

CS306: Introduction to IT Security

Fall 2018

Lecture 2: Symmetric encryption

Instructor: **Nikos Triandopoulos**

September 4, 2018



Last week

- ◆ Course logistics
- ◆ Introduction to the field of IT security
 - ◆ in-class discussion with a real-world example
 - ◆ secure outsourced computation
 - ◆ need for integrity & privacy protections
 - ◆ coverage of basic security concepts & terms
 - ◆ the “IT-security” game
 - ◆ core security properties
 - ◆ vulnerabilities, threats, controls

Today

- ◆ Symmetric-key cryptography (I)
 - ◆ symmetric-key encryption
 - ◆ classical symmetric-key ciphers
 - ◆ definition of perfect secrecy
 - ◆ the one-time pad

2.0 Announcements

CS306: Announcements

- ◆ CS306 is now **live on Canvas**
 - ◆ all course materials will be available there
 - ◆ lecture notes to be posted shortly before the class
 - ◆ e.g., posted on Tuesday morning & possibly slightly updated at later time
 - ◆ communication via Canvas (e.g., questions, clarifications, announcements, etc.)
- ◆ TA hours & instructor's office **hours start this week**
 - ◆ TA hours: TBA (scheduled towards end of week)
 - ◆ office hours: NB321, Tue 4-6pm & by appointment
 - ◆ **note:** occasional changes due to conflicting schedules (e.g., travel or departmental needs)

CS306: Announcements (continued)

- ◆ Lab sections **start on September 6th** (this week)
 - ◆ recitation of course materials, guided work on assignments, preparation for HWs
 - ◆ typically in form of **a quiz posted on Canvas**, occasionally as whiteboard discussion
 - ◆ please **bring your laptop** as computer use will be needed
 - ◆ lab sessions are **required!**
 - ◆ attendance is kept
 - ◆ please send us advice notice for known future absences

CS306: Lectures & labs

CS306 is offered in **2 required sessions**, each offered in **multiple sections**

- ◆ lectures

- ◆ *Skip lunch* CS306-A, Tue 12:00pm - 14:30pm, Kidde 360 12/60
- ◆ *Skip dinner* CS306-B, Tue 18:15pm - 20:45pm, Babbio 104 7/60

- ◆ labs

- ◆ *Stay focused* CS306-Lx, sometime & somewhere on Thursdays

x	A	B	C	D	E	F
time	8 - 8:50	9:25 - 10:15	10:50 - 11:40	12:15 - 13:05	13:40 - 14:30	15:05 - 15:55
place	NB101	Morton 105	Kidde 360	Babbio 220	Babbio 304	Burchard 514
capacity	13/20	6/20	0/20	0/20	0/20	0/20

PLEASE contact me if you have enrolled to CS306-LA

CS306: Tentative Syllabus

Week	Date	Topics	Reading	Assignment
1	Aug 28	Introduction	Ch. 1	-
2	Sep 4	Symmetric encryption	Ch. 2 & 12	Lab 1
3	Sep 11	Symmetric-key crypto II		
4	Sep 18	Public-key crypto I		
5	Sep 25	Public-key crypto II		
6	Oct 2	Access control & authentication		
–	Oct 9	No class (Monday schedule)		
7	Oct 16	Midterm (closed books)	All materials covered	

CS306: Tentative Syllabus

(continued)

Week	Date	Topics	Reading	Assignment
8	Oct 23	Software & Web security		
9	Oct 30	Network security		
10	Nov 6	Database security		
11	Nov 13	Cloud security		
12	Nov 20	Privacy		
13	Nov 27	Economics		
14	Dec 4	Legal & ethical issues		
15	Dec 11 (or later)	Final (closed books)	All materials covered*	

CS306: Course outcomes

- ◆ **Terms**

- ◆ describe common security terms and concepts

- ◆ **Cryptography**

- ◆ state basics/fundamentals about secret and public key cryptography concepts

- ◆ **Attack & Defense**

- ◆ acquire basic understanding for attack techniques and defense mechanisms

- ◆ **Impact**

- ◆ acquire an understanding for the broader impact of security and its integral connection to other fields in computer science (such as software engineering, databases, operating systems) as well as other disciplines including STEM, economics, and law

- ◆ **Ethics**

- ◆ acquire an understanding for ethical issues in cyber-security

Questions?

