Lecture 7
October 6, 2021

Symmetric Cryptography

- Sender, receiver share common key
  - Keys may be the same, or trivial to derive from one another
  - Sometimes called *secret key cryptography*
- Two basic types
  - Transposition ciphers
  - Substitution ciphers
  - Combinations are called *product ciphers*

Transposition Cipher

- Rearrange letters in plaintext to produce ciphertext
- Example (Rail-Fence Cipher)
  - Plaintext is `HELLO WORLD`
  - Rearrange as

```
HLOOL
ELWRD
```

  - Ciphertext is `HLOOL ELWRD`

Attacking the Cipher

- Anagramming
  - If 1-gram frequencies match English frequencies, but other *n*-gram frequencies do not, probably transposition
  - Rearrange letters to form *n*-grams with highest frequencies

Example

- Ciphertext: HLOOLELWRD
- Frequencies of 2-grams beginning with H
  - HE 0.0305
  - HO 0.0043
  - HL, HW, HR, HD < 0.0010
- Frequencies of 2-grams ending in H
  - WH 0.0026
  - EH, LH, OH, RH, DH ≤ 0.0002
- Implies E follows H

Example

- Arrange so the H and E are adjacent

```
HE
LL
OW
OR
LD
```

- Read across, then down, to get original plaintext

Substitution Ciphers

- Change characters in plaintext to produce ciphertext
- Example (Caesar cipher)
  - Plaintext is HELLO WORLD
  - Change each letter to the third letter following it (X goes to A, Y to B, Z to C)
    - Key is 3, usually written as letter 'D'
  - Ciphertext is KHOOR ZRUOG

Attacking the Cipher

- Exhaustive search
  - If the key space is small enough, try all possible keys until you find the right one
  - Caesar cipher has 26 possible keys
- Statistical analysis
  - Compare to 1-gram model of English

Statistical Attack

- Compute frequency of each letter in ciphertext:

`G`	0.1	`H`	0.1	`K`	0.1	`O`	0.3
`R`	0.2	`U`	0.1	`Z`	0.1		

- Apply 1-gram model of English
  - Frequency of characters (1-grams) in English is on next slide

Character Frequencies

a	0.07984	h	0.06384	n	0.06876	t	0.09058
b	0.01511	i	0.07000	o	0.07691	u	0.02844
c	0.02504	j	0.00131	p	0.01741	v	0.01056
d	0.04260	k	0.00741	q	0.00107	w	0.02304
e	0.12452	l	0.03961	r	0.05912	x	0.00159
f	0.02262	m	0.02629	s	0.06333	y	0.02028
g	0.02013					z	0.00057

Statistical Analysis

- $f(c)$ frequency of character c in ciphertext
- $\varphi(i)$ correlation of frequency of letters in ciphertext with corresponding letters in English, assuming key is i
  - $\varphi(i) = \Sigma_{0 \le c \le 25} f(c)p(c - i)$ so here,

    $\varphi(i) = 0.1\ p(6 - i) + 0.1\ p(7 - i) + 0.1\ p(10 - i) + 0.3\ p(14 - i) + 0.2\ p(17 - i) + 0.1\ p(20 - i) + 0.1\ p(25 - i)$
    - $p(x)$ is frequency of character x in English

Correlation: φ(*i*) for $0 \le i \le 25$

i	**φ(*i*)**	***i***	**φ(*i*)**	***i***	**φ(*i*)**	***i***	**φ(*i*)**
0	0.0469	7	0.0461	13	0.0505	19	0.0312
1	0.0393	8	0.0194	14	0.0561	20	0.0287
2	0.0396	9	0.0286	15	0.0215	21	0.0526
3	0.0586	10	0.0631	16	0.0306	22	0.0398
4	0.0259	11	0.0280	17	0.0386	23	0.0338
5	0.0165	12	0.0318	18	0.0317	24	0.0320
6	0.0676					25	0.0443

The Result

- Most probable keys, based on φ:
  - $i = 6$, $\varphi(i) = 0.0676$
    - plaintext `EBIIL TLOLA`
  - $i = 10$, $\varphi(i) = 0.0631$
    - plaintext `AXEEH PHKEW`
  - $i = 14$, $\varphi(i) = 0.0561$
    - plaintext `WTAAD LDGAS`
  - $i = 3$, $\varphi(i) = 0.0586$
    - plaintext `HELLO WORLD`
- Only English phrase is for $i = 3$
  - That's the key (3 or 'D')

