03 Cryptography I

Engr 399/599: Hardware Security

Grant Skipper, Ph.D.
Indiana University

Adapted from: Mark Tehranipoor of University of Florida

Course Website

engr599.github.io

Write that down!

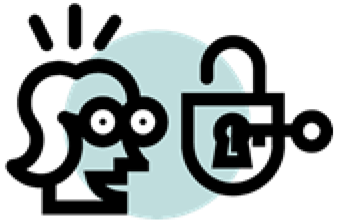
Some Basic Definitions

- **Intellectual property** represents the property of your mind or intellect - proprietary knowledge

- The four legally defined forms of IP
  - **Patents** When you register your invention with the government, you gain the legal right to exclude anyone else from manufacturing or marketing it
  - **Trademarks** A trademark is a name, phrase, sound or symbol used in association with services or products
  - **Copyrights** Copyright laws protect written or artistic expressions fixed in a tangible medium
  - **Trade secrets** A formula, pattern, device or compilation of data that grants the user an advantage over competitors

Some Basic Definitions (Cont’d)

- **Cryptography**:
  - crypto (secret) + graph (writing)
    - the science of locks and keys
  - The keys and locks are mathematical
  - Underlying every security mechanism, there is a “secret”…
  - We are going to talk some about the traditional crypto, but we will also show new forms of security based on other forms of HW-based secret



What Does Secure Mean?

- It has to do with an asset that has some value – think of what can be an asset!
- There is no static definition for “secure”
- Depends on what is that you are protecting your asset from
- Protection may be sophisticated and unsophisticated
- Typically, breach of one security makes the protection agent aware of its shortcoming

Typical Cycle in Securing a System

- Predict potential breaches and vulnerabilities
- Consider possible countermeasures, or controls
- Either actively pursue identifying a new breach, or wait for a breach to happen
- Identify the breach and work out a protected system again

Computer Security

- No matter how sophisticated the protection system is – simple breaches could break-in
- A computing system is a collection of hardware (HW), software (SW), storage media, data, and human interacting with them
- Security of SW, data, and communication
- HW security, is important and challenging
    - Manufactured ICs are obscure
    - HW is the platform running SW, storage and data
    - Tampering can be conducted at many levels
    - Easy to modify because of its physical nature

Definitions

- **Vulnerability**: Weakness in the secure system

- **Threat**: Set of circumstances that has the potential to cause loss or harm

- **Attack**: The act of a human exploiting the vulnerability in the system

- **Computer security aspects**
  - **Confidentiality**: the related assets are only accessed by authorized parties
  - **Integrity**: the asset is only modified by authorized parties
  - **Availability**: the asset is accessible to authorized parties at appropriate times

Hardware Vulnerabilities

- Physical Attacks
- Trojan Horses
- IP Piracy
- IC Piracy & Counterfeiting
- Backdoors
- Tampering
- Reverse Engineering

Adversaries

- **Individual, group or governments**
  - Pirating the IPs – illegal use of IPs
  - Inserting backdoors, or malicious circuitries
  - Implementing Trojan horses
  - Reverse engineering of ICs
  - Spying by exploiting IC vulnerabilities
- **System integrators**
  - Pirating the IPs
- **Fabrication facilities**
  - Pirating the IPs
  - Pirating the ICs
- **Counterfeiting parties**
  - Recycling, cloned, etc.

Hardware Controls for Secure Systems

- Hardware implementations of encryption
  - Encryption has to do with scrambling to hide
- Design locks or physical locks limiting the access
- Devices to verify the user identities
- Hiding signatures in the design files
- Intrusion detection
- Hardware boards limiting memory access
- Tamper resistant
- Policies and procedures
- More …

Secret

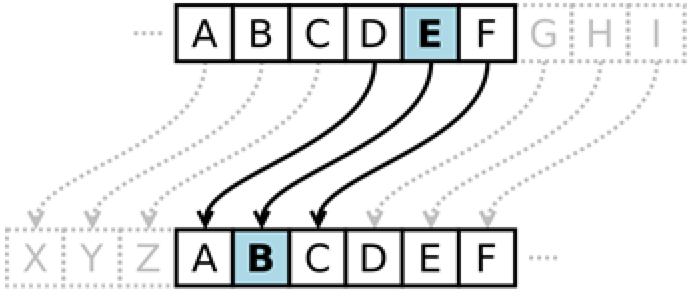
- **Underlying most security mechanisms or protocols is the notion of a “secret”**
  - Lock and keys
  - Passwords
  - Hidden signs and procedures
  - Physically hidden

Cryptography – History

- Has been around for 2000+ years
- In 513 B.C, Histiaeus of Miletus, shaved the slave’s head, tattooed the message on it, let the hair grow

Cryptography – Pencil & Paper Era

- Caesar's cipher: shifting each letter of the alphabet by a fixed amount!
  - ❑ Easy to break



Plaintext: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

- Cryptoquote: simple substitution cipher, permutations of 26 letters
  - ❑ Using the dictionary and the frequencies, this is also easy to break

Caesar Cipher Example

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

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

- What is K? K = ?

- Encrypt INDIANA with K

Caesar Cipher Example

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

- What is K? K = ?

  3
- Encrypt INDIANA with K

  LQGLDQD

Breaking Caesar’s Cipher

Naive way to break caesar cipher?

Assume PT/CT is english alphabet.

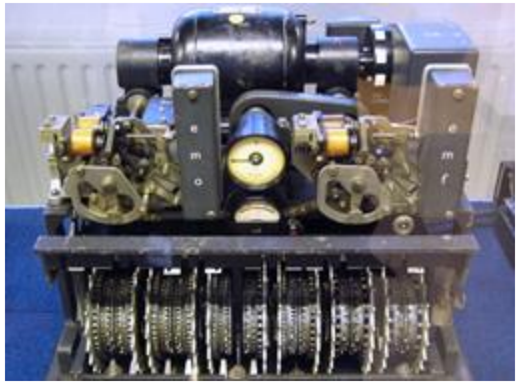
Thoughts?

Cryptography – Mechanical Era

- Around 1900, people realized cryptography has math and stat roots
- German’s started a project to create a mechanical device to encrypt messages
- Enigma machine □ supposedly unbreakable
- A few polish mathematicians got a working copy
- The machine later sold to Britain, who hired 10,000 people to break the code!
- They did crack it! The German messages were transparent to enemies towards the end of war
  - **Estimated that it cut the war length by about a year**
- British kept it secret until the last working Enigma!

