Cryptanalysis
(암호분석)

Simple Ciphers

2020.3

Contents

- Introduction
- Monoalphabetic ciphers
- Polyalphabetic ciphers
- Breaking monoalphabetic/polyalphabetic ciphers

Introduction

- The Topic of Cryptanalysis
  - Rapidly developed in the past 30 years with more powerful computers
    → much to learn
  - Important papers are written with many diverse goals and audiences
    → difficult to understand

"There is only one way to become a good cryptanalyst – to practice breaking codes."

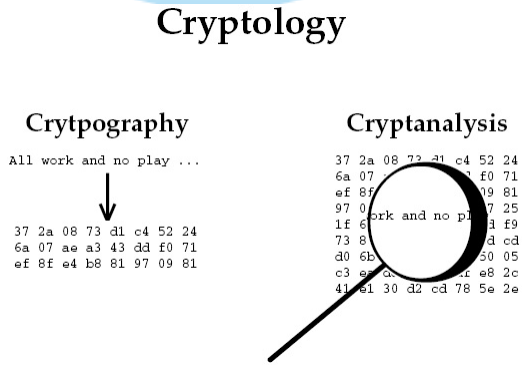
(Bruce Schneier, 2000)

Concepts of security

- Security
  - freedom from danger, risk, and loss particularly related to information
  - violations of security: to steal credit card information, SSN, bank account, login passwords, etc.

- Information security
  - keeps information free from danger of being exposed to unauthorized parties
  - Three principles: **confidentiality, integrity, availability**
  - Additional requirement: authenticity

Cryptology



- Cryptology
  - Science of secure or confidential exchange of information

Cryptology = Cryptography + Cryptanalysis

- Cryptography
  - science of understanding , implementing, and using information obfuscation techniques
  - techniques: cryptographic codes, cryptosystems, ciphers, encryption/decryption

- Cryptanalysis
  - study of defeating/strengthening cryptographic techniques
  - finding, exploiting, and correcting weaknesses

Principles of good cryptography

- by Claude Shannon

- The amount of security necessary should dictate how much effort we put into securing or encrypting our data.
  - Do not pay more for the security than the value of the information.
- The size of the ciphertext should be less than or equal to the size of the plaintext.
  - The longer(more bits) ciphertext may contain, more room there to derive the original information.
- The cryptographic system should be simple.
  - A lot of complexity makes lots of room for errors.
- Errors should not propagate. (old concept)
  - A transmission error should have as limited an impact as possible.

Other principles

- **security by obscurity**
  - It is possible that, without enough people evaluating the cryptosystem, there will be undiscovered errors in the algorithm.
- **Kerckhoff's principle**
  - A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

Simple ciphers

- Monoalphabetic ciphers
  - Caesar cipher
  - ROT13
  - single-letter substitution cipher
- Polyalphabetic ciphers
  - Vigenere Tableau
  - Transposition cipher(permutation cipher)

Monoalphabetic Ciphers

- Encryption/Decryption method
  - one letter is replaced by a single new alphabet
  - using a single lookup table

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

Caesar cipher

Monoalphabetic cipher – Keying

- Caesar cipher
  - 26 different keys possible (eg: key=+3)
- ROT13
  - Caesar cipher with a shift of key=13
  - The encryption and decryption operations are identical.
- Keyed Alphabets
  - Choose a keyword (eg: swordfish) and appending remaining alphabets as a substitution table
  - Disadvantages: relation (keyword ↔ ABCDE…), invariant part (at the end of alphabets)

