

EE 4215

Security Technology

Lecture 2: Classical Cryptography

2018/19 Semester B

Course Structure

- Week 1: Security Fundamental
- Week 2: Classical Cryptography (114 students)
- Week 3: Number Theory
- Week 4: Public-key Cryptography
- Week 5: Secret-key Cryptography
- **Week 5: Quiz 1**
- Week 6: Hashing Algorithm
- Week 7: PKI, SSL/TLS
- Week 8: Web Security
- Week 9: Wi-Fi & P2P Security
- Week 10: Cloud Computing Security
- **Week 10: Quiz 2**
- Week 11: Smart-card Security
- Week 12: Blockchain Security Technology
- Week 13: Secure-code, Trusted Hardware, Review

Lecture 2:

Classical Cryptography

- Terminology
- Classical Encryption Techniques
- Key Distribution
- Steganography

Definitions

Plaintext

- An original message

Ciphertext

- The coded message

Enciphering/encryption

- The process of converting from plaintext to ciphertext

Deciphering/decryption

- Restoring the plaintext from the ciphertext

Cryptography

- The area of study of the many schemes used for encryption

Cryptographic system/cipher

- A scheme

Cryptanalysis

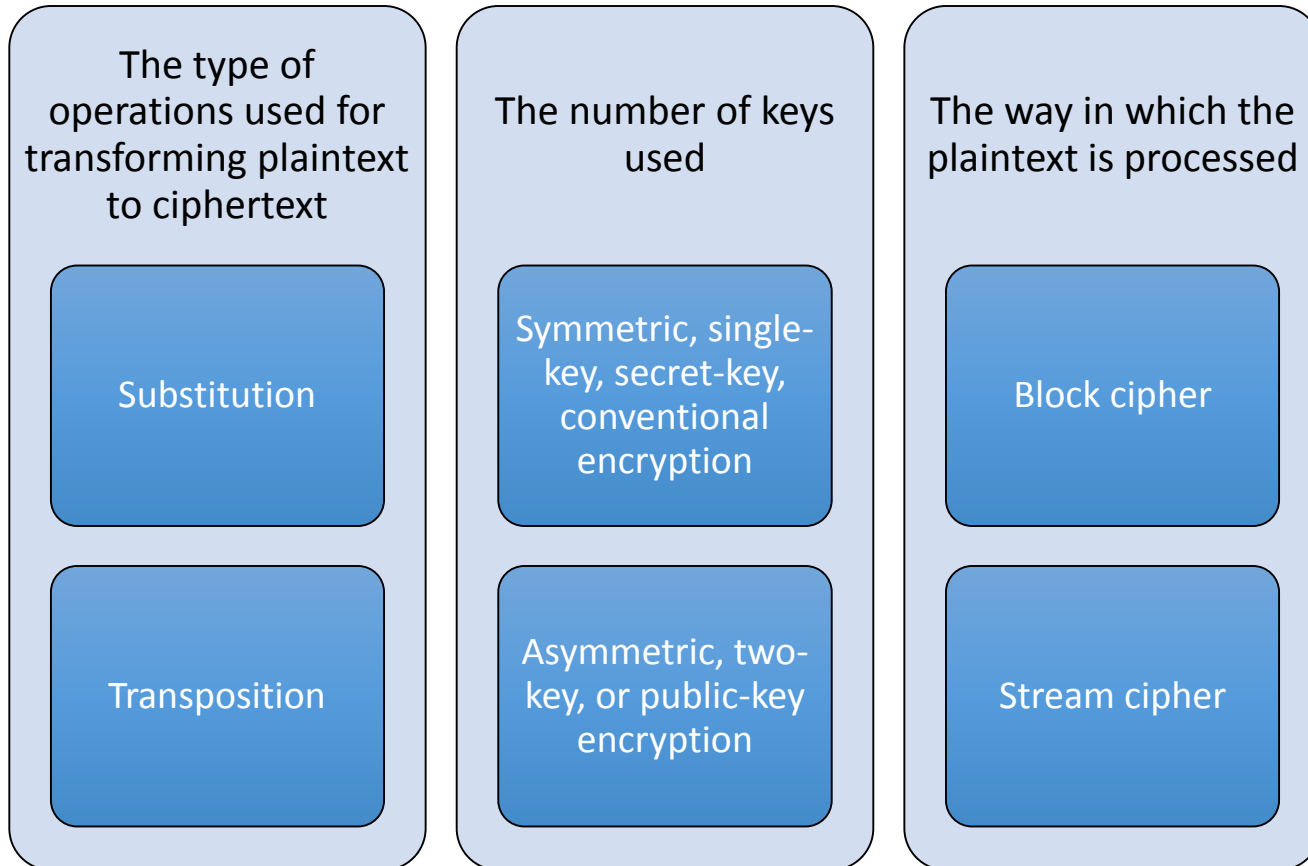
- Techniques used for deciphering a message without any knowledge of the enciphering details

Cryptology

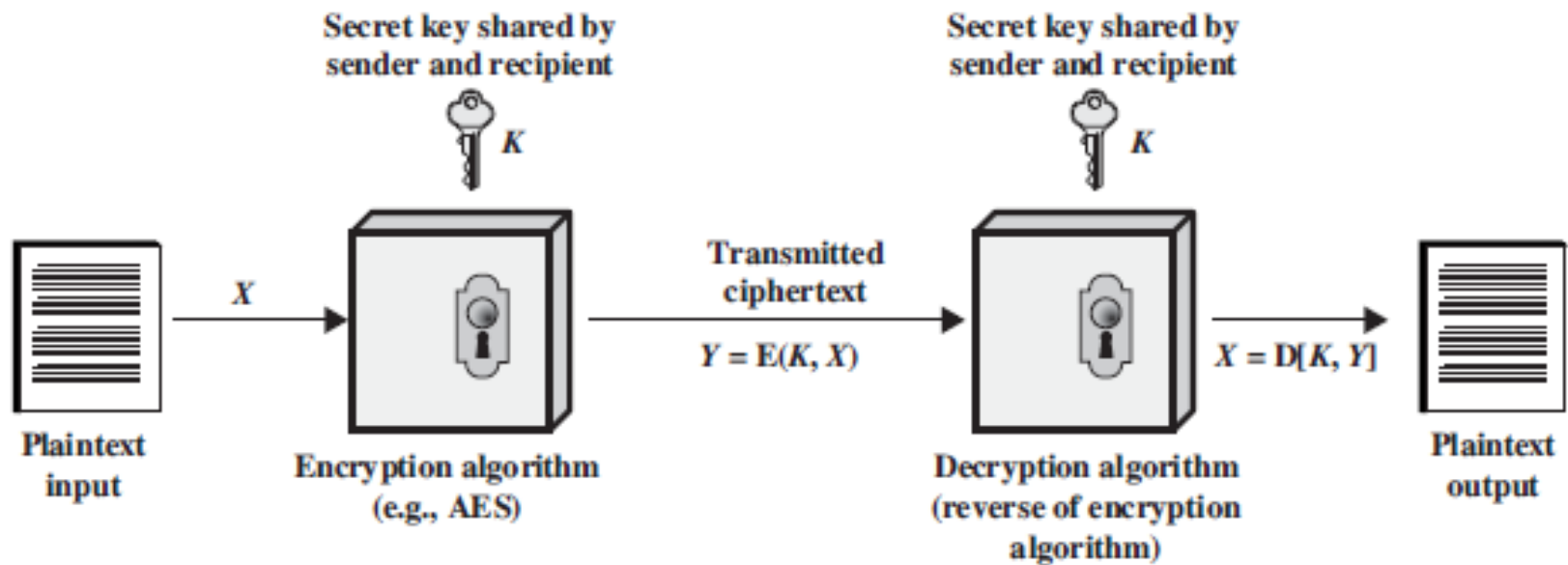
- The areas of cryptography and cryptanalysis

Cryptographic Systems

- Characterized along three independent dimensions:



Model of Symmetric Encryption



Some Basic Terminology

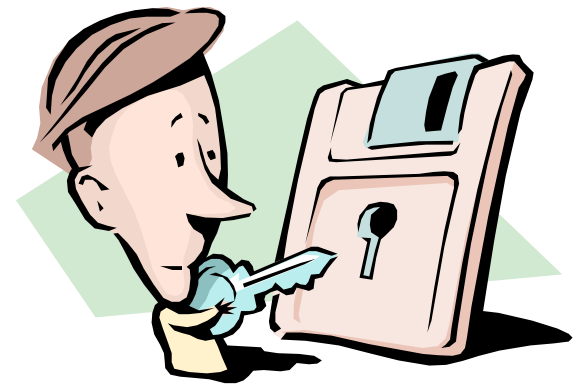
- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext into ciphertext
- **key** - info used in cipher known only to sender/receiver

- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext

- **cryptography** - study of encryption principles/methods
- **cryptanalysis (code-breaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - cryptography + cryptanalysis

Symmetric Cipher Model

- There are two requirements for secure use of conventional encryption:
 - A strong encryption algorithm
 - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure

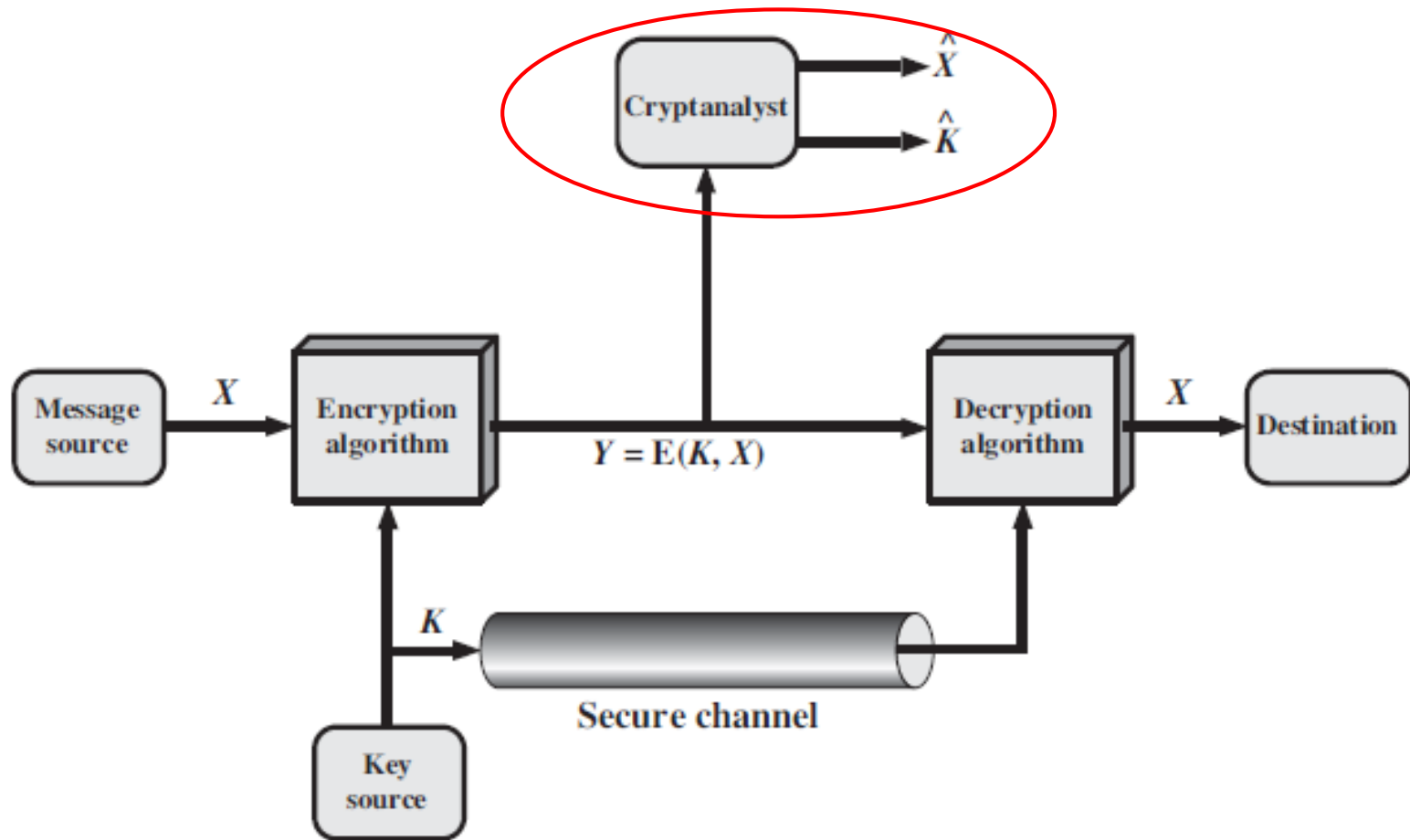


Cryptography



- characterize cryptographic systems by:
 - type of encryption operations used
 - substitution / transposition
 - product : combines two or more transformations
 - number of keys used
 - single-key or private (symmetric) / two-key or public (asymmetric)
 - way in which plaintext is processed
 - block / stream

Cryptanalysis 密碼分析



Cryptanalysis

- objective to recover key not just message
- general approaches:
 - cryptanalytic attack
 - brute-force attack



Cryptanalytic Attacks



Type of Attack	Known to Cryptanalyst
Ciphertext Only	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

More Definitions

- unconditional security
 - no matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- computational security
 - given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken

Cryptanalysis and Brute-Force Attack

Cryptanalysis

- Attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext
- Attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used

Brute-force attack

- Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained
- On average, half of all possible keys must be tried to achieve success

Brute Force Search 蠻力

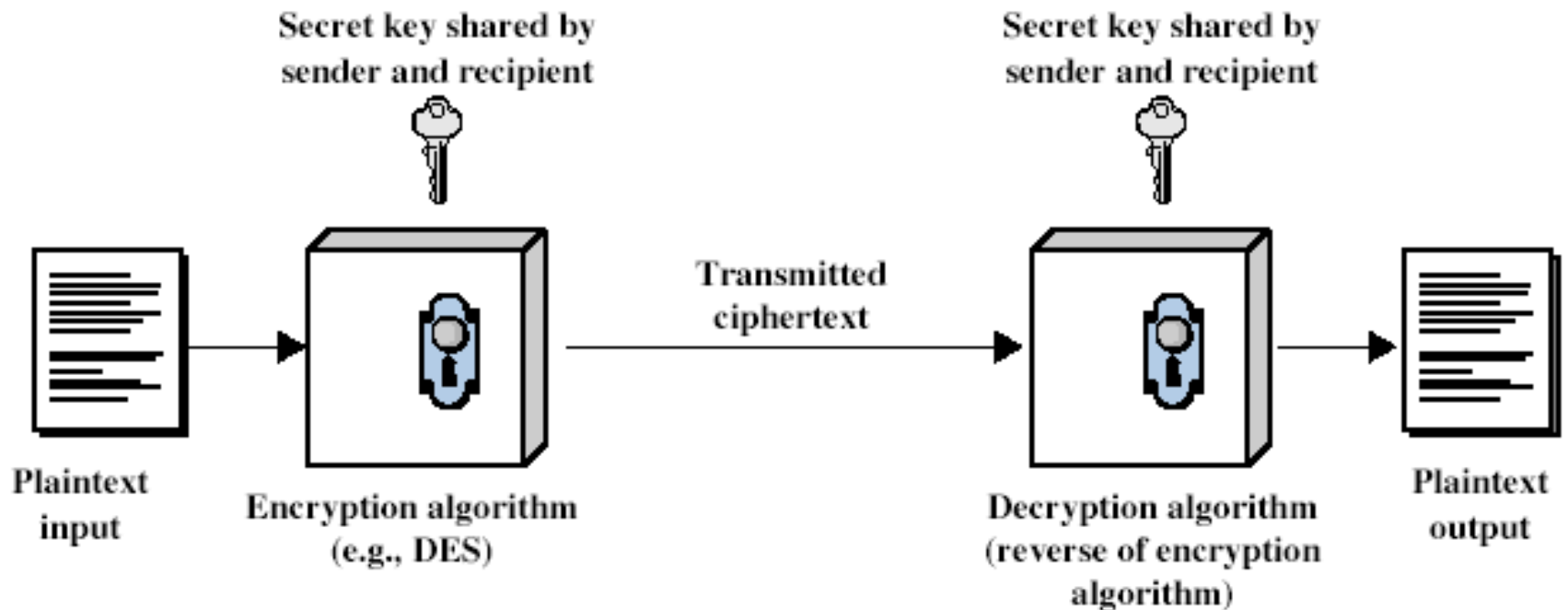
- try every single key
- most basic attack, proportional to key size
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/ μ s	Time required at 10^6 decryptions/ μ s
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{s} = 35.8 \text{ minutes}$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{s} = 1142 \text{ years}$	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{s} = 5.4 \times 10^{24} \text{ years}$	$5.4 \times 10^{18} \text{ years}$
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36} \text{ years}$	$5.9 \times 10^{30} \text{ years}$
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12} \text{ years}$	$6.4 \times 10^6 \text{ years}$

Symmetric Encryption

- aka conventional / private-key / single-key encryption
- all classical encryption algorithms are secret key based
 - classical means “prior to invention of public-key encryption” in 1970’s
- and by far most widely used

Symmetric Cipher Model



Symmetric Cryptosystem

- A cryptosystem is pair of algorithms that take a key and convert plaintext to ciphertext and back
- Symmetric cryptosystem
 - sender and recipient share a **common secret key**
- Mathematically:
$$Y = E_K(X)$$
$$X = D_K(Y)$$

Symmetric Cryptosystem

- Two requirements:
 - a strong encryption algorithm
 - a secret key known **only** to sender and receiver
- One problem
 - how to securely agree with the common key? Need a secure channel to distribute the common key

Encryption Algorithms

- Encryption algorithm is assumed to be known
 - **Security through obscurity** : attempts to use secrecy of design or implementation to provide security
 - “Security through obscurity” is not a good tactics
 - Kerckhoffs's principle (19th century):
 - A cryptosystem should be secure even if everything about the system, except the key, is known
 - Security depends on the secrecy of the key, not the secrecy of the algorithm

Lecture 2:

Classical Cryptography

- Terminology
- Classical Encryption Techniques
- Key Distribution
- Steganography

Classical ciphers

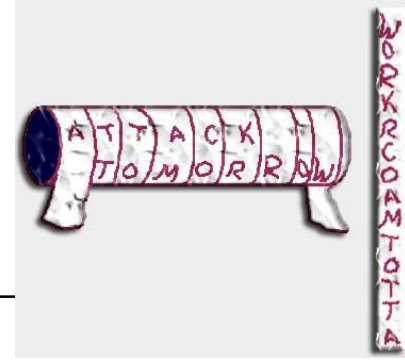
Converting plain text to cipher text

- **Substitution**
 - Substitute letters of the plain text with other letters/symbol
 - Mono/polyalphabetic ciphers
 - One-time pad
- **Transposition**
 - Rearrange plaintext
- Combo of both transposition and substitution

Substitution

- letters of plaintext are replaced by other letters or by numbers or symbols
- **monoalphabetic** unique mapping of plaintext alphabet to ciphertext alphabet
 - e.g. Caesar, Hill, Playfair
- **polyalphabetic** plaintext mapped to ciphertext based on key to select alphabet
 - e.g. Vigenere, enigma
- **stream** keystream is generated and used to map plaintext to ciphertext
 - e.g. one-time pad

Transposition



- re-arrange characters (permute)
- plain text letters re-arranged, arrays or rail-fence

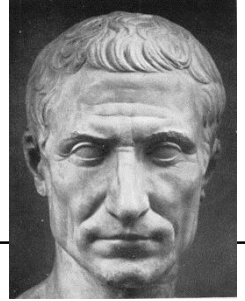
T	H	I	S	A	S
E	C	R	E	T	M
E	S	S	A	G	E

teehcsirsseaatgsme

Others

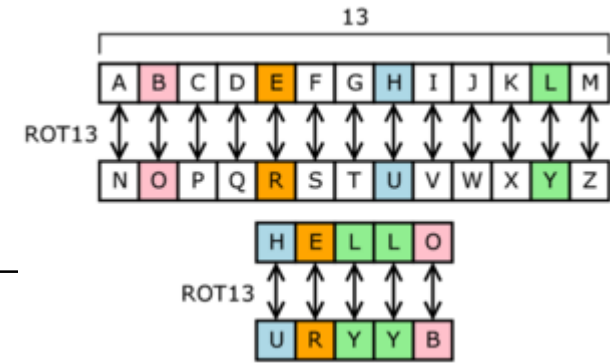
- **product cipher**: combining two or more simple transposition or substitution operations, hoping to result a more secure encryption
- **Super-encryption**: an encryption operation for which the plaintext input to be transformed is the ciphertext output of a previous encryption operation
 - encrypting encrypted information

Caesar Cipher



- earliest known substitution cipher
- by Julius Caesar
 - (13 July 100 BC – 15 March 44 BC)
- first attested use in military affairs
- replaces each letter by 3rd letter on
 - or key = 3
- example:
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB

Caesar Cipher



- can define transformation as:

