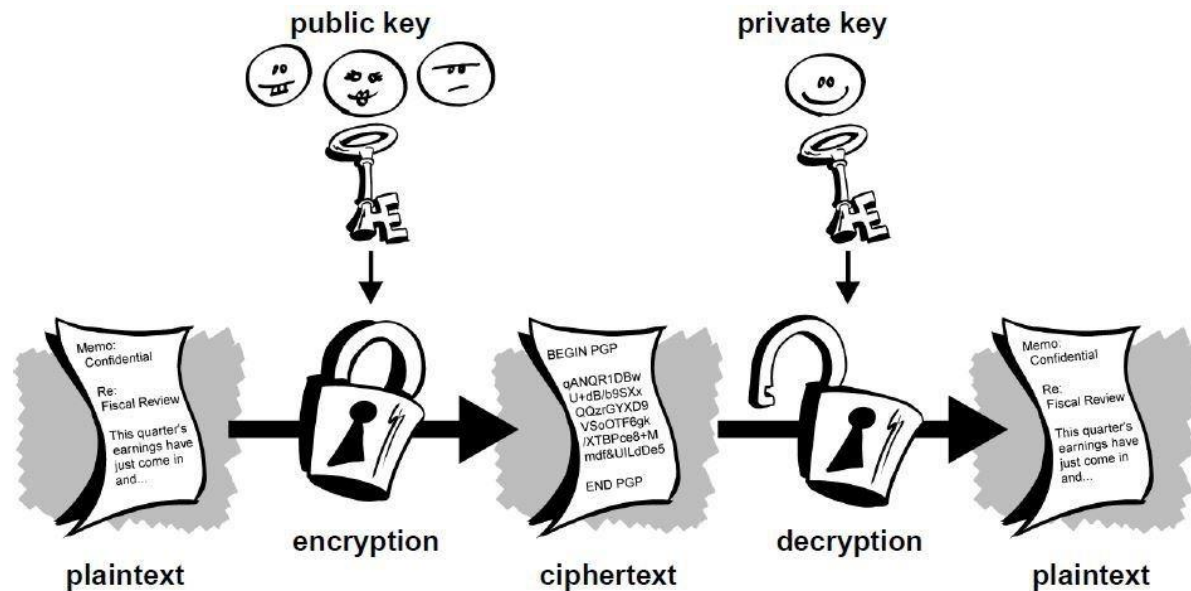


# AI and Machine Learning

## Vipul Goyal



# Some Basics

My Name: Vipul Goyal

Home page: <http://www.cs.cmu.edu/~goyal/>

Some resources for the class:

- Lecture notes from my (university) class at CMU:  
<http://www.cs.cmu.edu/~goyal/s18/15503.html>
- Katz and Lindell book: Introduction to Modern Cryptography
- Nice video lectures at coursera.org, lot of material on YouTube and Wikipedia

# Course Basics

- Lecture sessions every week
- Mute your mic during the lectures
- Slides will be provided immediately after the lecture
- Go through the slides on your own **before** the next class
- If you have any questions during the lectures, please type them in the chat window
- We will take periodic breaks and I will answers the typed questions

# Course Basics Contd..

- Students will be divided into groups later in the course
- Each group will take up a project. Examples:
  - Digital Signatures
  - RSA encryption
  - Bitcoin and cryptocurrencies
  - Applications of Blockchains to other areas
- At the end of the course: final presentation + final report by each group
- You will get a letter grade at the end of the course

# Goal of this Course

- Life becoming more digital
  - Crypto: defines the rules of digital world
  - Imagine a society without rules
- My goal: give you an overview of the very basic concepts. Get you interested. Prepare you to dive deeper on your own.

# Cryptography

- Cryptography and computer security: **very fast growing** topics within computer science
- Online attackers are growing in sophistication. New websites are hacked every month.
- **Severe shortage of talent in this area. Software companies simply cannot find enough people to hire.**
- Very active area of research as well. Exciting new things being invented all the time. Latest Revolution: Bitcoin, cryptocurrencies, ...

