

# Cryptography

## CS 555


### **Week 1:**

- Course Overview & What is Cryptography
- Historical Ciphers (& How to Break Them)
- Perfect Secrecy

**Readings:** Katz and Lindell Chapter 1-2 + Appendix A.3 (background)

# Topic 1: Course Overview & What is Cryptography

# crypt·tog·ra·phy


/krip'təgrəfē/ 

*noun*

noun: **cryptography**

the art of writing or solving codes.

Translate cryptography to

Choose language 

Use over time for: cryptography

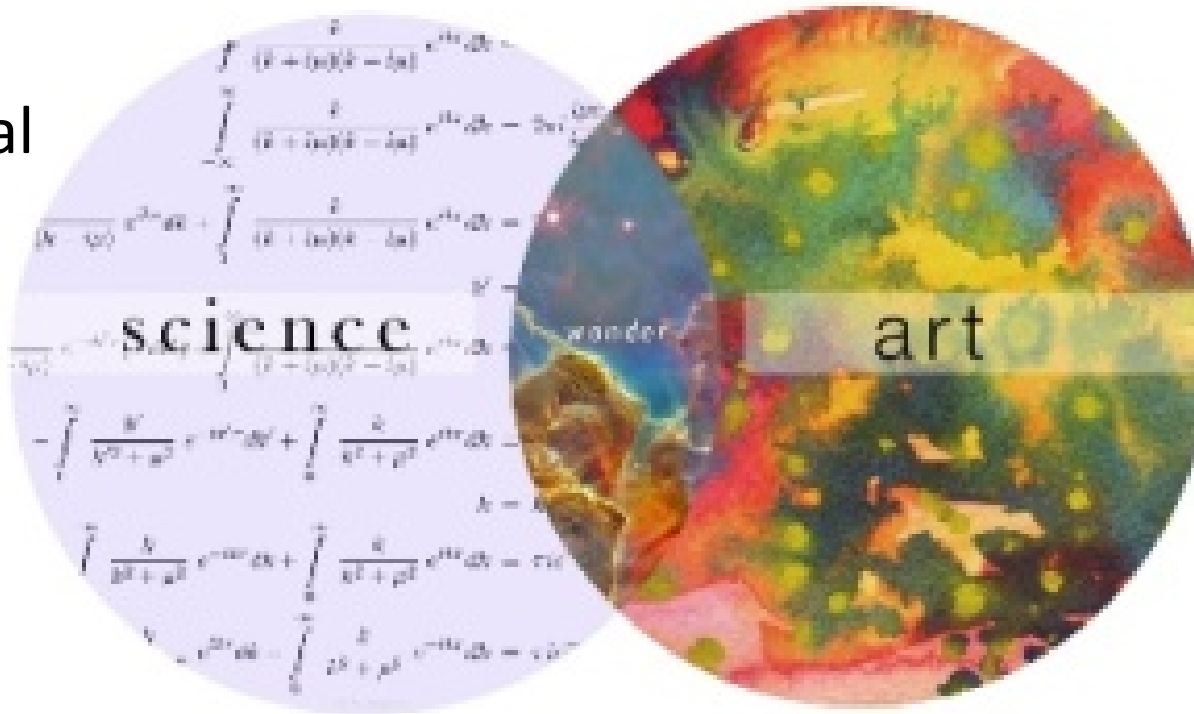


Show less

# What is Cryptography?

“the art of writing or solving codes” – Concise Oxford English Dictionary

- Precise Mathematical Security Definitions
- Specific Algorithmic Assumptions
- Formal Security Reductions/Proofs



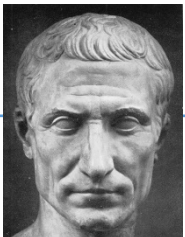
- Experience
- Intuition
- Creativity

# What is Cryptography?

“the art of writing or solving codes” – Concise Oxford English Dictionary

“The study of mathematical techniques for *securing digital information*, systems and distributed computation against adversarial attacks.”

-- Intro to Modern Cryptography



Art



Late 20<sup>th</sup> century

Science

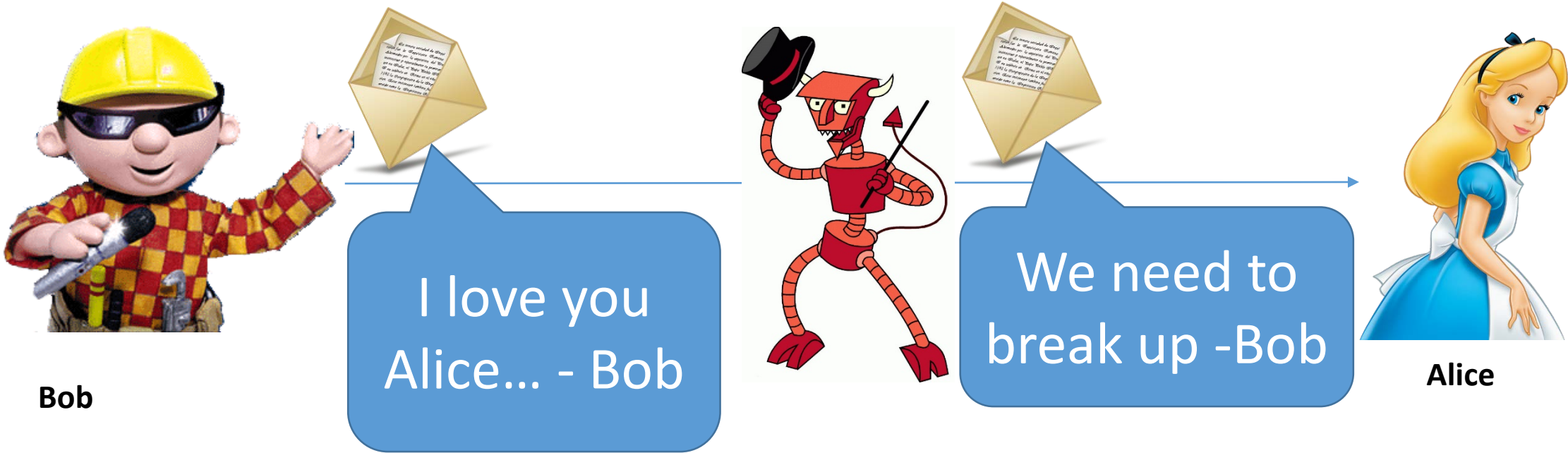
# What Does It Mean to “Secure Information”

- Confidentiality (Security/Privacy)
  - Only intended recipient can see the communication



# What Does It Mean to “Secure Information”

- Confidentiality (Security/Privacy)
  - Only intended recipient can see the communication
- Integrity (Authenticity)
  - The message was actually sent by the alleged sender



# Two Attacker Models

- **Passive Attacker (Eve)**
  - Attacker can eavesdrop
  - Protection Requires?
    - Confidentiality
  
- **Active Attacker (Mallory)**
  - Has full control over communication channel
  - Protection Requires?
    - Confidentiality & Integrity





# Steganography vs Cryptography

- Steganography
  - Goal: Hide existence of a message
    - Invisible Ink, Tattoo Underneath Hair, ...



- Assumption: Method is secret

# Steganography vs Cryptography

- Steganography
  - **Goal:** Hide existence of a message
    - Invisible Ink, Tattoo Underneath Hair, ...
  - **Assumption:** Method is secret
- Cryptography
  - **Goal:** Hide the meaning of a message
  - Depends only on secrecy of a (short) key
  - **Kerckhoff's Principle:** Cipher method should not be required to be secret.



