Class CS 6903, Lecture n. 1

- Welcome to the class!
- Please check the syllabus online (it should contain all you need to know about this class)

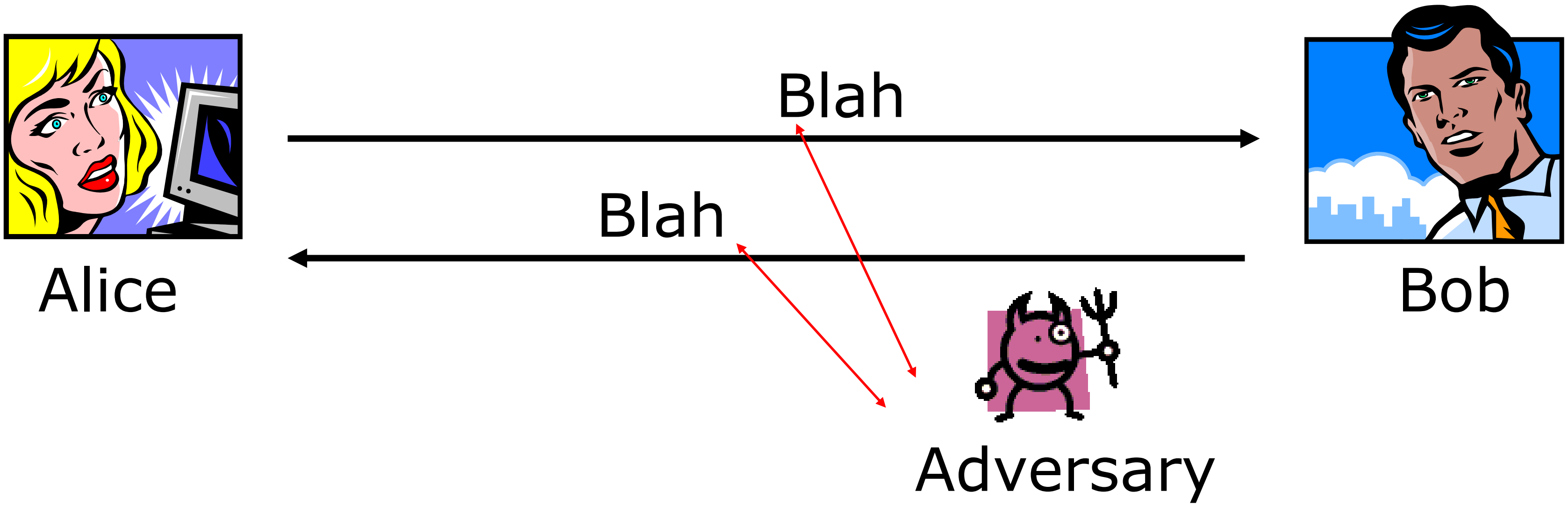
Summary of Lecture 1

- Cryptography: main problem + history
- Some Probability Background
- Some Background on Algorithms
- Classical Cryptography
- Encryption with Perfect Secrecy

Cryptography: context

- Fundamental and critical part of larger security systems
- Cryptography is only the beginning towards solving your security problems
- Cryptography is a very difficult topic
- We made cryptography much easier
- Thinking like a cryptographer makes you a better security expert

Cryptography: main problem



- Main problem statement: Two parties, commonly referred as Alice and Bob, want to exchange messages
  - privately from any unauthorized parties that are allowed to eavesdrop the exchanged communication
- Modern Cryptography studies this problem (solved using encryption) + many more related ones

Cryptography: history

- Need for cryptography was recognized since ancient times:
  - About 4000 years ago egyptians “seemed to encrypt” some hieroglyphic writings (1st documented use of cryptography)
  - Roman emperor Julius Ceasar used “some encryption” to communicate with his commanders
  - Etc…
  - See Kahn’s book, *The Codebreakers,* for a detailed non-technical history
- History of Cryptography can be divided into Classical Cryptography and Modern Cryptography
  - Historically: classical cryptography was used by military and intelligence organizations
  - Today: modern cryptography is everywhere, in most computer systems, used by most computer users (often unknowingly)

Applications and Definition

- Application scenarios:
  - Imagine {Allies, politicians, business people, lovers…} communicating during {battles, negotiations, deals, all of the above…}, respectively
  - User revealing credit card number, password, PIN
  - User accessing e-mail stored on a website
  - Multiple users communicating via phone, video, chats, etc
- Cryptography definitions:
  - Concise Oxford Dictionary: art of writing and solving codes
  - Merriam-Webster Dictionary: the computerized encoding and decoding of information
  - [KL] (variant): scientific study of techniques for securing digital information as well as multi-party computations and transactions

Summary of Lecture

- Cryptography: main problem + history
- Some Probability Background
- Some Background on Algorithms
- Classical Cryptography
- Encryption with Perfect Secrecy