Cryptography – Mechanical Era

- Another German-invented code was Tunny (Lorenz cipher system)
- Using a pseudorandom number generator, a seed produced a key stream *ks*
- The key stream xor'd with plain text p to produce cipher c: $c=p\oplus ks$
- How was this code cracked by British cryptographers at Bletchley Park in Jan 1942?
- A lucky coincidence!



German rotor stream cipher machines used by the German Army during World War II

Summary

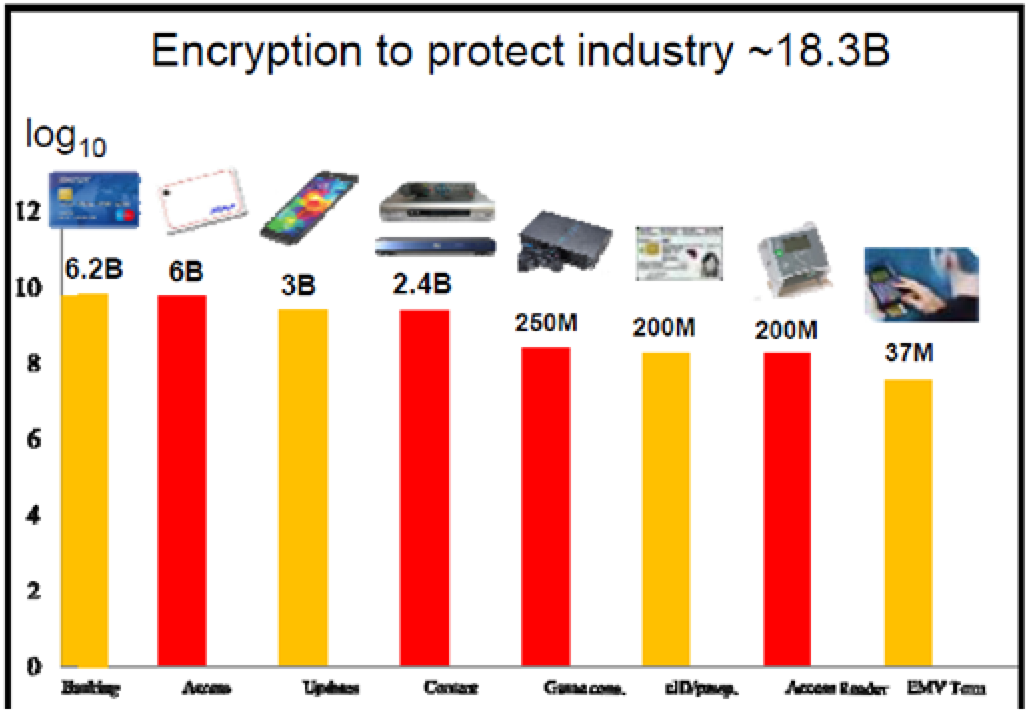
- Substitution ciphers
- Permutations
- Making good ciphers
- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)

Side note: Information Theory - good to familiarize yourself!
https://en.wikipedia.org/wiki/Information_theory

Slides are courtesy of Leszek T. Lilien from WMich
http://www.cs.wmich.edu/~llilien/

Cryptography will play an increasingly Important Role …

- Crypto principles see growing usage in information protection
- A locking approach



Cryptographic algorithms protects critical infrastructure and assets!

Threats to Messages

- Interception
- Interruption
  - Blocking msgs
- Modification
- Fabrication / Forging

"A threat is blocked by control of a vulnerability"

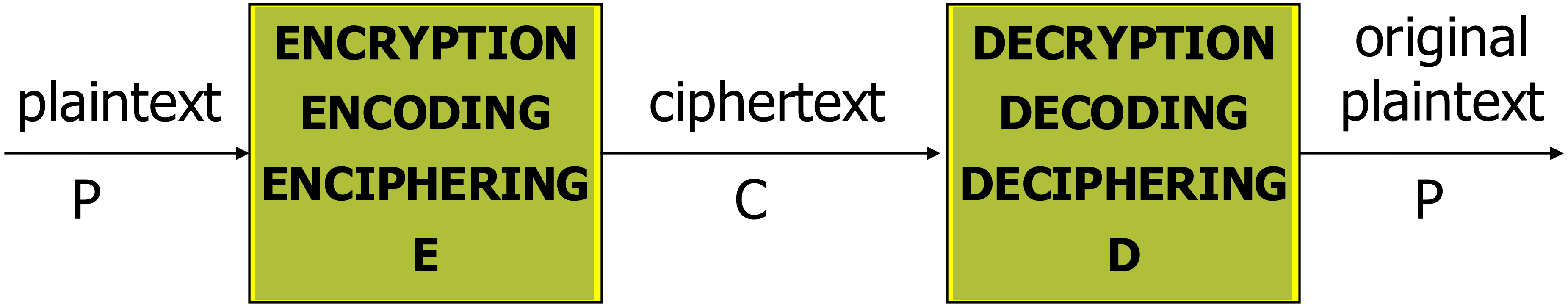
[Pfleeger & Pfleeger]

[cf. B. Endicott-Popovsky, U. Washington]

Basic Terminology & Notation

- **Cryptology:**
  - cryptography + cryptanalysis
- **Cryptography:**
  - art/science of keeping message secure
- **Cryptanalysis:**
  - art/science of breaking ciphertext
    - *Enigma* in world war II
      - Read the real story – not fabrications!