# Symmetric Key Encryption

- What cryptography has historically been all about (Pre 1970)
- Two parties (sender and receiver) share secret key
- Sender uses key to encrypt (“scramble”) the message before transmission
- Receiver uses the key to decrypt (“unscramble”) and recover the original message

# Encryption: Basic Terminology

- Plaintext
  - The original message  $m$
- Plaintext Space (Message Space)
  - The set  $\mathcal{M}$  of all possible plaintext messages
  - Example 1:  $\mathcal{M} = \{ 'attack', 'retreat', 'hold\ current\ position' \}$
  - Example 2:  $\mathcal{M} = \{0,1\}^n$  --- all  $n$ -bit messages
- Ciphertext  $c \in \mathcal{C}$ 
  - An encrypted (“scrambled”) message  $c \in \mathcal{C}$  (ciphertext space)
- Key/Keyspace  $k \in \mathcal{K}$

# Private Key Encryption Syntax

- Message Space:  $\mathcal{M}$
- Key Space:  $\mathcal{K}$
- Three Algorithms  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ 
  - $\text{Gen}(R)$  (Key-generation algorithm)
    - **Input:** Random Bits  $R$
    - **Output:** Secret key  $k \in \mathcal{K}$
  - $\text{Enc}_k(m)$  (Encryption algorithm)
    - **Input:** Secret key  $k \in \mathcal{K}$  and message  $m \in \mathcal{M}$
    - **Output:** ciphertext  $c$
  - $\text{Dec}_k(c)$  (Decryption algorithm)
    - **Input:** Secret key  $k \in \mathcal{K}$  and a ciphertext  $c$
    - **Output:** a plaintext message  $m \in \mathcal{M}$
- **Invariant:**  $\text{Dec}_k(\text{Enc}_k(m))=m$

Typically picks  $k \in \mathcal{K}$   
uniformly at random

Trusted Parties (e.g., Alice and Bob)  
must run Gen in advance to obtain  
secret  $k$ .

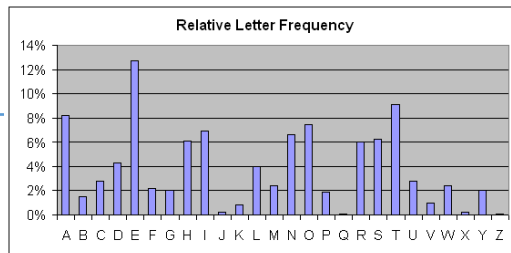
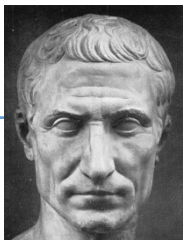
Assumption: Adversary does not get  
to see output of Gen

# Cryptography History

- 2500+ years
- Ongoing battle
  - Codemakers and codebreakers

**Formalization of Modern Crypto (1976+)**

## Caesar Shift Cipher (50 BC)

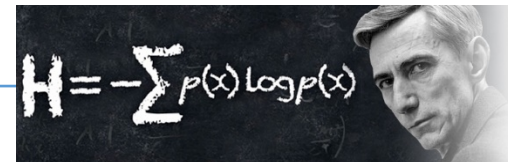


Frequency Analysis

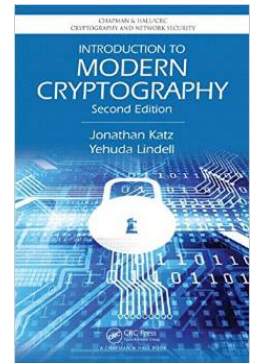
## Shannon Entropy/Perfect Secrecy (~1950)



Cipher Machines (1900s)


$$H = -\sum p(x) \log p(x)$$

1970s

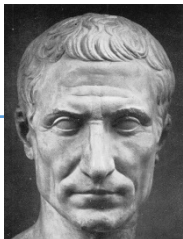


Public Key Crypto/RSA

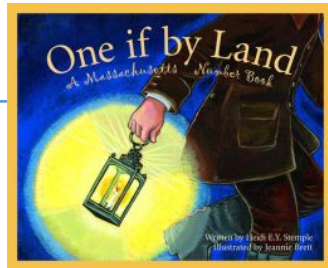
# Who Uses Cryptography

- Traditionally: Militias
- Modern Times: Everyone!

Caesar Shift Cipher (50 BC)



Revolutionary War



Modern Crypto



# Course Goals

- Understand the mathematics underlying cryptographic algorithms and protocols
- Understand the power (and limitations) of common cryptographic tools
- Understand the formal approach to security in modern cryptography



# Expected Background

- Basic Probability Theory
- Algorithms and Complexity
  - Most security proofs involve reductions
- General Mathematical Maturity
  - Quantifiers/Predicate Logic
  - Understand what is (is not) a proper definition
  - Know how to write a proof

# Recap: Lecture 1

- Syllabus
- What is cryptography
  - Science vs Art
- Authenticity vs Integrity
- Steganography vs Cryptography
  - Hiding existence vs. meaning of a message

# Review: Symmetric Key Encryption

- What cryptography has historically been all about (Pre 1970)
- Two parties (sender and receiver) share secret key
- Sender uses key to encrypt (“scramble”) the message before transmission
- Receiver uses the key to decrypt (“unscramble”) and recover the original message

# Review: Encryption: Basic Terminology

- Plaintext
  - The original message  $m$
- Plaintext Space (Message Space)
  - The set  $\mathcal{M}$  of all possible plaintext messages
  - Example 1:  $\mathcal{M} = \{ 'attack', 'retreat', 'hold\ current\ position' \}$
  - Example 2:  $\mathcal{M} = \{0,1\}^n$  - all  $n$  – bit messages
- Ciphertext  $c \in \mathcal{C}$ 
  - An encrypted (“scrambled”) message  $c \in \mathcal{C}$  (ciphertext space)
- Key/Keyspace  $k \in \mathcal{K}$

# Review: Private Key Encryption Syntax

- Message Space:  $\mathcal{M}$
- Key Space:  $\mathcal{K}$
- Three Algorithms  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ 
  - $\text{Gen}(R)$  (Key-generation algorithm)
    - **Input:** Random Bits  $R$
    - **Output:** Secret key  $k \in \mathcal{K}$
  - $\text{Enc}_k(m)$  (Encryption algorithm)
    - **Input:** Secret key  $k \in \mathcal{K}$  and message  $m \in \mathcal{M}$
    - **Output:** ciphertext  $c$
  - $\text{Dec}_k(c)$  (Decryption algorithm)
    - **Input:** Secret key  $k \in \mathcal{K}$  and a ciphertext  $c$
    - **Output:** a plaintext message  $m \in \mathcal{M}$
- **Invariant:**  $\text{Dec}_k(\text{Enc}_k(m))=m$

Typically picks  $k \in \mathcal{K}$   
uniformly at random

Trusted Parties (e.g., Alice and Bob)  
must run Gen in advance to obtain  
secret  $k$ .

