CryptoAnalysis

Upper case letters can be 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 mod 26 = C
X+4 mod 26 = B
```

...

Basic Types of Ciphers

- Substitution ciphers

  Letters of plaintext *P* are **replaced** with other letters by encryption algorithm E

- Transposition or permutation ciphers

  Order of letters in *P* are **rearranged** by E

- Product ciphers

  Combine two or more ciphers to enhance the security of the crypto-system

Substitution Ciphers

Outline:

a. The Caesar Cipher

b. Other Substitution Ciphers

c. One-Time Pads

The Caesar Cipher

$c_i = E(p_i) = (p_i+3) \bmod 26$ (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: $\pi(i) = (i+3) \bmod 26$

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

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

Key = 3, or key = 'D' (b/c D represents 3)

Caesar Cipher (Barbara Endicott-Popovsky, U. Washington)

Example

P (plaintext): `HELLO WORLD`

C (ciphertext): `khoor zruog`

Caesar Cipher is a monoalphabetic substitution cipher (a simple substitution cipher)

Exhaustive search

If the key space is small enough, try all possible keys until you find the right one

Caesar cipher has 25 possible keys (1 to 25) (assuming 0 would never be used!)

Statistical analysis (attack)

Compare to **1-gram** (*unigram*) model of English, which shows ***frequency*** of (single) characters in English

Statistical Attack

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

8/28/2006 [cf. Barbara Endicott-Popovsky, U. Washington]

Step 1: Statistical Attack

Compute frequency *f(c)* of each letter *c* in ciphertext

Example: *C* = 'khoor zruog'

10 characters: 'o': 3, 'r': 2, {k, h, z, u, g}: 1

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 all other c_i

Statistical Attack

Step 2: Statistical Analysis

$\phi(i)$: Correlation of frequency of letters in ciphertext with frequency of corresponding letters in English for a particular key *i*

For key *i*: $\phi(i) = \Sigma_{0 <= c <= 25}\, f(c) * p(c - i)$

c is representation of character (0-25)

f(c) is frequency of letter *c* in ciphertext *C*

p(x) is frequency of character *x* in English

This is correlation analysis, i.e., the value of *i* that generates the largest sum indicates the closest match between frequencies in alphabet and cipher text.

Statistical Attack

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
```

Convert letters to numbers:

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

Compute correlation value:

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

Step 2a: Statistical Attack Calculations

- Correlation $\varphi(i)$ for $0 \le i \le 25$

i	$\varphi(i)$	*i*	$\varphi(i)$	*i*	$\varphi(i)$	*i*	$\varphi(i)$
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

8/28/2006 [cf. Barbara Endicott-Popovsky, U. Washington] 25

Statistical Attack

Most probable keys are the largest $\phi(i)$ values:

$i = 6$, $\phi(i) = 0.0660$

Plaintext EBIIL TLOLA

$i = 10$, $\phi(i) = 0.0635$

Plaintext AXEEH PHKEW

$i = 3$, $\phi(i) = 0.0575$

Plaintext HELLO WORLD

$i = 14$, $\phi(i) = 0.0535$

Plaintext WTAAD LDGAS

The plaintext is 'legible English' only for the case when $i = 3$

So the key is 3 or 'D' and the code broken

Caesar’s Problem

Problem: Key is too short

Only used a 1-char key (monoalphabetic substitution)

- Can be found by exhaustive search
- Statistical frequencies not concealed well by the short key, i.e., ciphertext looks too much like the composition of ’regular’ English phrases

Solution: Make the key longer

n-char key (*n* >= 2) - *polyalphabetic* substitution

- Makes exhaustive search much more difficult
- Statistical frequencies concealed much better
- Makes cryptoanalysis harder

Other Substitution Ciphers

Vigenere Tableaux cipher is a ***polyalphabetic substitution*** cipher

Polyalphabetic Substitution Ciphers (J. Leiwo, VU, NL)

Attempts to flatten (diffuse) the frequency distribution of letters by combining high frequency letters with low frequency letters

Example: 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

Key1: ← skip 2 letters

Key2: ← skip 4 letters

Key definition:

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)

Encryption involves using *Key1* for first letter of plaintext, *Key2* for second letter, *Key1* again for third letter, etc.