Basic Cryptographic Scheme



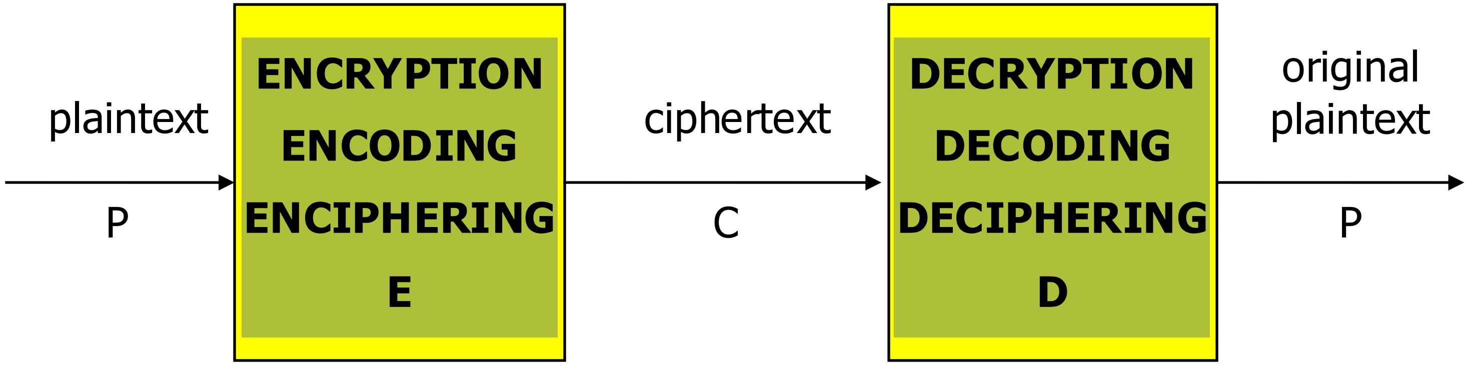
- $P = <p_1, p_2, \ldots, p_n>$ — p_i = i-th char of P
  - P = "DO NOT TELL ANYBODY" — p_1 ="D", p_2 = "O", etc.
  - By convention, cleartext in uppercase
- $C = <c_1, c_2, \ldots, c_n>$ — c_i = i-th char of C
  - C = "ep opu ufmm bozcpez" — c_1 ="e", c_2 ="p", etc.
  - By convention, ciphertext in lowercase

Benefits of Cryptography

- **Improvement not a Solution!**
  - Minimizes problems
  - Doesn’t solve them
    - Remember: There is *no* solution!
  - Adds an envelope (encoding) to an open postcard (plaintext or cleartext)

[cf. D. Frincke, U. of Idaho]

Formal Notation

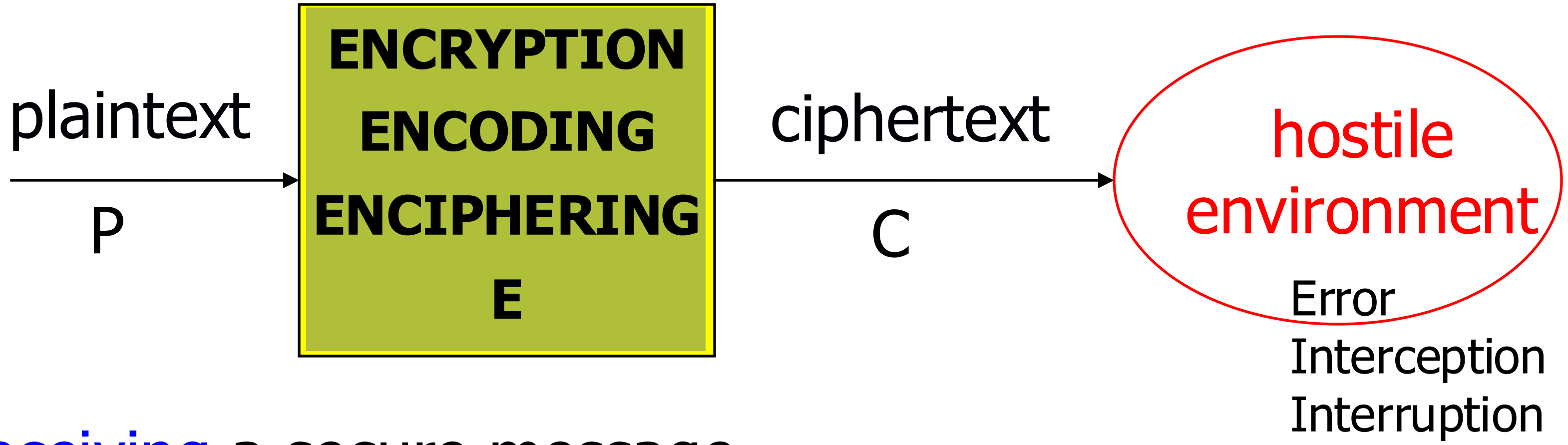


- C = E(P) E – encryption rule/algorithm
- P = D(C) D – decryption rule/algorithm

- We need a cryptosystem, where:
  - P = D(C)= D(E(P))
    - i.e., able to get the original message back