swordfihabcegjklmnpqtuvxyz

- single-letter substitution
  - possible keys: $26! \approx 2^{89}$

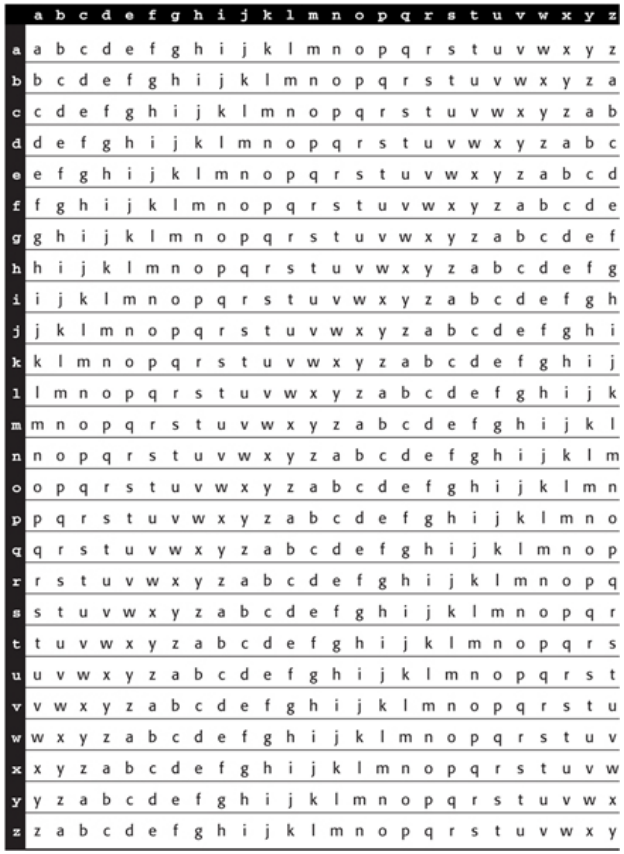
Polyalphabetic ciphers

- Improving monoalphabetic ciphers
  - needs more powerful encryption schemes
  - without increasing the complexity too much

- Polyalphabetic cipher (Naïve version)
  - Simply repeating a monoalphabetic cipher k times
  - Needs to share k tables for $(26!)^k$ different keys
  - Hard to memorize or manage the shared contents of tables

Vigenere Cipher

Vigenere Tableau



	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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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

- Vigenere cipher
  - Uses the Vigenere tableau (open to public)
  - Password (encryption key) selects rows for single-letter substitutions
- Advantages
  - Ciphertext is statistically random enough
  - Key is easy to remember
  - eg: key = **caesar**

plaintext: `the quick brown roman fox jumped over the lazy ostrogoth dog`

ciphertext: `VHI IUZEK FJONP RSEAE HOB BUDREH GVVT TLW LRBY SKTIQGSLH UQG`

Transposition ciphers

- Transposition vs Substitution
  - Substitution: characters are substituted for different ones
  - Transposition: characters are shuffled. The contents are preserved and changed in order.

- Security of transposition ciphers
  - The encryption mechanism cannot be so obvious.

  - eg: simplest scheme (reversing)

plaintext: `cryptology`  ciphertext: `ygolotpyrc`

Columnar transpositions

- Columnar transposition cipher
  - Set the row length as k (key)
  - Write plaintext to fill up a line of rectangle (row length = k)
  - Read the text(ciphertext) from top to bottom, left to right (often spaces can be removed)

  - eg: row length $k = 6$ (key)

plaintext: `all work and no play makes johnny a dull boy`

ciphertext: `AKPKNL LALENL LNASYB WDYJAO ONMODY ROAHU`

$k = 6$ (key)

```
a l l w o r
k a n d n o
p l a y m a
k e s j o h
n n y a d u
l l b o y
```

Double columnar transpositions

- Security problem of columnar transposition
  - easy to guess the key (enumerating all possibilities for k)