Discrete Probability Distributions

- An experiment is a procedure that yields one of a given set of outcomes
- The individual possible outcomes are called simple events
- The set of all possible outcomes is called the sample space
- Unless otherwise specified, we only consider discrete sample spaces S; that is, sample spaces with only finitely many possible outcomes, labeled $s_1,\ldots, s_n$
- A probability distribution P on S is a sequence of numbers $p_1,\ldots, p_n$ that are all non-negative and sum to 1
  - The number p_i is interpreted as the probability of s_i being the outcome of the experiment
- An event E is a subset of the sample space S
- The probability that event E occurs, denoted Pr(E), is the sum of the probabilities p_i of all simple events s_i which belong to E

Probability of Events

- If E is an event, let $\overline{E}$ denote its complement; that is, the event E that does not happen
- If E_1, E_2 are events, let $E_1 \vee E_2$ denote their disjunction; that is, the event that at least one of them happens
- If E_1, E_2 are events, let $E_1 \wedge E_2$ denote their conjunction; that is, the event that both of them happen
- We have that:
  - Pr [E] = 1 – Pr [$\overline{E}$]
  - Pr [$E_1 \vee E_2$] <= Pr [E_1] + Pr [E_2] (union bound)
  - Let $F, E_1 \ldots E_n$ be events such that Pr [$E_1 \vee \ldots \vee E_n$] = 1 and $\Pr[E_i \wedge E_j] = 0$ for all i,j. Then $\Pr[F] = \sum_{i=1}^{n} \Pr[F \wedge E_i]$

Conditioning Probability

- If E_1, E_2 are events such that $\Pr[E_2] \neq 0$, the conditional probability of event E_1 given E_2, denoted as $\Pr[E_1 \mid E_2]$, is defined as $\Pr[E_1 \mid E_2] = \dfrac{\Pr[E_1 \wedge E_2]}{\Pr[E_2]}$
- Note that $\Pr[E_1 \wedge E_2] = \Pr[E_1 \mid E_2] \cdot \Pr[E_2]$
- If E_1, E_2 are independent, then $\Pr[E_1 \wedge E_2] = \Pr[E_1] \cdot \Pr[E_2]$
- The conditioning rule says the following:
  - let $F, E_1 \ldots E_n$ be events such that Prob $[E_1 \vee \ldots \vee E_n]$ = 1 and $\Pr[E_i \wedge E_j] = 0$ for all i,j. Then $\Pr[F] = \sum_{i=1}^{n} \Pr[F \mid E_i] \cdot \Pr[E_i]$
- The Bayes Theorem says that if E_1, E_2 are events such that $\Pr[E_2] \neq 0$, it holds that $\Pr[E_1 \mid E_2] = \dfrac{\Pr[E_1] \cdot \Pr[E_2 \mid E_1]}{\Pr[E_2]}$

Estimating probabilities

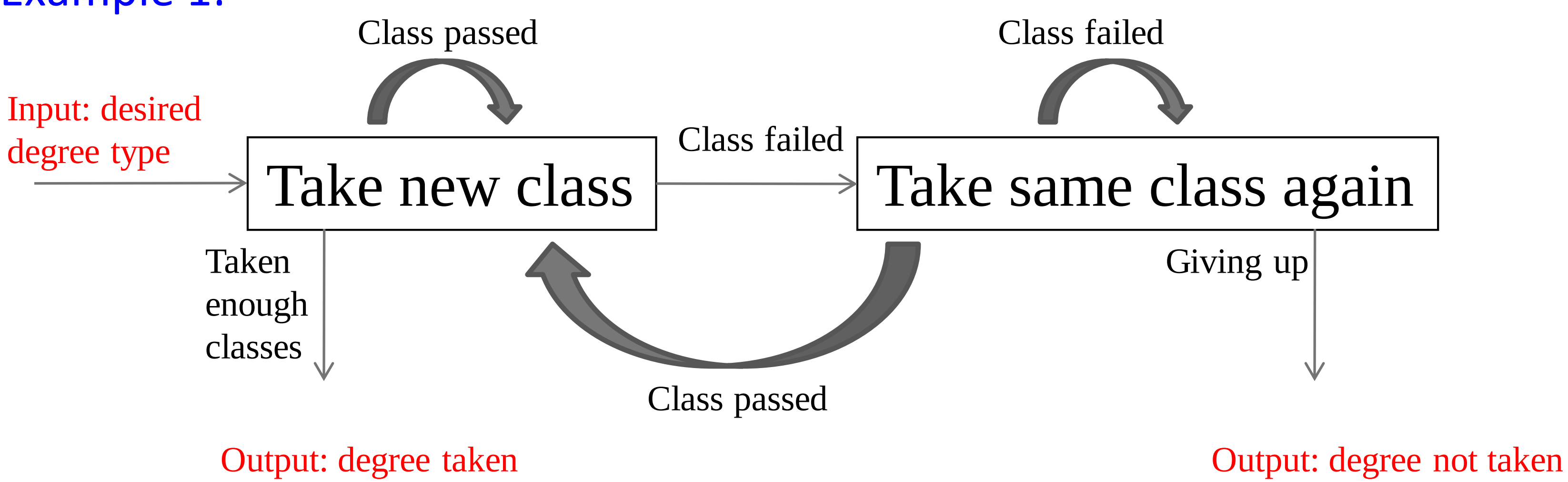
- Assume a discrete sample space S with outcomes $s_1,\ldots, s_n$
- How do we estimate $p_1,\ldots, p_n$?
  - Recall: p_i is the probability of s_i being the outcome of the experiment
- Basic approach:
  - Run the experiment many (i.e., m) times
  - Record the number of occurrences o_i with outcome s_i, for $i=1,\ldots,n$
  - Define $p'_i = o_i / m$, for $i=1,\ldots,n$
- Results from statistics imply:
  - If m is large enough, then p'_i is a good enough estimate for p_i