2.1 Symmetric-key encryption

Recall: Confidentiality

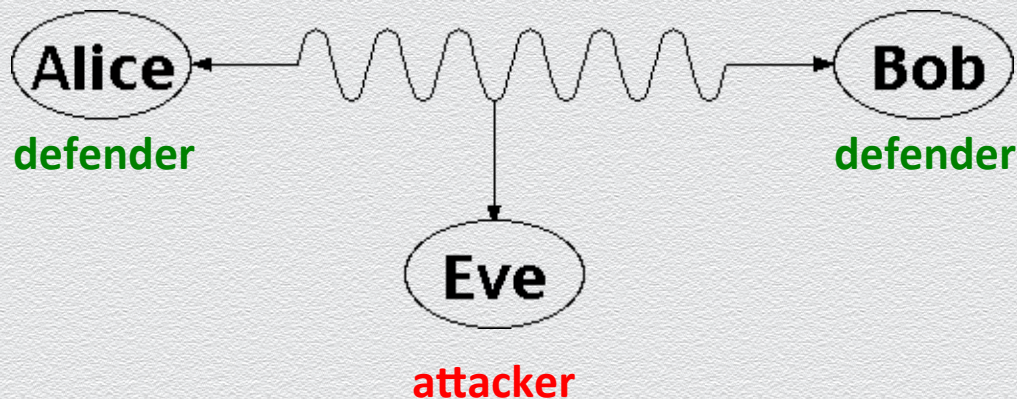
Fundamental security property

- ◆ an asset is viewed only by authorized parties
- ◆ “C” in the CIA triad

*“computer security seeks to prevent **unauthorized viewing (confidentiality)** or modification (integrity) of **data** while preserving access (availability)”*

Eavesdropping

- ◆ main threat against confidentiality of **in-transit** data



Problem setting: Secret communication

Two parties wish to communicate over a channel

- ◆ Alice (sender/source) wants to send a message m to Bob (recipient/destination)

Underlying channel is unprotected

- ◆ Eve (attacker/adversary) can eavesdrop any sent messages
- ◆ e.g., packet sniffing over networked or wireless communications



Solution concept: Symmetric-key encryption

Main idea

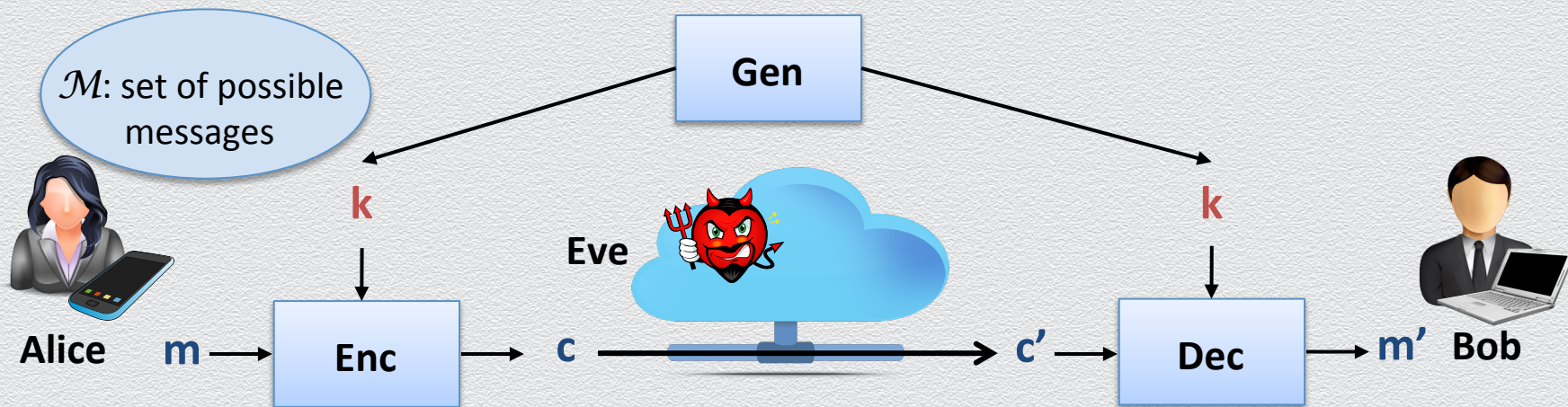
- ◆ secretly transform message so that it is **unintelligible** while in transit
 - ◆ Alice **encrypts** her message m to **ciphertext** c , which is sent instead of **plaintext** m
 - ◆ Bob **decrypts** received message c to original message m
 - ◆ Eve can intercept c but “**cannot learn**” m from c
 - ◆ Alice and Bob share a **secret key** k that is used for both message transformations



