a conversation.
"she opened the window just enough to **eavesdrop on** the conversation outside"

Similar: listen in | spy | intrude | monitor | tap | wiretap | record | overhear

Faythe the "Faithful"

Carnegie Mellon Africa

Classic friendly actor

faith·ful

/ˈfāTHfəl/

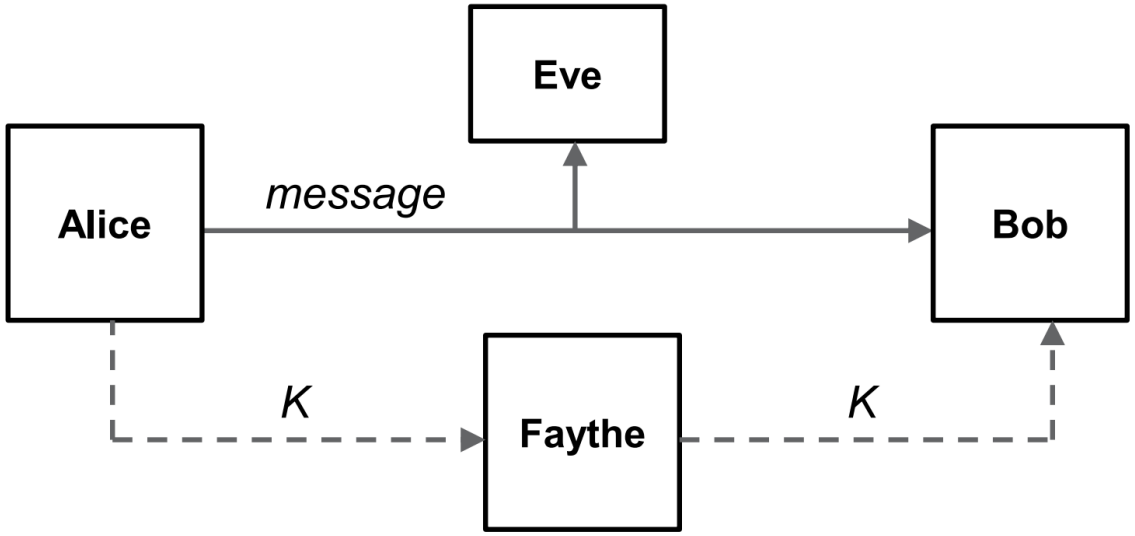
adjective

1. remaining loyal and steadfast.
   "throughout his career, he remained **faithful to** the principles of Classical art"

Similar: loyal | constant | true | devoted | true-blue | truehearted

Cryptography Application One

- **Symmetric Encryption: securely send a *message* from Alice to Bob, who know a shared secret**



Eve, the "eavesdropper", will see *message* but shouldn't be able to read it. Faythe, the "faithful", previously delivered a secret key, *K*, from Alice to Bob

Additional Cryptographic Objectives

- **Other goals include**
  - Allow Bob to verify that a contact claiming to be Alice truly is Alice...and vice versa
    - ✓ Allow Alice to establish a secure connection to Bob before Bob knows who Alice is
    - ✓ Allow Bob to securely verify that Alice knows a previously shared secret (e.g. login credentials)
  - Allow anyone to authenticate that a message claiming to originate from Alice truly came from Alice (e.g. a signed contract, a press release)
  - Detect subtle message corruption, either accidental or intentional (e.g. in downloaded software)

Solution Components for Symmetric Encryption

- **A *plaintext* space, *P***
  - These are messages to be sent
  - Example: letters in the alphabet
- **A *ciphertext* space, *C***
  - These are symbols that are actually sent
  - Example: letters in the alphabet, numbers, bits, etc.
- **A *keyspace*, *K***
  - This is the set of possible shared secrets
  - Example: an integer

Symmetric encryption: same key is known by both Alice and Bob

Solution Components for Symmetric Encryption

- **An *encryption function* space, *E***
  - These functions map plaintext to ciphertext
  - There is a function for each key, *k*
  - We denote the function $y = e_k(x) = e(k, x)$
  - Example: next up
- **A *decryption function* space, *D***
  - These functions map ciphertext to plaintext
  - There is a function for each key, *k*
  - We denote the function $x = d_k(y)$
  - Example: next up