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

- mathematically give each letter a number

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

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

- then have Caesar cipher as:

$$c = E(p) = (p + k) \bmod (26)$$

$$p = D(c) = (c - k) \bmod (26)$$

Cryptanalysis of Caesar Cipher

- only have 26 possible ciphers
 - A maps to A,
 - A maps to B, ...
 - A maps to Z
- could simply try each in turn (a **brute force search**)
 - given ciphertext, just try all shifts of letters
 - do need to recognize when have plaintext
 - eg. break ciphertext "GCUA VQ DTGCM"

Caesar Cipher Algorithm

- Can define transformation as:

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

- Mathematically give each letter a number

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- Algorithm can be expressed as:

$$c = E(3, p) = (p + 3) \bmod (26)$$

- A shift may be of any amount, so that the general Caesar algorithm is:

$$C = E(k, p) = (p + k) \bmod 26$$

- Where k takes on a value in the range 1 to 25; the decryption algorithm is simply:

$$p = D(k, C) = (C - k) \bmod 26$$

Brute-Force Cryptanalysis of Caesar Cipher

	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
KEY						
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrp	rfe	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	objv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlg
12	dvvk	dv	rwkvi	kyv	kfxr	grikp
13	cuuj	cu	qvjuh	jxu	jewq	fqhjo
14	btti	bt	putg	iwt	idvp	epgin
15	assh	as	othsf	hvs	hcuo	dofhm
16	zrrg	zr	nsgre	gur	gbtn	cnegl
17	yqqf	yq	mrfqd	ftq	fasm	bmdfk
18	xppe	xp	lqepc	esp	ezrl	alcej
19	wood	wo	kpdob	dro	dyqk	zkbdi
20	vnn	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxc

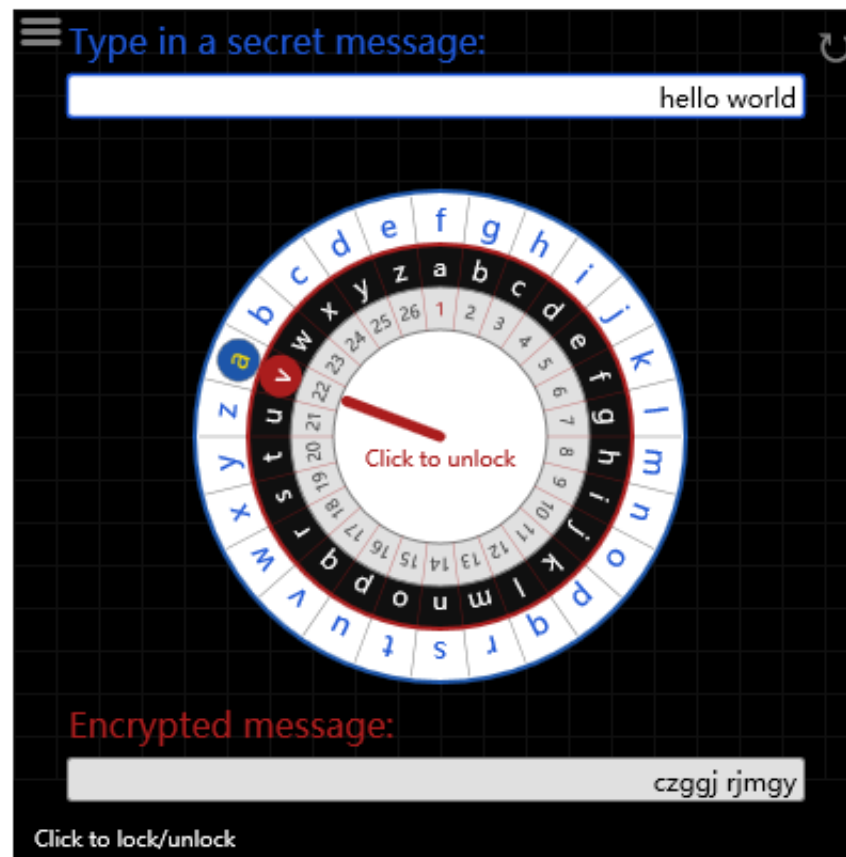
Caesar Cipher Decoder Ring



Caesar Cipher Ring

- <https://www.khanacademy.org/computing/computer-science/cryptography/crypt/p/caesar-cipher-exploration>

Caesar Cipher Exploration



Bute-forcing Caesar Cipher

- Maximum 25 keys to try for decryption

KEY	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrpc	rfc	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	objv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewul	ew	svlwi	lzw	lows	bsila

Online Tool

- E.g.

http://www.simonsingh.net/The_Black_Chamber/caesar.html

- Can you break the Caesar code "GCUA VQ DTGCM"?

Sage Python Exercise

- How can we implement Caesar Cipher using Sage?
- Here are the helper functions.

```
▼
1  alpha = "abcdefghijklmnopqrstuvwxyz"
2
3  def is_alpha_char(c):
4      return (c.lower() in alpha)
5
6  def char_to_num(c):
7      return alpha.index(c.lower())
8
9  def num_to_char(x):
10     return alpha[x % 26]
```

Sage – Caesar Cipher

```
def CaesarEncrypt(k, plaintext):
    ciphertext = ""
    for j in xrange(len(plaintext)):
        p = plaintext[j]
        if is_alpha_char(p):
            x = (k + char_to_num(p)) % 26
            c = num_to_char(x)
        else:
            c = p
        ciphertext += c
    return ciphertext

def CaesarDecrypt(k, ciphertext):
    plaintext = ""
    for j in xrange(len(ciphertext)):
        c = ciphertext[j]
        if is_alpha_char(c):
            x = (char_to_num(c) - k) % 26
            p = num_to_char(x)
        else:
            p = c
        plaintext += p
    return plaintext
```

```
CaesarEncrypt(3, "Hello")
CaesarDecrypt(3, "khoor")
```

```
CaesarEncrypt(5, "Hello")
CaesarDecrypt(5, "mjqqt")
```

```
'khoor'
'hello'
'mjqqt'
'hello'
```

Monoalphabetic Cipher

- rather than just shifting the alphabet
- could shuffle (jumble) the letters arbitrarily
- each plaintext letter maps to a different random ciphertext letter
- hence key is 26 letters long

Plain: ab cd efg h i j k l mno pq r s t
uvw x yz

Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext: If **w**e **I**sh **t**o replace **l**etters

Ciphertext: WI **R**FRWAJ **U**HYFTSDVFSFUUFYA

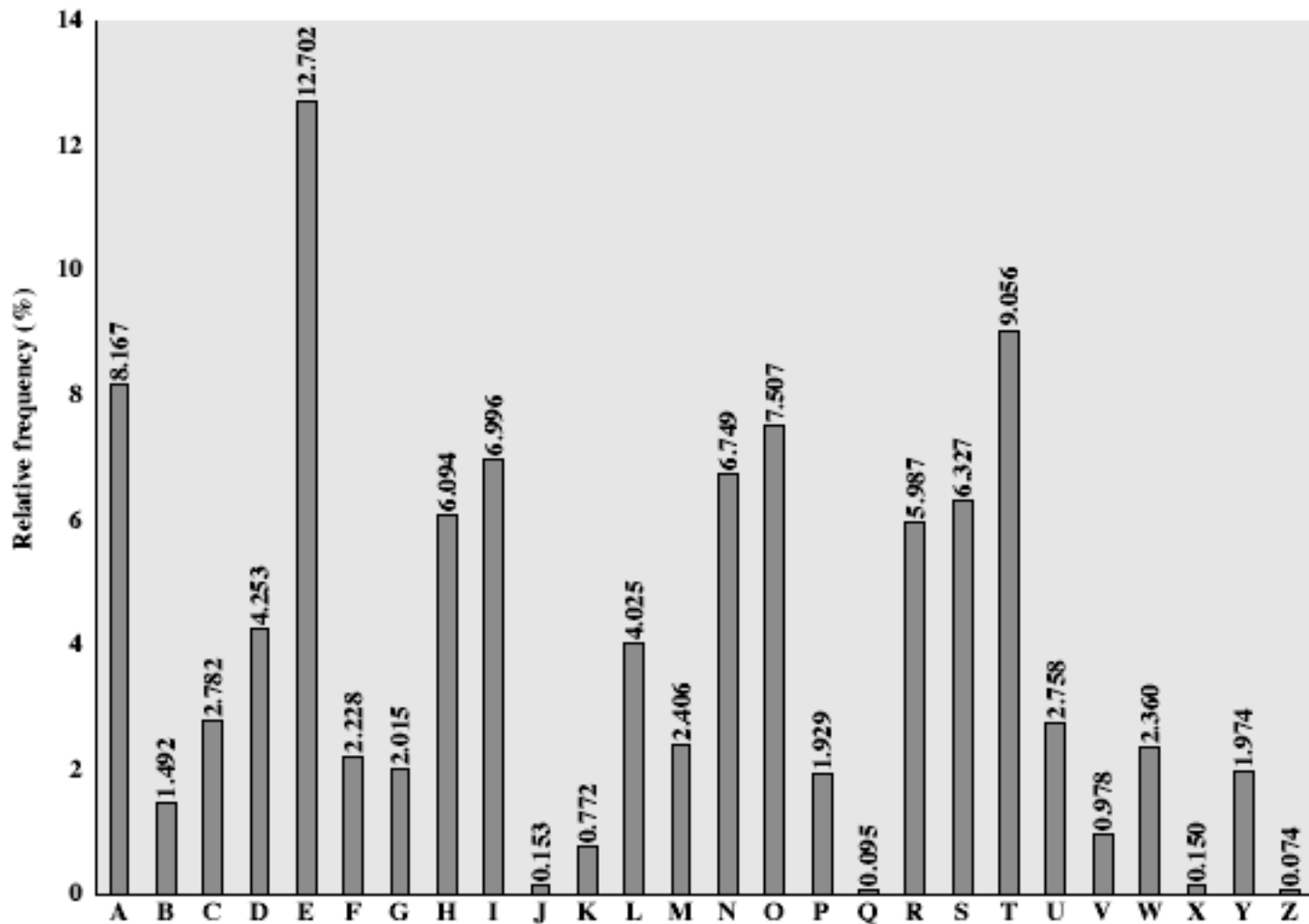
Monoalphabetic Cipher Security

- now have a total of $26! \approx 4 \times 10^{26}$ keys
- with so many keys, might think is secure
- but would be **!!!WRONG!!!**
- cryptanalysis can exploit the language characteristics in redundancy

Language Redundancy and Cryptanalysis

- human languages are **redundant**
- letters are not equally distributed
- in English **E** is by far the most common letter
 - followed by **T**, **A**, **O**, ...
- other letters like **Z**, **J**, **K**, **Q**, **X** are fairly rare
- there are tables of single, double & triple letter frequencies for various languages
- E.g. *<http://www.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html>*

English Letter Frequencies



English Letter Frequencies

- Common Bigrams: **th, he, in, er, an, re, nd, on, ...**
- Common Trigrams: **the, and, ing, her, hat, his, tha, ere, for, ent**
- Common Quadgrams: **that, ther, with, tion, here, ould, ight, have, hich**
- Reference *<http://norvig.com/mayzner.html>*
- Online tool: *<http://practicalcryptography.com/cryptanalysis/text-characterisation/monogram-bigram-and-trigram-frequency-counts/>*

Use in Cryptanalysis

- key concept - **monoalphabetic substitution ciphers do not change relative letter frequencies**
- discovered by Arabian scientists in the 9th century
- calculate letter frequencies for ciphertext
- compare counts/plots against known values
- cryptanalysis of monoalphabetic ciphers: must identify **each** letter
 - tables of letter frequencies, and common double/triple letters help

Example Cryptanalysis

- given ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- count relative letter frequencies
- guess **P** & **Z** are **e** and **t** respectively
- guess **ZW** is **th** and hence **ZWP** is **the**
- proceeding with trial and error finally get:

it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in moscow

Reading Assignment

- Some on-line tool might help the process
 - *http://cryptoclub.math.uic.edu/substitutioncipher/frequency_txt.htm*
- Please see a more complete example at *<http://abrickshort.wordpress.com/2005/08/04/deciphering-monoalphabetic-substitution-cipher/>*
- Are you able to write a program to assist the cryptanalysis process?