Security tool: Symmetric-key encryption scheme

Abstract cryptographic primitive, **a.k.a. cipher**, defined by

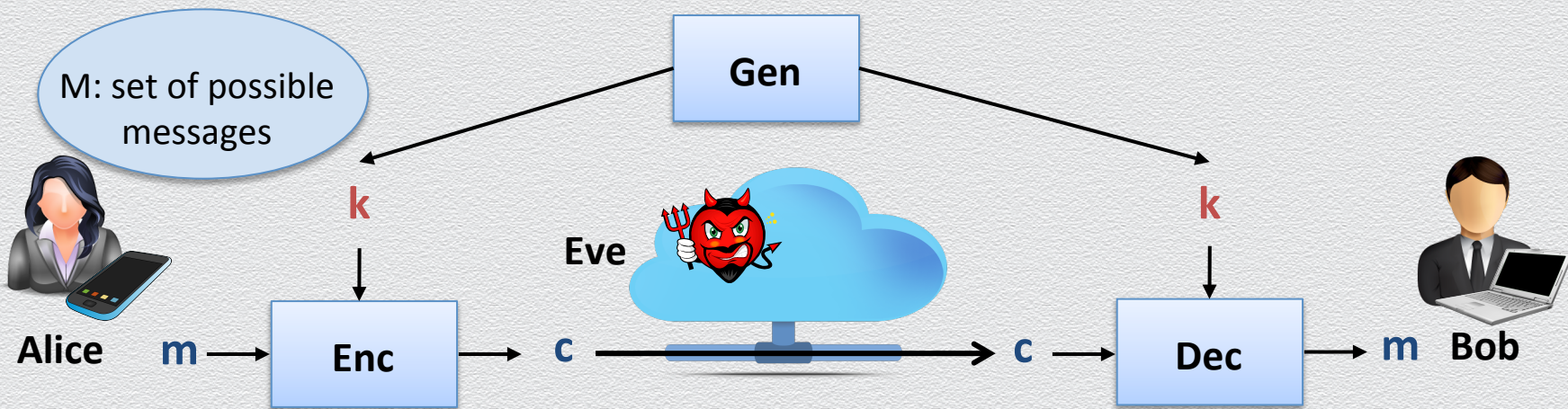
- ◆ a **message space** \mathcal{M} ; and
- ◆ a triplet of algorithms **(Gen, Enc, Dec)**
 - ◆ Gen, Enc are probabilistic algorithms, whereas Dec is deterministic
 - ◆ Gen outputs a uniformly random key k (from some key space \mathcal{K})



Desired properties for symmetric-key encryption scheme

By design, any symmetric-key encryption scheme should satisfy the following

- ◆ **efficiency:** key generation & message transformations “are fast”
- ◆ **correctness:** for all m and k , it holds that $\text{Dec}(\text{Enc}(m, k), k) = m$
- ◆ **security:** one “cannot learn” plaintext m from ciphertext c



Kerckhoff's principle

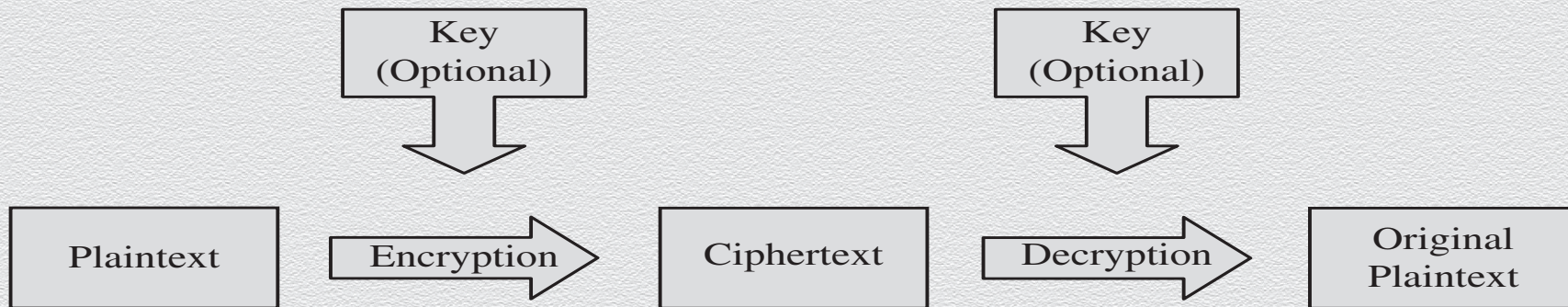
“The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.”

Reasoning

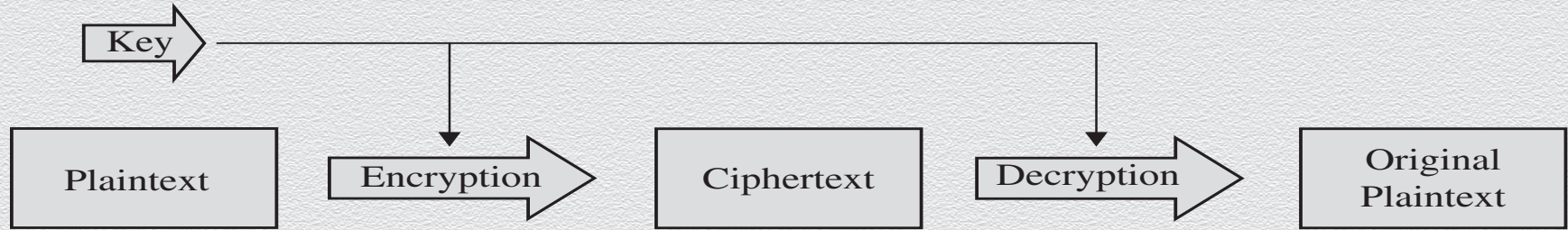
- ◆ due to security & correctness, Alice & Bob must share some secret info
- ◆ if no shared key captures this secret info, it must be captured by Enc, Dec
- ◆ but keeping Enc, Dec secret is problematic
 - ◆ harder to keep secret an algorithm than a short key (e.g., after user revocation)
 - ◆ harder to change an algorithm than a short key (e.g., after secret info is exposed)
 - ◆ riskier to rely on custom/ad-hoc schemes than publicly scrutinized/standardized ones