- Double columnar transposition cipher
  - Uses 2 columnar transpositions one right after the other
  - Encryption key = (k_1, k_2) for each columnar transposition
  - eg: key = $(k_1, k_2) = (6, 8)$

plaintext: `all work and no play makes johnny a dull boy`

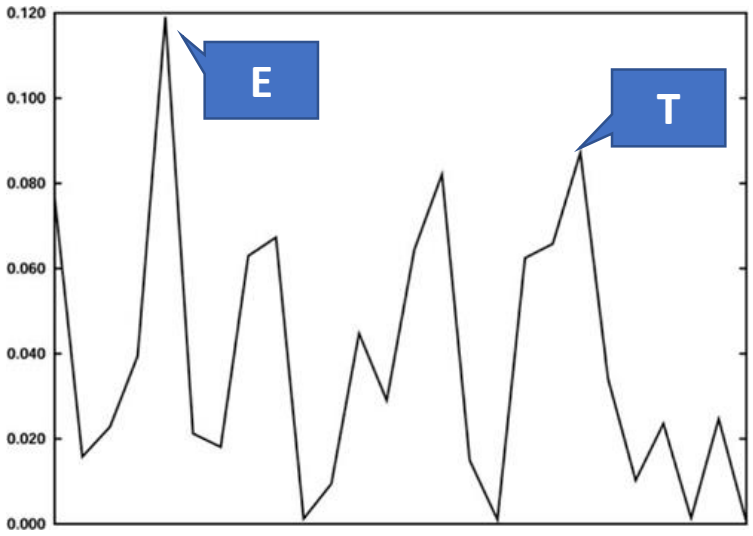
```
allwor
kandno
playma
kesjoh
nnyadu
llboy
```
$(k_1 = 6)$

```
akpknlla
lenllnas
ybwdyjao
onmodyro
ahu
```
$(k_2 = 8)$

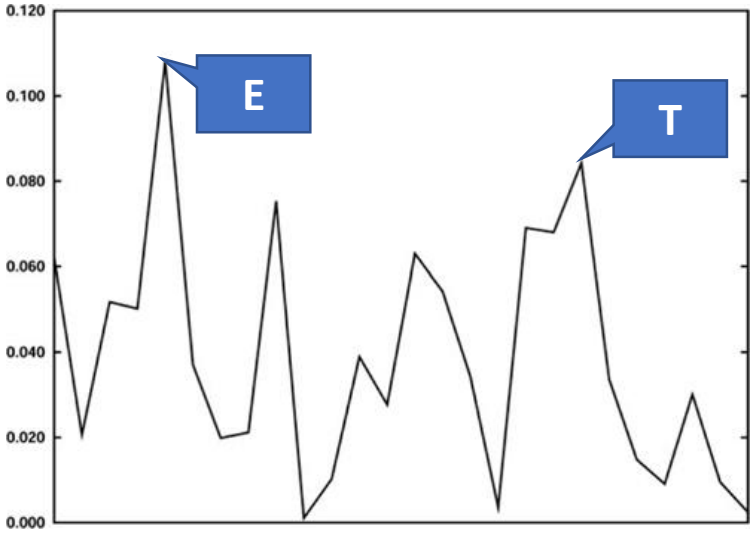
ciphertext: `ALYOA KEBNH PNWMU KLDON LYDLN JYLAA RASOO`

Breaking monoalphabetic ciphers

- Frequency analysis
  - Counting how often individual letters appear in the ciphertext
  - Monoalphabetic ciphers preserve the frequencies (only shuffle the alphabets)



The works of William Shakespeare



Linux source code (205MB)

Breaking monoalphabetic ciphers

- Index of coincidence I_c
  - A measure of how evenly distributed the character frequencies are.
  - How likely it is to draw two matching letters by randomly selecting two letters from a given text.

$$I_c = \sum_{c \in \text{alphabet}} \frac{n_c(n_c-1)}{N(N-1)} = \frac{n_a(n_a-1)}{N(N-1)} + \frac{n_b(n_b-1)}{N(N-1)} + \cdots + \frac{n_z(n_z-1)}{N(N-1)}$$

  - A monoalphabetic cipher preserves the index of coincidence. ($I_c(\text{plaintext}) = I_c(\text{ciphertext})$)
  - The lower, the more evenly distributed.
    (The theoretical minimum is $I_c = \lim_{n\to\infty} \sum \frac{n(n-1)}{26n(26n-1)} = \frac{1}{26} \approx 0.038$)
  - eg:
    - The complete works of Shakespeare: $I_c \approx 0.0639$
    - Linux kernel source code: $I_c \approx 0.0583$