Java Code Reference

- <https://codereview.stackexchange.com/questions/104863/monoalphabetic-cipher-in-java-with-gui>

Monoalphabetic Cipher in Java With GUI



An easy, terribly written but working, Caeasar (Monoalphabetic) cipher implementation, with a GUI (Graphical User Interface).

2

Cipher.java



```
class Cipher {  
  
    static final String alphabet = "abcdefghijklmnopqrstuvwxyz";  
  
    public static char shift(char ch, int amount) {  
        return alphabet.charAt((alphabet.indexOf(ch) + amount) % alphabet.length());  
    }  
  
    public static String caesarCipher(String text, int amount) {  
        String result = "";  
        for (char ch : text.toCharArray()) {  
            if (alphabet.indexOf(ch) >= 0) {  
                result += shift(ch, amount);  
            }  
        }  
        return result;  
    }  
  
    public static void main(String[] args) {  
        assert shift('a', 1) == 'b';  
        assert shift('z', 2) == 'b';  
        assert caesarCipher("abc", 2).equals("cde");  
    }  
}
```

Gui.java

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
public class Gui {
```

Playfair Cipher

- Large number of keys in a monoalphabetic cipher cannot provide security
- How to improve?
- One approach is to substitute multiple letters
- **Playfair Cipher** is an example
- invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair

Playfair Key Matrix

- 26 letters are filled into a 5x5 matrix of letters starting with a keyword
 - letters I and J count as one letter
 - fill in letters of keyword (ignore duplicates)
 - fill rest of matrix with other letters
- eg. using the keyword **MONARCHY**

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

Encryption

➤ Plaintext:

- MEET PRIME MINISTER AFTER DINNER

➤ Step 1: Prepare the plaintext

- rewrite the plaintext message in pairs
 - ME ET PR IM EM IN IS TE RA FT ER DI NN ER
- all J are replaced with I
- double letters occurring in a pair must be divided by an X or a Z
 - e.g. NN becomes NXN
 - ME ET PR IM EM IN IS TE RA FT ER DI NX NE R
- if there are an odd number of letters in the plaintext, an extra letter chosen by the person writing the cipher is added to the end
 - ME ET PR IM EM IN IS TE RA FT ER DI NX NE RX

Encryption

➤ Prepared plaintext:

- ME ET PR IM EM IN IS TE RA FT ER DI NX NE RX

➤ Step 2: choose a key, e.g. MONARCHY and create the key matrix:

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

Encrypting

- Step 3: plaintext is encrypted **two letters** at a time
 - a. if both letters fall in the same row in the key matrix, replace each with letter to right (wrapping back to start from end)
 - b. if both letters fall in the same column in the key matrix, replace each with the letter below it (again wrapping to top from bottom)
 - c. otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair

Encryption

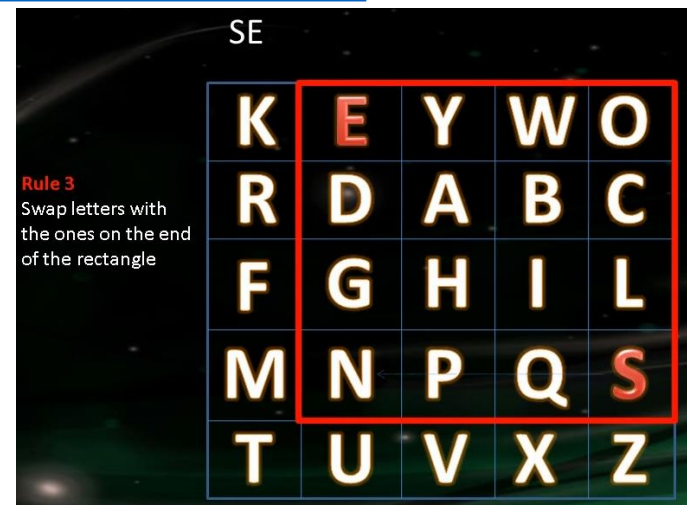
➤ Step 3: encrypt

- ME ET PR IM EM IN IS TE RA FT ER DI NX NE RX
- ME → same column → CL (rule b)
- ET → form a rectangle → KL (rule c)
- ...
- RA → same row → MR (rule a)

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

Encrypting

- Step 3: eventually the result becomes
CL KL TO EA LC GA SX LK MR KP KM BK AW MG AZ
- Online tool that can help:
 - <http://www.crosswordman.com/cgi-bin/playfair>
 - *Example video:*
<https://www.youtube.com/watch?v=quKhvu2tPy8>



Decryption

- Step 1: Break the ciphertext into pairs of letters
- Step 2: Create the key matrix
- Step 3: Transform the pairs of letters in the opposite direction from that used for encryption
- Exercise: decrypt **CL KL TO EA LC GA SX LK MR KP KM BK AW MG AZ** using the key **MONARCHY**

Security of Playfair Cipher

- security is much improved over monoalphabetic since there are $26 \times 26 = 676$ digrams
- would need a 676 entry frequency table to analyse (verses 26 for monoalphabetic)
- was widely used for many years
 - eg. by US & British military in WW1
- it **can** be broken, given a few hundred letters since it still has much of plaintext structure

Playfair Cipher in Java

- https://rosettacode.org/wiki/Playfair_cipher#Java

Java [edit]

```
import java.awt.Point;
import java.util.Scanner;

public class PlayfairCipher {
    private static char[][] charTable;
    private static Point[] positions;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        String key = prompt("Enter an encryption key (min length 6): ", sc, 6);
        String txt = prompt("Enter the message: ", sc, 1);
        String jti = prompt("Replace J with I? y/n: ", sc, 1);

        boolean changeJtoI = jti.equalsIgnoreCase("y");

        createTable(key, changeJtoI);

        String enc = encode(prepareText(txt, changeJtoI));

        System.out.printf("\nEncoded message: %s\n", enc);
        System.out.printf("\nDecoded message: %s\n", decode(enc));
    }

    private static String prompt(String promptText, Scanner sc, int minLen) {
        String s;
        do {
            System.out.print(promptText);
            s = sc.nextLine().trim();
        } while (s.length() < minLen);
        return s;
    }
}
```

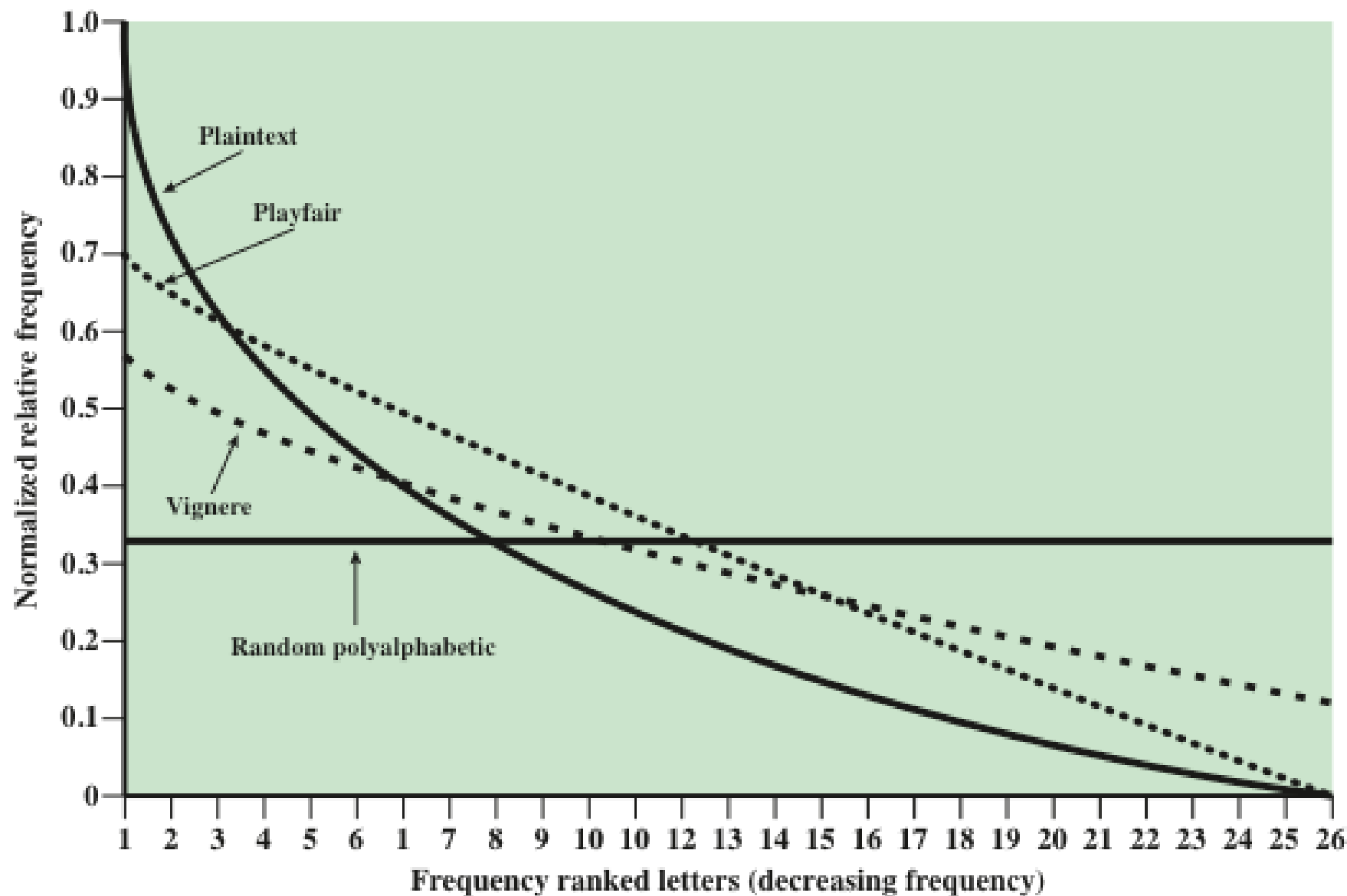


Figure 3.6 Relative Frequency of Occurrence of Letters

Polyalphabetic Substitution Ciphers

- improve security using multiple cipher alphabets
 - make cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- Write the key across the top of the plaintext, repeating it as many times as necessary
- use the key alphabet to determine which alphabet is used to substitute for each letter of the message

Vigenère Cipher

- simplest polyalphabetic substitution cipher
 - effectively multiple Caesar ciphers
- key is multiple letters long $K = k_1 k_2 \dots k_d$
 - Write the key across the top of the plaintext, repeating it as many times as necessary
- i^{th} letter in the key specifies the plaintext letter should be substituted with the corresponding i^{th} Caesar letter
- decryption simply works in reverse

Example of Vigenère Cipher

- write down the plaintext
- write the keyword repeatedly above the plaintext
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

key: *deceptive* *deceptive* *deceptive*

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

- online tool: <http://rumkin.com/tools/cipher/vigenere-keyed.php>

Vigenère Table

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

Slide rule ciphers

St Cyr 1880
key: position

ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ



Security of Vigenère Ciphers

- have multiple ciphertext letters for each plaintext letter
- hence letter frequencies are obscured
- but not totally lost
- start with letter frequencies
 - see if look monoalphabetic or not
- if not, then need to determine number of alphabets in the key first

Vigenere Ciphers in Haskell

- https://rosettacode.org/wiki/Vigen%C3%A8re_cipher#Haskell

Haskell [edit]