Cryptography in Practice

- Sending a secure message



plaintext

P

ENCRYPTION
ENCODING
ENCIPHERING
E

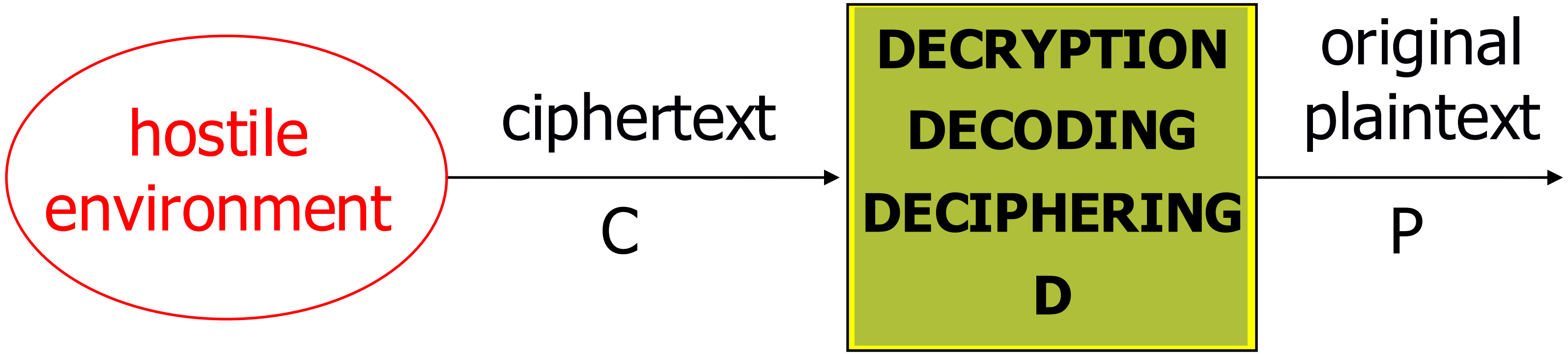
ciphertext

C

hostile environment

Error
Interception
Interruption

- Receiving a secure message



hostile environment

ciphertext

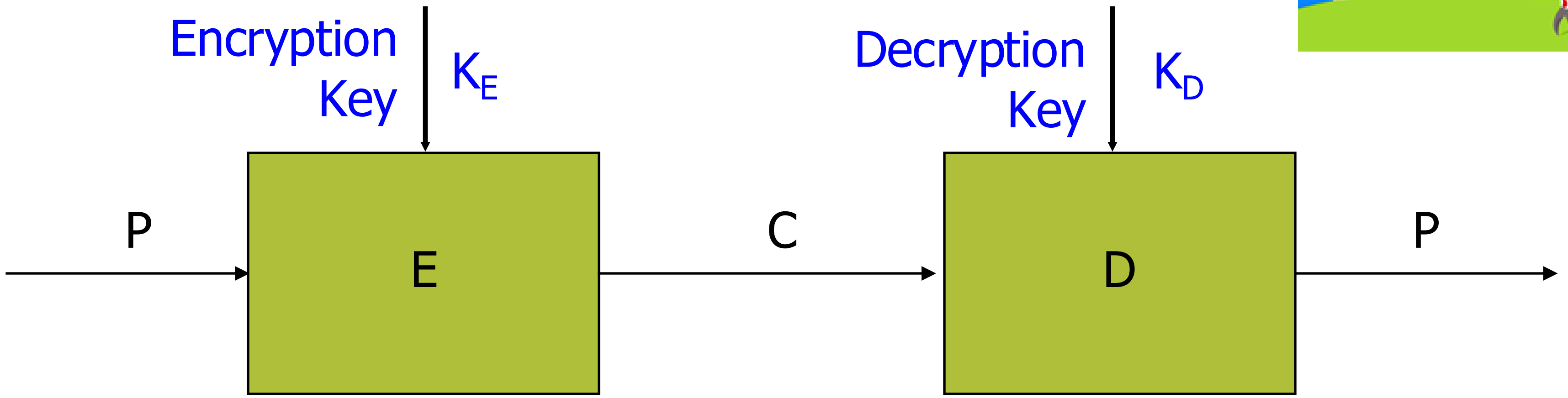
C

DECRYPTION
DECODING
DECIPHERING
D

original plaintext

P

Crypto System with Keys



- $C = E(K_E, P)$
  - E = *set* of encryption algorithms / K_E selects $E_i \in E$
- $P = D(K_D, C)$
  - D = *set* of decryption algorithms / K_D selects $D_j \in D$
- Crypto algorithms and keys are like door locks and keys
- We need: $P = D(K_D, E(K_E, P))$

Classification of Cryptosystems w.r.t. Keys

- Keyless cryptosystems exist (e.g., Caesar’s cipher)
  - Less secure
- Symmetric cryptosystems: $K_E = K_D$
  - Classic
  - Encipher and decipher using the same key
    - Or one key is easily derived from other
- Asymmetric cryptosystems: $K_E \neq K_D$
  - Public key system
  - Encipher and decipher using different keys
    - Computationally infeasible to derive one from other

[cf. B. Endicott-Popovsky, U. Washington]

Cryptanalysis (1)

- **Cryptanalysts goals:**
  - Break a single msg
  - Recognize patterns in encrypted msgs, to be able to break the subsequent ones
  - Infer meaning w/o breaking encryption
    - Unusual volume of msgs between enemy troops may indicate a coming attack
    - Busiest node may be enemy headquarters
  - Deduce the key, to facilitate breaking subsequent msgs
  - Find vulnerabilities in implementation or environment of an encryption algorithm
  - Find a general weakness in an encryption algorithm

Cryptanalysis (2)

- **Information for cryptanalysts:**
  - Intercepted encrypted msgs
  - Known encryption algorithms
  - Intercepted plaintext
  - Data known or suspected to be ciphertext
  - Math or statistical tools and techniques
  - Properties of natural languages
    - Esp. adversary’s natural language
      - To confuse the enemy, Americans used Navajo language in WW2
  - Propertiers of computer systems
- Role of ingenuity / luck
- There are *no* rules!!!

Breakable Encryption (1)

- **Breakable encryption**
  - *Theoretically*, it is possible to devise unbreakable cryptosystems
  - *Practical* cryptosystems almost always are breakable, given adequate time and computing power
  - The trick is to make breaking a cryptosystem hard enough for the intruder

[cf. J. Leiwo, VU, NL]

Breakable Encryption (2)

- Example: Breakability of an encryption algorithm

  Msg with just 25 characters

  - 26^{25} possible decryptions ~ 10^{35} decryptions
  - Only one is the right one
  - Brute force approach to find the right one:
    - At 10^{10} (10 bln) decryption/sec => 10^{35} / 10^{10} = 10^{16} sec = 10 bln yrs !
    - Infeasible with current technology

How can we constrain the problem and reduce state space we need to check?

- Be smarter – use ingenuity
  - ❑ Could reduce 26^{25} to, say, 10^{15} decryptions to check

    At 10^{10} decr./sec => 10^{15} / 10^{10} = 10^{5} sec = ~ 1 day

Requirements for Crypto Protocols

- Messages should get to destination
- Only the recipient should get it
- Only the recipient should see it
- Proof of the sender’s identity
- Message shouldn’t be corrupted in transit
- Message should be sent/received once

[cf. D. Frincke, U. of Idaho]

- Proofs that message was sent/received (non-repudiation)

Representing Characters

- Letters (uppercase only) represented by numbers 0-25 (modulo 26).

A	**B**	**C**	**D**	**. . .**	**X**	**Y**	**Z**
0	**1**	**2**	**3**	**. . .**	**23**	**24**	**25**

- Operations on letters:

A + 2 = C

X + 4 = B (circular!)

. . .

Basic Types of Ciphers

- Substitution ciphers
  - Letters of P replaced with other letters by E
- Transposition (permutation) ciphers
  - *Order* of letters in P rearranged by E
- Product ciphers
  - E "=" E_1 "+" E_2 "+" ... "+" E_n
    - Combine two or more ciphers to enhance the security of the cryptosystem

Substitution Ciphers

- **Substitution Ciphers:**
  - **Letters of P replaced with other letters by E**

The Caesar Cipher (1)

- $c_i=E(p_i)=p_i+3 \bmod 26$ (*26* letters in the English alphabet)

  Change each letter to the third letter following it (circularly)

  A □ D, B □ E, ... X □ A, Y □ B, Z □ C

- Can represent as a permutation π: $\pi(i) = i+3 \bmod 26$

  $\pi(0)=3, \pi(1)=4, \ldots,$

  $\pi(23)=26 \bmod 26=0, \pi(24)=1, \pi(25)=2$

- Key = 3, or key = ‘D’ (because D represents 3)

The Caesar Cipher (2)

- Example [cf. B. Endicott-Popovsky]
  - P (plaintext): HELLO WORLD
  - C (ciphertext): khoor zruog