Modular Arithmetic

- **Definition of *a* mod *n***
  - Let $a,n \in \mathbb{Z}$ and $n > 0$. We define *a* mod *n* to be the remainder of *a* divided by *n* (Note: remainders are always positive)
    - ✓ Example: 31 mod 5 = 1
      - – Reason: 31 = 6*5 + 1 so "1" is the remainder of 31 divided by 5
    - ✓ Example: -12 mod 5 = 3
      - – Reason: -12 = (-3)*5 + 3
- **In the general case we can write**
  - $a = kn + r$ and $a \bmod n = r$
    - ✓ *k* may be negative, but *r* is always positive

Example Cryptosystem: Shift Cipher

- **Let $P = C = K = \mathbb{Z}_{26}$**
  - $\mathbb{Z}_{26} = \{0, 1, 2, ..., 25\}$
  - Define $y = e_k(x) = (x + k) \bmod 26$
  - Define $d_k(y) = (y - k) \bmod 26$
- **We see that P maps naturally to the alphabet**
  - $A \rightarrow 0$
  - $B \rightarrow 1$
  - etc.

Example Cryptosystem: Shift Cipher

- **Let *k = 3* (this special case is called the *Caesar cipher*)**
  - $e_3(0) = 3$ (in other words $e_3(A) = D$)
  - ...
  - $e_3(25) = 2$ (in other words $e_3(Z) = C$)
- **The message "MEET AT MIDNITE" (ignoring spaces) encrypts to**
  - PHHWDWPLGQLWH

Shift Cipher Insecure

- **The shift cipher can be easily broken by an exhaustive key search**
  - There are only 26 different keys, *k*
  - Try them all and see when you can read the message

Course Logistics

Class Tools

- **Slack**
  - Use Slack to communicate with me about all class related issues
    - ✓ can directly message me one-on-one
  - There will be public channels
    - ✓ items for discussion may be posted on Slack such as articles
    - ✓ please reply “in thread”
- **Canvas**
  - Posting of class announcements
  - Posting of various course materials such as lecture notes
  - Submission of assignments

Class Tools

- **Email**
  - Only use email to contact me about things unrelated to this class
- **Zoom**
  - Please leave your video on
    - ✓human feedback and social interaction are important
  - Remotely taught class sessions will be recorded and the link will be shared via Canvas and/or Slack announcements

Grading Philosophy

Grading

The quality of your code, reports, and other work are important parts of your grade. Your final grade will be based on the following as manifested by work:

- **A** – Exceptional mastery of the course material and concepts. Clear, well organized, and essentially error-free work that goes the extra mile. "A" work is more than just complete and adequate, it is impressive.
- **B** – Adequate understanding of nearly all the concepts covered in the course. Well organized and proficient work. "B" work is both complete and respectable.
- **C** – Reasonable effort was put into the class, however, substantial gaps in conceptual understanding exist and/or the work is poorly organized, perfunctory, or contains significant errors.
- **D** – Little effort was put into the class, and little was learned.

Perfunctory (adj.): Carried out with minimum effort or reflection. Performed merely as a routine duty. Hasty and superficial. Lacking interest, care, or enthusiasm. Indifferent or apathetic

Grading Cutoffs

Grade	Percentage
A	≥ 93
A-	≥ 90
B+	≥ 87
B	≥ 83
B-	≥ 80
C+	≥ 77
C	≥ 73
C-	≥ 70
D+	≥ 67
D	≥ 63
D-	≥ 60
R (F)	< 60

Safety Valves

- **I will create two assignments on Canvas for every lab. One is considered late and will be 24 hours after the other**
  - You may submit to either assignment
  - If you submit to the late assignment, your grade will be **reduced**
    - ✓5 points the first time
    - ✓15 points the second time
    - ✓30 points the third time
    - ✓No credit after that!

