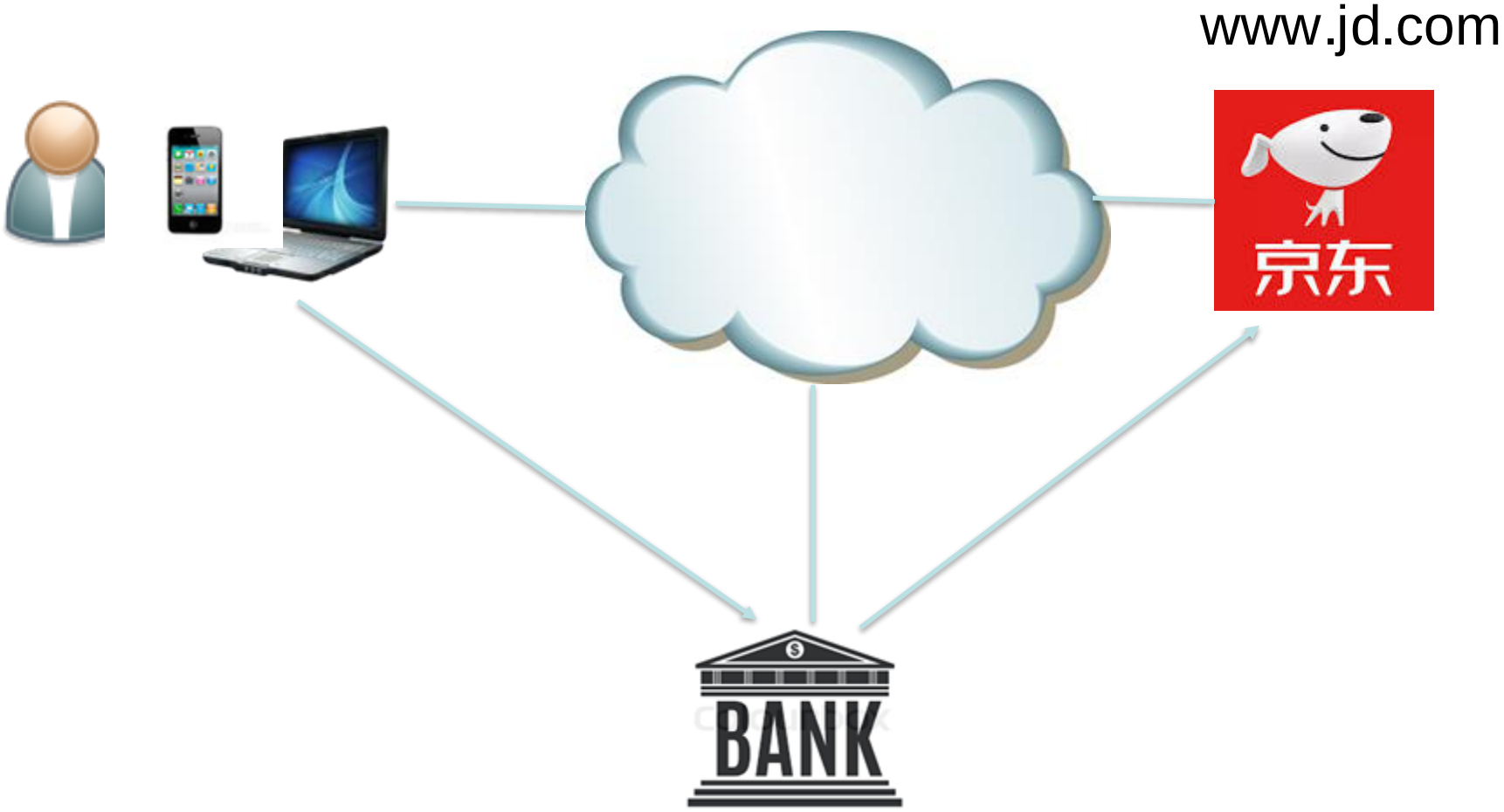
密码学基础与应用

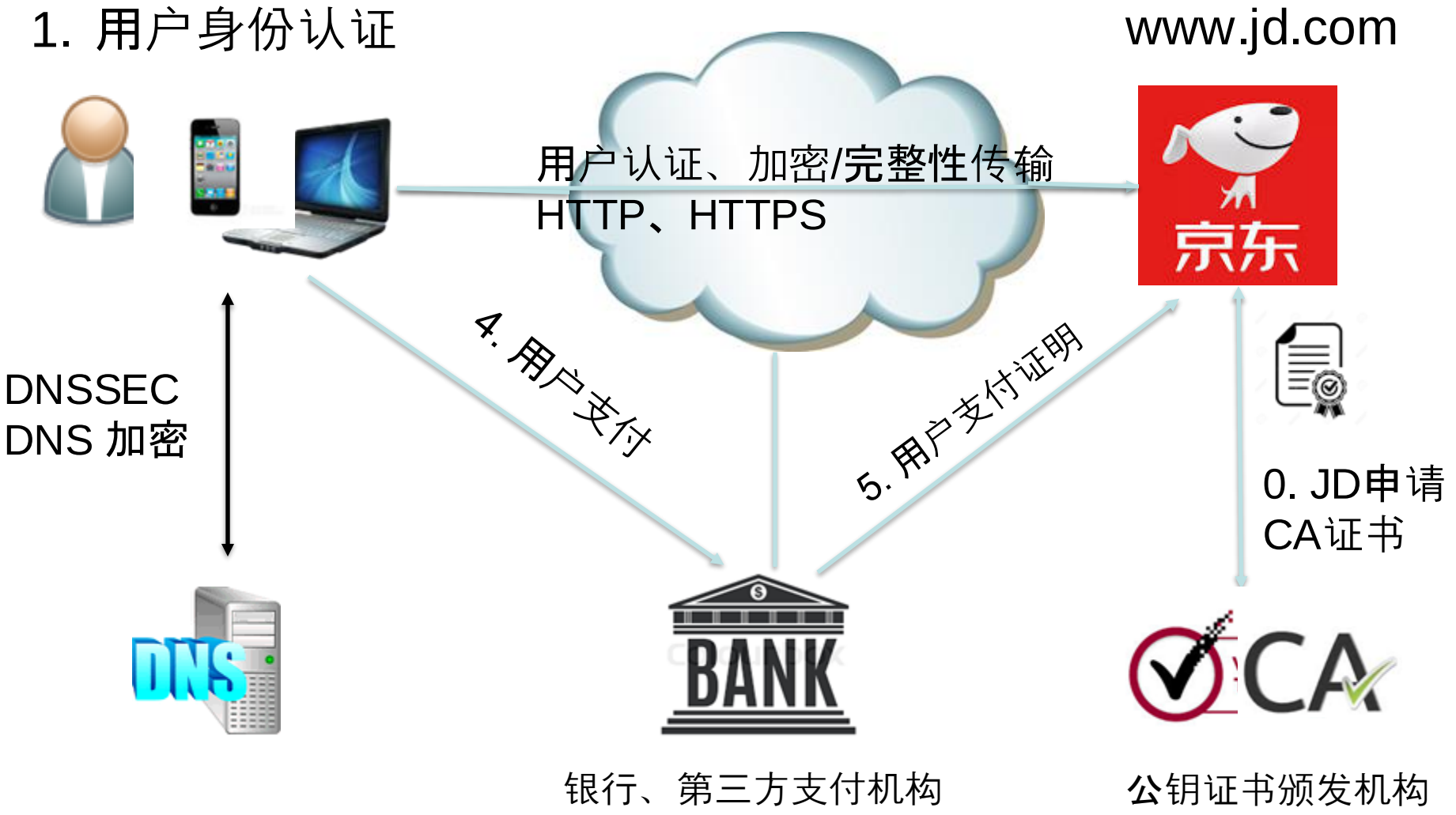