Symmetric-key encryption

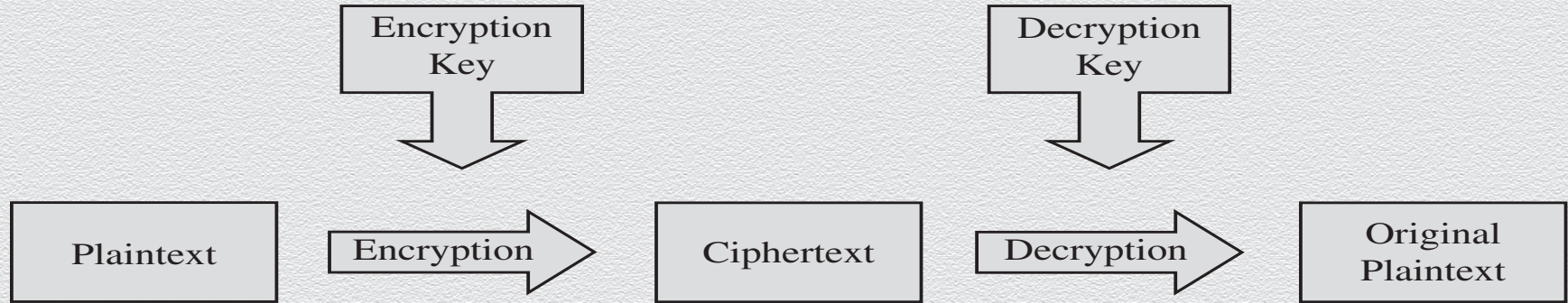
- ◆ Also referred to as simply “symmetric encryption”



Symmetric Vs. Asymmetric encryption



(a) Symmetric Cryptosystem



(b) Asymmetric Cryptosystem

Main application areas

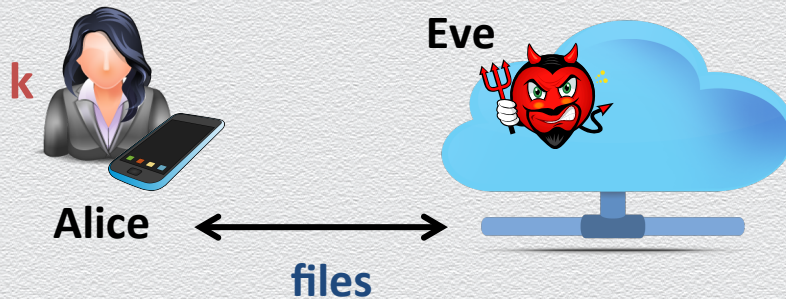
Secure communication

- ◆ **encrypt messages** sent among parties
- ◆ assumption
 - ◆ Alice and Bob **securely generate, distribute & store shared key k**
 - ◆ attacker does not learn key k



Secure storage

- ◆ **encrypt files** outsourced to the cloud
- ◆ assumption
 - ◆ Alice **securely generates & stores key k**
 - ◆ attacker does not learn key k



Brute-force attack

Generic attack

- ◆ given a captured ciphertext c and known key space \mathcal{K} , Dec
- ◆ strategy is an **exhaustive search**
 - ◆ for all possible keys k in \mathcal{K}
 - ◆ determine if $\text{Dec}(c, k)$ is a likely plaintext m
- ◆ **requires some knowledge on the message space \mathcal{M}**
 - ◆ i.e., structure of the plaintext (e.g., PDF file or email message)

Countermeasure

- ◆ key should be a **random** value from a **sufficiently large** key space \mathcal{K} to make exhaustive search attacks **infeasible**

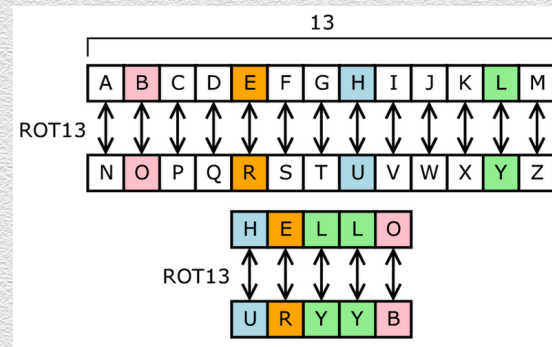


2.2 Classical ciphers

Substitution ciphers

Large class of ciphers

- ◆ each letter is uniquely replaced by another
- ◆ there are $26!$ possible substitution ciphers
 - ◆ e.g., one popular substitution “cipher” for some Internet posts is ROT13
- ◆ historically
 - ◆ all classical ciphers are of this type