- Caesar Cipher is a monoalphabetic substitution cipher (= simple substitution cipher)
  One key is used
  One letter substitutes the letter in P

Attacking a Substitution Cipher

- Exhaustive search
  - If the key space is small enough, try all possible keys until you find the right one
  - Cæsar cipher has 26 possible keys
    from A to Z OR: from 0 to 25
- Statistical analysis (attack)
  - Compare to so called 1-gram (unigram) model of English
    - 1-gram: It shows frequency of (single) characters in English
  - The longer the C, the more effective statistical analysis would be

[cf. Barbara Endicott-Popovsky, U. Washington]

1-grams (Unigrams) for English

a	0.080	h	0.060	n	0.070	t	0.090
b	0.015	i	0.065	o	0.080	u	0.030
c	0.030	j	0.005	p	0.020	v	0.010
d	0.040	k	0.005	q	0.002	w	0.015
e	0.130	l	0.035	r	0.065	x	0.005
f	0.020	m	0.030	s	0.060	y	0.020
g	0.015					z	0.002

[cf. Barbara Endicott-Popovsky, U. Washington]

Statistical Attack – Step 1

- Compute frequency f(c) of each letter c in ciphertext
- Example: c = ‘khoor zruog’
  - 10 characters: 3 * ‘o’, 2 * ‘r’, 1 * {k, h, z, u, g}
  - f(c):

    f(g)=0.1 f(h)=0.1 f(k)=0.1 f(o)=0.3 f(r)= 0.2

    f(u)=0.1 f(z)=0.1 $f(c_i)$ = 0 for any other ci
- Apply 1-gram model of English
  - Frequency of (single) characters in English
  - 1-grams on previous slide

[cf. Barbara Endicott-Popovsky, U. Washington]

Statistical Analysis – Step 2

- phi $\phi(i)$ - correlation of frequency of letters *in ciphertext* with frequency of corresponding letters *in English* —for key i
- For key i: $\phi(i) = \Sigma_{0 \le c \le 25} f(c) * p(c - i)$
  - *c* representation of character (a-0, ..., z-25)
  - f(c) is frequency of letter c in ciphertext C
  - $p(x)$ is frequency of character x in English
  - Intuition: sum of probabilities for words in P, if i were the key

c is a letter in ciphertext thus c-i is the letter in plaintext.

- Example: C = 'khoor zruog' (P = 'HELLO WORLD')

f(c): f(g)=0.1, f(h)=0.1, f(k)=0.1, f(o)=0.3, f(r)=0.2, f(u)=0.1, f(z)=0.1

c: g - 6, h - 7, k - 10, o - 14, r - 17, u - 20, z - 25

$$\phi(i) = 0.1p(6 - i) + 0.1p(7 - i) + 0.1p(10 - i) + \\ + 0.3p(14 - i) + 0.2p(17 - i) + 0.1p(20 - i) + \\ + 0.1p(25 - i)$$

[cf. Barbara Endicott-Popovsky, U. Washington]

Statistical Attack – Step 2a (Calculations)

- Correlation ϕ(i) for 0≤ i ≤25

i	**ϕ(*i*)**	***i***	**ϕ(*i*)**	***i***	**ϕ(*i*)**	***i***	**ϕ(*i*)**
0	0.0482	7	0.0442	13	0.0520	19	0.0315
1	0.0364	8	0.0202	14	0.0535	20	0.0302
2	0.0410	9	0.0267	15	0.0226	21	0.0517
3	0.0575	10	0.0635	16	0.0322	22	0.0380
4	0.0252	11	0.0262	17	0.0392	23	0.0370
5	0.0190	12	0.0325	18	0.0299	24	0.0316
6	0.0660					25	0.0430

[cf. Barbara Endicott-Popovsky, U. Washington]

Statistical Attack – Step 3 (The Result)

- Most probable keys (largest ϕ(i) values):
  - i = 6, ϕ(i) = 0.0660
    - plaintext EBIIL TLOLA
  - i = 10, ϕ(i) = 0.0635
    - plaintext AXEEH PHKEW
  - i = 3, ϕ(i) = 0.0575
    - plaintext HELLO WORLD
  - i = 14, ϕ(i) = 0.0535
    - plaintext WTAAD LDGAS
- Only English phrase is for *i* = 3
  - That's the key (3 or 'D') – code broken

[cf. Barbara Endicott-Popovsky, U. Washington]

Caesar's Problem

- Conclusion: Key is too short
    - 1-char key – monoalphabetic substitution
        - Can be found by exhaustive search
        - Statistical frequencies not concealed well by short key
            - They look too much like 'regular' English letters
- Solution: Make the key longer
    - n-char key (n ≥ 2) – polyalphabetic substitution
        - Makes exhaustive search much more difficult
        - Statistical frequencies concealed much better
            - Makes cryptanalysis harder

[cf. Barbara Endicott-Popovsky, U. Washington]

Other Substitution Ciphers

n-char key:

- Polyalphabetic substitution ciphers
- Vigenere Tableaux cipher

Polyalphabetic Substitution - Examples

- Flatten (diffuse) *somewhat* the frequency distribution of letters by combining high and low distributions

- Example – 2-key substitution:

	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
Key1:	a	d	g	j	m	p	s	v	y	b	e	h	k	n	q	t	w	z	c	f	i	l	o	r	u	x
Key2:	n	s	x	c	h	m	r	w	b	g	l	q	v	a	f	k	p	u	z	e	j	o	t	y	d	i

- **Question**:

  How Key1 and Key2 were defined?

[cf. J. Leiwo, VU, NL]

Polyalphabetic Substitution - Examples

- ...