Caesar's Problem

- Key is too short
  - Can be found by exhaustive search
  - Statistical frequencies not concealed well
    - They look too much like regular English letters
- So make it longer
  - Multiple letters in key
  - Idea is to smooth the statistical frequencies to make cryptanalysis harder

Vigènere Cipher

- Like Caesar cipher, but use a phrase
  - So it's effectively multiple Caesar ciphers
- Example
  - Message `A LIMERICK PACKS LAUGHS ANATOMICAL`
  - Key `BENCH`
  - Encipher using Caesar cipher for each letter:

```
key    BENCHBENCHBENCHBENCHBENCHBENCH
plain  ALIMERICKPACKSLAUGHSANATOMICAL
cipher BPVOLSMPMWBGXUSBYTJZBRNVVNMPCS
```

Relevant Parts of Tableau

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

- Tableau shown has relevant rows, columns only
  - Columns correspond to letters from the key
  - Rows correspond to letters from the message
- Example encipherments:
  - key `B`, letter `R`: follow `B` column down to `R` row (giving “`S`”)
  - Key `H`, letter `L`: follow `H` column down to `L` row (giving “`S`”)

Useful Terms

- *period*: length of key
  - In earlier example, period is 3
- *tableau*: table used to encipher and decipher
  - Vigènere cipher has key letters on top, plaintext letters on the left
- *polyalphabetic*: the key has several different letters
  - Caesar cipher is monoalphabetic

Attacking the Cipher

- Approach
  - Establish period; call it *n*
  - Break message into *n* parts, each part being enciphered using the same key letter
  - Solve each part; you can leverage one part from another
- We will show each step

The Target Cipher

- We want to break this cipher:

```
ADQYS MIUSB OXKKT MIBHK IZOOO EQOOG IFBAG KAUMF
VVTAA CIDTW MOCIO EQOOG BMBFV ZGGWP CIEKQ HSNEW
VECNE DLAAV RWKXS VNSVP HCEUT QOIOF MEGJS WTPCH
AJMOC HIUIX
```

Establish Period

- Kasiski: *repetitions in the ciphertext occur when characters of the key appear over the same characters in the plaintext*
- Example:

```
key    VIGVIGVIGVIGVIGV
plain  THEBOYHASTHEBALL
cipher OPKWWECIYOPKWIRG
```

Note the key and plaintext line up over the repetitions (underlined). As distance between repetitions is 9, the period is a factor of 9 (that is, 1, 3, or 9)

Repetitions in Example

Letters	Start	End	Gap Length	Gap Length Factors
OEQOOG	24	54	30	2, 3, 5
MOC	50	122	72	2, 2, 2, 3, 3

Estimate of Period

- OEQOOG is probably not a coincidence
  - It’s too long for that
  - Period may be 1, 2, 3, 5, 6, 10, 15, or 30
- MOC is also probably not a coincidence
  - Period may be 1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, or 72
- Period of 2 or 3 is probably too short (but maybe not)
- Begin with period of 6

Check on Period

- Index of coincidence is probability that two randomly chosen letters from ciphertext will be the same
- Tabulated for different periods:

1	0.0660
2	0.0520
3	0.0473
6	0.0427

Compute IC for an Alphabet

- IC = $[n (n - 1)]^{-1} \sum_{0 \le i \le 25} [F_i (F_i - 1)]$
    - where n is length of ciphertext and F_i the number of times character i occurs in ciphertext
- For the given ciphertext, IC = 0.0433
    - Indicates a key of length 5 or 6
    - A statistical measure, so it can be in error, but it agrees with the previous estimate (which was 6)

Splitting Into Alphabets

alphabet 1: AIKHOIATTOBGEEERNEOSAI

alphabet 2: DUKKEFUAWEMGKWDWSUFWJU

alphabet 3: QSTIQBMAMQBWQVLKVTMTMI

alphabet 4: YBMZOAFCOOFPHEAXPQEPOX

alphabet 5: SOIOOGVICOVCSVASHOGCC

alphabet 6: MXBOGKVDIGZINNVVCIJHH

- ICs (#1, 0.0692; #2, 0.0779; #3, 0.0779; #4, 0.0562; #5, 0.1238; #6, 0.0429) indicate all alphabets have period 1, except #4 (between 1 and 2) and #6 (between 5 and 6); assume statistical variance