Breaking polyalphabetic ciphers

- Attack strategy for Vigenere cipher
  - Guess the key length
  - Then break the ciphertext into a smaller set of monoalphabetic ciphertexts
  - Derive the key for each cipher using frequency analysis

- How to guess the key length

  (method 1) Using the index of coincidence

  (method 2) Analysis of the distance among the repeated occurrences of a common word (eg. *the*)

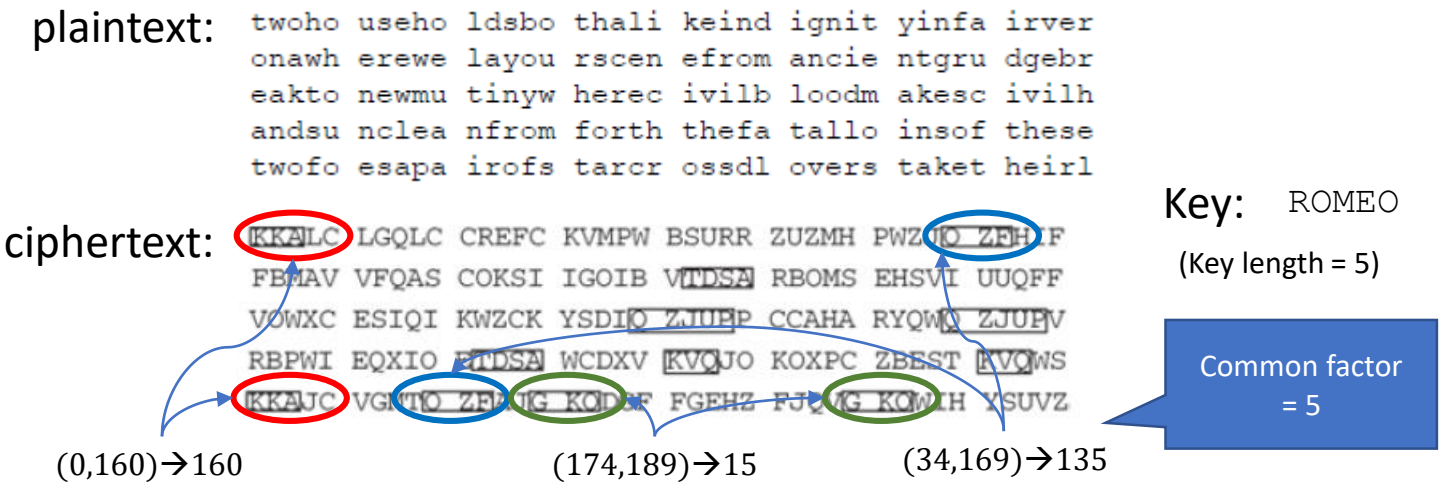
Guessing the key length

- (Method 1) using the index of coincidence
  - use the relation between the key length of Vigenere cipher and resulting index of coincidence of the ciphertext

KEY LENGTH	APPROXIMATE I_C
1	0.0639
2	0.0511
3	0.0468
4	0.0446
5	0.0438
6	0.0426
7	0.0423
8	0.0417
9	0.0412
10	0.0410
...	...
∞	0.0384

Guessing the key length

- (Method 2) using repeated occurrences of common words
  - A common word appears multiple times in the plaintext.
  - Some of them are encrypted to the same word in the ciphertext (at least two).
  - Compute the common factor of all the distances between them.
  - eg:



Breaking Transposition Ciphers

- Attack Columnar Transposition Ciphers
  - The key space(the number of possible keys) is small.
  - Attack on digraph and trigraph analysis (use the probabilities of most common digraphs).