- Example:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key1:	a	d	g	j	m	p	s	v	y	b	e	h	k	n	q	t	w	z	c	f	i	l	o	r	u	x
Key2:	n	s	x	c	h	m	r	w	b	g	l	q	v	a	f	k	p	u	z	e	j	o	t	y	d	i

- Answer:

  Key1 – start with ‘a’, skip 2, take next,
  skip 2, take next letter, ... (circular)

  Key2 - start with ‘n’ (2nd half of alphabet), skip 4,
  take next, skip 4, take next, ... (circular)

[cf. J. Leiwo, VU, NL]

Polyalphabetic Substitution - Examples

– Example:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key1:	a	d	g	j	m	p	s	v	y	b	e	h	k	n	q	t	w	z	c	f	i	l	o	r	u	x
Key2:	n	s	x	c	h	m	r	w	b	g	l	q	v	a	f	k	p	u	z	e	j	o	t	y	d	i

– **Plaintext: TOUGH STUFF**

– **Ciphertext: ffirv zfjpm**

use n (=2) keys in turn for consecutive P chars in P

- Note:
  - Different chars mapped into the same one: **T, O** □ **f**
  - Same char mapped into different ones: **F** □ **p, m**
  - '**f**' most frequent in C (0.30); in English: f(**f**) = 0.02 << f(**e**) = 0.13

[cf. J. Leiwo, VU, NL]

Vigenere Tableaux (1)

Note: Row A – shift 0 (a->a)

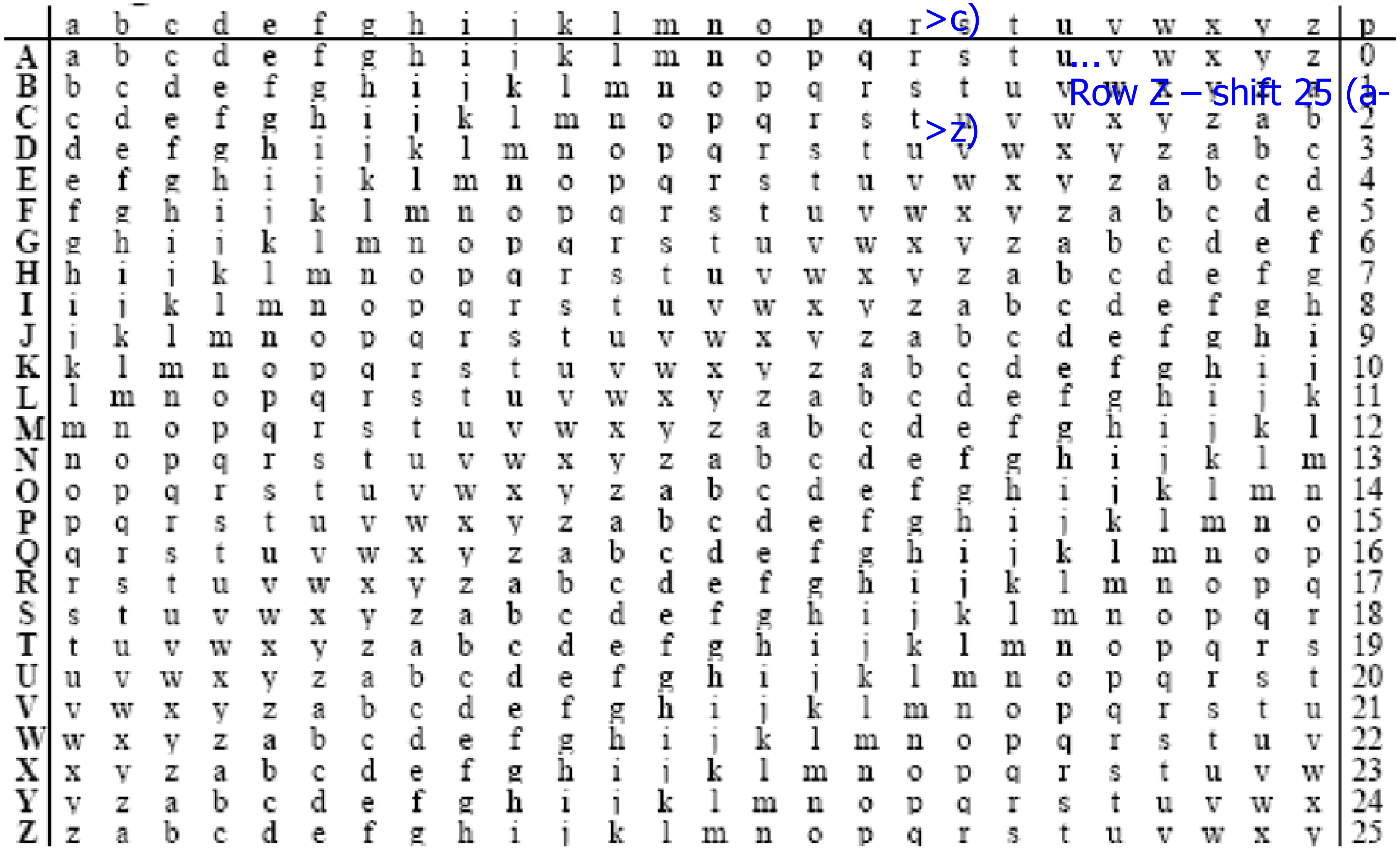
Row B – shift 1 (a->b)

Row C – shift 2 (a->c)

…

Row Z – shift 25 (a->z)

[cf. J. Leiwo, VU, NL]



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

Vigenère Tableaux (2)

- Example

  Key:

  EXODUS

  Plaintext P:

  YELLOW SUBMARINE FROM YELLOW RIVER

  Extended keyword (re-applied to mimic words in P):

  YELLOW SUBMARINE FROM YELLOW RIVER
  EXODUS EXODUSEXO DUSE XODUSE XODUS

  Ciphertext:

  cbxoio wlppujmks ilgq vsofhb owyyj

[cf. J. Leiwo, VU, NL]

Vigenère Tableaux (3)

- Example

  …

  Extended keyword (re-applied to mimic words in P):

  ```
  YELLOW SUBMARINE FROM YELLOW RIVER
  EXODUS EXODUSEXO DUSE XODUSE XODUS
  ```

  Ciphertext:

  ```
  cbzoio wlppujmks ilgq vsofhb owyyj
  ```

- Answer:

  c from P indexes row

  c from extended key indexes column

  e.g.: row Y and column e □ ‘c’
  row E and column x □ ‘b’
  row L and column o □ ‘z’
  …

[cf. J. Leiwo, VU, NL]

Transposition Ciphers (1)

- Rearrange letters in plaintext to produce ciphertext
- Example 1a and 1b: Columnar transposition
  - Plaintext: **HELLO WORLD**
  - Transposition onto: (a) 3 columns:

    HEL
    LOW
    ORL
    DXX **XX** - padding

    (b) onto 2 columns:

    HE
    LL
    OW
    OR
    LD
  - Ciphertext (read column-by column):

    (a) **hlodeorxlwlx** (b) **hloolelwrd**
  - What is the key?
    - Number of columns: (a) key = 3 and (b) key = 2

Transposition Ciphers (2)

- Example 2: Rail-Fence Cipher
  - Plaintext: **HELLO WORLD**
  - Transposition into 2 rows (rails) column-by-column:

    HLOOL

    ELWRD
  - Ciphertext: **hloolelwrd** (Does it look familiar?)