General structure of classical ciphers

Based on letter substitution

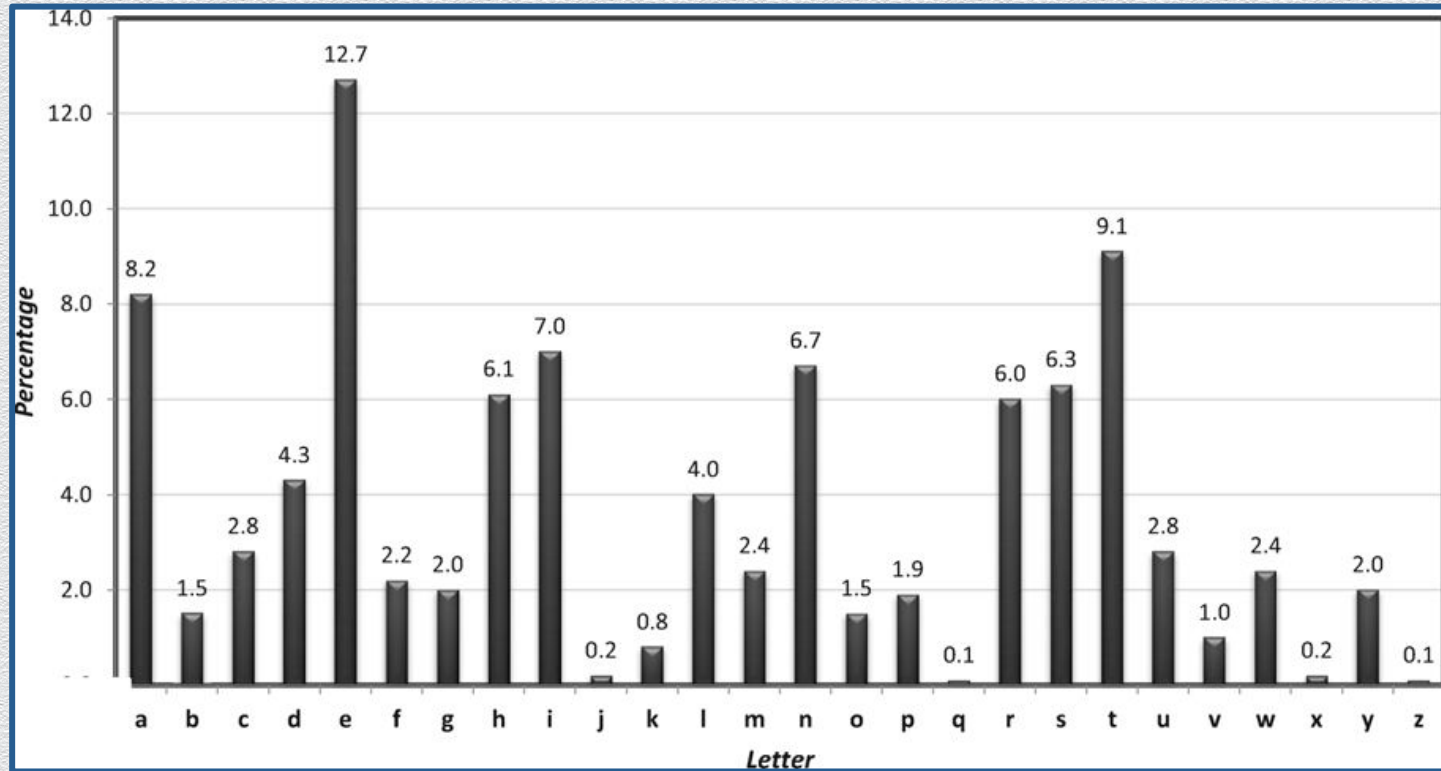
- ◆ message space \mathcal{M} is “valid words” from a given alphabet
 - ◆ e.g., English text without spaces, punctuation or numerals
 - ◆ characters can be represented as numbers in $[0:25]$
- ◆ encryption
 - ◆ mapping each plaintext character into another character
 - ◆ character mapping is typically defined as a “shift” of a plaintext character by a number of positions in a canonical ordering of the characters in the alphabet
 - ◆ character shifting occurs with “wrap-around” (using mod 25 addition)
- ◆ decryption
 - ◆ undo character shifting with “wrap-around” (using mod 25 subtraction)

Limitations of substitution ciphers

Generally, susceptible to frequency (and other statistical) analysis

- ◆ letters in a natural language, like English, are not uniformly distributed
- ◆ cryptographic attacks against substitution ciphers are possible
 - ◆ e.g., by exploiting knowledge of letter frequencies, including pairs and triples

Letter frequency in (sufficiently large) English text



Classical ciphers – examples

Caesar's cipher

- ◆ shift each character in the message by 3 positions
 - ◆ or by 13 position in ROT-13
- ◆ cryptanalysis
 - ◆ **no secret key is used** – based on “security by obscurity”
 - ◆ thus the code is trivially insecure once knows Enc (or Dec)

Classical ciphers – examples (II)

Shift cipher

- ◆ **keyed extension** of Caesar's cipher
- ◆ randomly set key k in $[0:25]$
 - ◆ shift each character in the message by k positions
- ◆ cryptanalysis
 - ◆ **brute-force attacks** are effective given that
 - ◆ **key space is small** (26 possibilities or, actually, 25 as 0 should be avoided)
 - ◆ message space M is **restricted to “valid words”**
 - ◆ e.g., corresponding to valid English text