Assumption: Adversary does not get  
to see output of Gen

# Example: Shift Cipher (Multiple Characters)

- Key Space:  $\mathcal{K} = \{0, 1, \dots, 25\}$
- Message Space:  $\mathcal{M} = \{a, b, c, \dots, z\}^*$

$$\text{Enc}_k(m_1 \circ \dots \circ m_n) = RS_k(m_1) \circ \dots \circ RS_k(m_n)$$
$$\text{Dec}_k(c_1 \circ \dots \circ c_n) = LS_k(c_1) \circ \dots \circ LS_k(c_n)$$

- **Note:**

$$\text{Dec}_k(\text{Enc}_k(m_1 \circ \dots \circ m_n)) = m_1 \circ \dots \circ m_n$$

since

$$LS_k(RS_k(m_i)) = m_i$$

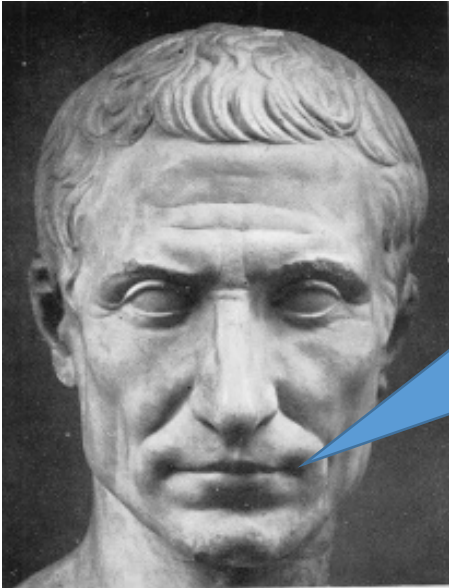
# Topic 2: Historical Ciphers (& How to Break Them)

# Shift Cipher

- Key Space:  $\mathcal{K} = \{0, 1, \dots, 25\}$
- Message Space:  $\mathcal{M} = \{a, b, c, \dots, z\}^*$
- Right Shift Operation
  - $RS_1(a) = b$
  - $RS_1(b) = c$
  - ...
  - $RS_1(z) = ?$
  - $RS_{i+1}(a) = RS_i(b)$
- $Enc_k(m_1 \circ \dots \circ m_n) = RS_k(m_1) \circ \dots \circ RS_k(m_n)$ 
  - Each letter in plaintext message  $m = m_1 \circ \dots \circ m_n$  is right shifted  $k$  times  $RS_k$
- **Question:** what is ciphertext space  $\mathcal{C}$ ?



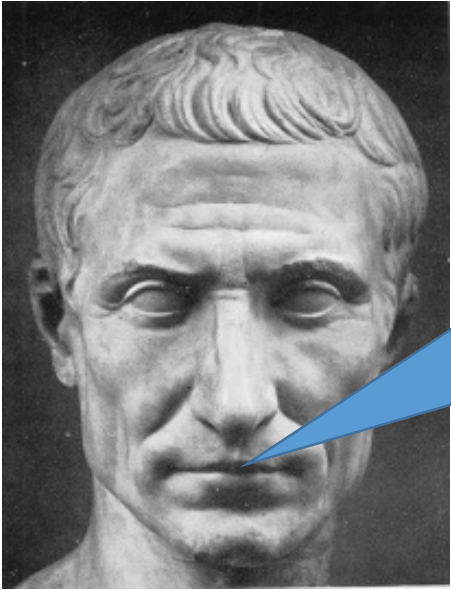
# Caesar Cipher



Three shall be the number of thy shifting and the number of thy shifting shall be three. Four shalt thou not shift, neither shift thou two, excepting that thou then proceed to three. Five is right out.....

Caesar adopted the shift cipher with secret key  $k=3$

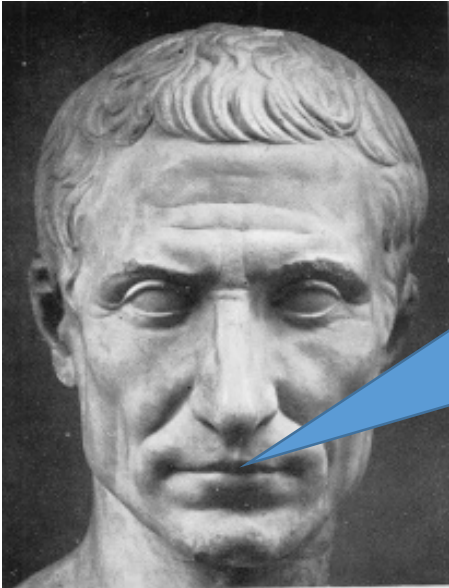
# Caesar Cipher (Example)



BEGIN THE ATTACK NOW  
→  
EHJLQWKHDWWDFNQRZ

Caesar adopted the shift cipher with secret key  $k=3$

# Caesar Cipher (Example)



BEGINTHEATTACKNOW  
→  
EHJLQWKHDWWDFNQRZ

Immediate Issue: anyone who knows method can decrypt  
(since  $k=3$  is fixed)

# Modern Application: Avoid Spoilers (ROT13)

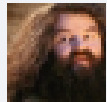


**Harry Potter**

I was shocked and horrified when Snape killed Dumbledore.

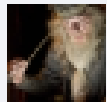
Like · Comment · 32 minutes ago · 🌐

👍 26 people like this.



**Hagrid** Me too!

31 minutes ago · Like · 👍 3



**Dumbledore** Thanks for ruining the plot, jerk!

15 minutes ago · Like · 👍 34



Write a comment ...



# Modern Application: Avoid Spoilers (ROT13)



**Harry Potter**

[ROT13 to avoid spoilers] V jnf fubpxrq naq ubeevsvrq jura Fancr xvyyrq Qhzoyrqber.

Like · Comment · 32 minutes ago ·

20 people like this.



**Dumbledore** I am dying to find out what will happen, but I will wait to decrypt until after I read the book.

15 minutes ago · Like · 23



Write a comment ...

# Shift Cipher: Brute Force Attack

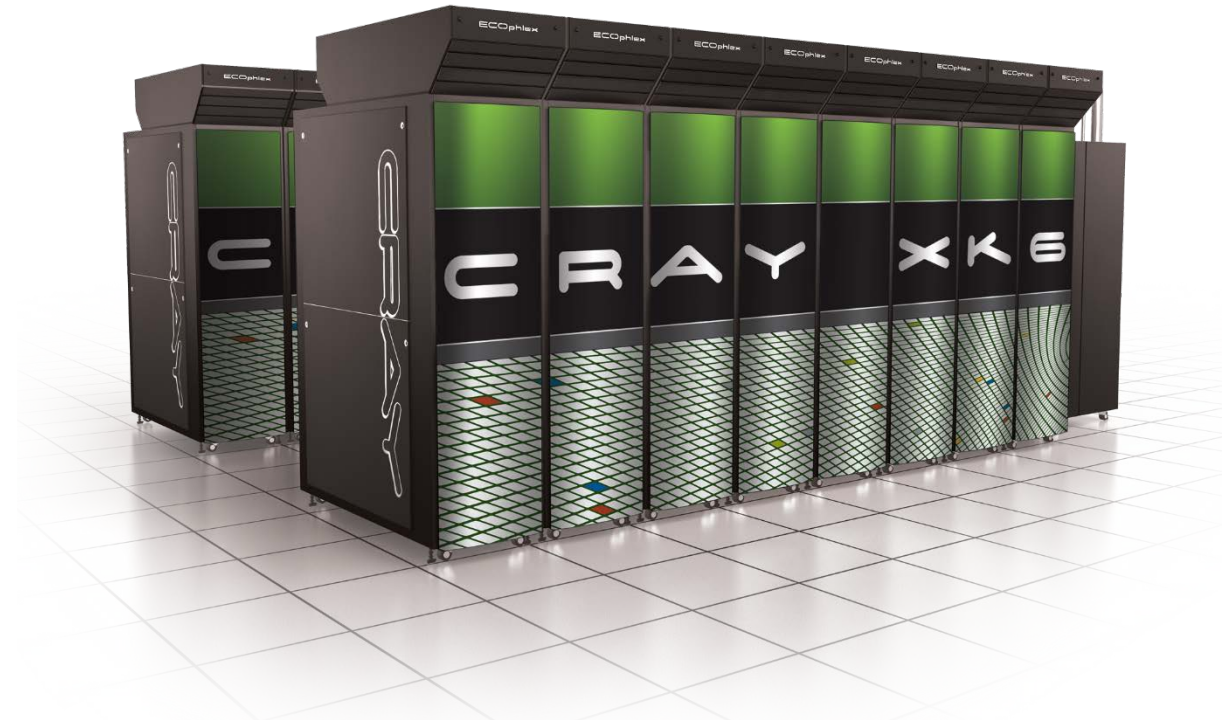
- Ciphertext: “lwyrw ztn sd ndj iwxcz xh gxvwi?”
  - $k=1 \rightarrow m =$  “mxysx auo te oek jxyda yi hywxj?”
  - $k=2 \rightarrow m=$ “nyzty bvp uf pfl kyzeb zj izxyk?”
  - $k=3 \rightarrow m=$ “ozauz cwq vg qgm lzafc ak jayzl?”
  - $k=4 \rightarrow m =$  “pabva dxr wh rhn mabgd bl kbzam?”
  - $k=5 \rightarrow m=$ “qbcwb eys xi sio nbche cm lcabn?”
  - $k=6 \rightarrow m=$ “rcdxc fzt yj tjp ocdif dn mdbco?”