Frequency Examination

#	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
1	3	1	0	0	4	0	1	1	3	0	1	0	0	1	3	0	0	1	1	2	0	0	0	0	0	0
2	1	0	0	2	2	2	1	0	0	1	3	0	1	0	0	0	0	0	1	0	4	0	4	0	0	0
3	1	2	0	0	0	0	0	0	2	0	1	1	4	0	0	0	4	0	1	3	0	2	1	0	0	0
4	2	1	1	0	2	2	0	1	0	0	0	0	1	0	4	3	1	0	0	0	0	0	0	2	1	1
5	1	0	5	0	0	0	2	1	2	0	0	0	0	0	5	0	0	0	3	0	0	2	0	0	0	0
6	0	1	1	1	0	0	2	2	3	1	1	0	1	2	1	0	0	0	0	0	0	3	0	1	0	1
	H	M	M	M	H	M	M	H	H	M	M	M	M	H	H	M	L	H	H	H	M	L	L	L	L	L

The last row has general letter frequencies (H high, M medium, L low)

Begin Decryption

- First matches characteristics of unshifted alphabet
- Third matches if `I` shifted to `A`
- Sixth matches if `V` shifted to `A`
- Substitute into ciphertext (bold are substitutions)

AD**I**YS **RI**U**K**B O**CK**K**L** MI**GH**K **A**ZO**TO** E**I**OO**L** **I**F**T**AG
PAU**E**F V**AT**A**S** CI**IT**W **E**OC**NO** E**I**OO**L** **B**M**T**FV **EG**G**O**P
C**NE**K**I** HS**SE**W **N**EC**SE** D**D**AA**A** **R**W**C**XS **AN**S**N**P H**HE**U**L**
QO**NO**F **E**EG**OS** W**L**PC**M** **A**J**E**OC **MI**U**A**X

Look For Clues

- **A**J**E** in last line suggests "are", meaning second alphabet maps A into S:

ALIYS **RICK**B O**CKSL** MI**GHS** **A**ZO**TO** **MI**OO**L** **INT**AG

PACEF V**ATIS** CI**ITE** **E**OC**NO** **MI**OO**L** **BUT**FV **EGOO**P

C**NESI** HS**SEE** **N**EC**SE** **LD**AA**A** **REC**XS **ANAN**P H**HECL**

QO**NON** **E**EG**OS** **EL**PC**M** **ARE**OC **MICA**X

Next Alphabet

- **MICA**X in last line suggests “mical” (a common ending for an adjective), meaning fourth alphabet maps O into A:

ALIMS **RICKP** O**CKSL** **A**I**GHS** **AN**O**TO** **MIC**O**L** **INTO**G

PACET V**ATIS** **Q**I**ITE** **EC**C**NO** **MIC**O**L** **BUTT**V **EGOOD**

C**NESI** **V**S**SEE** **NS**C**SE** **LDO**A**A** **RECL**S **ANAND** H**HECL**

EO**NON** **ES**G**OS** **ELD**C**M** **AREC**C **MICAL**

Got It!

- **Q**I means that U maps into I, as Q is always followed by U:

  ALIME RICKP ACKSL AUGHS ANATO MICAL INTOS
  PACET HATIS QUITE ECONO MICAL BUTTH EGOOD
  ONESI VESEE NSOSE LDOMA RECLE ANAND THECL
  EANON ESSOS ELDOM ARECO MICAL

One-Time Pad

- A Vigenère cipher with a random key at least as long as the message
  - Provably unbreakable
  - Why? Look at ciphertext `DXQR`. Equally likely to correspond to plaintext `DOIT` (key `AJIY`) and to plaintext `DONT` (key `AJDY`) and any other 4 letters
  - Warning: keys *must* be random, or you can attack the cipher by trying to regenerate the key
    - Approximations, such as using pseudorandom number generators to generate keys, are *not* random

Overview of the DES

- A block cipher:
  - encrypts blocks of 64 bits using a 64 bit key
  - outputs 64 bits of ciphertext
- A product cipher
  - basic unit is the bit
  - performs both substitution and transposition (permutation) on the bits
- Cipher consists of 16 rounds (iterations) each with a 48 bit round key generated from the user-supplied key

Structure of the DES

- Input is first permuted, then split into left half (L) and right half (R), each 32 bits
- Round begins; R and round key run through function *f*, then xor'ed with L
  - *f* expands R to 48 bits, xors with round key, and then each 6 bits of this are run through S-boxes (substitution boxes), each of which gives 4 bits of output
  - Those 32 bits are permuted and this is the output of *f*
- R and L swapped, ending the round
  - Swapping does not occur in the last round
- After last round, L and R combined, permuted, forming DES output