Summary of Lecture

- Cryptography: main problem + history
- Some Probability Background
- Some Background on Algorithms
- Classical Cryptography
- Perfect Secrecy for Encryption

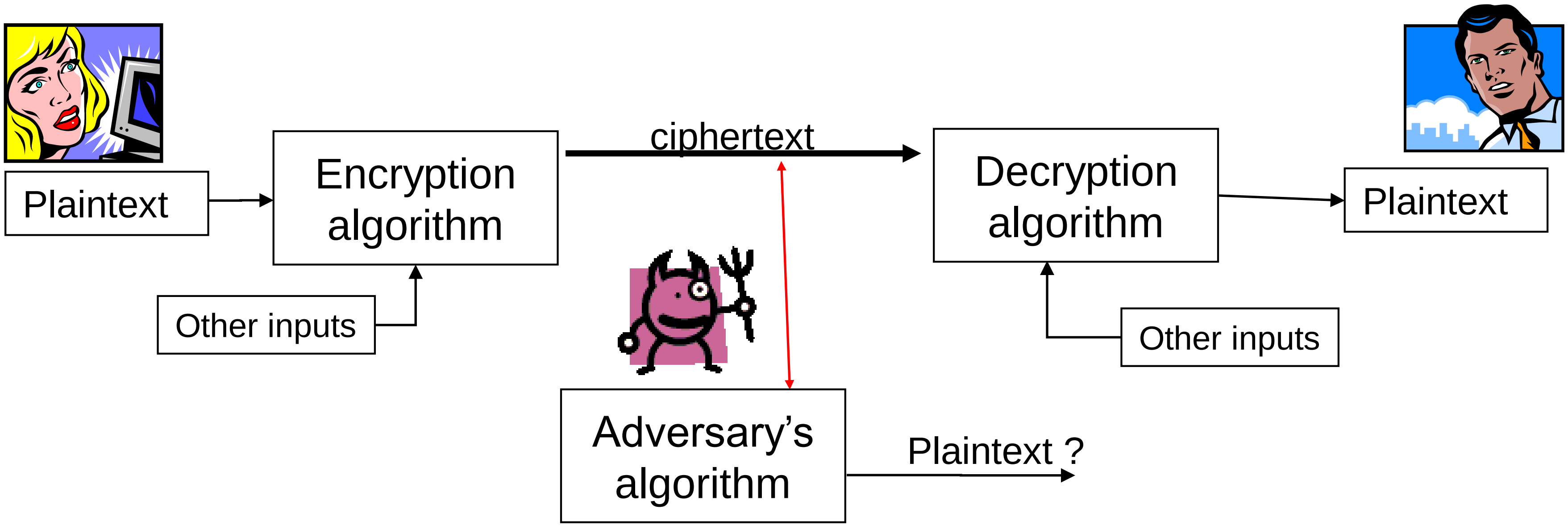
Defining Algorithms

- Definition: finite sequence of "primitive" instructions, expressed in an "unambiguous" language, that start from a (possibly variable) input value and end with a (possibly variable) output value
- Note: Definition implies that if primitive instructions compute a (mathematical) function, then entire algorithm computes a related (mathematical) function
- Example 1:



- Example 2: on input value x, scan all elements of n-element array until you find one equal to x

Algorithms in Encryption



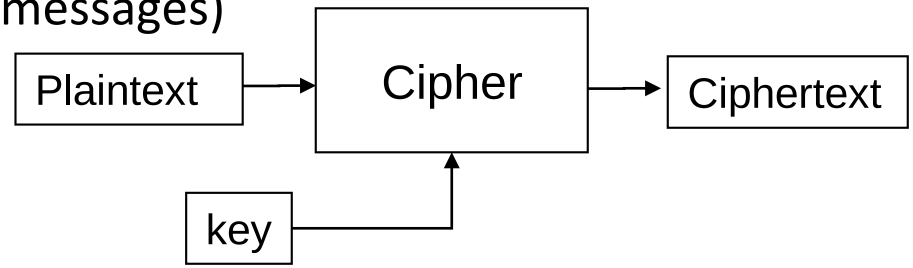
- Alice uses an encryption algorithm to send an encrypted message to Bob
  - On input plaintext and other inputs, this algorithm returns a ciphertext
- Bob uses a decryption algorithm to receive the message sent by Alice
  - On input ciphertext and other inputs, this algorithm returns a plaintext
- The adversary himself uses an algorithm to process the ciphertext returned by the encryption algorithm (and try to derive the plaintext)
  - On input ciphertext + other inputs, it returns a candidate plaintext or a failure symbol