```
import Data.Char
import Text.Printf

-- Perform encryption or decryption, depending on f.
crypt f key = map toLetter . zipWith f (cycle key)
  where toLetter = chr . (+) (ord 'A')

-- Encrypt or decrypt one letter.
enc k c = (ord k + ord c) `mod` 26
dec k c = (ord c - ord k) `mod` 26

-- Given a key, encrypt or decrypt an input string.
encrypt = crypt enc
decrypt = crypt dec

-- Convert a string to have only upper case letters.
convert = map toUpper . filter isLetter

main = do
  let key = "VIGENERECIPHER"
      text = "Beware the Jabberwock, my son! The jaws that bite, "
          ++ "the claws that catch!"
      encr = encrypt key $ convert text
      decr = decrypt key encr
  printf "    Input: %s\n    Key: %s\nEncrypted: %s\nDecrypted: %s\n"
    text key encr decr
```

Output:

```
    Input: Beware the Jabberwock, my son! The jaws that bite, the claws that catch!
    Key: VIGENERECIPHER
Encrypted: WMCEEIKLGRPIFVMEUGXQPWQVIOIAVEYXUEKFKBTALVXTGAFXEYVKPAGY
Decrypted: BEWARETHEJABBERWOCKMYSONTHEJAWSTHATBITETHECLAWSTHATCATCH
```

Rotor Machines

- before modern ciphers, rotor machines were most common complex ciphers in use
- widely used in WW2
 - German Enigma, Allied Hagelin, Japanese Purple
- implemented a very complex, varying substitution cipher
- used a series of cylinders, each giving one substitution, which rotated and changed after each letter was encrypted
- with 3 cylinders have $26^3=17576$ alphabets

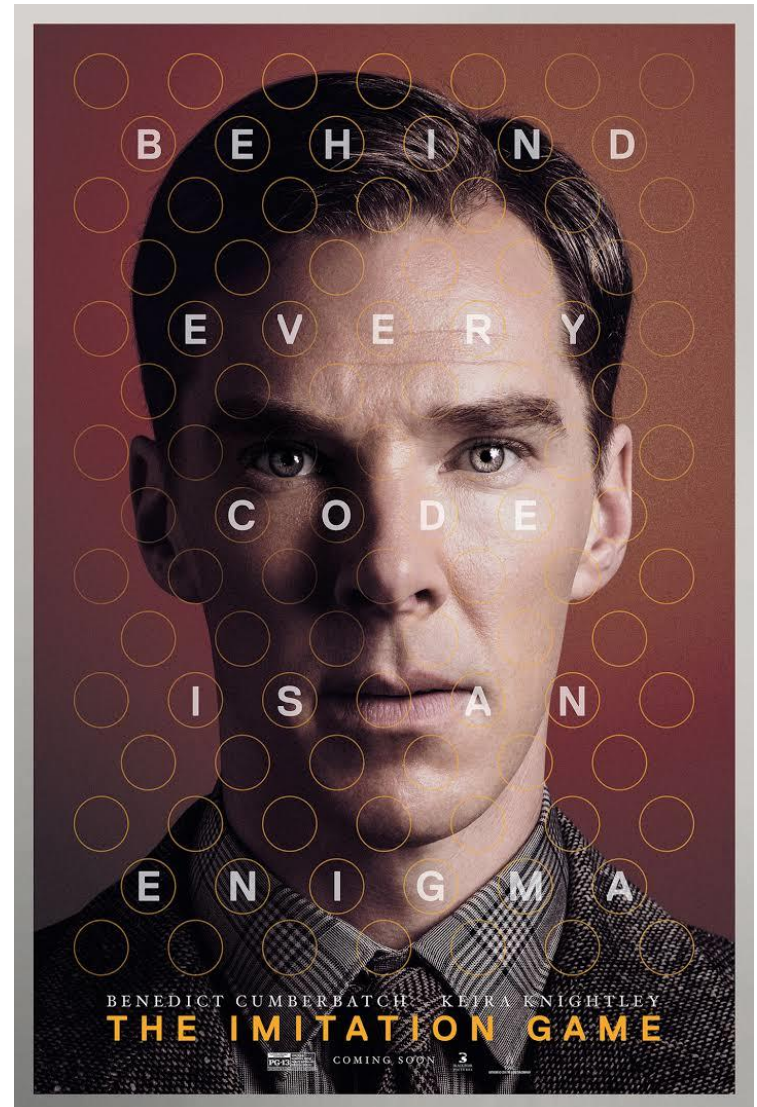
German Enigma Encryption Machine



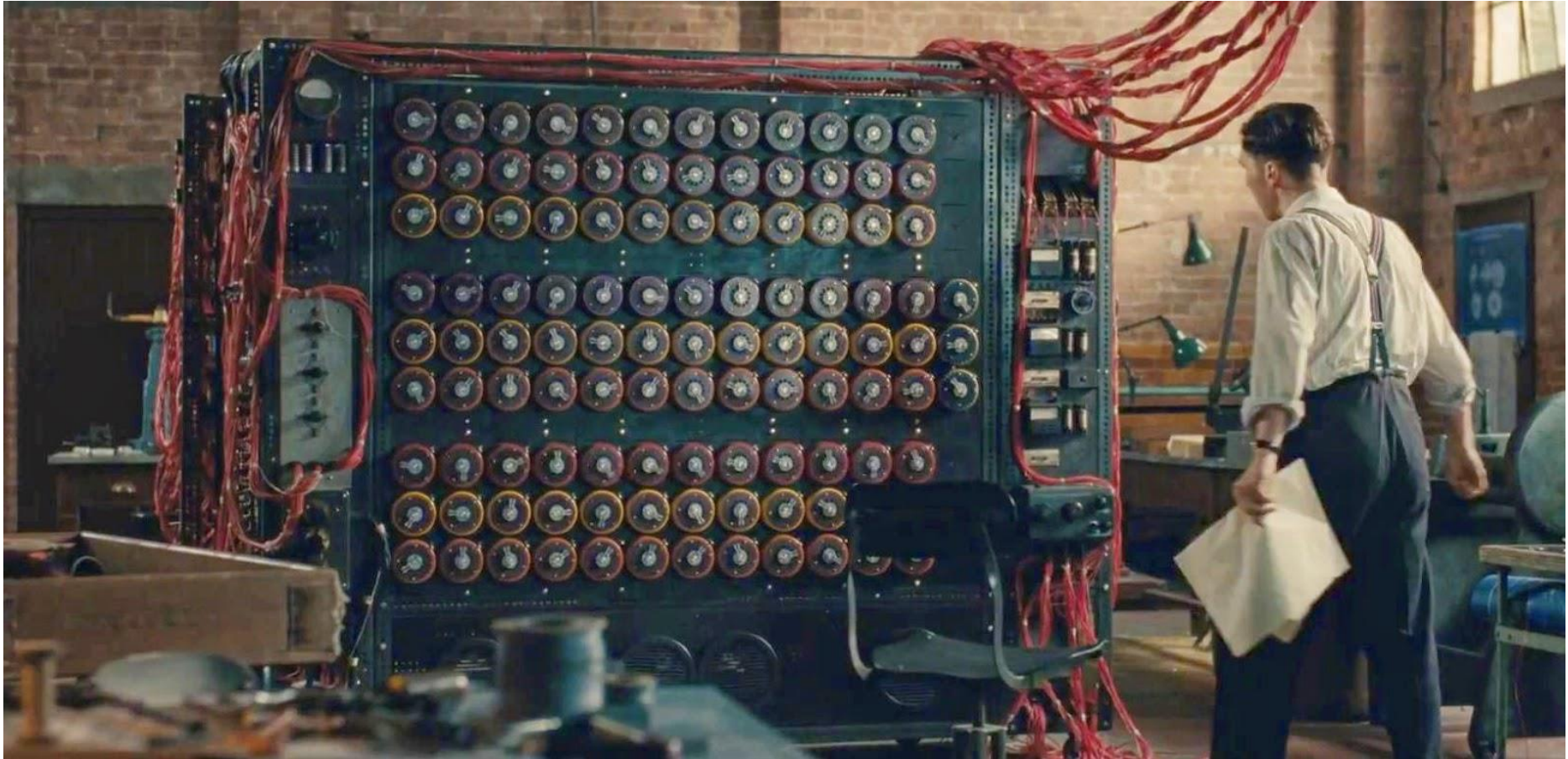
Movie – Alan Turning

In 1939, newly created British intelligence agency MI6 recruits Cambridge mathematics alumnus Alan Turing (Benedict Cumberbatch) to crack Nazi codes, including Enigma -- which cryptanalysts had thought unbreakable. Turing's team, including Joan Clarke (Keira Knightley), analyze Enigma messages while he builds a machine to decipher them. Turing and team finally succeed and become heroes, but in 1952, the quiet genius encounters disgrace when authorities reveal he is gay and send him to prison.

Wikipedia



Trailer - Movie

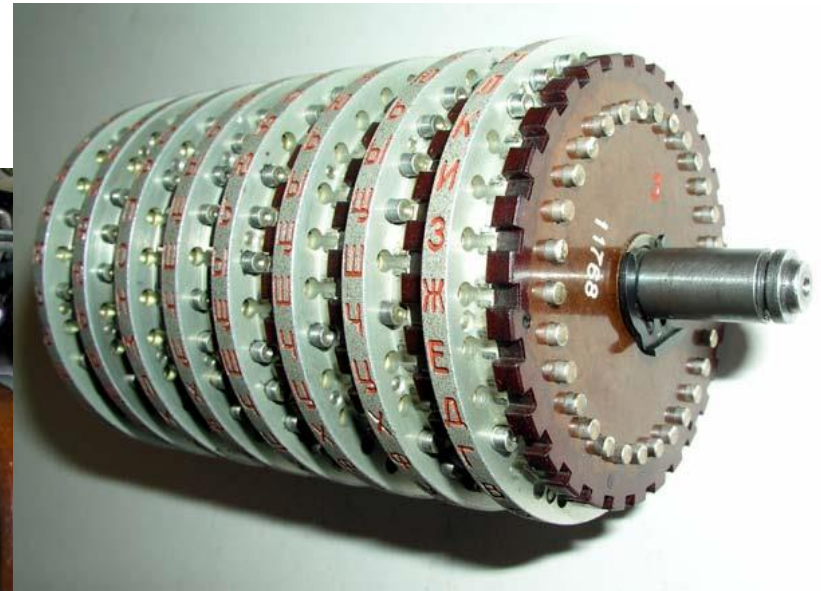
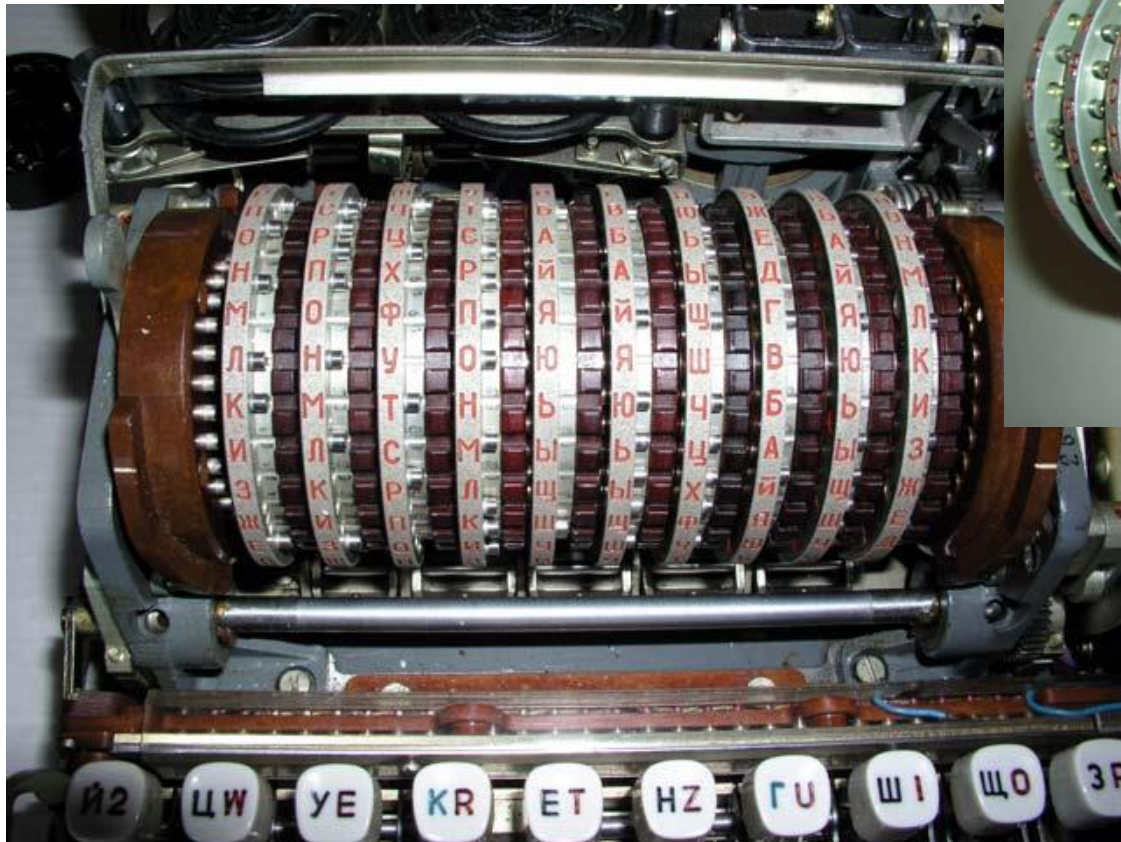


<https://www.youtube.com/watch?v=S5CjKEFb-sM>