Haixin Duan

安全通信的应用场景：网上购物



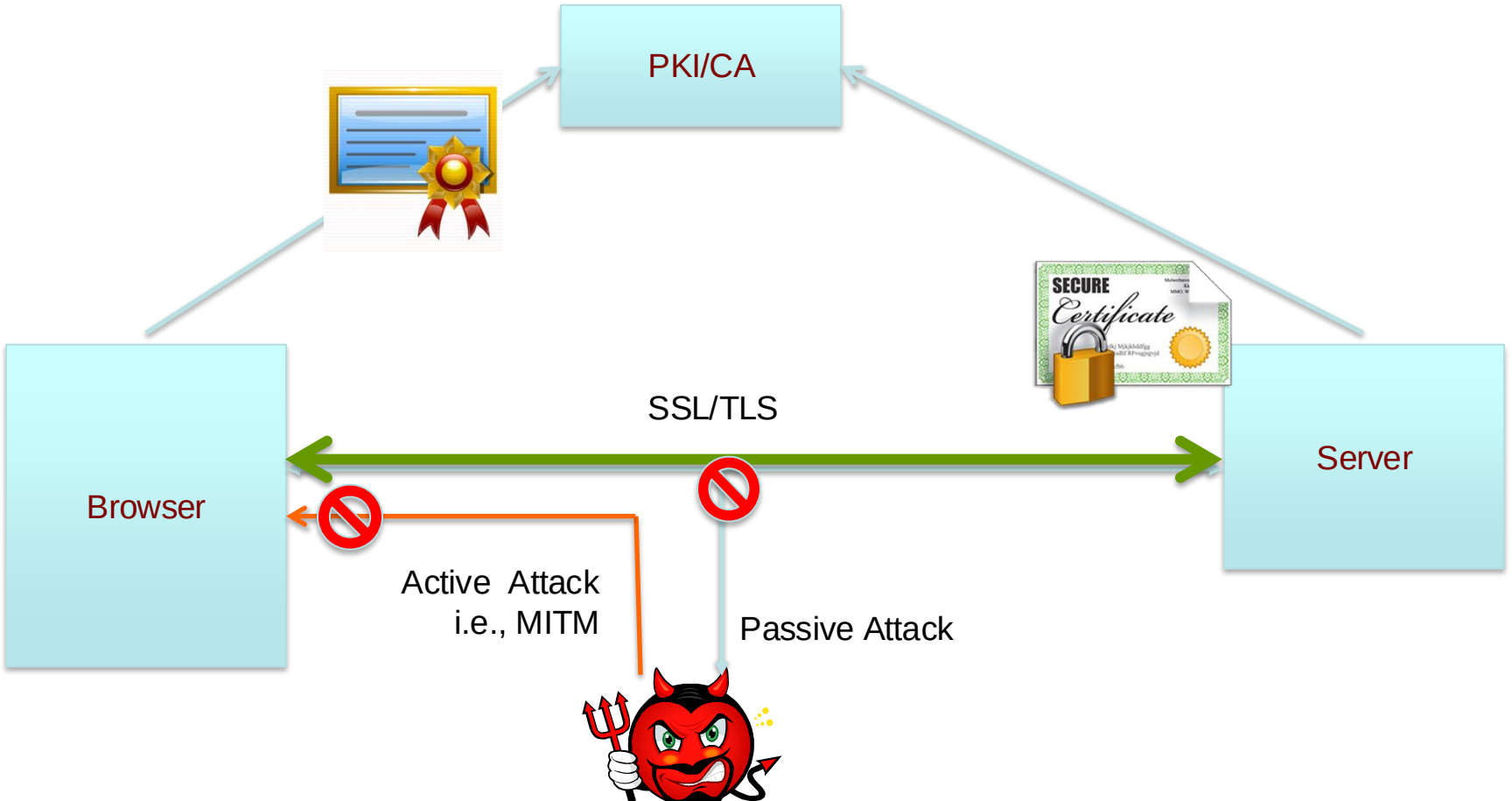