It is very important that you do all the labs!

Exceptions

- **I will make exceptions only for *extreme* compassionate reasons**
  - Family tragedy
  - Documented major illness
- **These examples are compassion arousing, but NOT sufficient to justify an exception**
  - Studying for a test in another class
  - Working on another class's homework
  - Preparing for a visiting dignitary or CMU Africa event
  - Colds, headaches, upset stomach, and the like
    - ✓ Seeing a doctor, having a doctor's note specifying you need "rest", getting a prescription for an anti-biotic, etc. isn't enough
  - Providing support to a struggling friend
  - etc.

Why are you here?

- **Purpose of University: create and disseminate knowledge**
  - You are (mostly) here to Learn
    - ✓knowledge creation via research is (mostly) a Ph.D activity
- **Antifragility**
  - Don’t cheat yourself

Independent Work

- **It is, of course, OK to talk with your fellow students and to learn from each other**
- **However, the work you turn in must be your own and NOT a copy!**
  - If work is copied, I may, at my discretion
    - ✓ Split the points between the individuals
    - ✓ Give both parties zero points
    - ✓ Take more serious disciplinary action

Independent Work cont.

- **Students are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments. Any assistance, though, must be limited to discussion of the problem and sketching general approaches to a solution. Each student must write out his or her own solutions. Consulting another student's solution is prohibited, and submitted solutions may not be copied from any source.**

- **Copying from someone else's solution (including previous years' solutions), lab write-ups, or exams—or allowing another student to copy your work—is cheating. Any form of collaboration is strictly prohibited on exams and is cheating. If you have any question about whether some activity would constitute cheating, ask first.**

- **It is an academic integrity violation to give unauthorized assistance even if the giver does not benefit and even if no specific receiver is caught or punished. There can be substantial peer pressure to help each other out. If you are feeling that pressure, go to any of the faculty or staff to talk about it.**

Course Organization

- **This course is a hybrid lab / lecture**
  - Classes may have both a lecture time and lab time. The lab time will sometimes just be coding. Other times we'll have an outside speaker
  - Components of grade
    - ✓ Labs (80%)
    - ✓ Miscellaneous (20%)
      - – Attendance
      - – Quizzes
      - – Professionalism
      - – Etc.

Topics we will cover

- **Symmetric encryption**
  - Basic symmetric ciphers
    - ✓Permutation, substitution
  - AES (a standard)
    - ✓iterated ciphers
    - ✓substitution / permutation networks
  - Block ciphers vs stream ciphers and various operating modes
- **Perfect secrecy**
  - Discrete probability review
  - One-time pad

Topics we will cover

- **Hash functions**
  - Security
    - ✓Adversary's goals
    - ✓Las Vegas algorithms
    - ✓Random oracle model
  - SHA 256
- **Message Authentication codes**
  - HMAC
  - CBC-MAC
  - Security
    - ✓forgery
  - Authenticated encryption (CCM)

Topics we will cover

- **Public Key Cryptography**
  - RSA
  - Public key infrastructure
- **Digital Signatures**
  - A clever application of public key cryptography
- **Applications**
  - Secure storage of passwords
  - TLS and HTTPS
  - Secure logon procedures
- **Blockchain**
  - Basic overview

Topics we will cover

- **Quantum crytpography**
  - Algorithm complexity
  - New standards

Let's get started!