Hagelin Rotor Machine

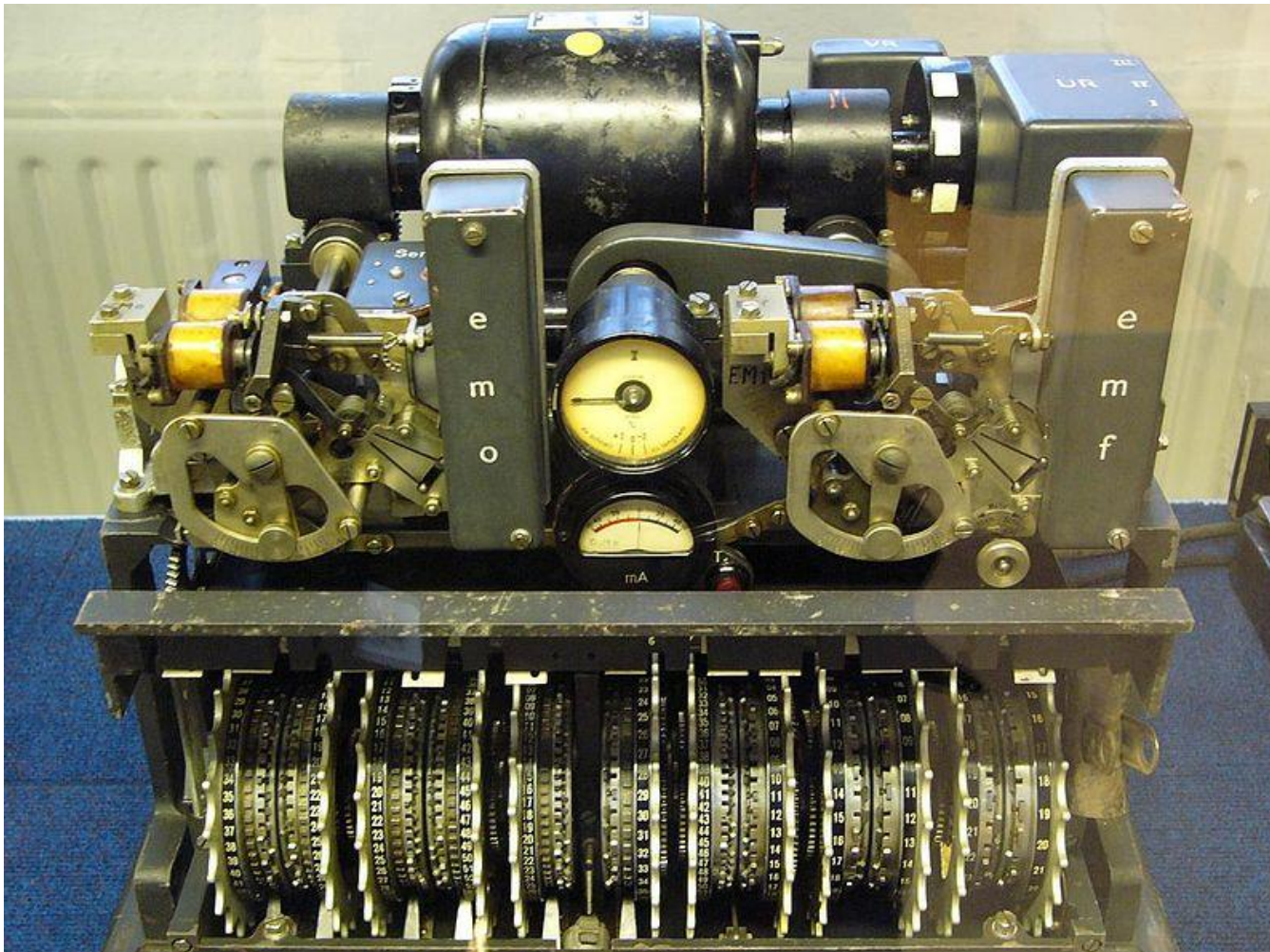


Soviet cipher machine - Fialka (M-125)



Cipher machines

German Lorenz cipher



German Lorenz cipher machine, used in World War II to encrypt Army Commands messages

ACM Turing Award



CELEBRATING 50 YEARS
OF COMPUTING'S GREATEST ACHIEVEMENTS

Vernam Cipher

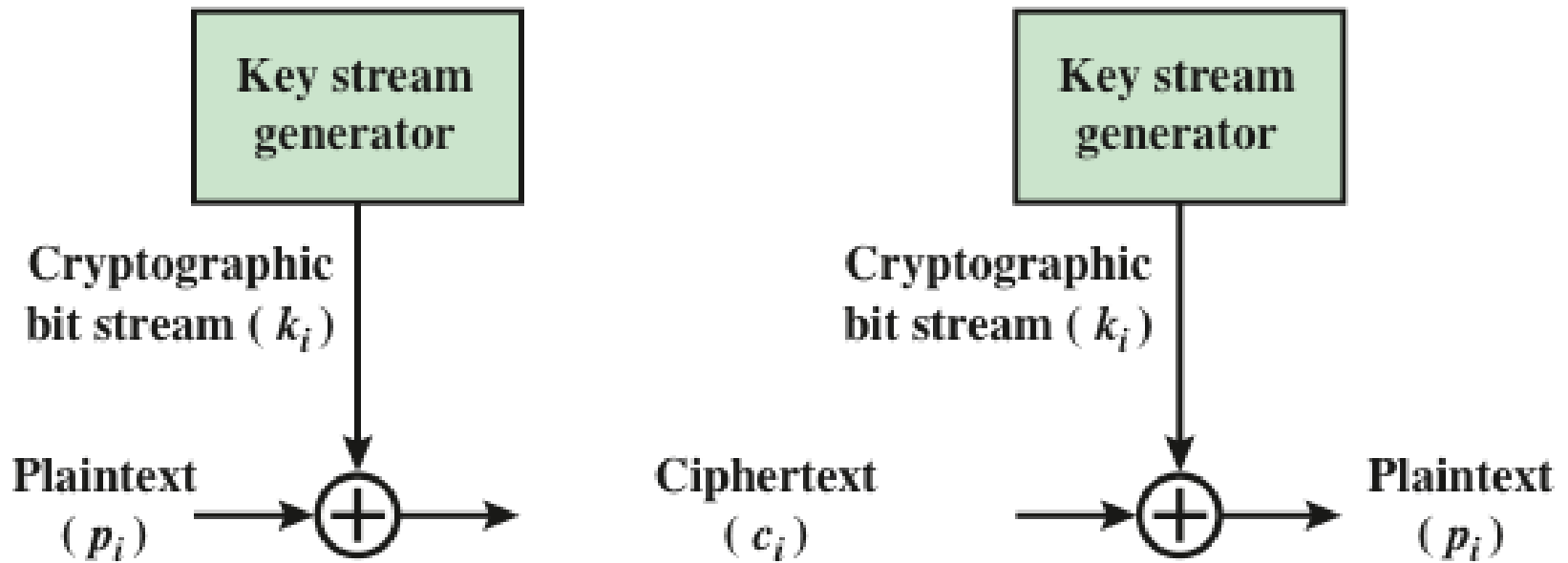


Figure 3.7 Vernam Cipher

One-Time Pad (OTP)

- The key is a truly random key as long as the message
- The cipher will be unconditionally secure / unbreakable since ciphertext bears no statistical relationship to the plaintext
- can only use the key **once**
- Limitations: generation & secure distribution of key

One-Time Pad (OTP)

- Improvement to Vernam cipher proposed by an Army Signal Corp officer, Joseph Mauborgne
- Use a random key that is as long as the message so that the key need not be repeated
- Key is used to encrypt and decrypt a single message and then is discarded
- Each new message requires a new key of the same length as the new message
- Scheme is unbreakable
 - Produces random output that bears no statistical relationship to the plaintext
 - Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code



Difficulties

- The one-time pad offers complete security but, in practice, has two fundamental difficulties:
 - There is the practical problem of making large quantities of random keys
 - Any heavily used system might require millions of random characters on a regular basis
 - Mammoth key distribution problem
 - For every message to be sent, a key of equal length is needed by both sender and receiver
- Because of these difficulties, the one-time pad is of limited utility
 - Useful primarily for low-bandwidth channels requiring very high security
- The one-time pad is the only cryptosystem that exhibits *perfect secrecy*

Transposition Ciphers

- now consider classical **transposition** or **permutation** ciphers
- these hide the message by rearranging the letter order
- without altering the actual letters used
- can recognise these since have the same frequency distribution as the original text

Rail Fence cipher

- write message letters out diagonally over a number of rows
- then read off cipher row by row

- write message in two rails:

```
m e m a t r h t g p r y  
e t e f e t e o a a t
```

- giving ciphertext

MEMATRHTGPRYETEFETEOAAT

Row Transposition Ciphers

- a more complex transposition
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

Key:	4	3	1	2	5	6	7
Plaintext:	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	a	m	x	y	z

Ciphertext: TTNA APTM TSUO AODW COIX KNLY PETZ

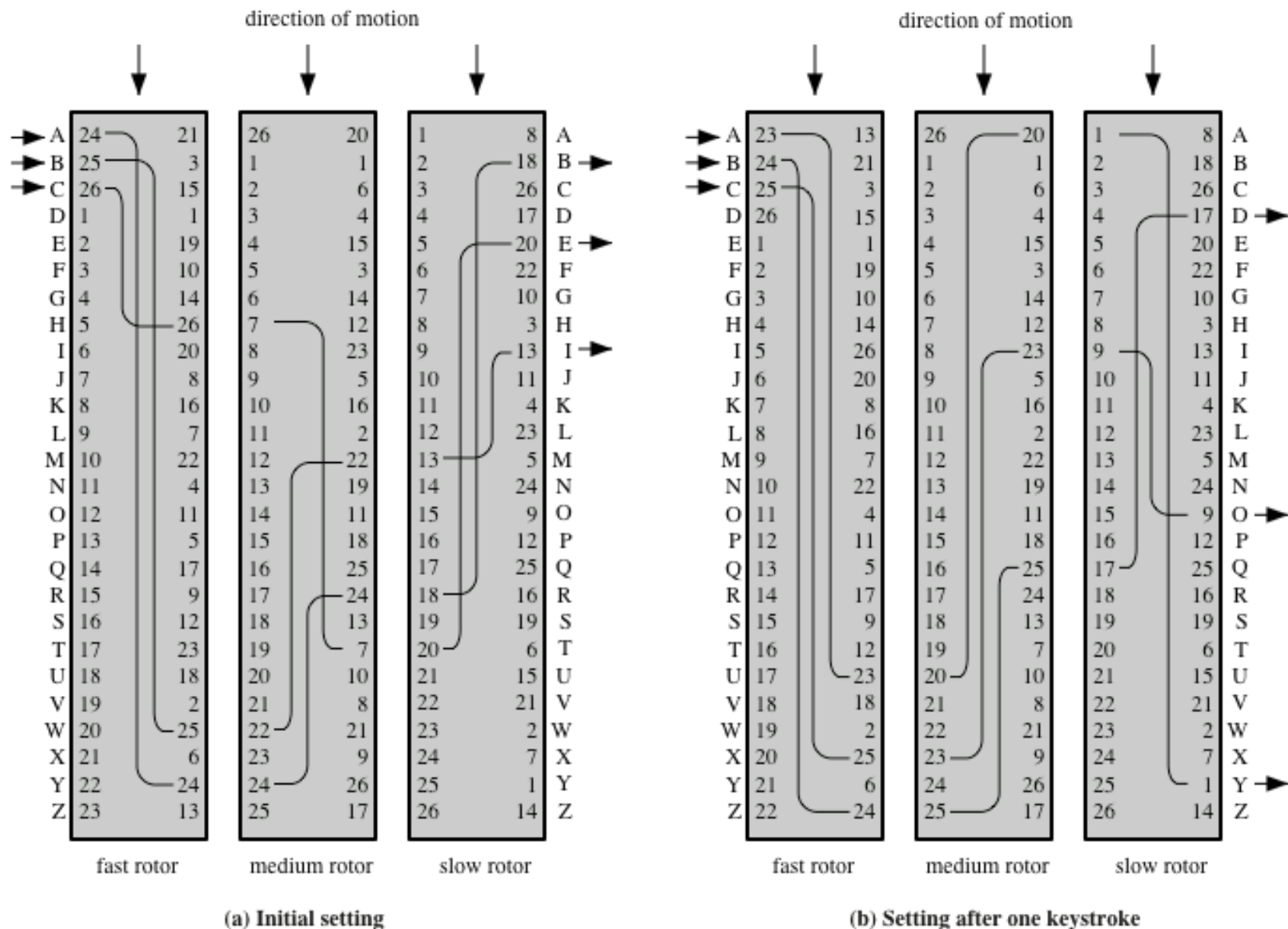


Figure 3.8 Three-Rotor Machine With Wiring Represented by Numbered Contacts

Lecture 2:

Classical Cryptography

- Terminology
- Classical Encryption Techniques
- Key Distribution
- Steganography

Key Distribution

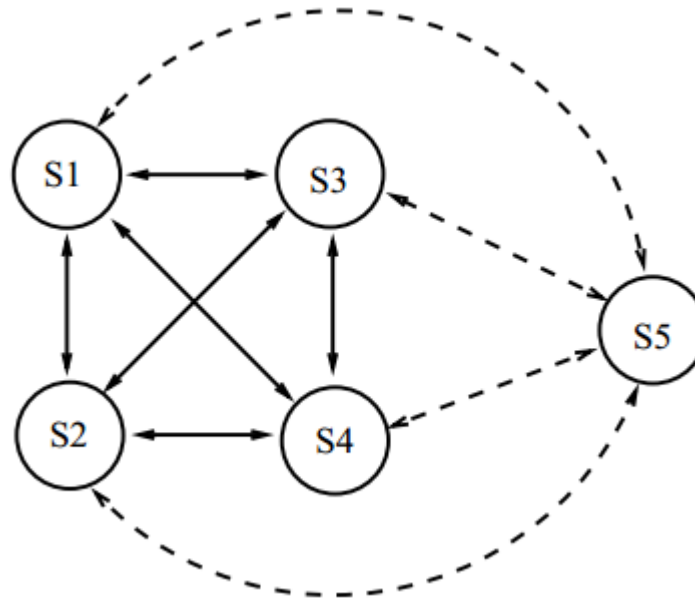
- Session key (short-term keys):
 - at the conclusion of a session, the key is destroyed
 - if compromised → not affecting the long-term security of the system
- Static key (long-term keys):
 - keys that are to be in use for a long time period
 - if compromised → a major problem

Key Distribution

- How many keys are needed for pairwise secure communication among N parties using symmetric encryption?
 - require $N(N-1)/2$ secret keys
 - Exercise: Prove the above statement

Key Distribution

- When the n^{th} user is added to the system, he needs to share a new key with EACH of the previous $(n - 1)$ users



Key Distribution

- Key distribution is problematic for symmetric key cryptosystem
 - both parties must exchange the key prior to using any encryption
 - what-if when these two parties cannot meet?
- Will re-visit the problem in later lectures
 - Exercise: can you propose a solution?
 - Public Key Cryptography

Lecture 2:

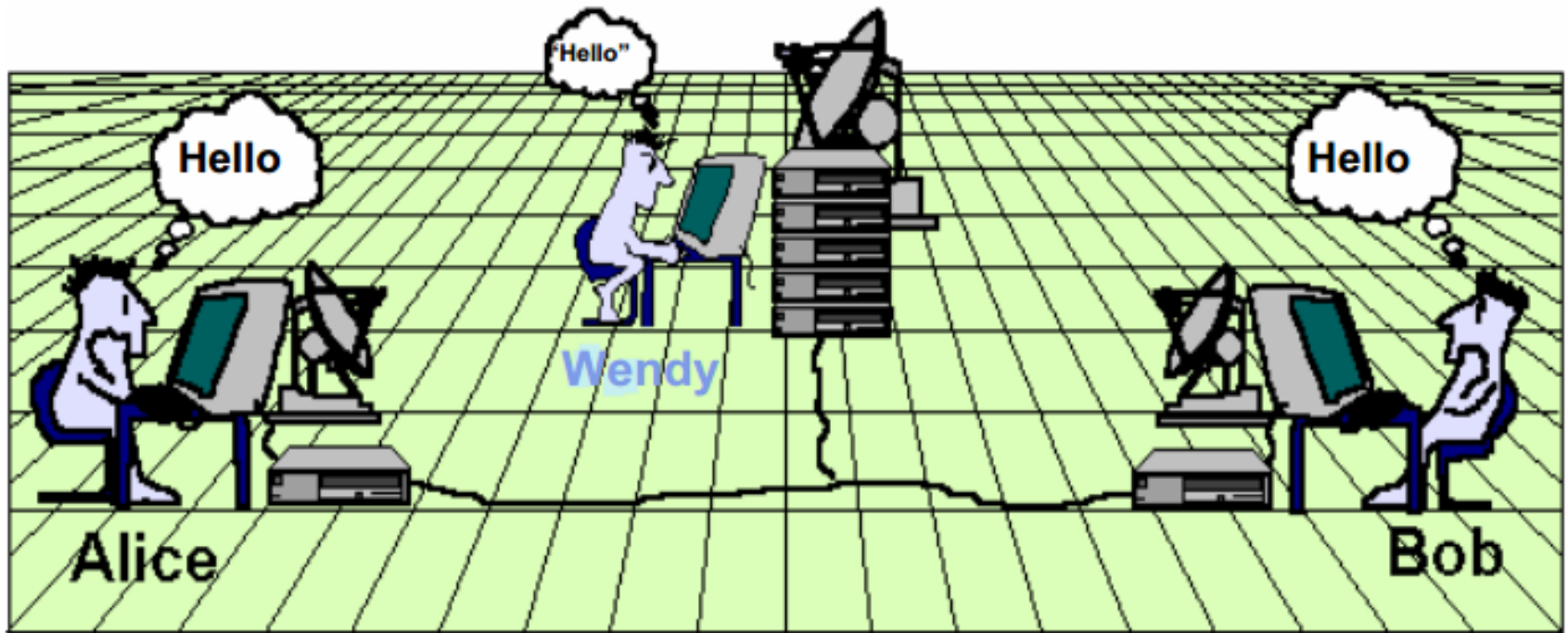
Classical Cryptography

- Terminology
- Classical Encryption Techniques
- Key Distribution
- Steganography

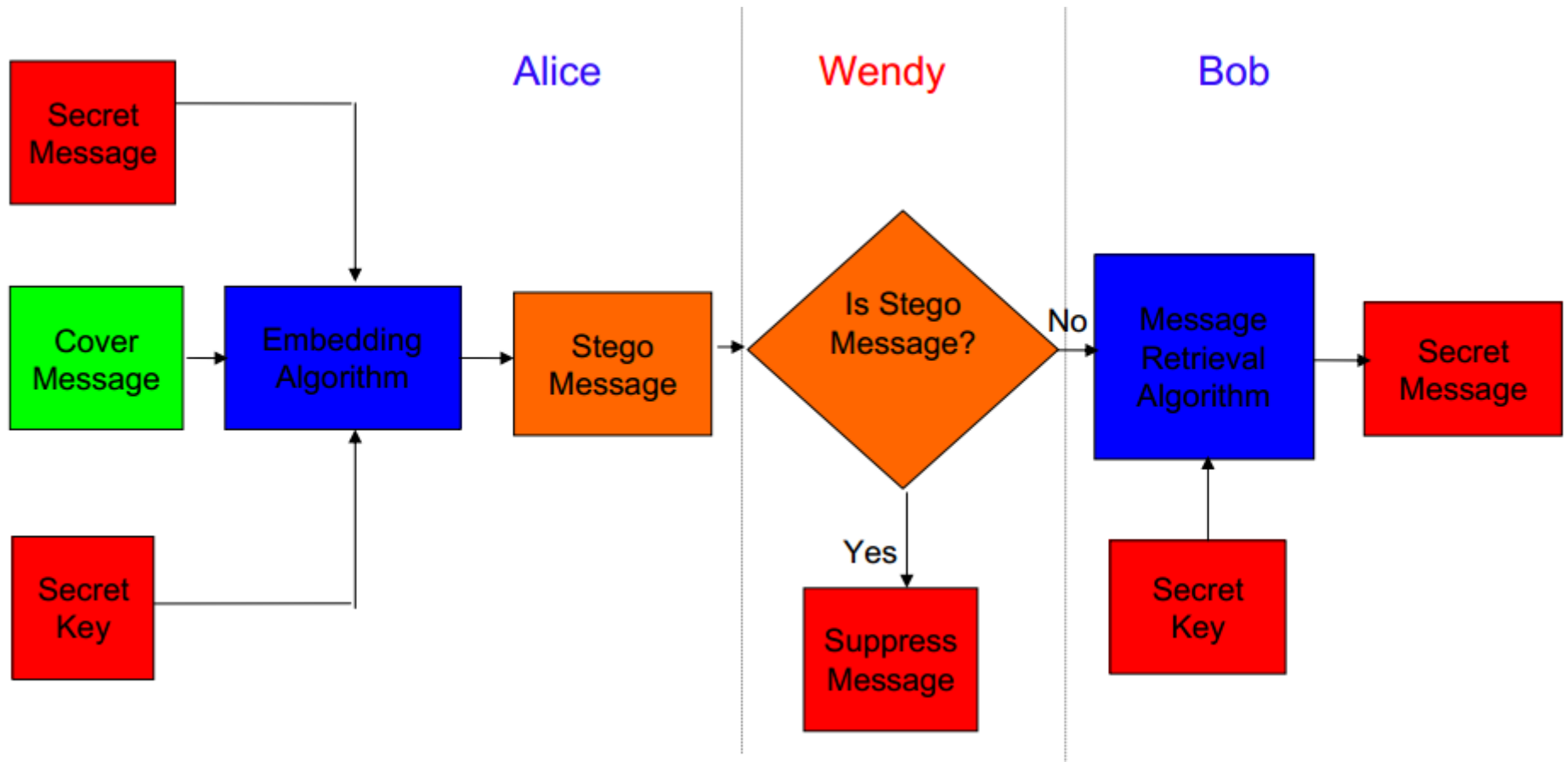
Steganography

- hides existence of message
 - using only a subset of letters/words in a longer message marked in some way
 - using invisible ink
 - hiding in LSB in graphic image or sound file
- has drawbacks
 - high overhead to hide relatively few info bits
- encryption vs. steganography
 - Encryption: you might be forced to surrender your key
 - Steganography: protection against detection

Scenario: Prisoners' Problem



How it works?



Example

Bob Smith, my assistant programmer, can always be found hard at work in his cubicle. Bob works independently, without wasting company time talking to colleagues. Bob never thinks twice about assisting fellow employees, and he always finishes given assignments on time. Often Bob takes extended measures to complete his work, sometimes skipping coffee breaks. Bob is a dedicated individual who has absolutely no vanity in spite of his high accomplishments and profound knowledge in his field. I firmly believe that Bob can be classed as a high-caliber employee, the type which cannot be dispensed with. Consequently, I duly recommend that Bob be promoted to executive management, and a proposal will be sent away as soon as possible.

Sd/-

Project Leader

Example

- A memo was soon sent following the letter:

That stupid idiot was reading over my shoulder when I wrote the report sent to you earlier today. Kindly read every second line (i.e. 1, 3, 5, 7, 9, ..) for my true assessment of him.

Example