    [cf. Barbara Endicott-Popovsky, U. Washington]
  - What is the key?
    - Number of rails key = 2

Product Ciphers

- A.k.a. combination ciphers
- Built of multiple blocks, each is:
  - Substitution

or:

  - Transposition
- Example: two-block product cipher
  - $E_2(E_1(P, K_{E1}), K_{E2})$
- Product cipher might *not* necessarily be stronger than its individual components used separately!
  - Might not be even as strong as individual components

Criteria for “Good” Ciphers

- “Good” depends on intended application
  - Substitution
    - C hides chars of P
    - If > 1 key, C dissipates high frequency chars
  - Transposition
    - C scrambles text => hides n-grams for n > 1
  - Product ciphers
    - Can do all of the above
  - What is more important for your app?
    What facilities available to sender/receiver?
    - E.g., no supercomputer support on the battlefield

Criteria for “Good” Ciphers

- **Commercial Principles of Sound Encryption Systems**
  1. Sound mathematics
     - Proven vs. not broken so far
  2. Verified by expert analysis
     - Including outside experts
  3. Stood the test of time
     - Long-term success is not a guarantee
       - Still. Flows in many E’s discovered soon after their release

- Examples of popular commercial encryption:
  - DES / RSA / AES

DES = Data Encryption Standard
RSA = Rivest-Shamir-Adelman
AES = Advanced Encryption Standard (rel. new)

[cf. A. Striegel]

Stream and Block Ciphers (1)

a. Stream ciphers

b. Problems with stream ciphers

c. Block ciphers

d. Pros / cons for stream and block ciphers

Stream Ciphers (1)

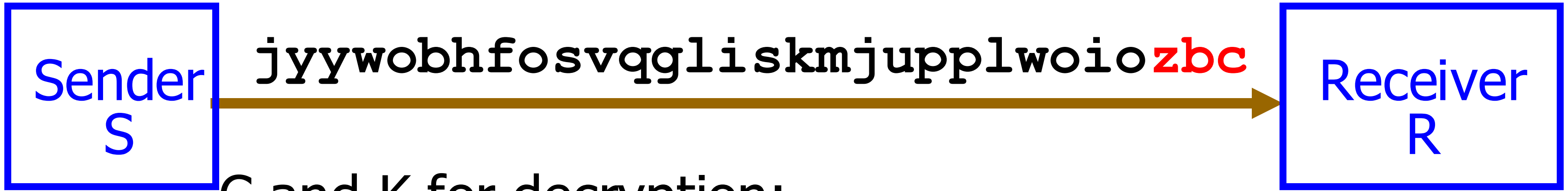
- Stream cipher: 1 char from P □ 1 char for C
  - Example: polyalphabetic cipher
    - P and K (repeated 'EXODUS'):
      YELLOWSUBMARINEFROMYELLOWRIVER
      EXODUSEXODUSEXODUSEXODUSEXODUS
    - Encryption (char after char, using Vigenère Tableaux):
      (1) E(**Y**, **E**) □ **c** (2) E(**E**, **X**) □ **b** (3) E(**L**, **O**) □ **z** . . .
    - C: **cbzoiowlppujmksilgqvsofhbowyyj**
    - C as sent (in the right-to-left order):
      jyywobhfosvqgliskmjupplwoiozbc

Sender
S
Receiver
R

Stream Ciphers (2)

– Example: polyalphabetic cipher - cont.

- C as received (in the right-to-left order):



jyywobhfosvqgliskmjupplwoiozbc

- C and K for decryption:

  cbzoiowlppujmksilgqvsofhbowyyj

  EXODUSEXODUSEXODUSEXODUSEXODUS

- Decryption:

  (1) D(c, E) □ Y (2) D(b, X) □ E (3) D(z, O) □ L ...

- Decrypted P:

  YEL...

Q: Do you know how D uses Vigenère Tableaux?

A: Finds c under column e □ Y

Problems with Stream Ciphers (1)

- Problems with stream ciphers
  - Dropping a char from key K results in wrong decryption
  - Example:
    - P and K (repeated `EXODUS') with a char in K missing:

YELLOWSUBMARINEFROMYELLOWRIVER
<u>EO</u>DUSEXODUSEXODUSEXODUSEXODUSE

missing X in K ! (no errors in repeated K later)

- Encryption (using VT):

1) E(**Y**,**E**) □
c

2) E(**E**,**O**) □
s

3) E(**L**,**D**) □

- Ciphertext: **cso...**

C in the order as sent (right-to-left):
...osc

Problems with Stream Ciphers (2)

- C as received (in the right-to-left order):

  ...osc

- C and correct K (‘EXODUS’) for decryption:

  cso...

  EXO...

- Decryption (using VT, applying correct key):

  1) D(c, E) □ Y

  2) D(s, X) □ V

  3) D(o, O) □ A

  ...

- Decrypted P:

  YVA... - Wrong!

  – We know it’s wrong, Receiver might not know it *yet*!

What if message is corrupted in a noisy area?

Problems with Stream Ciphers (3)

- The problem might be recoverable
  - Example:

    If R had more characters decoded, R might be able to detect that S dropped a key char, and R could recover

    - E.g., suppose that R decoded:

      YELLOW SUBMAZGTR

      - R could guess, that the 2nd word should really be:

        SUBMARINE

      - => R would know that S dropped a char from K after sending "**SUBMA**"
      - => R could go back 4 chars, drop a char from K ("recalibrate K with C"), and get "resynchronized" with S

Block Ciphers (1)

- We can do better than using recovery for stream ciphers
  - Solution: use block ciphers

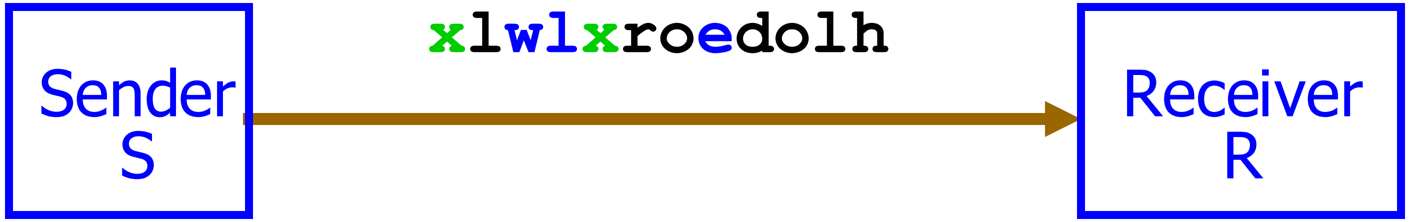
- Block cipher:
  1 *block* of chars from P □ 1 *block* of chars for C
  - Example of block cipher: columnar transposition
  - Block size = "o(message length)" (informally)