银行、第三方支付机构

安全通信的应用场景：网上购物

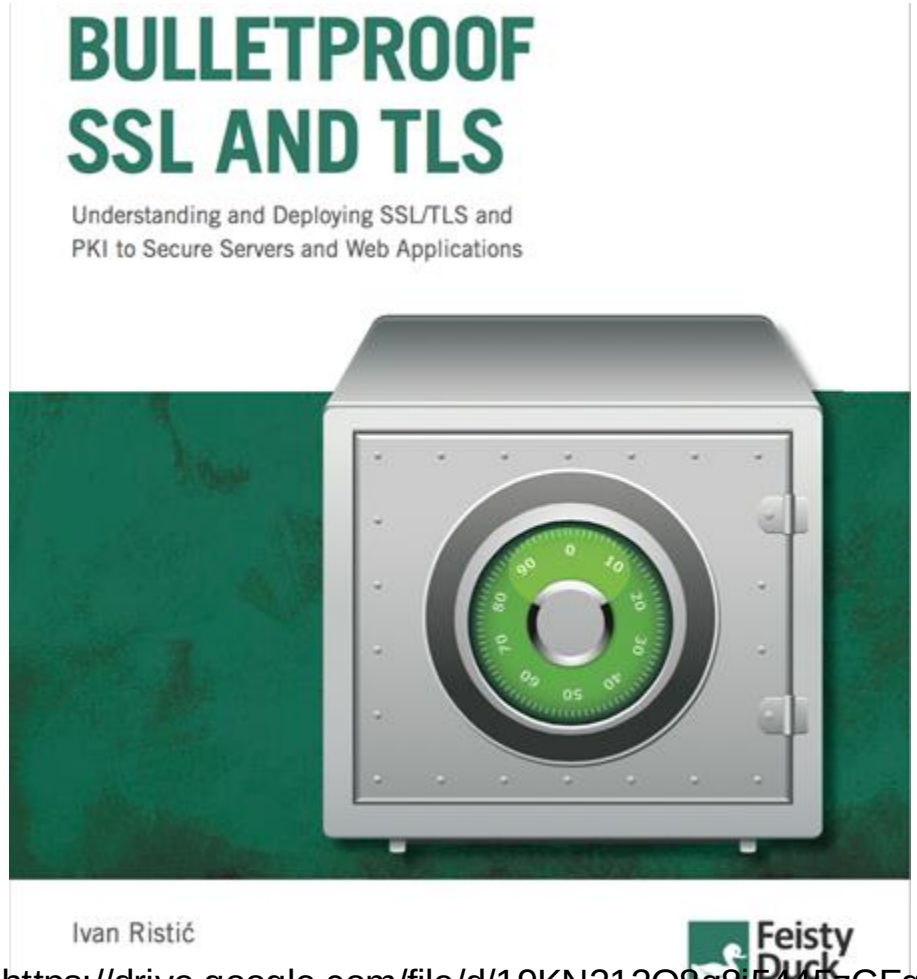