# First goal: Secret Communication



Adversary  
Eavesdropper

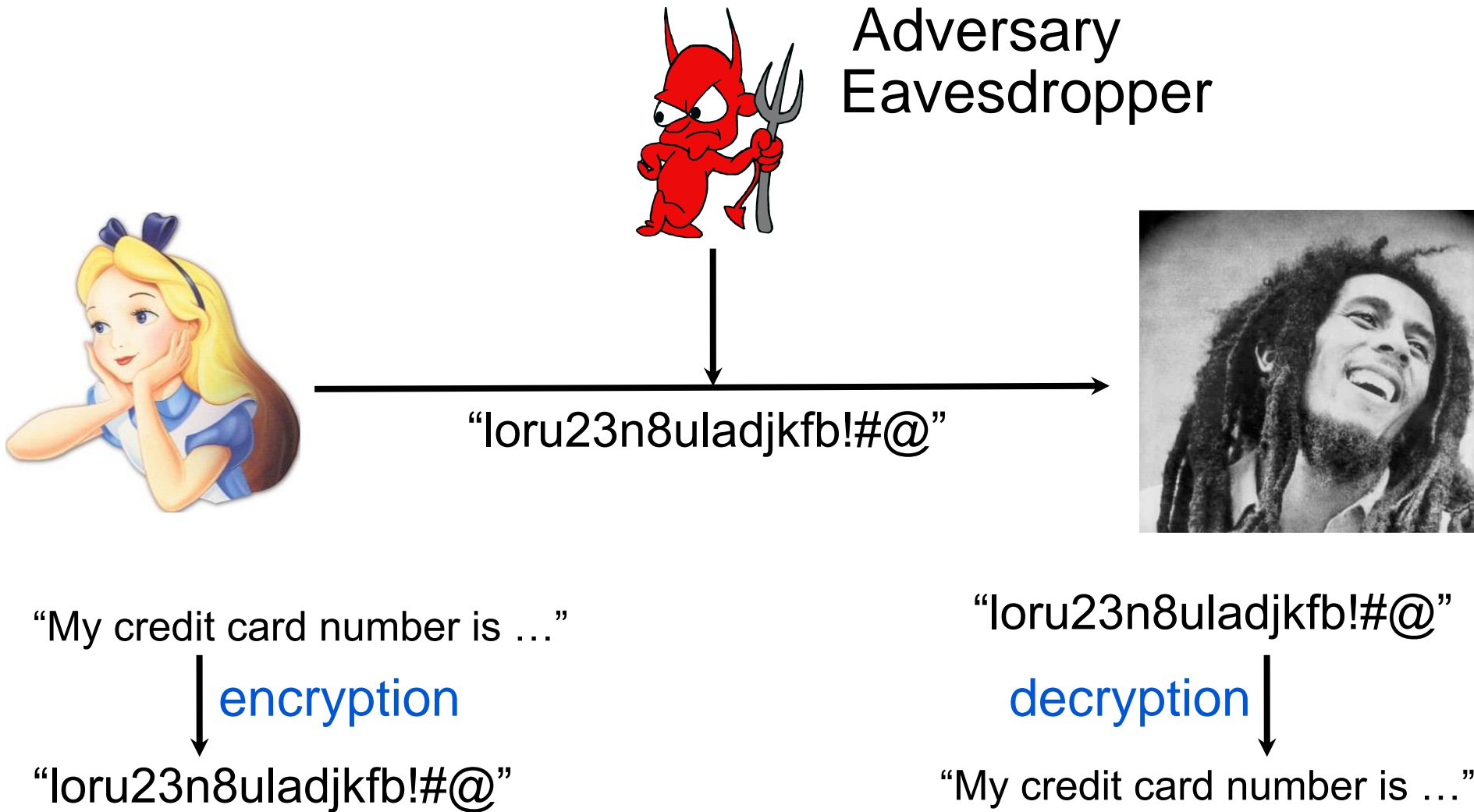


"My credit card number is ..."



"My credit card number is ..."

# First goal: Secret Communication





## Private Key Encryption (Ciphers)

# Private key cryptography



Parties must agree on a key pair beforehand.

# Private key cryptography

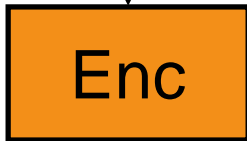
[illegible] $C$ 

***K***

 $M$  (plaintext)

***K***

***K, M***



$C$  (ciphertext)

[illegible] $K, C$ 
$$M$$

# A more Formal Definition

A secret key encryption (SKE) consists of 3 algorithms:

- 1) **Gen** called key generation algorithm. It doesn't take any input. You can run Gen and it outputs a key  $K$ . (many times,  $K$  will just be a large number)
- 2) **Enc** called the encryption algorithm. Enc takes a message  $m$ , and, a key  $K$  as input. Enc outputs a ciphertext  $C$
- 3) **Dec** called the decryption algorithm. Dec takes ciphertext  $C$ , and, a key  $K$  as input. Dec output the message  $m$ .

# A note about security

Better to consider worst-case conditions

Assume the adversary knows everything except the key(s)  
Adv might even know (part of) message (attack/defend)

Completely sees ciphertext  $C$

Completely knows the algorithms **Enc** and **Dec**

# History of Cipher Design

Is cryptography a branch of computer science?

YES and NO

- Computer Science = 50 year old
- Cryptography = 2000 year old
- Very useful in military conquests. Want to hide your strategy from the enemy.