Polyalphabetic Substitution Ciphers

Plaintext: `TOUGH STUFF`

Ciphertext: `ffirv zfjpm`

Obtained by mapping T->f using *Key1*, O->f using *Key2*, U->i using *Key1*, etc.

	A	B	C	D	E	F	G	H	I	J	K	L	M	
Key1:	a	d	g	j	m	p	s	v	y	b	e	h	k	←skip 2 letters
Key2:	n	s	x	c	h	m	r	w	b	g	l	q	v	←skip 4 letters

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key1:	n	q	t	w	z	c	f	i	l	o	r	u	x
Key2:	a	f	k	p	u	z	e	j	o	t	y	d	i

Characteristics:

- Different chars mapped into the same one: **T**, **O -> f**
- Same char mapped into different ones: **F -> p**, **m**
- 'f' most frequent in Ciphertext => 0.30

  In English: f(**f**) = 0.02 << f(**e**) = 0.13

Vigenere Tableaux

Key:

EXODUS

Plaintext:

YELLOW SUBMARINE FROM YELLOW RIVER

Extended keyword: Re-applied to match length of plaintext:

YELLOW SUBMARINE FROM YELLOW RIVER

EXODUS EXODUSEXO DUSE XODUSE XODUS

Ciphertext:

cbzoio wlppujmks ilgq vsofhb owyyj

How does this work?

Char from plaintext indexes row and char from extended key indexes column

For example,

- row Y and column E: ’c’
- row E and column X: ’b’
- row L and column O: ’z’

Vigenere Tableaux

Vigenere Tableaux (1)

[cf. J. Leiwo, VU, NL]

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)

	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

One-Time Pads

OPT: Variant of using Vigenere Tableaux

Designed to fix problem its problem that the key used might be too short

Above: ’EXODUS’ is only 6 chars

Sometimes considered a **perfect** cipher

Used extensively during Cold War

One-Time Pad:

Large, non-repeating set of long keys on pad sheets/pages

Sender and receiver have identical pads

Example:

300-char msg to send, 20-char key per sheet

Use & tear off 300/20 = 15 pages from the pad

Encryption:

Sender writes letters of consecutive 20-char keys *above* the letters of the plain-text

One-Time Pads

Encryption:

Sender creates ciphertext by adding the plaintext and key characters in each of the columns and takes the sum *mod 26*

And then destroys the used keys

Decryption:

Receiver constructs columns in the same way with ciphertext and the key characters from the same 15 consecutive pages of the pad

Receiver subtracts key characters from ciphertext mod 26 and destroys the keys

Characteristics:

- The key is as long as the message

- The key is always changing (and destroyed after use)

Weaknesses:

- Requires **perfect** synchronization required between S and R

  Intercepted or dropped messages can destroy synchronization

One-Time Pads

Weaknesses:

- Need lots of keys
- Need to distribute pads securely

Transposition Ciphers

Rearrange letters in plaintext to produce ciphertext

Example of **columnar transposition**

Plaintext: HELLO WORLD

(a) Transposition into 3 columns:

```
HEL
LOW
ORL
DXX        XX - padding
```

Transposition Ciphers

(b) Transposition into 2 columns:

```
HE
LL
OW
OR
LD
```

Ciphertext is constructed by reading table column-wise:

(a) **hlodeorxlwlx**

(b) **hloolelwrd**

What is the key?

Number of columns: (a) key = 3 and (b) key = 2

Example 2: **Rail-Fence Cipher**

Plaintext: HELLO WORLD

Transposition Ciphers

Transposition into 2 rows (**rails**) column-by-column:

```
HLOOL
ELWRD
```

Ciphertext:

hloolelwrd (Does it look familiar?)

What is the key?

Number of rails: key = 2

Attacking Transposition Ciphers

Anagramming

n-gram: *n*-char strings in English

Digrams (*2-grams*) for English alphabet are: aa, ab, ac, ...az, ba, bb, bc, ..., zz ($26^2 = 676$ rows in table)

Trigrams are: aaa, aab, ... ($26^3 = 17{,}576$ rows in table)

Attacking Transposition Ciphers

Anagramming

4-grams are: aaaa, aaab, ...