# Shift Cipher: Brute Force Attack

- Ciphertext: “lwyrw ztn sd ndj iwxcz xh gxvwi?”
  - ...
  - $k=7 \rightarrow m=$ “sdeyd gau zk ukq pdejg eo necdp?”
  - $k=8 \rightarrow m=$ “tefze hbv al vlr qefkh fp ofdeq?”
  - $k=9 \rightarrow m =$  “ufgaf icw bm wms rfgli gq pgefr?”
  - $k=10 \rightarrow m=$ “vghbg jdx cn xnt sghmj hr qhfgs?”
  - $k=11 \rightarrow m=$  “which key do you think is right?”
  - $k=12 \rightarrow m=$  “xijdi lfz ep zpv uijol jt sjhiu?”

# Sufficient Key Space Principle

“Any secure encryption scheme *must* have a key space that is sufficiently large to make an exhaustive search attack infeasible.”





# Sufficient Key Space Principle

“Any secure encryption scheme *must* have a key space that is sufficiently large to make an exhaustive search attack infeasible.”

**Question 1:** How big is big enough? Complicated question....

**Question 2:** If the key space is large is the encryption scheme necessarily secure?

# Substitution Cipher

- Secret key K is permutation of the alphabet
  - Example:
    - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
    - X E U A D N B K V M R O C Q F S Y H W G L Z I J P T
- **Encryption:** apply permutation K to each letter in message
  - TELLHIMABOUTME → GDOOKVCXEFLGCD
- **Decryption:** reverse the permutation

# Substitution Cipher

- Secret key  $K$  is a permutation of the alphabet

- Example:

- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- X E U A D N B K V M R O C Q F S Y H W G L Z I J P T

- **Question:** What is the size of the keyspace  $\mathcal{K}$ ?

$$|\mathcal{K}| = 26! \approx 2^{88}$$

# Tuesday's Crypto Answers

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	9	23	21	18	11	1	17	19	8	4	5	22	7	3	14	12	6	15	25	13	10	26	24	20	16

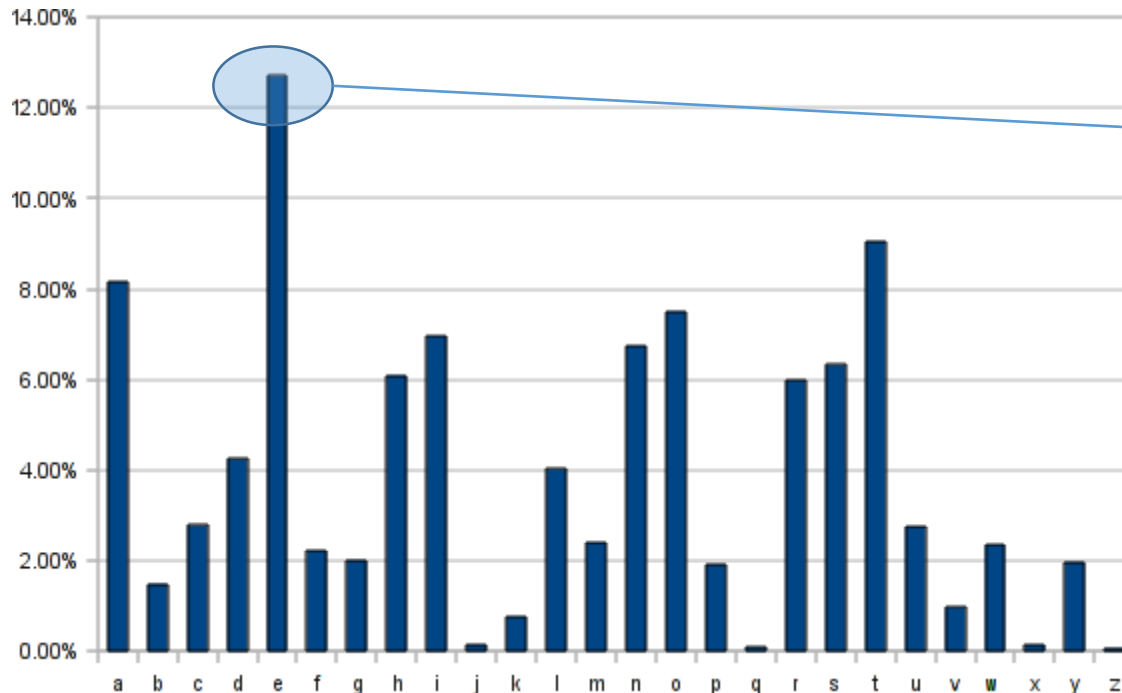
T H E      O N L Y      T I M E      Y O U      R U N      O U T  
25 17 18      3 7 5 20      25 19 22 18      20 3 13      6 13 7      3 13 25

O F      C H A N C E S      I S      W H E N      Y O U  
3 11      23 17 2 7 23 18 15      19 15      26 17 18 7      20 3 13

S T O P      T A K I N G      T H E M .  
15 25 3 14      25 2 4 19 7 1      25 17 18 22

# Frequency Analysis

- **Observation 1:** If e is mapped to d then every appearance of e in the plaintext results in the appearance of a d in the ciphertext
- **Observation 2:** Some letters occur much more frequently in English.
- **Observation 3:** Texts consisting of a few sentences tend to have a distribution close to average.



Step 1: Find letter in ciphertext that occurs with frequency > 11%. This letter is probably e...