# History of Cipher Design

- A large number of ciphers designed over the last 2000 years
- Oldest recorded cipher: Caesar Cipher
- All ciphers designed prior to 1950: classical ciphers
- Every classical cipher: broken

# Role of Encryption/Ciphers

- Very interesting movie: The Imitation Game (about WW2 and life of Alan Turing)
- Germans: using Enigma machine to encrypt their messages
- British tried but couldn't break using traditional pen and paper approaches
- Turing: built a huge machine which was capable of trying thousands of keys every minute



# Role of Encryption/Ciphers

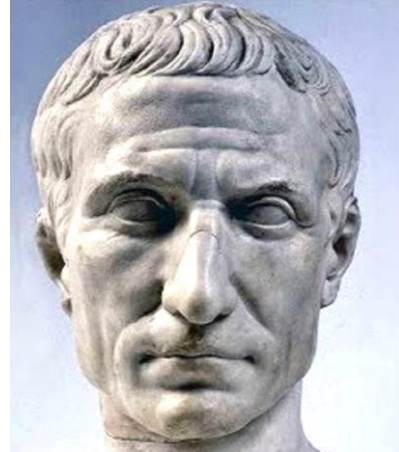
- This machine: can be thought of as first ever computer
- Estimates: Breaking the ciphers resulted in war becoming shorter by 18 months. 9 million lives saved!

Some Classical Ciphers (and why they are bad)

# Caesar cipher

Example: shift by 3

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c



(similarly for capital letters)

“Dear Math, please grow up and solve your own problems.”

↓  
“Ghdu Pdwk, sohdivh jurz xs dqg vroyh brxu rzq sureohpv.”

# Caesar cipher

Example: shift by 3

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

(similarly for capital letters)

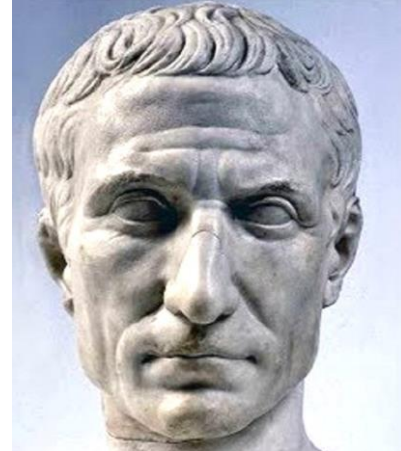


: the shift number

If key = 3:

To encrypt: simply shift forward each letter by 3

To decrypt: shift backward each letter by 3



# Breaking Caesar Cipher

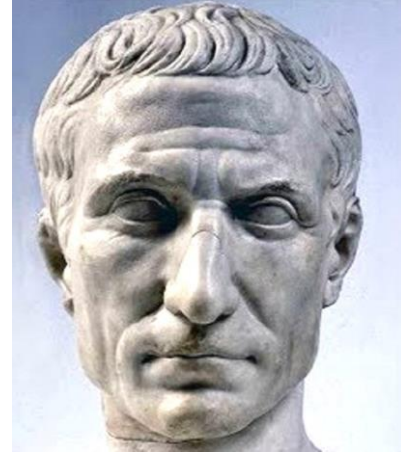
Total number of possible keys?

- Only 26

How to break:

- Adversary sees a ciphertext
- Tries every possible key
- Sees if the decrypted text makes sense

Moral of the story: keys should be large



# Other type of characters?

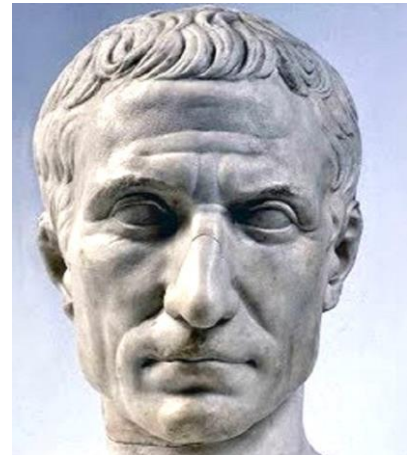
- What about space, comma, semi-colon?
- What about another language such as Chinese?

Basic principal remains the same:

- Write down all possible characters
- To encrypt, shift by a certain amount



: the shift number



# Substitution cipher



: a table mapping the input letter to an output letter

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
j	k	b	d	e	l	m	c	f	g	n	o	x	y	r	s	v	w	z	a	t	u	p	q	h	i

Say:

- message = acd
- Ciphertext = jbd

# Substitution cipher



: a table mapping the input letter to an output letter

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
j	k	b	d	e	l	m	c	f	g	n	o	x	y	r	s	v	w	z	a	t	u	p	q	h	i

To encrypt:


Replace each letter of the message by the corresponding letter from the table

To decrypt:

Simply read back what each letter means from the table



# Substitution cipher



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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
j	k	b	d	e	l	m	c	f	g	n	o	x	y	r	s	v	w	z	a	t	u	p	q	h	i

- Total number of keys:  $26!$  (where  $!$  = factorial function)

Why?