Summary of Lecture

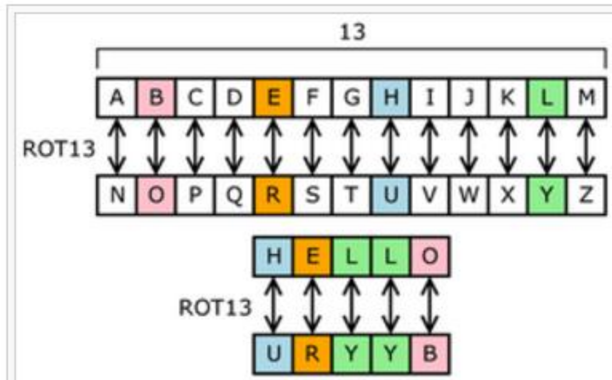
- Cryptography: main problem + history
- Some Probability Background
- Some Background on Algorithms
- Classical Cryptography
- Perfect Secrecy for Encryption

Classical Cryptography: one early cipher

- Problem: design “cipher” algorithms to encrypt messages into ciphertext (and decrypt ciphertext back into original messages)



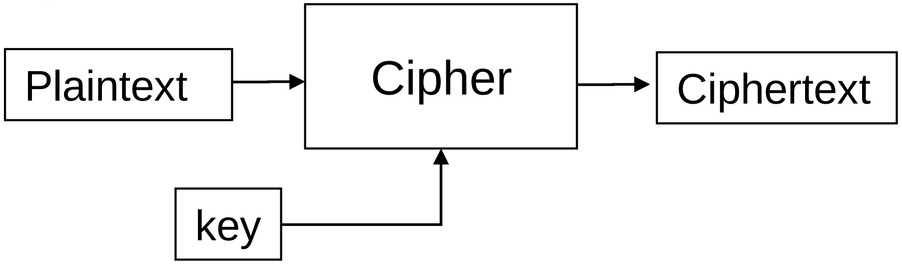
- Julius Ceasar’s cipher:
  - Ex. 1: “easy” → “hdvb”
  - Encryption algorithm: shift message letters by 3 forward positions
  - Decryption algorithm: shift ciphertext letters by 3 backward positions
  - Message secrecy properties: very limited secrecy
    - any adversary knowing the algorithm can decrypt immediately
    - secrecy, if any, is based on obscurity
  - ROT-13 (a variant with a shift of 13 positions) is used today in online forums to prevent accidental decryption



ROT13 is a Caesar cipher, a type of substitution cipher. In ROT13, the alphabet is rotated 13 steps.

A second cipher

- Preliminary observation:
  - In ciphers like Ceasar's cipher encryption is always the same
  - There was no random (or real) key



- A second cipher – let us introduce:
  - some randomness in the algorithm (using a random key k); here, set k=11:

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

A second cipher – the shift cipher

- The shift cipher:
  - Let k be a random number in {0,…,25}
  - Encryption algorithm: shift message letters by k forward positions
  - Decryption algorithm: shift ciphertext letters by k backward positions
  - Ex.: k=11, message="easy class", ciphertext="qmel oxmee"
  - Message secrecy properties: very limited secrecy
    - Secrecy based on the randomness of the key
    - <u>But</u> any adversary knowing the algorithm can try all 26 values for k and find the one for which it can easily decrypt into a meaningful plaintext
- The large key space principle:
  - The above was an exhaustive (or brute-force) search attack
  - Key space being small → attack was efficient
  - Necessary (but not necessarily sufficient) principle: a secure encryption scheme must have a key space not vulnerable to exhaustive search

A third cipher

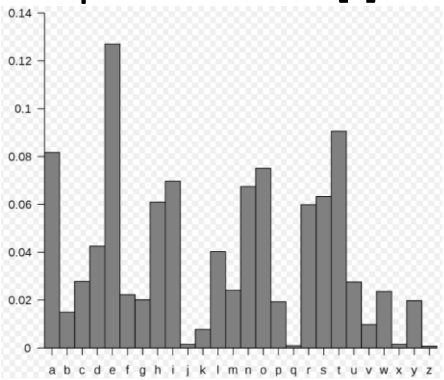
- Preliminary observations:
  - Key space in the shift cipher was small
  - In particular, all message characters were shifted by the same (secret) amount
  - What happens if this amount differs for each character?
- A third cipher – let us add:
  - More randomness in the algorithm (and a large key space); here, consider the following permutation:

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

A third cipher - description

- The mono-alphabetic substitution cipher:
  - Let k[0],…,k[25] be distinct random numbers in {0,…,25}
  - Encryption algorithm: for each message letter, if the letter is equal to the i-th alphabet letter, where i is in {0,…,25}, map it to letter k[i]
  - Decryption algorithm: decrypt analogously by computing the inverse (random) permutation; i.e., for each ciphertext letter, if the letter is equal to the k[i]-th alphabet letter, where i is in {0,…,25}, map it to letter i.
  - Ex.: k permutation from previous slide,

    m="easy class", c="hywv uoyww"
  - Message secrecy properties: very limited secrecy
    - Secrecy based on key randomness + high key space (26!)
    - <u>But</u> the mapping of each letter is fixed, and the probability distributions of individual letters in the English language is known (prob[e]>prob[t]>prob[a]>...)
    - Thus, the adversary uses average letter frequencies in English language to derive k[i] values (see previous approach to estimate probabilities from occurrences) and decrypt ciphertexts into meaningful plaintext



- Conclusion:
  - The above cipher had high key space but was still insecure

A fourth cipher

- Preliminary observation:
  - In the mono-alphabetic cipher any message character was mapped to the same (secret) value
  - what if we impose that particular mapping to differ over time?
- A fourth cipher – let us add:
  - Mapping different instances of the same plaintext character to different ciphertext characters
  - More randomness in the algorithm using a (short) random key; here, set as "mykey":

Plaintext	H	E	L	L	O	C	R	Y	P	T	O	S	T	U	D	E	N	T
Key	M	Y	K	E	Y	M	Y	K	E	Y	M	Y	K	E	Y	M	Y	K
Ciphertext	U	D	W	Q	N	P	Q	J	U	S	B	R	E	Z	C	R	M	E

A fourth cipher – definition

- The Vigenere (poly-alphabetic substitution) cipher:
  - Let k[1],…,k[t] be random numbers in {1,…,26}, for some small t
  - Encryption algorithm: for each group of t consecutive message letters, the letter in the i-th position in the group is shifted by k[i] forward positions, for i=1,…,t
  - Decryption algorithm: shift analogously by k[i] backward positions
  - Message secrecy properties:
    - previous attack is now not possible, because the same English letter is shifted by different characters;
    - <u>however</u>, all ciphertext characters c[j], c[j+t],… at distance t from each other are shifted using the same random number k[j] for some j in {1,..,t};
    - this number can be found by studying the frequency of ciphertext characters for each value of j (see previous approach to estimate probabilities from occurrences) and check which value for k[j] gives "right" probability distribution
    - once all k[j] are found, the adversary can easily decrypt the ciphertext
    - That assumed t was known; when unknown, try for all t=1,…,max key length
- Conclusion: The above cipher had large key space but was still insecure

Conclusions on classical cryptography

- Current view of classical Cryptography:
  - Originally considered as an art of designing and breaking “secrecy codes”
  - Art built on heuristic techniques (i.e., techniques with minimal, if at all, rigorously formulated security guarantees)
  - Based on predecessors of today’s popular concepts such as “confusion”
  - Based on today’s less popular concepts such as “obscurity”
  - Only worked for a very limited set of applications and adversary scenarios
  - Only worked until someone was able to break the scheme
- Lessons learned:
  - Large key space principle
  - A large key space is necessary but not sufficient
  - Designing secure encryption schemes is a hard task

Question set 1

- Why is ROT-13 a better or worse choice than Ceasar's cipher when the choice criteria is making accidental decryption harder?
- Assume a meaningful plaintext is encrypted using the shift cipher. How many decryption attempts are sufficient for an exhaustive (or brute-force) search attack to find the plaintext with probability 1? How does the number change if we only require probability 1/2 of success?
- How would you estimate the probability that a given English letter appears in an unencrypted message? What would you expect about the probability that a given English letter appears in an encrypted message, when a "secure" encryption scheme is used?