Block Ciphers (2)

- Why block size = “o(message length)” ?
  - Because R must wait for “almost” the entire C before R can decode some characters near beginning of P
  - E.g., for P = ‘**HELLO WORLD**’, block size is “o(10)”
  - Suppose that Key = 3 (3 columns):

```
HEL
LOW
ORL
DXX
```

  - C as sent (in the right-to-left order):



Block Ciphers (3)

- C as received (in the right-to-left order): **xlwlxroedolh**
- R knows: K = 3, block size = 12 (=> 4 rows)

```
123
456
789
abc
```

a=10
b=11
c=12

=> R knows that characters wil be sent in the order:
1st-4th-7th-10th--2nd-5th-8th-11th--3rd-6th-9th-12th

- R must wait for at least:
  - 1 char of C to decode 1st char of P (‘**h**’)
  - 5 chars of C to decode 2nd char of P (‘**he**’)
  - 9 chars of C to decode 3rd, 4th, and 5th chars of P (‘**hello**’)
  - 10 chars of C to decode 6th, 7th, and 8th chars of P (‘**hello wor**’)
  - etc.

Block Ciphers (4)

- *Informally*, we might call ciphers like the above example columnar transposition cipher "weak-block" ciphers
  - R can get some (even most) but not all chars of P before entire C is received
    - R can get one char of P immediately
      - » the 1st-after 1 of C (delay of 1 - 1 = 0)
    - R can get some chars of P with "small" delay
      - » e.g., 2nd-after 5 of C (delay of 5 - 2 = 3)
    - R can get some chars of P with "large" delay
      - » e.g., 3rd-after 9 of C (delay of 9 – 3 = 6)

- There are block ciphers when R cannot even start decoding C before receiving the entire C
  - *Informally*, we might call them "strong-block" ciphers

Pros / Cons for Stream and Block Ciphers (1)

- Pros / cons for stream ciphers
  - + Low delay for decoding individual symbols
    - Can decode as soon as received
  - + Low error propagation
    - Error in E(c_1) does not affect E(c_2)
  - - Low diffusion
    - Each char separately encoded => carries over its frequency info
  - - Susceptibility to malicious insertion / modification
    - Adversary can fabricate a new msg from pieces of broken msgs, even if he doesn't know E (just broke a few msgs)

Pros / Cons for Stream and Block Ciphers (2)

- Pros / cons for block ciphers
  - + High diffusion
    - Frequency of a *char* from P diffused over (a few chars of) a *block* of C
  - + Immune to insertion
    - Impossible to insert a char into a block without easy detection (block size would change)
    - Impossible to modify a char in a block without easy detection (if checksums are used)

Pros / Cons for Stream and Block Ciphers (3)

- Pros / cons for block ciphers — Part 2
  - - High delay for decoding individual chars
    - See example for ‘**hello worldxx**’ above
      - For some E can’t decode even the 1st char before whole k chars of a block are received
  - - High error propagation
    - It affects the block, not just a single char

Cryptanalysis (1)

- What cryptanalysts do when confronted with unknown?

  Four possible situations:

  1) C available
  2) Full P available
  3) Partial P available
  4) E available (or D available)

Control the situation!

- (1) – (4) suggest 5 different approaches

Cryptanalysis (2)

- Cryptanalyst approaches

1) Ciphertext-only attack
   - We have shown examples for such attacks
     - E.g., for Caesar’s cipher, columnar transposition cipher

2) Known plaintext attack
   - Analyst have C and P
     - Needs to deduce E such that C=E(P), then finds D

3) Probable plaintext attack
   - Partial decryption provides partial match to C
     - This provides more clues

Cryptanalysis (3)

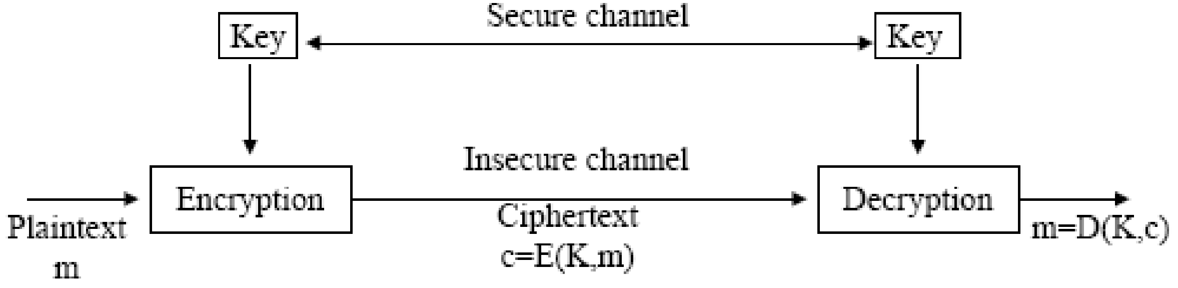
- Cryptanalyst approaches – cont.

4) Chosen plaintext attack
   - Analyst able to fabricate encrypted msgs
     - Then observe effects of msgs on adversary's actions
       - » This provides further hints

5) Chosen ciphertext attack
   - Analyst has both E and C
   - Run E for many candidate plaintexts to find P for which E(P) = C
     - Purpose: to find K_E

Symmetric and Asymmetric Cryptosystems (1)

- Symmetric encryption = secret key encryption
  - $K_E = K_D$ — called a secret key or a private key
  - Only sender S and receiver R know the key



[cf. J. Leiwo]

  - As long as the key remains secret, it also provides authentication (= proof of sender's identity)

Symmetric and Asymmetric Cryptosystems (3)

- Asymmetric encryption = public key encryption (PKE)
  - – $K_E \neq K_D$ — public and private keys
- PKE systems eliminate symmetric encryption problems
  - – Need no secure key distribution channel
    - => easy key distribution

Symmetric and Asymmetric Cryptosystems (4)

- One PKE approach:
  - R keeps her private key K_D
  - R can distribute the correspoding public key K_E to anybody who wants to send encrypted msgs to her
    - No need for secure channel to send K_E
    - Can even post the key on an open Web site — it is public!
  - Only private K_D can decode msgs encoded with public K_E!
    - Anybody (K_E is public) can encode
    - Only owner of K_D can decode