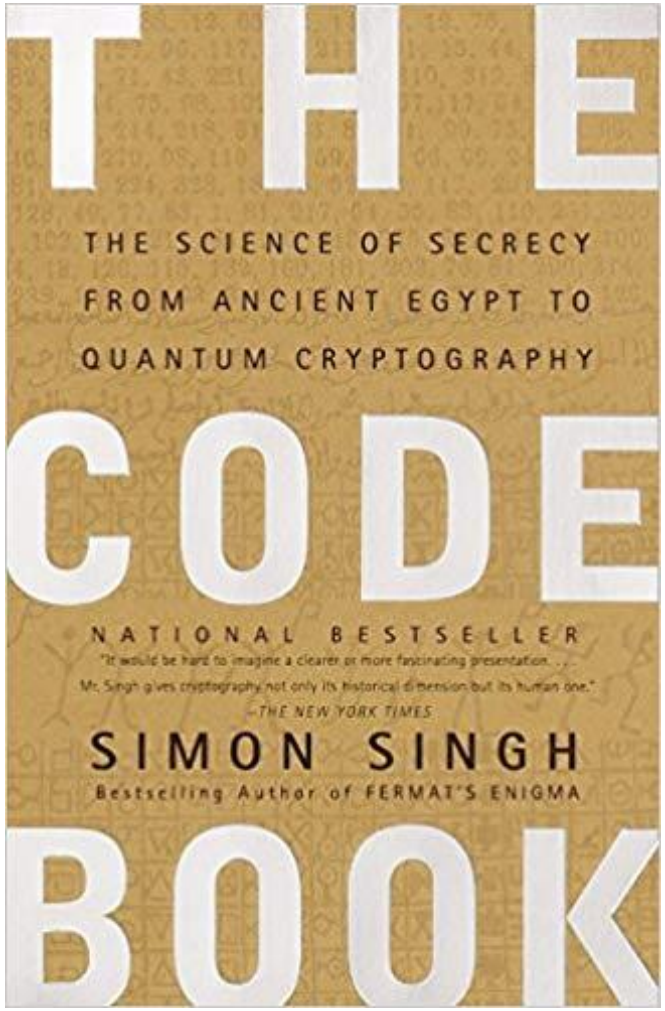
Build/Understand it ,Break it and Fix it