Bob Smith, my assistant programmer, can always be found hard at work in his cubicle. Bob works independently, without wasting company time talking to colleagues. Bob never thinks twice about assisting fellow employees, and he always finishes given assignments on time. Often Bob takes extended measures to complete his work, sometimes skipping coffee breaks. Bob is a dedicated individual who has absolutely no vanity in spite of his high accomplishments and profound knowledge in his field. I firmly believe that Bob can be classed as a high-caliber employee, the type which cannot be dispensed with. Consequently, I duly recommend that Bob be promoted to executive management, and a proposal will be sent away as soon as possible.

Sd/-
Project Leader

Another Example

News Eight Weather: Tonight increasing snow. Unexpected precipitation smothers eastern towns. Be extremely cautious and use snowtires especially heading east. The highway is not knowingly slippery. Highway evacuation is suspected. Police report emergency situations in downtown ending near Tuesday.

First letter of each word yields:

Newt is upset because he thinks he is President.

From WWII German spy (Kahn):

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on, by ejecting suets and vegetable oils.

Second letter of each word yields:

Pershing sails from NY June 1.

Steganography

- The art of covered writing
 - conceal what you are communicating
 - double meaning
 - hidden message can be encrypted first

Stego – hidden message

Crypto – visible but encrypted



AP

U.S. officials say Osama bin Laden is posting instructions for terrorist activities on sports chat rooms, pornographic bulletin boards and other Web sites.

Classical Techniques

- Linguistic
 - first letter of each word (null cipher)
 - letter positions on page
 - Anagrams 易位構詞遊戲, e.g. earth = heart
 - Try <http://www.anagramgenius.com/server.html>
 - ...
- Technical
 - microdots
 - invisible ink
 - typewriter correction ribbon
 - spread spectrum (RF)
 - ...

Digital Techniques

- hide messages (including encrypted message) in images or videos
- hide data on disk
- hide data/services on Internet
 - hackers use non-standard ports for "telnet"
 - shell over ICMP (the ping protocol)
 - hide bits in network packet headers
 - ...
- font alterations, word/line shifts
- embed in email headers, pdf file, html doc, etc

Steganalysis

- detect/prove stegomessage
- read stegomessage
- remove/jam the stegomessage
- develop a better statistical model of the cover than the sender

Steganography vs. Encryption

- Steganography has a number of drawbacks when compared to encryption
 - It requires a lot of overhead to hide a relatively few bits of information
 - Once the system is discovered, it becomes virtually worthless

- The advantage of steganography
 - It can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered
 - Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide

Lecture 2: What you have learnt?

- Terminology
- Classical Encryption Techniques
- Key Distribution
- Steganography
- Symmetric Cipher Model
 - Cryptography
 - Cryptanalysis and Brute-Force Attack
- Transposition techniques
- Rotor machines
- Substitution techniques
 - Caesar cipher
 - Monoalphabetic ciphers
 - Playfair cipher
 - Polyalphabetic ciphers
 - One-time pad
- Steganography

Your mission



- Understand classical cryptography
- Build up the interests
- Get ready for upcoming Homework #1