# Vigenère Cipher

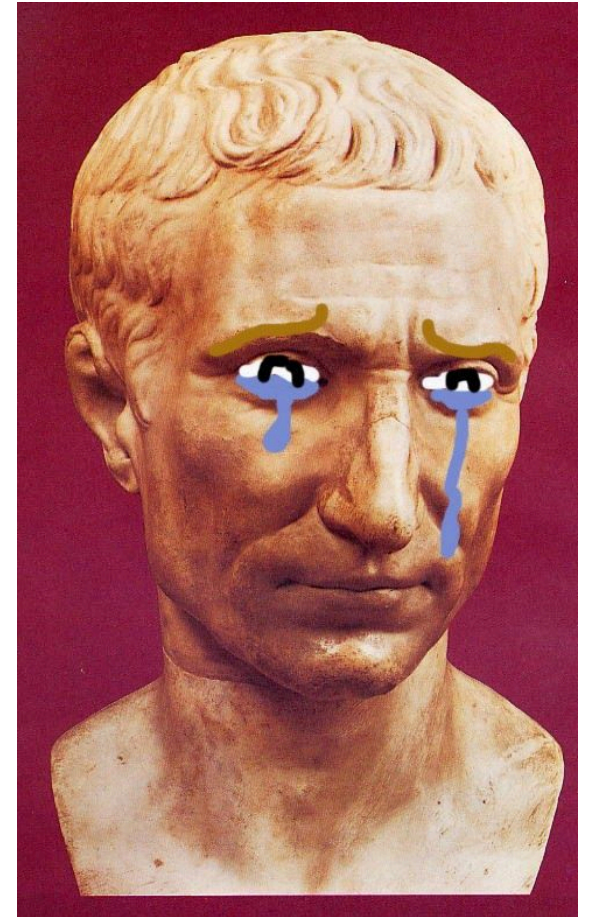
- Generalizes Shift Cipher
- $K = k_1, \dots, k_t$
- $\text{Enc}_K(m)$ 
  - Shift first letter right  $k_1$  times
  - Shift second letter right  $k_2$  times
  - ...
  - Shift  $t^{\text{th}}$  letter right  $k_t$  times
  - Shift  $t+1^{\text{st}}$  letter right  $k_1$  times
  - ...
- **Question:** Size of key-space?
- Answer:  $26^t$  (brute force may not be useful)

# Vigenère Cipher

- Still vulnerable to frequency analysis
- Good guess: Select  $K=k_1, \dots, k_t$  to maximize number of e's in resulting ciphertext
  - See Katz and Lindell 1.3 for even more sophisticated heuristics.
- Attack works when the initial message  $m$  is sufficiently long
- Vigenère is “perfectly secret” if the message  $m$  is at most  $t$  letters long.

# Conclusions

- Designing secure ciphers is hard
- Vigenère remained “unbroken” for a long time
- Complex schemes are not secure
- All historical ciphers have fallen





# Topic 3: Perfect Secrecy + One-Time-Pads

# Principles of Modern Cryptography

- Need formal definitions of “security”

*If you don't understand what you want to achieve, how can you possibly know when (or if) you have achieved it?*

- Attempt 1: Impossible for attacker to recover secret key  $K$

- $\text{Enc}_k(m) = m$

- Attempt 2: Impossible for attacker to recover entire plaintext from ciphertext?

- Ok to decrypt 90% of message?

- Attempt 3: Impossible for attacker to figure out any particular character of the plaintext from the ciphertext?

- [Too Weak] Does employee make more than \$100,000 per year?
  - [Too Strong] Lucky guess? Prior Information? (e.g., letters always begin “Dear ....”)

# Principles of Modern Cryptography

- Need formal definitions of “security”

*If you don't understand what you want to achieve, how can you possibly know when (or if) you have achieved it?*

- Final Attempt: Regardless of information an attacker *already* has, a ciphertext should leak no *additional information* about the underlying plaintext.
  - This is the “*right*” approach
  - Still need to *formalize* mathematically
- Security definition includes goal and threat-model

# Principles of Modern Cryptography

- Proofs of Security are critical
  - Iron-clad guarantee that attacker will not succeed (relative to definition/assumptions)
- Experience: intuition is often misleading in cryptography
  - An “intuitively secure” scheme may actually be badly broken.
- Before deploying in the real world
  - Consider definition/assumptions in security definition
  - Does the threat model capture the attackers true abilities?

# Perfect Secrecy Intuition

- Regardless of information an attacker *already* has, a ciphertext should leak no *additional information* about the underlying plaintext.
- We will formalize this intuition
  - And show how to achieve it

# Private Key Encryption Syntax

- Message Space:  $\mathcal{M}$
- Key Space:  $\mathcal{K}$
- Three Algorithms  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ 
  - $\text{Gen}(R)$  (Key-generation algorithm)
    - **Input:** Random Bits  $R$
    - **Output:** Secret key  $k \in \mathcal{K}$ .
  - $\text{Enc}_k(m)$  (Encryption algorithm)
    - **Input:** Secret key  $k \in \mathcal{K}$  and message  $m \in \mathcal{M}$
    - **Output:** ciphertext  $c$
  - $\text{Dec}_k(c)$  (Decryption algorithm)
    - **Input:** Secret key  $k \in \mathcal{K}$  and a ciphertext  $c$
    - **Output:** a plaintext message  $m \in \mathcal{M}$
- **Invariant:**  $\text{Dec}_k(\text{Enc}_k(m))=m$

Typically picks  $k \in \mathcal{K}$   
uniformly at random

Trusted Parties (e.g., Alice and Bob)  
must run Gen in advance to obtain  
secret  $k$ .

Assumption: Adversary does not get  
to see output of Gen

# An Example

- Enemy knows that Caesar likes to fight in the rain and it is raining today

$$\Pr[m = \textit{wait}] = 0.3$$
$$\Pr[m = \textit{attack}] = 0.7$$

- Suppose that Caesar sends  $c = \text{Enc}_K(m)$  to generals and that the attacker calculates

$$\Pr[m = \textit{wait} | c = \text{Enc}_K(m)] = 0.2$$
$$\Pr[m = \textit{attack} | c = \text{Enc}_K(m)] = 0.8$$

- Did the attacker learn anything useful?

# Perfect Secrecy

**Definition 1:** An encryption scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  with message space  $\mathcal{M}$  is perfectly secret if for *every* probability distribution  $\mathcal{D}$  over  $\mathcal{M}$ , every message  $m \in \mathcal{M}$  and every ciphertext  $c \in \mathcal{C}$  for which  $\Pr[C =$



# Proof (one direction):

Suppose first that  $(\text{Gen}, \text{Enc}, \text{Dec})$  does not satisfy definition 2. Then there exists  $m, m' \in \mathcal{M}$  and  $c \in \mathcal{C}$  such that

$$\Pr[\text{Enc}_K(m) = c] \neq \Pr[\text{Enc}_K(m') = c] \quad (1).$$

We will now prove that definition 1 does not hold. Define  $\mathcal{D}$  such that

$$\Pr[M=m] = \Pr[M=m'] = \frac{1}{2}$$

Assume for the sake of contradiction that Definition 1 were satisfied then we would have

$$\Pr[M = m | C = c] = \Pr[M = m] = \frac{1}{2}, \quad \text{and}$$
$$\Pr[M = m' | C = c] = \Pr[M = m'] = \frac{1}{2}$$

which implies

$$\Pr[M = m | C = c] = \Pr[M = m' | C = c] \quad (*)$$

# Proof (one direction):

Suppose first that  $(\text{Gen}, \text{Enc}, \text{Dec})$  does not satisfy definition 2. Then there exists  $m, m' \in \mathcal{M}$  and  $c \in \mathcal{C}$  such that

$$\Pr[\text{Enc}_K(m) = c] \neq \Pr[\text{Enc}_K(m') = c] \quad (1).$$

Define  $\mathcal{D}$  such that  $\Pr[M=m]=\Pr[M=m']=\frac{1}{2}$

## Bayes Rule (1)

$$\begin{aligned} \Pr[M = m | \text{Enc}_K(M) = c] &= \frac{\Pr[C = c | M = m] \Pr[M=m]}{\Pr[C=c]} \\ &= \frac{1}{2} \frac{\Pr[\text{Enc}_K(m) = c]}{\Pr[C=c]} \end{aligned} \quad (2)$$