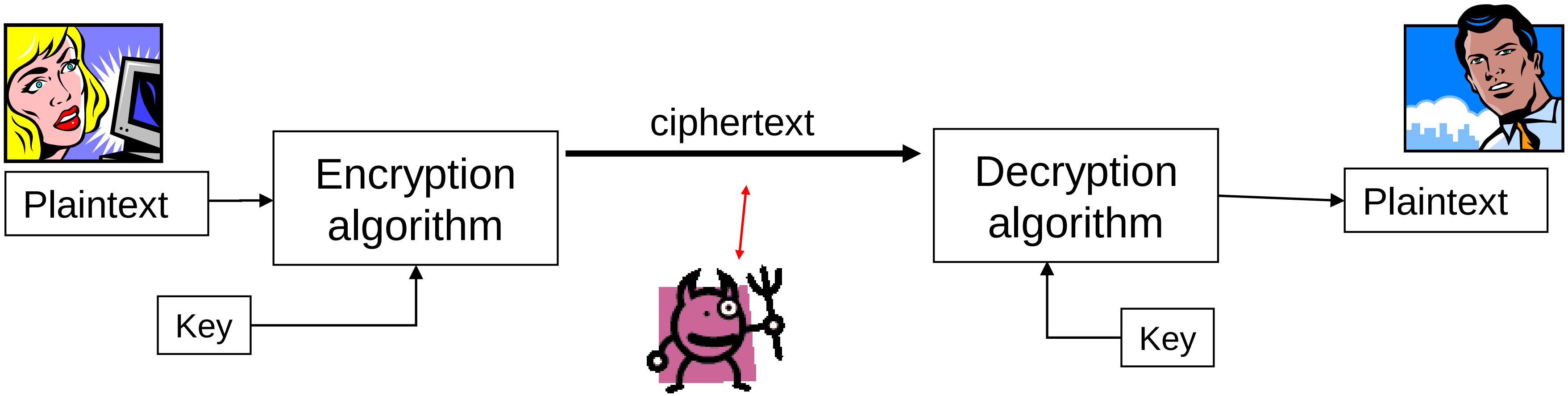
Summary of Lecture

- Cryptography: main problem + history
- Some Probability Background
- Some Background on Algorithms
- Classical Cryptography
- Perfect Secrecy for Encryption

Perfectly-secret encryption: introduction

- Historically speaking, we are on a transition from classical cryptography to modern cryptography
  - Some features of this area are in common with each one of the two
- Perfectly-secret encryption:
  - Was developed before the modern cryptography revolution of mid-70s and 80s
  - Can be based on precise mathematical definitions
  - Can be accompanied with rigorous proofs of properties
  - Has limitations that will be later overcome by modern cryptography methods
  - Will be useful to understand modern cryptography
  - Will provide useful tools for some modern cryptography solutions

Perfectly-secret encryption: setting and goal



- Setting:
  - Alice and Bob have computers that can run encryption and decryption algorithms
  - Alice and Bob somehow share a secret key k
- Goal: exchange messages that remain private against an adversary that can eavesdrop and run powerful algorithms

Syntax and requirements

- Syntax of perfectly-secret encryption:
  - Message space M
  - Key space K, ciphertext space C
  - Key-generation algorithm Gen
  - Encryption algorithm Enc
  - Decryption algorithm Dec
  - Gen is a probabilistic algorithm that generates a key k in K with some distribution
  - On input message m from M, and k from K, Enc returns ciphertext c in C
  - On input ciphertext c, and k from K, Dec returns message m' or a failure symbol
- Requirements:
  - (Perfect) Correctness: For any k,m,m', it holds that Prob [m'=m] = 1
  - (Perfect) Secrecy: for any m in M, and any c in C such that Prob[c] > 0, it holds that Prob [m|c] = Prob[m]

Perfect secrecy: intuition

- Recall: Prob [m|c] = Prob[m]
- Intuition behind perfect secrecy:
  - Consider an *a priori* probability distribution D_0 on the message space (i.e., defining the probability that a message is chosen for encrypted communication)
  - Consider an *a posteriori* probability distribution D_1 on the message space conditioned by the ciphertext (i.e., defining the probability that a message was chosen for encrypted communication given the ciphertext)
  - Perfect secrecy requires that D_0 and D_1 are identical $\rightarrow$ ciphertext does not change the probability of the original message

One equivalent formulation

- Theorem: An encryption scheme over message space M is perfectly secret if and only if for any distribution over M, any m in M and any ciphertext c with prob[c]>0, it holds that

$$\text{Prob}\,[c|m] = \text{Prob}[c]$$

- Proof:
  - Need to show two statements:
    - "if" statement: Prob [c|m] = Prob[c] implies Prob [m|c] = Prob[m] (original formulation) and
    - "only if" statement: viceversa; i.e., original formulation → new one
  - Assume Prob [c|m] = Prob[c], multiply both sides by Prob[m]/Prob[c], and obtain

$$\text{Prob}\,[c|m] * \text{Prob}[m] \,/\, \text{Prob}[c] = \text{Prob}[m]$$

  - By Bayes Theorem, lhs is = Prob[m|c] → perfect secrecy ("if" statement) follows
  - Viceversa ("only if" statement) is proved in essentially the same way

A second equivalent formulation (perfect indistinguishability)