- 理解当前最流行的安全通信协议TLS，它是怎么工作的？又是如何被攻击的？如何改进的？



参考书





https://drive.google.com/file/d/19KN212O8g8i5445zCFgd-Ysv-jnzYs8K/view?usp=drive_link

https://simonsingh.net/books/the-code-book/

Content

- Cryptography overview
- PKI and PKI attacks
- TLS overview
- TLS attacks

从凯撒密码到量子通信

实例分析密码技术的演进

Duan Haixin

Encryption and decryption example

- DefCon CTF 2009 qualifier, http://shallweplayaga.me/crypto/
- Question

ASI JL DUJZTED SA J EJZD JVV NBTODI, VDD FOD AHB VBFD:

OFFT://YYY.AHB.MSK/TJMD2/CDN08/NSCD_122908.OFZE,

PSQD GSW MWDVV? LS, JNFWJE AHB RWBX.

Encryption and decryption example

ASI JL DUJZTED SA J EJZD JVV NBTODI, VDD FOD AHB VBFD: **OFFT**://**YYY.**AHB.MSK/TJMD2/CDN08/NSCD_122908.**OFZE**, PSQD GSW MWDVV? LS, JNFWJE AHB RWBX.

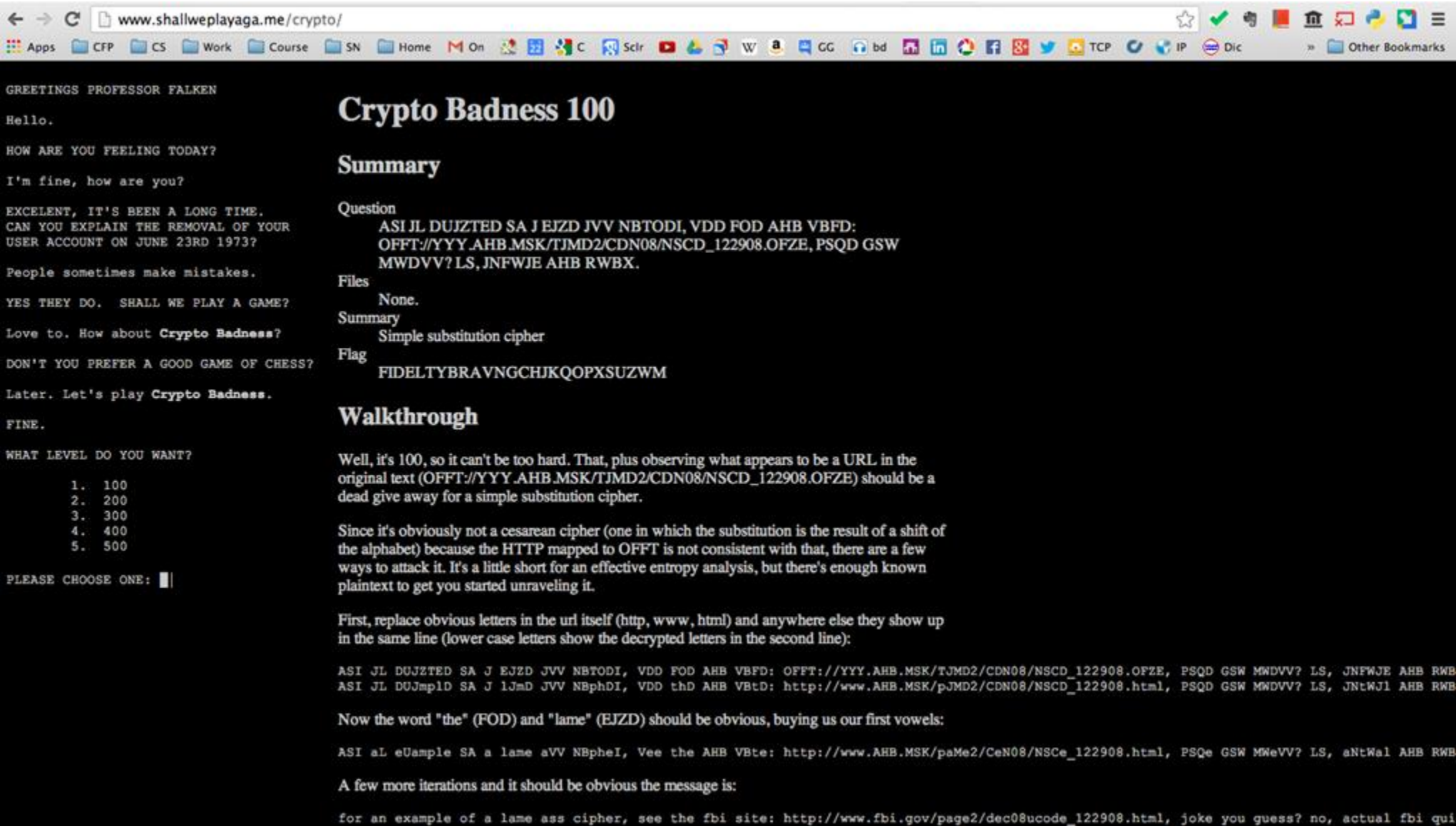
Cipher: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Plaintext: L T H P W M