- First entry in table = 26 possible letters
- Second entry in table = 25 possible letters
- ....
- Total ways of constructing the table =  $26 \cdot 25 \cdot 24 \cdot \dots \cdot 1$
- This is too large! Hard to try every possible key

# Breaking Substitution cipher?



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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
j	k	b	d	e	l	m	c	f	g	n	o	x	y	r	s	v	w	z	a	t	u	p	q	h	i

- Still possible to break! No need to try all keys.
- Idea: frequency analysis
- Some letters in English more frequent than others
  - e = 12.5 %
  - t = 9.28 %
  - a = 8.04 %
  - ...

# Frequency Analysis



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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
j	k	b	d	e	l	m	c	f	g	n	o	x	y	r	s	v	w	z	a	t	u	p	q	h	i

- Some letters in English more frequent than others  
e = 12.5 %  
t = 9.28 %  
a = 8.04 % ...
- Now say in ciphertext, frequency of b = 12%  
=> This means e was mapped to b
- Repeating this: we can learn many entries of the table

# Frequency Analysis Contd..



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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
j	k	b	d	e	l	m	c	f	g	n	o	x	y	r	s	v	w	z	a	t	u	p	q	h	i

- Frequency of Bigrams  
th = 3.56 %  
in = 2.43 % ...
- If we see lot of rj in ciphertext  
=> t mapped to r, h mapped to j
- Frequency of double letters  
ee = 0.38  
oo = 0.21

# Frequency Analysis Contd..



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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
j	k	b	d	e	l	m	c	f	g	n	o	x	y	r	s	v	w	z	a	t	u	p	q	h	i

- Attack requires the adversary to collect a large amount of ciphertext. If you have encryption of a single sentence, not enough to break.
- Excellent post. Read as homework:  
<http://norvig.com/mayzner.html>

# Vigenere cipher



: a random string (say BAE)

## Encryption:

- Repeat key to make it equal to message length
- Shift each letter of message by the letter of the key
- Sometimes easier to write each character of the key as a number rather than letter

M = ATTACK

K = BAEBAE

C = CUYCDP

# Vigenere cipher



: a random string (say BAE)

## Decryption:

- Repeat key to make it equal to ciphertext length
- Shift **back** each letter of message by the letter of the key

C = CUYCDP

K = BAEBAE

M = ATTACK

# Vigenere cipher security?



: a random string (say BAE)

Total number of keys:  $26 \cdot 26 \cdot 26 \dots$   
 $= 26^n$  (assuming key has  $n$  letters)

- Could be huge if  $n$  is large (say  $n = 50$ )
- Hard to try all possible keys!
- Secure?



# Vigenere cipher (in)security



: a random string (say BAE)

Again: frequency analysis to the rescue

Step 1: guess length of the key (try again if fail). Say length = 3.

Step 2: divide ciphertext into 3 parts

CT = FWJUOIFDSLKFSLSLJ.....

First part = FUFLSL

Step 3: Apply frequency analysis separately on each part

# Vigenere cipher (in)security



: a random string (say BAE)

CT = FWJUOIFDSLKFSFSLJ.....

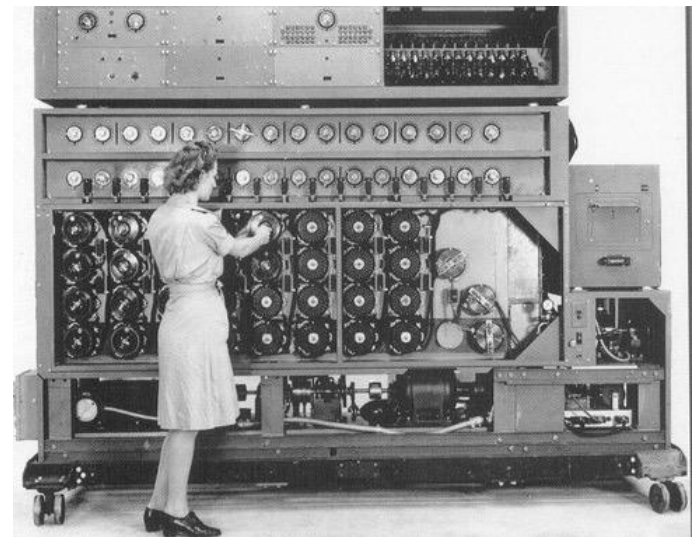
First part = FUFLSL

Step 3: Apply frequency analysis separately on each part

- **Observation**: each letter in first part is shifted by the same letter (i.e. by B)
- Say we notice frequency of G is about 12%
- Then: E was mapped to G
- **Hence: key = B (shift of 2)**
- Repeat to find each letter of the key

# Enigma (WW2)

A much more complex cipher: Still broken



Questions so far?