Classical ciphers – examples (III)

Mono-alphabetic substitution cipher

- ◆ **generalization** of shift cipher
- ◆ key space defines **permutation** on alphabet
 - ◆ use a **1-1 mapping between characters** in the alphabet to produce ciphertext
 - ◆ i.e., shift each **distinct** character in the plaintext (by some appropriate number of positions defined by the key) to get a **distinct** character in the ciphertext
- ◆ cryptanalysis
 - ◆ key space is large (of the order of $26!$ or $\sim 2^{88}$) but cipher is vulnerable to attacks
 - ◆ character mapping is **fixed** by key so **plaintext & ciphertext exhibit same statistics**

Alternative attack against “shift cipher”

- ◆ brute-force attack + inspection if English “make sense” is quite **manual**
- ◆ a better **automated** attack is based on statistics
 - ◆ if character i (in $[0:25]$) in the alphabet has frequency p_i (in $[0..1]$), then
 - ◆ from known statistics, we know that $\sum_i p_i^2 \approx 0.065$, so
 - ◆ since character i (in plaintext) is mapped to character $i + k$ (in ciphertext)
 - ◆ if $L_j = \sum_i p_i q_{i+j}$, then we expect that $L_k \approx 0.065$
- ◆ thus, a brute-force attack can **test** all possible keys w.r.t. the **above criterion**
 - ◆ the search space **remains the same**
 - ◆ yet, the condition to finish the search **becomes much simpler**

2.3 Perfect secrecy

A formal view of symmetric encryption

A symmetric-key encryption scheme is defined by



- ◆ a **message space** \mathcal{M} , $|\mathcal{M}| > 1$, and a triple (**Gen**, **Enc**, **Dec**)
- ◆ **Gen**: probabilistic key-generation algorithm, defines **key space** \mathcal{K}
 - ◆ $\text{Gen} \rightarrow k \in \mathcal{K}$
- ◆ **Enc**: probabilistic encryption algorithm, defines **ciphertext space** \mathcal{C}
 - ◆ $\text{Enc}: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, $\text{Enc}(k, m) = \text{Enc}_k(m) \rightarrow c \in \mathcal{C}$
- ◆ **Dec**: deterministic decryption algorithm
 - ◆ $\text{Dec}: \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$, $\text{Dec}(k, c) = \text{Dec}_k(c) := m \in \mathcal{M}$

A view through the lens of probability

Symmetric-key encryption scheme: **message space** \mathcal{M} & triple (**Gen**, **Enc**, Dec)

- ◆ **Gen** defines **key space** \mathcal{K} i.e., $\text{Gen} \rightarrow k \in \mathcal{K}$
- ◆ **Enc** defines **ciphertext space** \mathcal{C} i.e., $\text{Enc}_k(m) \rightarrow c \in \mathcal{C}$

Assumption

- ◆ **messages** & **keys** are chosen **independently** according to prob. distributions $\mathcal{D}_{\mathcal{M}}, \mathcal{D}_{\mathcal{K}}$
- ◆ if M, K are random variables denoting the chosen message and key respectively
 - ◆ for any $m \in \mathcal{M}$, $\mathcal{D}_{\mathcal{M}}$ defines $\Pr[M = m]$  *a priori probability that m is sent*
 - ◆ for any $k \in \mathcal{K}$, $\mathcal{D}_{\mathcal{K}}$ defines $\Pr[K = k]$  *typically uniform*

Fact

- ◆ given $\mathcal{D}_{\mathcal{M}}, \mathcal{D}_{\mathcal{K}}$ & **internally used randomness**, Enc imposes a prob. distr. $\mathcal{D}_{\mathcal{C}}$ (over \mathcal{C})
- ◆ if C denotes the sent ciphertext, then for any $c \in \mathcal{C}$, $\mathcal{D}_{\mathcal{C}}$ defines $\Pr[C = c]$

Perfect correctness

For any $k \in \mathcal{K}$, $m \in \mathcal{M}$ and any ciphertext c output of $\text{Enc}_k(m)$,
it holds that

$$\Pr[\text{Dec}_k(c) = m] = 1$$

Towards defining perfect security

- ◆ defining security for an encryption scheme is not trivial
 - ◆ e.g., what we mean by << Eve “cannot learn” m (from c) >> ?
- ◆ our setting so far is a random experiment
 - ◆ a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
 - ◆ a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
 - ◆ $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

how to define security?

Attempt 1: Protect the key k!

- ◆ Security means that

- ◆ a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- ◆ a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- ◆ $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

the adversary should **not** be able to **compute the key k**

- ◆ Intuition

- ◆ it'd better be the case that the key is protected!...



necessary condition

- ◆ Problem

- ◆ this definition fails to exclude clearly insecure schemes
- ◆ e.g., the key is never used, such as when $\text{Enc}_k(m) := m$



but not
sufficient condition!

Attempt 2: Don't learn m !