ASI JL DUJmplD SA **J** LJmD JVV NBpODI, V**DD thD** AHB VBtD: http://www.AHB.MSK/pJMD2/CDN08/NSCD_122908.html, PSQE GSW MWDVV? LS, JNtWJl AHB RWBX.

Cipher: 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
Plaintext: E L T H P W M

for an example of **a** lame ass cipher, see **the** fbi site: HTTP://WWW.fbi.MoK/PaMe2/Cec08/coCe_122908.HTML, PSQl GSW MWess? LS, aNtWaL fbi RWiX.

Cipher: 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
Plaintext: F I E L T BRA C H O P X S W MA



http://shallweplayaga.me/crypto/

Content

- 古典密码（Classical cipher）：
  - 代换（Substitution）
  - 换位（Transposition）
- 密码算法和破解
  - mono-alphabetic substitution
  - Polyalphabetic substitution: Vigenère cipher
  - Cipher in World War I
    - ADFGVX CIPHERS
  - Cipher in World War II
    - Enigma and cracking
  - WEP in Wi-Fi

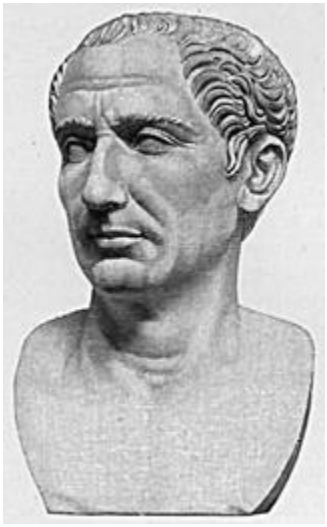
substitution cipher（替代，代换）

- mono-alphabetic substitution cipher
  - Caesar cipher

Code book

ABCDEFGHIGKLMNOPQRSTUVWXYZ

DEFGHIGKLMNOPQRSTUVWXYZABC



Gaius Iulius Caesar
前100年－前44年

明文　Caesar was a great soldier

密文　Fdhvdu zdv d juhdw vroglhu

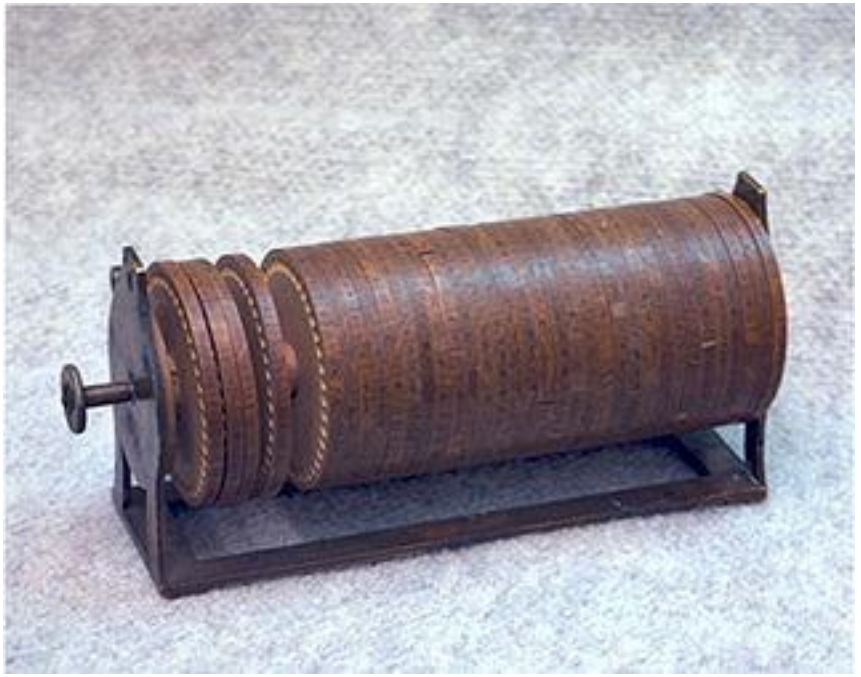
Encryption ： $C_i = (P_i + K) \text{ Mod } 26$

Decryption ： $P_i = (C_i - K) \text{ Mod } 26$

Brute force attack 蛮力攻击

Decryption shift	Candidate plaintext
0	exxegoexsrgi
1	dwwdfndwrqfh
2	cvvcemcvqpeg
3	buubdlbupodf
4	attackatonce
5	zsszbjzsnmbd
6	yrryaiyrmlac
...	
23	haahjrhavujl
24	gzzgiqgzutik
25	fyyfhpfytshj

Jefferson disk



Frequency analysis, 9th century by Al-Kindi

Abu Yūsuf Ya'qūb ibn 'Isḥāq aṣ-Ṣabbāḥ al-Kindī

艾布·优素福·叶尔孤白·本·伊斯哈格·本·萨巴赫·肯迪

中世纪阿拉伯的著名哲学家、自然科学家、音乐家、医生、黑客，亚里士多德学派的主要代表人物之一