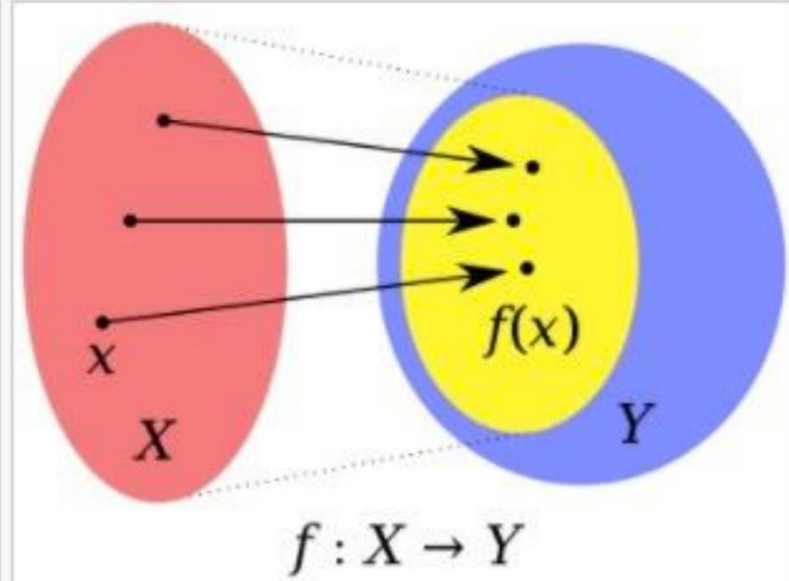
Function Terminology

- **A function has three elements**
  - Domain
    - ✓ The set of all possible inputs to the function
    - ✓ The function assigns one and only one value to each element of the domain
  - Image (Range)
    - ✓ The set obtained by letting *x* range over all elements in the domain, and looking at which *f(x)* values occur
  - Codomain
    - ✓ A set in which all the function outputs fall
    - ✓ The image is a subset of the codomain

Function Terminology

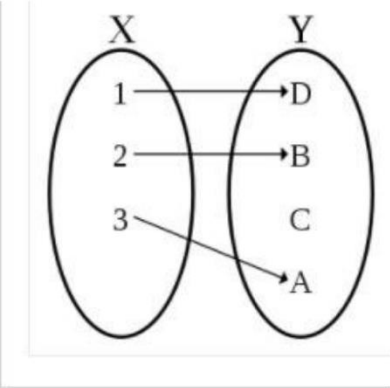
- **The term *range* is occassionally used to refer to the codomain, but normally people mean the image**

Example: $y = x^2$
Domain: $\mathbb{R}$
Image: $\{y \in \mathbb{R} : y \geq 0\}$
Codomain: $\mathbb{R}$



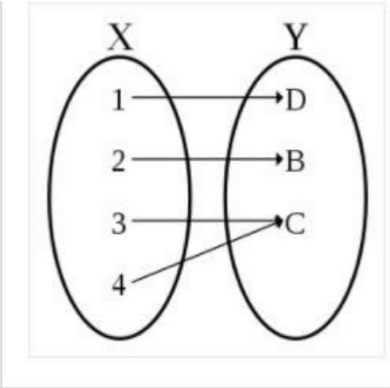
A function f from X to Y. The blue oval Y is the codomain of f. The yellow oval inside Y is the image of f.

Function Types



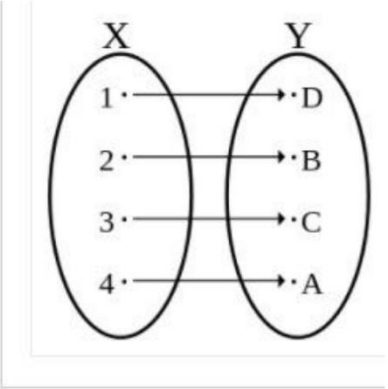
An injective non-surjective function (injection, not a bijection)

one-to-one
(injective)



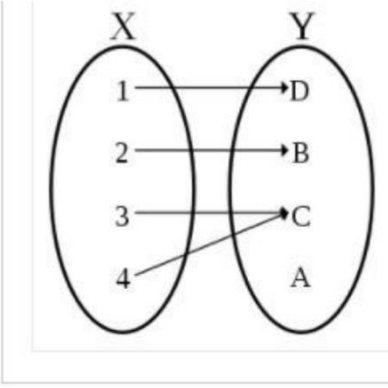
A non-injective surjective function (surjection, not a bijection)

onto
(surjective)



An injective surjective function (bijection)

one-to-one
and
onto
(bijective)



A non-injective non-surjective function (also not a bijection)

neither
one-to-one
nor
onto

Cryptosystem Definition (Symmetric)