DIGRAPH	PROBABILITY	TRIGRAPH	PROBABILITY
th	3.16%	the	1.45%
he	2.28%	and	0.87%
an	1.63%	you	0.58%
er	1.62%	her	0.53%
ou	1.47%	hat	0.50%
in	1.45%	tha	0.48%
ha	1.27%	ing	0.48%
es	1.27%	eth	0.41%
nd	1.24%	our	0.40%
st	1.24%	his	0.38%
re	1.24%	thi	0.37%
en	1.19%	for	0.35%
ea	1.14%	ere	0.34%
or	1.07%	ith	0.33%
at	1.02%	ent	0.32%
is	1.01%	oth	0.31%

Attack Columnar Transposition Ciphers

- Sliding window technique
  - Finding adjacent positions for digraphs in the plaintext.
  - eg:

plaintext: `all work and no play makes johnny a dull boy`

ciphertext: `AKPKNL LALENL LNASYB WDYJAO ONMODY ROAHU`

$k = 6$ (key)

a	l	l	w	o	r
k	a	n	d	n	o
p	l	a	y	m	a
k	e	s	j	o	h
n	n	y	a	d	u
l	l	b	o	y	

p_0	p_1	p_2	p_3	p_4	p_5
p_6	p_7	p_8	p_9	p_{10}	p_{11}
p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}
p_{18}	p_{19}	p_{20}	p_{21}	p_{22}	p_{23}
p_{24}	p_{25}	p_{26}	p_{27}	p_{28}	p_{29}
p_{30}	p_{31}	p_{32}	p_{33}	p_{34}	

c_0	c_6	c_{12}	c_{18}	c_{24}	c_{30}
c_1	c_7	c_{13}	c_{19}	c_{25}	c_{31}
c_2	c_8	c_{14}	c_{20}	c_{26}	c_{32}
c_3	c_9	c_{15}	c_{21}	c_{27}	c_{33}
c_4	c_{10}	c_{16}	c_{22}	c_{28}	c_{34}
c_5	c_{11}	c_{17}	c_{23}	c_{29}	

c_0		p_0	
c_1		p_6	
c_2		p_{12}	
c_3		p_{18}	
c_4		p_{24}	
c_5		p_{30}	
c_0	c_6	p_0	p_1
c_1	c_7	p_6	p_7
c_2	c_8	p_{12}	p_{13}
c_3	c_9	p_{18}	p_{19}
c_4	c_{10}	p_{24}	p_{25}
c_5	c_{11}	p_{30}	p_{31}
c_6	c_{12}	p_1	p_2
c_7	c_{13}	p_7	p_8
c_8	c_{14}	p_{13}	p_{14}
c_9	c_{15}	p_{19}	p_{20}
c_{10}	c_{16}	p_{25}	p_{26}
c_{11}	c_{17}	p_{31}	p_{32}
c_{12}	c_{18}	p_2	p_3
c_{13}	c_{19}	p_8	p_9
c_{14}	c_{20}	p_{14}	p_{15}
c_{15}	c_{21}	p_{20}	p_{21}
c_{16}	c_{22}	p_{26}	p_{27}
c_{17}	c_{23}	p_{32}	p_{33}
c_{18}	c_{24}	p_3	p_4
c_{19}	c_{25}	p_9	p_{10}
c_{20}	c_{26}	p_{15}	p_{16}
c_{21}	c_{27}	p_{21}	p_{22}
c_{22}	c_{28}	p_{27}	p_{28}
c_{23}	c_{29}	p_{33}	p_{34}
c_{24}	c_{30}	p_4	p_5
c_{25}	c_{31}	p_{10}	p_{11}
c_{26}	c_{32}	p_{16}	p_{17}
c_{27}	c_{33}	p_{22}	p_{23}
c_{28}	c_{34}	p_{28}	p_{29}
c_{29}		p_{34}	
c_{30}		p_5	
c_{31}		p_{11}	
c_{32}		p_{17}	
c_{33}		p_{23}	
c_{34}		p_{29}	