# Proof (one direction):

Suppose first that  $(\text{Gen}, \text{Enc}, \text{Dec})$  does not satisfy definition 2. Then there exists  $m, m' \in \mathcal{M}$  and  $c \in \mathcal{C}$  such that

$$\Pr[\text{Enc}_K(m) = c] \neq \Pr[\text{Enc}_K(m') = c] \quad (1).$$

Define  $\mathcal{D}$  such that  $\Pr[M=m] = \Pr[M=m'] = \frac{1}{2}$

## Bayes Rule (2)

$$\begin{aligned} \Pr[M = m' | \text{Enc}_K(M) = c] &= \frac{\Pr[C = c | M = m'] \Pr[M=m']}{\Pr[C=c]} \\ &= \frac{1}{2} \frac{\Pr[\text{Enc}_K(m') = c]}{\Pr[C=c]} \end{aligned} \quad (3)$$

# Proof (one direction):

Suppose first that  $(\text{Gen}, \text{Enc}, \text{Dec})$  does not satisfy definition 2. Then there exists  $m, m' \in \mathcal{M}$  and  $c \in \mathcal{C}$  such that

$$\Pr[\text{Enc}_K(m) = c] \neq \Pr[\text{Enc}_K(m') = c] \quad (1).$$

Define  $\mathcal{D}$  such that  $\Pr[M=m]=\Pr[M=m']=\frac{1}{2}$

**Combining equations (2) and (3), Bayes Rule implies that**

$$\begin{aligned} \Pr[M = m' | \text{Enc}_K(M) = c] &= \frac{1}{2} \frac{\Pr[\text{Enc}_K(m') = c]}{\Pr[C=c]} \\ &\neq \frac{1}{2} \frac{\Pr[\text{Enc}_K(m) = c]}{\Pr[C=c]} = \Pr[M = m | \text{Enc}_K(M) = c] \quad (**) \end{aligned}$$

Proof (one direction):

**Thus, Bayes Rule implies that**

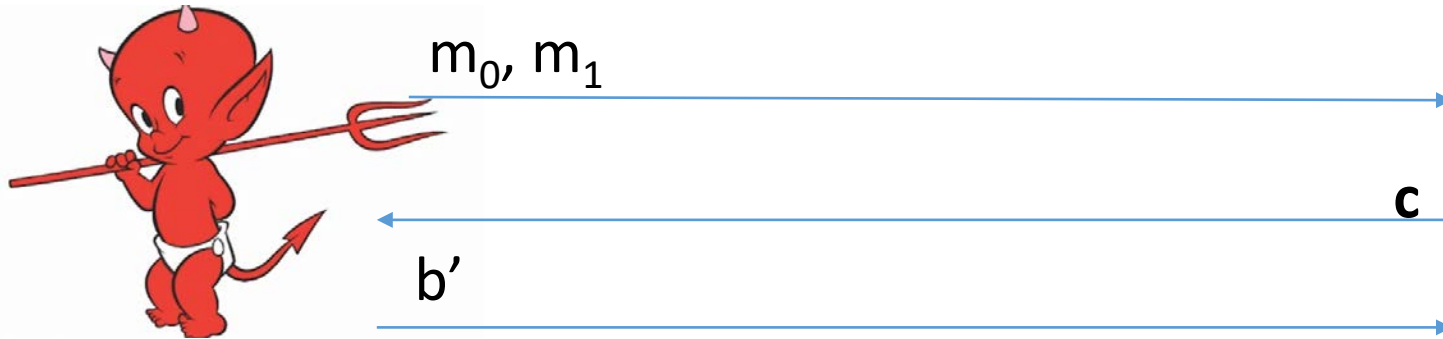
$$\begin{aligned} \Pr[M = m' | \text{Enc}_K(M) = c] &= \frac{1 \Pr[\text{Enc}_K(m') = c]}{2 \Pr[C=c]} \\ &\neq \frac{1 \Pr[\text{Enc}_K(m) = c]}{2 \Pr[C=c]} = \Pr[M = m | \text{Enc}_K(M) = c] \quad (**) \end{aligned}$$

We previously showed that definition 2 implies

$$\Pr[M = m | C = c] = \Pr[M = m' | C = c] \quad (*)$$

Contradiction!

# Another Equivalent Definition (Game)



Random bit  $b$   
 $K \leftarrow \text{Gen}(\cdot)$   
 $c = \text{Enc}_K(m_b)$



$$\Pr \left[ \text{Devil guesses } b' = b \right] = \frac{1}{2}$$

# Another Equivalent Definition (Game)



*Formally, let  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  denote the encryption scheme, and let  $A$  denote an eavesdropping attacker.*

*Call the game the adversarial indistinguishability experiment and define a random variable  $\text{PrivK}_{A,\Pi}^{\text{eav}}$  as follows*

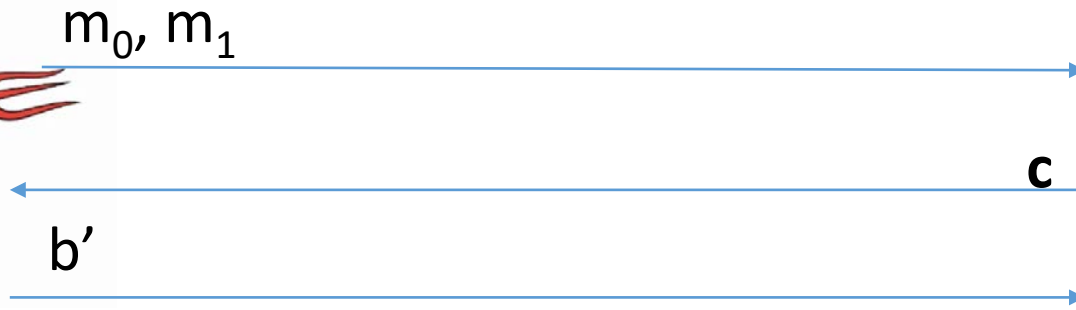
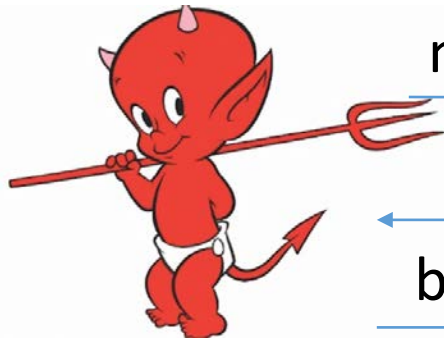
$$\text{PrivK}_{A,\Pi}^{\text{eav}} = \begin{cases} 1 & \text{if } b = b' \text{ (attacker is correct)} \\ 0 & \text{otherwise (attacker is not correct)} \end{cases}$$

*$\Pi$  has indistinguishable encryptions in the presence of an eavesdropper if for all eavesdropping adversaries  $A$  we have*

$$\Pr[\text{PrivK}_{A,\Pi}^{\text{eav}} = 1] = \frac{1}{2}$$

om bit b  
en(.)  
c<sub>K</sub>(m<sub>b</sub>)

# Another Equivalent Definition (Game)



Random bit  $b$   
 $K \leftarrow \text{Gen}(\cdot)$   
 $c = \text{Enc}_K(m_b)$

Suppose we have  $m, m', c'$  s.t.  $\Pr[\text{Enc}_K(m) = c'] > \Pr[\text{Enc}_K(m') = c']$  then the adversary can win the game w.p  $> \frac{1}{2}$ . How?

What else do we need to establish to prove that the definitions are equivalent?