- **A cryptosystem is a five-tuple {P, C, K, E, D} satisfying the following conditions**
  - P is a finite set of *plaintexts*
  - C is a finite set of *ciphertexts*
  - K is a finite set of *keys*
  - for each $k \in K$, there is an encryption function, $e_k( ) \in E$, and a corresponding decryption function, $d_k( ) \in D$ such that $d_k(e_k(x)) = x$ for all $x \in P$
    - ✓ $e_k( ) : P \rightarrow C$
    - ✓ $d_k( ) : C \rightarrow P$
    - ✓ $e_k( )$ is invertible, so $e_k( )$ is "one-to-one" (*injective*) but it need not be "onto" (*surjective*)

Protocol for Symmetric Encryption

- **Alice generates a key, *K***
- **Alice gives the key to Bob via a secure channel**
  - They might exchange keys in person
  - She might send it using a *believed secure* channel
    - ✓A prior face-to-face meeting
    - ✓Via a courier (“Faythe”)
    - ✓Via a different secure electronic communication link
    - ✓They might have secretly agreed, in person, on an obscure way to jointly derive the key
      - – Example: look at the NY times homepage at a specific time each day. Download the first news picture. Offset N/2 bytes into the picture where N is the image’s length. Starting at that byte in the binary image file, grab as many bits as needed to create a key

Protocol for Symmetric Encryption

- **Associated with the key is both an encrypt and a decrypt function, $e_K( )$ and $d_K( )$**
  - The same key is used for both encryption and decryption, hence the term *symmetric* encryption
- **Alice encrypts her message via $y = e_K(x)$**
- **Alice sends *y* to Bob**
- **Bob decrypts *y* using $x = d_K(y)$**

Mallory the Malicious

Classic evil actor

ma·li·cious

/məˈliSHəs/

adjective

characterized by malice; intending or intended to do harm.
"the transmission of malicious software such as computer viruses"

Similar: spiteful | malevolent | hostile | bitter | venomous | poisonous

Attacking a Cryptosystem

- **The adversary is "Mallory the malicious"**
- **We adopt *Kerckhoff's principle,* and assume Mallory knows the encryption algorithm**
  - Most real-world encryption is based on standards
- **Mallory's most ambitious goal is a *total break*—determining the encryption key**
  - If she succeeds, she can decipher every message she intercepts
  - Mallory may have more limited, but still dangerous, goals
- **The *attack model* specifies the information available to Mallory**

Attack Models

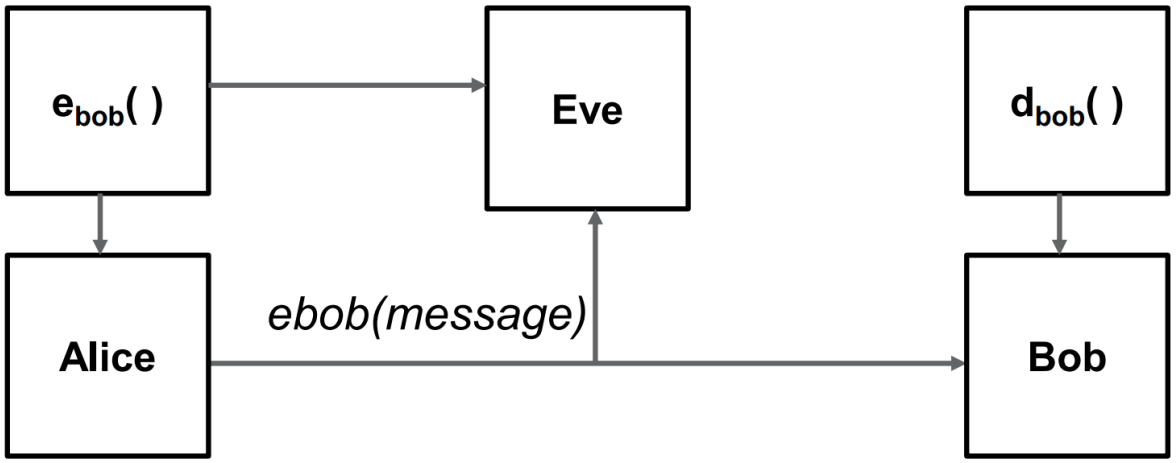
- **Ciphertext only**
  - Mallory possesses a string of ciphertext, *y*
- **Known plaintext**
  - Mallory possesses a string of plaintext, *x*, and its corresponding ciphertext, *y*
- **Chosen plaintext**
  - Mallory chooses *x* and then gets to see the corresponding ciphertext *y*
- **Chosen ciphertext**
  - Mallory chooses *y* and then gets to see the corresponding plaintext *x*