- ◆ Security means that

- ◆ a message m is chosen according to \mathcal{D}_M
- ◆ a key k is chosen according to \mathcal{D}_K
- ◆ $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

the adversary should **not** be able to **compute the message m**

- ◆ Intuition
 - ◆ it'd better be the case that the message m is not learned...
- ◆ Problem
 - ◆ this definition fails to exclude clearly undesirable schemes
 - ◆ e.g., those that protect m partially, i.e., they reveal the least significant bit of m

Attempt 3: Learn nothing!

- ◆ Security means that

- ◆ a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- ◆ a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- ◆ $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

the adversary should **not** be able to **learn any information about m**

- ◆ Intuition

- ◆ it seems close to what we should aim for perfect secrecy...

- ◆ Problem

- ◆ this definition ignores the adversary's prior knowledge on \mathcal{M}
 - ◆ e.g., distribution $\mathcal{D}_{\mathcal{M}}$ may be known or estimated
 - ◆ m is a valid text message, or one of "attack", "no attack" is to be sent

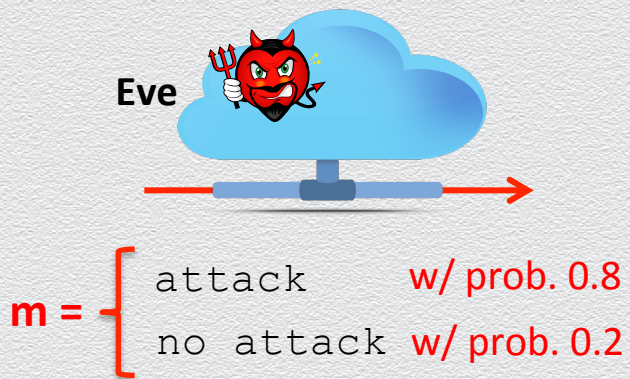
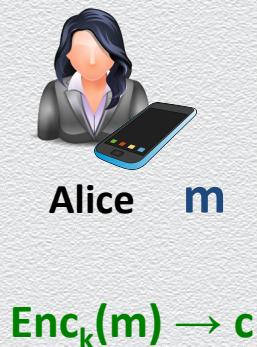
Attempt 4: Learn nothing more!

- ◆ Security means that

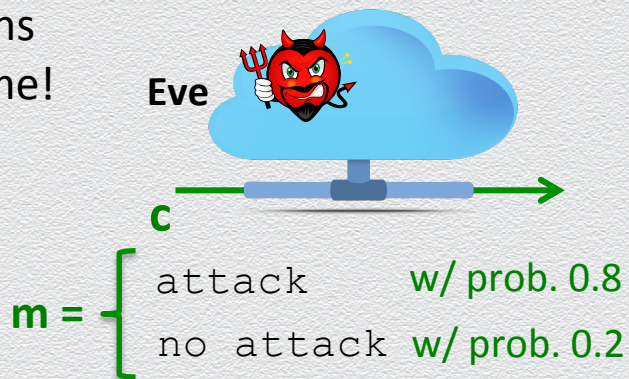
- ◆ a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- ◆ a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- ◆ $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

the adversary should **not** be able to **learn any additional information on m**

- ◆ How can we formalize this?



Eve's view
remains
the same!



Perfect secrecy (or information-theoretic security)

Definition 1

A symmetric-key encryption scheme (Gen, Enc, Dec) with message space \mathcal{M} , is **perfectly secret** if for every $\mathcal{D}_{\mathcal{M}}$, every message $m \in \mathcal{M}$ and every ciphertext $c \in \mathcal{C}$ for which $\Pr [C = c] > 0$, it holds that

$$\Pr[M = m \mid C = c] = \Pr [M = m]$$

- ♦ intuitively
 - ♦ the *a posteriori* probability that any given message m was actually sent is the **same** as the *a priori* probability that m would have been sent
 - ♦ observing the ciphertext reveals **nothing** about the underlying plaintext

Alternative view of perfect secrecy

Definition 2

A symmetric-key encryption scheme $(\text{Gen}, \text{Enc}, \text{Dec})$ with message space \mathcal{M} , is **perfectly secret** if for every messages $m, m' \in \mathcal{M}$ and every $c \in \mathcal{C}$, it holds that

$$\Pr[\text{Enc}_K(\textcolor{brown}{m}) = c] = \Pr [\text{Enc}_K(\textcolor{blue}{m}') = c]$$

- ◆ intuitively
 - ◆ the probability distribution \mathcal{D}_C **does not depend** on the plaintext
 - ◆ i.e., M and C are **independent** random variables
 - ◆ the ciphertext contains “**no information**” about the plaintext
 - ◆ “**impossible to distinguish**” an encryption of $\textcolor{brown}{m}$ from an encryption of $\textcolor{blue}{m}'$

The two definitions of perfect secrecy are equivalent

a posteriori = a priori

\sim

C is independent of M

For every $\mathcal{D}_{\mathcal{M}}$, $m \in \mathcal{M}$ and $c \in \mathcal{C}$, for which $\Pr[C = c] > 0$, it holds that

$$\Pr[M = m \mid C = c] = \Pr[M = m]$$

For every $m, m' \in \mathcal{M}$ and $c \in \mathcal{C}$, it holds that

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

random
experiment

$$\mathcal{D}_{\mathcal{M}} \rightarrow m = M$$
$$\mathcal{D}_{\mathcal{K}} \rightarrow k = K$$
$$\text{Enc}_k(m) \rightarrow c = C$$


Eve's view
remains
the same!