# One Time Pad [Vernam 1917]

$$\text{Enc}_K(m) = K \oplus m$$

$$\text{Dec}_K(c) = K \oplus c$$

$$\text{Example} = 1011 \oplus 0011 = ???$$

**Theorem:** The one-time pad encryption scheme is perfectly secret

The following calculation holds for any  $c, m$

$$\Pr[\text{Enc}_K(m)=c] = \Pr[K \oplus m = c] = \Pr[K=c \oplus m] = 1/|\mathcal{K}|.$$

Thus, for any  $m, m', c$  we have

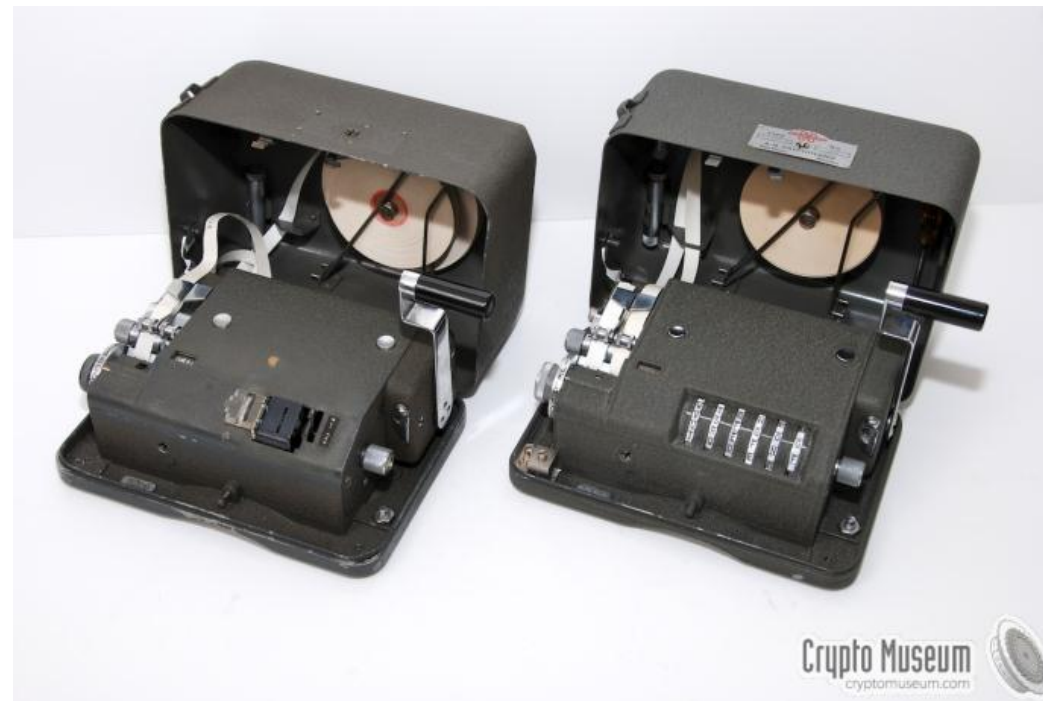
$$\Pr[\text{Enc}_K(m)=c] = 1/|\mathcal{K}| = \Pr[\text{Enc}_K(m')=c].$$

# One Time Pad [Vernam 1917]

$$\text{Enc}_K(m) = K \oplus m$$

$$\text{Dec}_K(c) = K \oplus c$$

**Example =  $1011 \oplus 0011 = ???$**



# One Time Pad



# Perfect Secrecy Limitations

**Theorem:** If  $(\text{Gen}, \text{Enc}, \text{Dec})$  is a perfectly secret encryption scheme then

$$|\mathcal{K}| \geq |\mathcal{M}|$$

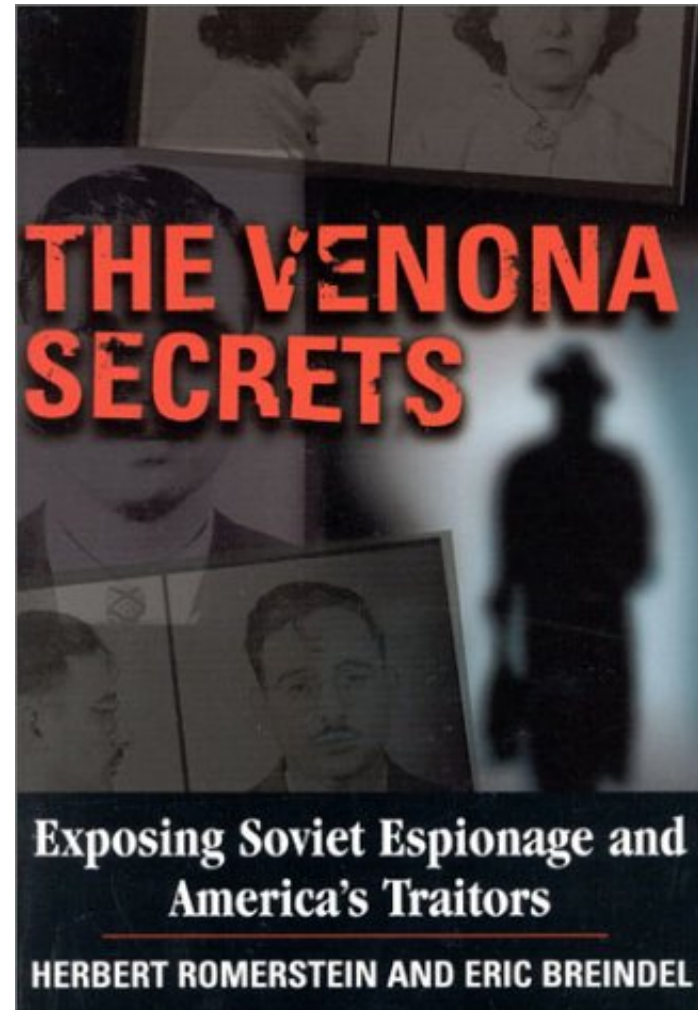
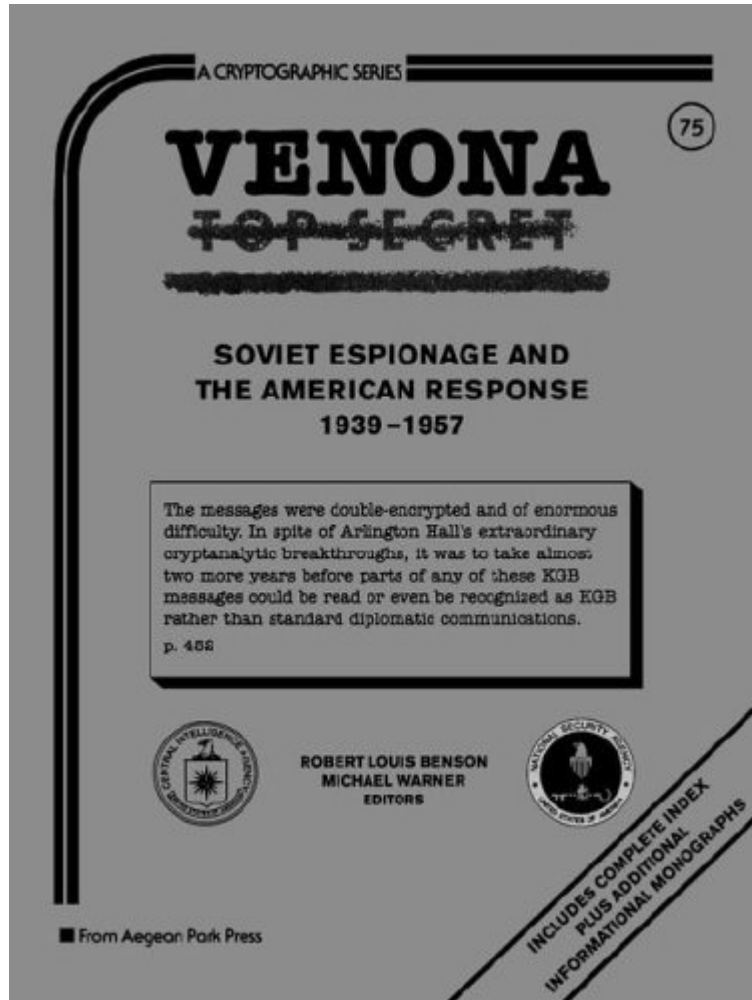
# One Time Pad Limitations