- Theorem: A symmetric encryption scheme over message space M is perfectly secret if and only if for any distribution over M, any $m_{0,}$ m_1 in M and any ciphertext c with prob[c]>0, it holds that

$$\text{Prob}\ [c|m_0] = \text{Prob}[c|m_1]$$

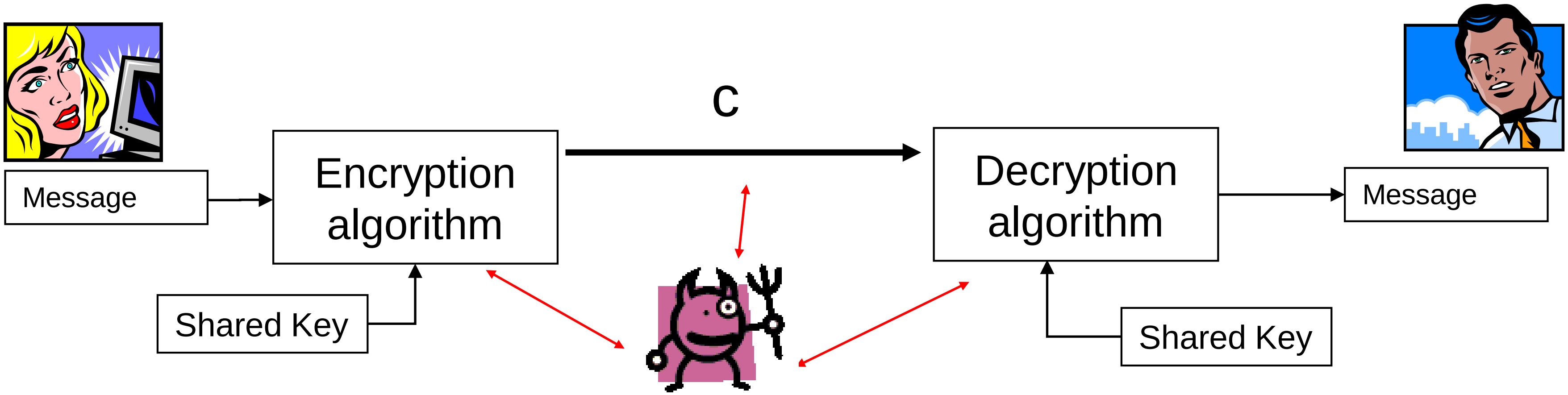
- Proof:
  - Need to show: Prob [m|c] = Prob[m] (original definition) implies Prob [c|m_0] = Prob[c|m_1] and viceversa
  - “only if” statement:
    - Use first equivalent formulation to obtain that perfect secrecy implies that Prob [c|m_b] = Prob[c] for b=0,1
    - Then observe that Prob [c|m_0] = Prob[c] = Prob[c|m_1]
  - See [KL] for other direction (“if” statement)
    - Intuition: rewrite Prob[c] via conditioning rule with Prob[m] and use again first equivalent formulation

A third equivalent formulation (adversarial indistinguishability)

- This formulation is based on a probability experiment involving an adversary A and formalizes A's (lack of) success in distinguishing the encryptions of two plaintexts
- Theorem: A symmetric encryption scheme over message space M is perfectly secret if and only if

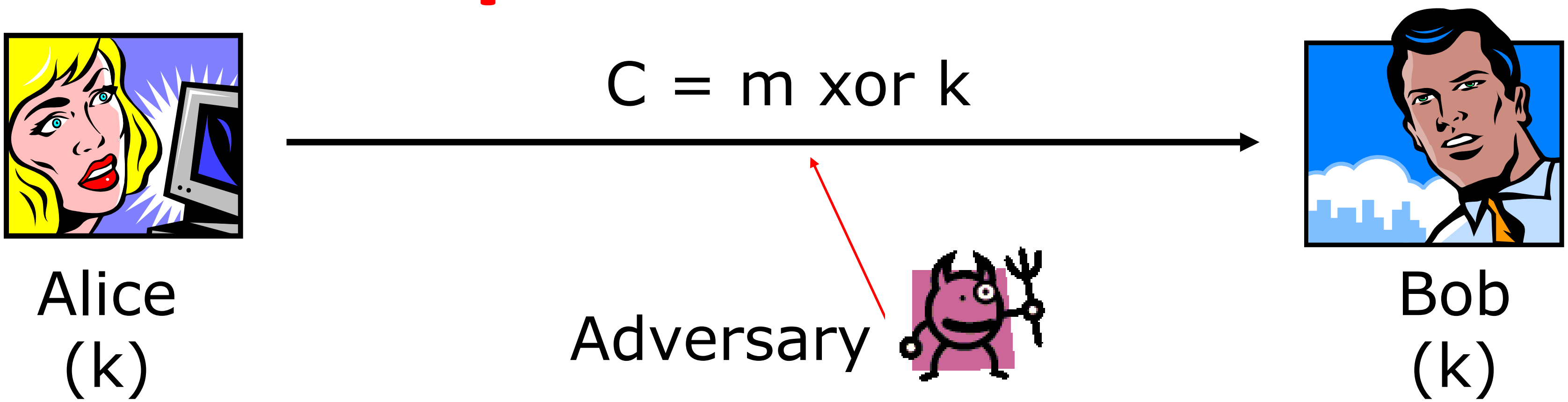
  Prob [adversary Adv succeeds in the following experiment] = 1/2
- Experiment:
  - Adv chooses two messages m_0,m_1 from M
  - Random key k is generated using Gen
  - Random bit b $\leftarrow$ {0,1} is chosen
  - Ciphertext c $\leftarrow$ Enc(k, m_b) is computed
  - Given c, Adv outputs b'
  - Adv succeeds if b'=b
- See [KL] for proof