2.4 The one-time pad

The one-time pad: A perfect cipher

A type of “substitution” cipher that is “absolutely unbreakable”

- ◆ invented in 1917 Gilbert Vernam and Joseph Mauborgne
- ◆ “substitution” cipher
 - ◆ **individually** replace plaintext characters with **shifted** ciphertext characters
 - ◆ **independently** shift each message character in a **random** manner
 - ◆ to encrypt a plaintext of length n , use n uniformly random keys k_1, \dots, k_n
- ◆ “absolutely unbreakable”
 - ◆ **perfectly secure** (when used correctly)
 - ◆ based on **independently random** shifts

The one-time pad (OTP) cipher

Fix t to be any positive integer; set $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^t$

- ◆ **Gen**: choose t bits uniformly at random (each bit independently w/ prob. .5)
 - ◆ $\text{Gen} \rightarrow \{0,1\}^t$
- ◆ **Enc**: given a key and a message of equal lengths, compute the bit-wise XOR
 - ◆ $\text{Enc}(k, m) = \text{Enc}_k(m) \rightarrow k \oplus m$ (i.e., mask the message with the key)
- ◆ **Dec**: compute the bit-wise XOR of the key and the ciphertext
 - ◆ $\text{Dec}(k, c) = \text{Dec}_k(c) := k \oplus c$
- ◆ **Correctness**
 - ◆ trivially, $k \oplus c = k \oplus k \oplus m = 0 \oplus m = m$

OTP is perfectly secure (using Definition 2)

For all k -bit long messages m_1 and m_2 and ciphertexts c , it holds that

$$\Pr[E_K(m_1) = c] = \Pr[E_K(m_2) = c],$$

where probabilities are measured over the possible keys chosen by Gen.

Proof

- ◆ the event “ $\text{Enc}_K(m_1) = c$ ” is the event “ $m_1 \oplus K = c$ ” or the event “ $K = m_1 \oplus c$ ”
- ◆ K is chosen at random, irrespectively of m_1 and m_2 , with probability 2^{-t}
- ◆ namely ciphertext does not reveal anything about the plaintext

OTP characteristics

A “substitution” cipher

- ◆ encrypt an n -symbol m using n uniformly random “shift keys” k_1, k_2, \dots, k_n

2 equivalent views

- ◆ $\mathcal{K} = \mathcal{M} = \mathcal{C}$

view 1 $\{0,1\}^n$

or

view 2 $G, (G, +)$ is a group

- ◆ “shift” method

bit-wise XOR ($m \oplus k$)

addition/subtraction ($m \pm k$)

Perfect secrecy

- ◆ since each shift is random, every ciphertext is equally likely for any plaintext

Limitations (on efficiency)

- ◆ “shift keys” (1) are **as long as messages** & (2) **can be used only once**

Perfect, but impractical

In spite of its perfect security, OTP has two notable weaknesses

- ◆ the key has to be **as long as** the plaintext
 - ◆ limited applicability
 - ◆ key-management problem
- ◆ keys **cannot be reused**
 - ◆ security vulnerability
 - ◆ e.g., reusing a key once, leaks the XOR of two plaintext messages
 - ◆ detriment to secure communication
 - ◆ securely distributing fresh keys is as hard as securely exchanging messages

Theoretical analysis of OTP – Shannon (1949)

Inherent limitations

- ◆ OTP is **optimal** in the class of **perfectly secret** symmetric encryption schemes
 - ◆ (1) For any perfect cipher with message space \mathcal{M} and key space \mathcal{K} : $|\mathcal{K}| \geq |\mathcal{M}|$
 - ◆ thus, OTP keys must be at least as large as the message length
 - ◆ (2) For any perfect cipher one-time key usage is **necessary**

Characterization

- ◆ A symmetric cipher such that $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$ is perfectly secure **if and only if**:
 - ◆ Every key $k \in \mathcal{K}$ is chosen with **equal** probability $1/|\mathcal{K}|$ (by algorithm Gen)
 - ◆ For every $m \in \mathcal{M}$, $c \in \mathcal{C}$, there exists a **unique** key $k \in \mathcal{K}$ such that $\text{Enc}_k(m)$ outputs c

Importance of OTP weaknesses

Inherent trade-off between efficiency / practicality Vs. perfect secrecy

- ◆ historically, OTP has been used efficiently & insecurely
 - ◆ repeated use of one-time pads compromised communications during the cold war
 - ◆ NSA decrypted Soviet messages that were transmitted in the 1940s
 - ◆ that was possible because the Soviets reused the keys in the one-time pad scheme
- ◆ modern approaches resemble OTP encryption
 - ◆ efficiency via use of pseudorandom OTP keys
 - ◆ “almost perfect” secrecy