Attack procedure:

If *1-gram* frequencies in C match their frequencies in English BUT other *n-gram* frequencies in C do **not** match their frequencies in English, THEN

It is probably a transposition encryption

Find *n-grams* with the highest frequencies in ciphertext then rearrange sub-strings in ciphertext to form *n-grams* with highest frequencies

Start with n=2

Ciphertext C:

hloolelwrd (from *Rail-Fence* cipher)

N-gram frequency check

1-gram frequencies in C **do** match their frequencies in English

Attacking Transposition Ciphers

N-gram frequency check

2-gram (hl, lo, oo, ...) frequencies in *C* **do not** match their frequencies in English

3-gram (hlo, loo, ool, ...) frequencies in *C* **do not** match their frequencies in English

...

=> Conclude it is probably a transposition

Frequencies in English for all *2-grams* from *C* starting with **h** (from table of frequencies of English digrams)

he 0.0305

ho 0.0043

hl, **hw**, **hr**, **hd** < 0.0010

Implies that in ***h*loo*l*e*l*wrd**, **e** follows **h**

Attacking Transposition Ciphers

Arrange *C* so that the **h** and **e** are adjacent

Since *2-gram* suggests a solution, cut *C* into 2 substrings with the 2nd substring starting with **e**:

hlool elwrd

Put them in 2 columns:

he
ll
ow
or
ld

Read row by row to get original plaintext: **HELLO WORLD**

Product Ciphers

Another name for **combination** ciphers

Built of multiple blocks, where each is based on *substitution* or *transposition*

Example: two-block product cipher

$E_2(E_1(P, K_{E1}), K_{E2})$

Product cipher might **not** be stronger than its individual components used separately!

Might not even be as strong as individual components!

Criteria for Good Ciphers (Claude Shannon's criteria (1949)

- Needed degree of secrecy should determine amount of labor
- Set of keys and enciphering algorithm should be free from complexity
- Implementation should be as simple as possible
- Size & storage of C should be restricted, e.g., size(C) should not be > size(P)

These were proposed at the dawn of computer era are still valid!

Criteria for Good Ciphers

Plus, one additional one

- Propagation of errors should be limited

Characteristics of good encryption schemes

- **Confusion**

  Interceptor **cannot** predict what will happen to *C* when she changes one character in *P*

  Encryptor with good confusion hides relationship between *P* + *K* and *C*

- **Diffusion**

  Changes in *P* spread out over **many parts** of *C*

  Encryptor with good diffusion requires attacker to collect/analyze a lot of *C*

Two basic types of Ciphers

- Stream
- Block

Stream and Block Ciphers

Stream Cipher:

1 char from *P* transformed into 1 char for *C*

The polyalphabetic cipher we saw earlier is an example, e.g., *P* and *K* (repeated "EXODUS")

YELLOWSUBMARINEFROMYELLOWRIVER
EXODUSEXODUSEXODUSEXODUSEXODUS

Encryption involves translating *P* one character at a time and transmitting to receiver

Problem: **dropping** a char results in **wrong** decryption

Block Ciphers

1 ***block*** of chars from *P* transformed to 1 ***block*** of chars for *C*

Example is the *columnar transposition* we saw earlier

Stream and Block Ciphers

Pros/Cons of **Stream Ciphers**

- Positive: Low delay for decoding individual symbols

  Can start decoding as soon as the *C* begins to be received

- Positive: Low error propagation

  Error in $E(c_1)$ does not affect $E(c_2)$

- Drawback: Low diffusion

  Each char encoded separately and therefore can reveal frequency information

- Drawback: Susceptibility to malicious insertion and manipulation

  Adversary can fabricate a new msg from pieces of broken msgs, even if he doesn't know E

Pros/Cons for **Block Ciphers**

- Positive: High diffusion

  Frequency of a chars in *P* diffused over a block of *C*

- Positive: 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 (checksums)

Stream and Block Ciphers

Pros/Cons for **Block Ciphers**

- Drawback: Large delay for decoding individual chars

  For some E, can **not** decode 1st char of C until entire block is received

- Drawback: High error propagation

  Errors affect the entire **block**, not just a single character