The first page of al-Kindi's manuscript "On Deciphering Cryptographic Messages", containing the oldest known description of cryptanalysis by frequency analysis.

On Decrypting Encrypted Correspondence

One way to solve an encrypted message, if we know its language, is to find a different plaintext of the same language long enough to fill one sheet or so, and then we count the occurrences of each letter. **We call the most frequently occurring letter the "first", the next most occurring letter the "second",** the following most occurring letter the "third", and so on, until we account for all the different letters in the plaintext sample. Then we look at the cipher text we want to solve and we also classify its symbols. We find the most occurring symbol and change it to the form of the "first" letter of the plaintext sample, the next most common symbol is changed to the form of the "second" letter, and the following most common symbol is changed to the form of the "third" letter, and so on, until we account for all symbols of the cryptogram we want to solve.

Letter frequency

From Wikipedia, the free encyclopedia

Letter frequency is simply the amount of times letters of the alphabet appear on average in written language. Letter frequency analysis dates back to the Arab mathematician Al-Kindi (c. 801–873 AD), who formally developed the method to break ciphers. Letter frequency analysis gained importance in Europe with the development of movable type in 1450 AD, where one must estimate the amount of type required for each letterform. Linguists use letter frequency analysis as a rudimentary technique for language identification, where it's particularly effective as an indication of whether an unknown writing system is alphabetic, syllabic, or ideographic.