Analyzing a Cryptosystem's Security

- **Analzyzing a cryptosystem's security means**
  - Picking a goal for Mallory
  - Picking an attack model
  - Determining the probability she can reach her goal under some relevant set of assumptions

Cryptographic Application Two

- **Asymmetric Encryption: securely send a *message* from Alice to Bob, who don't know a shared secret**



Alice and Eve both know Bob's encoding rule, $e_{bob}( )$, that can be used to encode a message that only Bob can read. In other words, only Bob knows the secret decoding rule $d_{bob}( )$. Alice can securely encrypt a message for Bob that Eve can't read.

Protocol for Asymmetric Encryption (i.e. public key / private key encryption

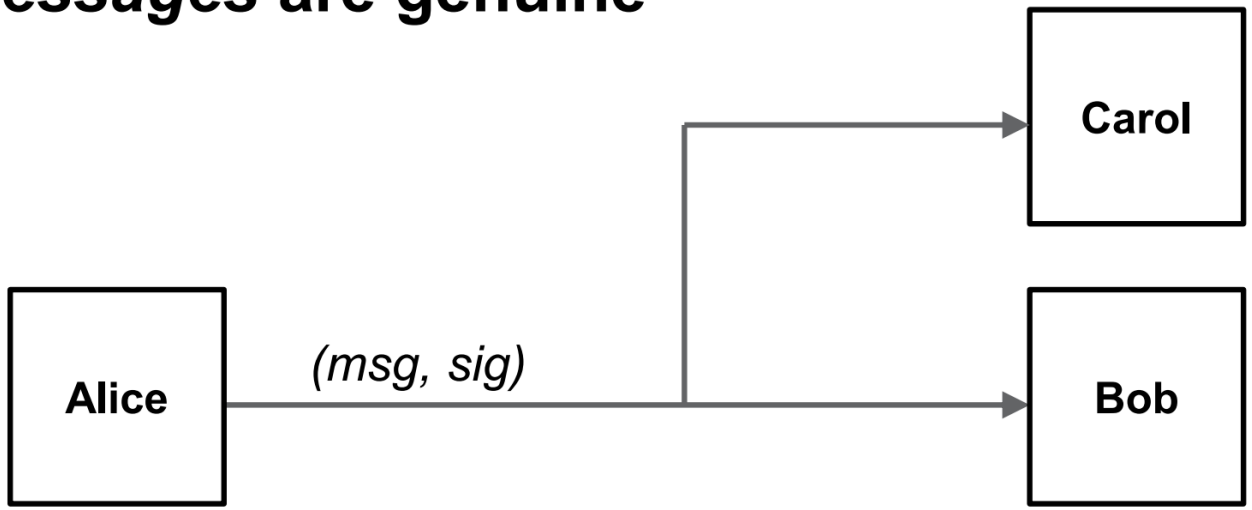
- **Bob generates a public encryption key, K_{Pub}, and a corresponging private decryption key, K_{Priv}**
  - Different keys are used for encryption and decryption, hence the term asymmetric encryption
- **Bob publishes K_{Pub} and the encryption algorithm to use (e.g. RSA)**
  - This makes clear $e_{K_{Pub}}( )$; there is a corresponding $d_{K_{Priv}}( )$
- **Alice can encrypt using $e_{K_{Pub}}( )$, but only Bob knows $d_{K_{Priv}}( )$ since it depends on K_{Priv}**

Question: why not use asymmetric encryption for all applications?