Perfect secrecy: one time pad



- A fundamental cryptographic protocol
  - invented by Vernam in 1918
  - apparently used very often in real-life since then
- Main Properties:
  - Satisfies perfect secrecy
  - Has optimal key length (Shannon, 1948)
  - Has essentially optimal efficiency

One-time pad definition



- Definition
  - Message space, key space, ciphertext space are = $\{0,1\}^L$ for some integer L>0
  - Gen generates a random string k from the key space
  - To encrypt L-bit message m, Enc returns C = k XOR m
  - To decrypt ciphertext C, Dec returns m' = C XOR k
- Example
  - Key = 1011010001, message = 0000011111, ciphertext = 1011001110
- Properties
  - Key length = message length
  - Simple encryption and decryption operations
  - Xor properties: what is b xor b, b xor 0, b xor 1, for all values of bit b?

One-time pad secrecy

- Theorem:
  - the one-time pad encryption scheme satisfies perfect secrecy
- Proof:
  - We have that Prob[C=c | M=m] = Prob[M xor K = c | M=m]
  - Note that Prob[M xor K = c | M = m] = Prob[m xor K = c]
  - Then note that Prob[m xor K = c] = Prob[K = m xor c]
  - The latter is $1/2^L$
  - The above holds for any m, thus we obtain that

    $$\text{Prob}[c \mid m_0] = 1/2^L = \text{Prob}\,[c|m_1]\,,$$

    which implies perfect indistinguishability and thus implies perfect secrecy
- Limitations:
  - Key space is at least as large as message space
  - Secrecy does not hold if key is used more than once
    - e.g. from c_0 = k xor m_0 and c_1 = k xor m_1 one can easily find m_0 xor m_1
  - The Venona U.S. project used this fact to decrypt foreign communication during WWII and, among other things, uncover spies on US territory

Perfect secrecy limitations

- One-time pad limitation on the key length is inherent
- Theorem: any perfectly-secret encryption scheme with message space M and key space K satisfies |K|>=|M|
- Proof:
  - We prove the (equivalent) contrapositive statement:

    if |K|<|M| then scheme is not perfectly secret
  - Consider uniform distribution over M, let c be a ciphertext occurring with probability > 0, and define

    M(c) = {m' | m'=Dec(k',c) for some k' in K}
  - Note that |M(c)|<=|K|
  - Thus, if |K|<|M|, there is m'' in M such that m'' is not in M(c)
  - Then Prob[m'' | c] = 0 while Prob[m''] is not
  - Thus, there exists a value m'' in M for which the perfect secrecy condition does not hold

Shannon’s theorem

- Fundaments of perfect secrecy are due to a pioneering paper by C. Shannon
- He also provided a characterization of perfectly-secret encryption schemes
- Theorem: an encryption scheme (Gen,Enc,Dec) with message space M, key space K and ciphertext space C such that |M|=|K|=|C| is perfectly secret if and only if
  - Gen returns every key in K with probability 1/|K|
  - For every m in M and c in C, there is exactly one key k in K such that Enc(k,m)=c
- See [KL] for proof
- Significance:
  - (1) characterization of perfectly-secret encryption scheme
  - (2) tool for easily proving or disproving perfect secrecy: the first condition is easy to work with and the second does not involve probabilities, and none of them involves the probability distribution over M

Question set 2

- Consider the one-time pad encryption scheme for message space, key space and ciphertext space equal to {0,1}. Assume message m is chosen with uniform distribution from the message space and let c denote the ciphertext. Compute the following probabilities:
  - Prob[m = 0], Prob[c = 1|m = 0], Prob[c = 0], Prob[m = 0 and c = 0]
- Consider using the shift cipher on message space {A,…,Z}. Analyze the security of the resulting encryption scheme.
- Consider an extension of the shift cipher where a random and independent shift (taken from the shared key) is applied for each message character. Analyze the security of this encryption scheme.
- Can you define a modification of the poly-alphabetic substitution cipher (without changing the algorithms, and based on setting a single parameter) so that the resulting schemes satisfies perfect secrecy?
- Consider modifying the XOR operation in the one-time pad into another arbitrary operation and check if the resulting scheme satisfies decryption correctness and perfect secrecy

Class CS 6903, End of Lecture n. 1

Reference → Topic ↓	[KL]	[MOV]	[FSK]
Cryptography, main problem + history	1.1	1.4	1
Some probability background	A.1, A.3	2.1	
Some background on algorithms			
Classical cryptography	1.1, 1.2, 1.3	2.2, 7.3	
Encryption with perfect secrecy	2		
	See https://www.nytimes.com/books/99/05/09/reviews/990509.09issermt.html for a review on a book on the Venona project		