- The key is as long as the message
  - How to exchange long messages?
  - Need to exchange/secure lots of one-time pads!
- OTPs can only be used once
  - As the name suggests
- VENONA project (US + UK)
  - Decrypt ciphertexts sent by Soviet Union which were mistakenly encrypted with portions of the same one-time pad over several decades



$$c \oplus c' = (m \oplus k) \oplus (m' \oplus k) = m \oplus m'$$

# VENONA project



# Shannon's Theorem

**Theorem:** Let  $(\text{Gen}, \text{Enc}, \text{Dec})$  be an encryption scheme with  $|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}|$ . Then the scheme is perfectly secret if and only if:

1. Every key  $k \in \mathcal{K}$  is chosen with (equal) probability  $1/|\mathcal{K}|$  by the algorithm  $\text{Gen}$ , and
2. For every  $m \in \mathcal{M}$  and every  $c \in \mathcal{C}$  there exists a unique key  $k \in \mathcal{K}$  such that  $\text{Enc}_k(m)=c$ .

# An Important Remark on Randomness

- In our analysis we have made (and will continue to make) a key assumption:
- We have access to true “randomness” to generate a secret key  $K$

Example:  $K$  = one time pad

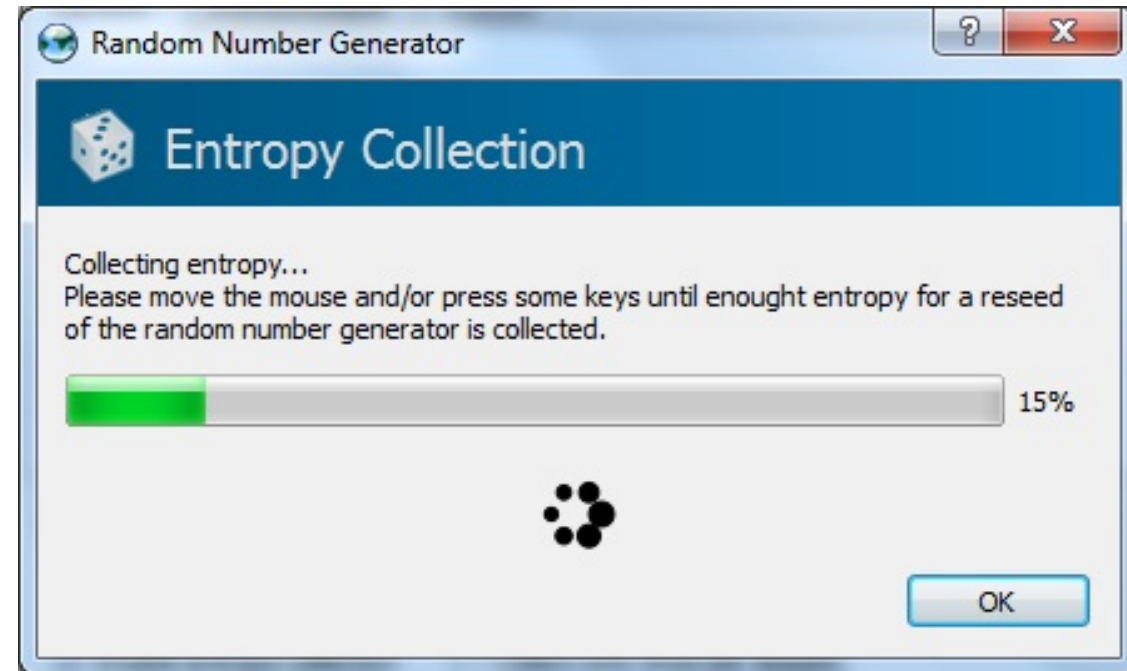
- Independent Random Bits
  - Unbiased Coin flips
  - Radioactive decay?





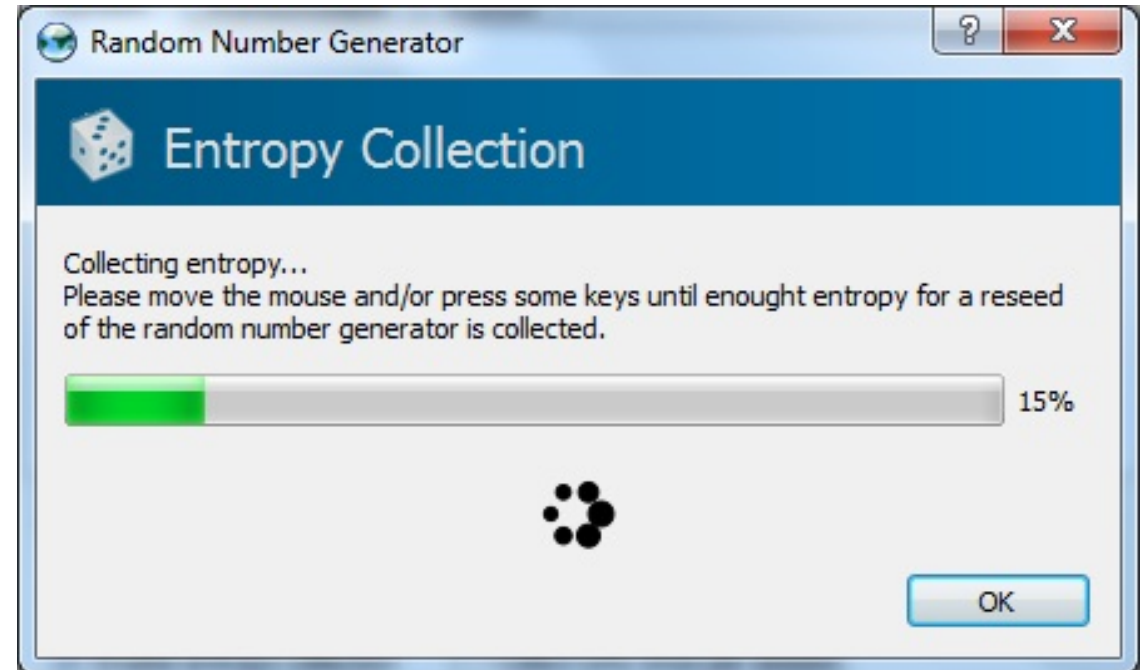
# In Practice

- Hard to flip thousands/millions of coins
- Mouse-movements/keys
  - Uniform bits?
  - Independent bits?
- Use Randomness Extractors
  - As long as input has high entropy, we can extract (almost) uniform/independent bits
  - Hot research topic in theory



# In Practice

- Hard to flip thousands/millions of coins
- Mouse-movements/keys
- Customized Randomness Chip?



# Caveat: Don't do this!

- Rand() in C stdlib.h is no good for cryptographic applications
- Source of many real world flaws



# Coming Up in Week 2...

- Computational Security
- Pseudorandomness + Stream Ciphers
- Chosen Plaintext Attacks and CPA Security

**Week 2 Reading:** Katz and Lindell 3.1-3.4