Answer: it is significantly more computationally complex than symmetric encryption

Cryptograph Application Three

- **<u>Digital Signing</u>: securely sign *messages*, without needing shared secrets, so that anyone can verify the *messages* are genuine**



Alice originates and "signs" *messages* such that Bob, Carol, and anyone else can verify they originated from her. Adversaries are unable to forge her signature on meaningful messages they create

Definition of a Group

- **Let $\mathbf{G}$ be a set of objects. Let "$\cdot$" be a binary operation between $\mathbf{G}$'s members. $\mathbf{G}$ is a *group* if it satisfies**
  - Closure
    - ✓ $\forall a,b \in \mathbf{G}$, $a \cdot b \in \mathbf{G}$
  - Associativity
    - ✓ $\forall a,b,c \in \mathbf{G}$, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
  - Existence of an identity
    - ✓ $\exists e \in \mathbf{G}$ s.t. $\forall a \in \mathbf{G}$, $e \cdot a = a \cdot e = a$
  - Existence of an inverse
    - ✓ $\forall a \in \mathbf{G}$, $\exists b \in \mathbf{G}$, s.t. $a \cdot b = b \cdot a = e$

If "$\cdot$" is also commutative, we say $\mathbf{G}$ is *Abelian*

Example of a Group

- **Let G = {0, 1, 2, ...,n-1} where n is an integer. Let $a \cdot b = (a + b) \bmod n \quad \forall a,b \in G$**
  - Closure: Yes, the remainder is always in the set **G**
  - Associativity: Yes (proof to follow)
  - Existence of an identity: Yes, *e=0*
  - Existence of an inverse: Yes, $a + (n - a) \bmod n = 0$ so $(n-a)$ is the inverse of *a*
    - ✓ Note: 0 is its own inverse (0 + 0 = *e*)

For more examples of groups see, e.g., http://www.cwladis.com/math100/Lecture7Groups.htm

Theorem 1.1

- **Theorem**
  (*a* + *b*) mod *n* = (*a* mod *n* + *b* mod *n*) mod *n*
- **proof**

$$
\begin{aligned}
a &= k_a n + r_a \\
b &= k_b n + r_b \\
a + b &= (k_a + k_b)n + (r_a + r_b) \\
(a + b) \bmod n &= (r_a + r_b) \bmod n \\
&= (a \bmod n + b \bmod n) \bmod n
\end{aligned}
$$

Theorem 1.2 (Associativity)

- **Theorem Let $a \cdot b = (a + b) \bmod n$. Then $(a \cdot b) \cdot c = a \cdot (b \cdot c)$**
- **proof**

We must show that

$$[(a+b) \bmod n + c] \bmod n = [a + (b+c) \bmod n] \bmod n$$

Let

$a = k_a n + r_a$

$b = k_b n + r_b$

$c = k_c n + r_c$

Consider the left-hand side (LHS)

$$[(a+b) \bmod n + c] \bmod n = [(r_a + r_b) \bmod n + k_c n + r_c] \bmod n$$

$$= [(r_a + r_b) \bmod n + r_c] \bmod n$$

$$= [(r_a + r_b) \bmod n + r_c \bmod n] \bmod n$$

$$= (r_a + r_b + r_c) \bmod n \quad \text{by Thm 1.1}$$

A similar analysis shows the RHS also equals

$$(r_a + r_b + r_c) \bmod n$$

// Q.E.D

Breakout Room Exercise

- **Now define "· " this way: $a \cdot b = (a * b) \bmod n$**
  - a * b is normal integer multiplication
- **Let G = {1, 2, ...,n-1} where n is an integer**
  - If n = 6 is this set closed under "· " ?
  - If n = 7 is this set closed under "· " ?
  - What, if anything, is different about 6 and 7?
- **Use Excel or google sheets to create a table showing {1, 2, ..., 6} * {1, 2, ..., 6} mod 7 (Look at the excel mod(a,b) function)**
  - Is this a group?
    - ✓ If so, what is the inverse of 5? What is the inverse of 6?