The use of letter frequencies and frequency analysis plays a fundamental role in cryptograms and several word puzzle games, including Hangman, Scrabble and the television game show *Wheel of Fortune*. One of the earliest descriptions in classical literature of applying the knowledge of English letter frequency to solving a cryptogram is found in Edgar Allan Poe's famous story *The Gold-Bug*, where the method is successfully applied to decipher a message instructing on the whereabouts of a treasure hidden by Captain Kidd.[1]

Letter frequencies also have a strong effect on the design of some keyboard layouts. The most frequent letters are on the bottom row of the Blickensderfer typewriter, and the home row of the Dvorak keyboard layout.

Contents [hide]

Background [edit]

The frequency of letters in text has been studied for use in cryptanalysis, and frequency analysis in particular, dating back to the Iraqi mathematician Al-Kindi (c. 801–873 AD), who formally developed the method (the ciphers breakable by this technique go back at least to the Caesar cipher invented by Julius Caesar, so this method could have been explored in classical times). Letter frequency analysis gained additional importance in Europe with the development of movable type in 1450 AD, where one must estimate the amount of type required for each letterform, as evidenced by the variations in letter compartment size in typographer's type cases.

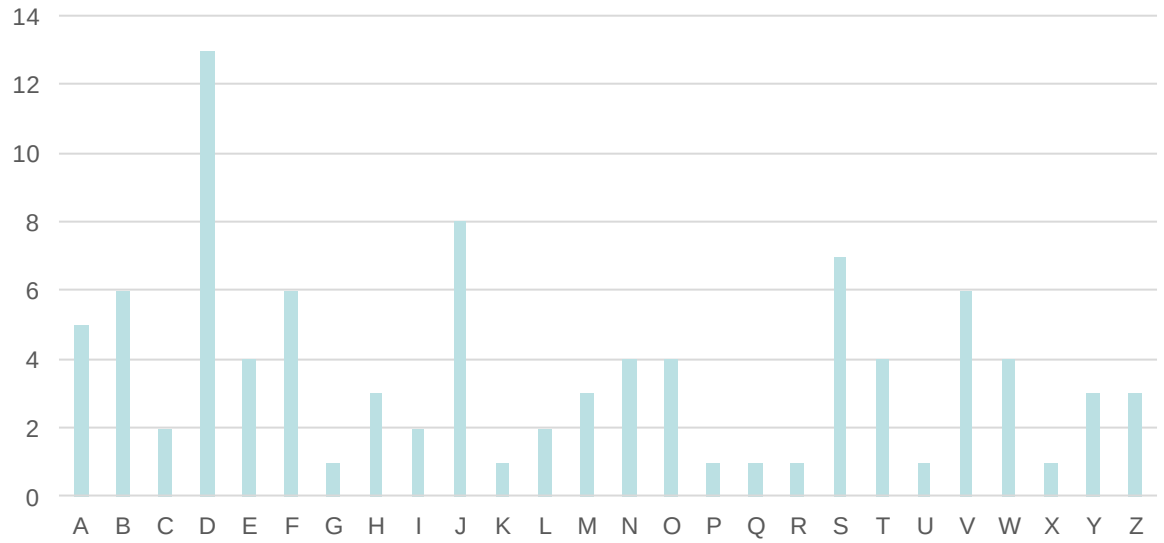
Letter	Relative frequency in the English language	
	Texts	Dictionaries
a	8.2%	7.8%
b	1.5%	2%
c	2.8%	4%
d	4.3%	3.8%
e	13%	11%
f	2.2%	1.4%
g	2%	3%
h	6.1%	2.3%
i	7%	8.6%
j	0.15%	0.21%
k	0.77%	0.97%
l	4%	5.3%
m	2.4%	2.7%
n	6.7%	7.2%
o	7.5%	6.1%
p	1.9%	2.8%
q	0.095%	0.19%
r	6%	7.3%
s	6.3%	8.7%
t	9.1%	6.7%
u	2.8%	3.3%
v	0.98%	1%
w	2.4%	0.91%
x	0.15%	0.27%
y	2%	1.6%
z	0.074%	0.44%

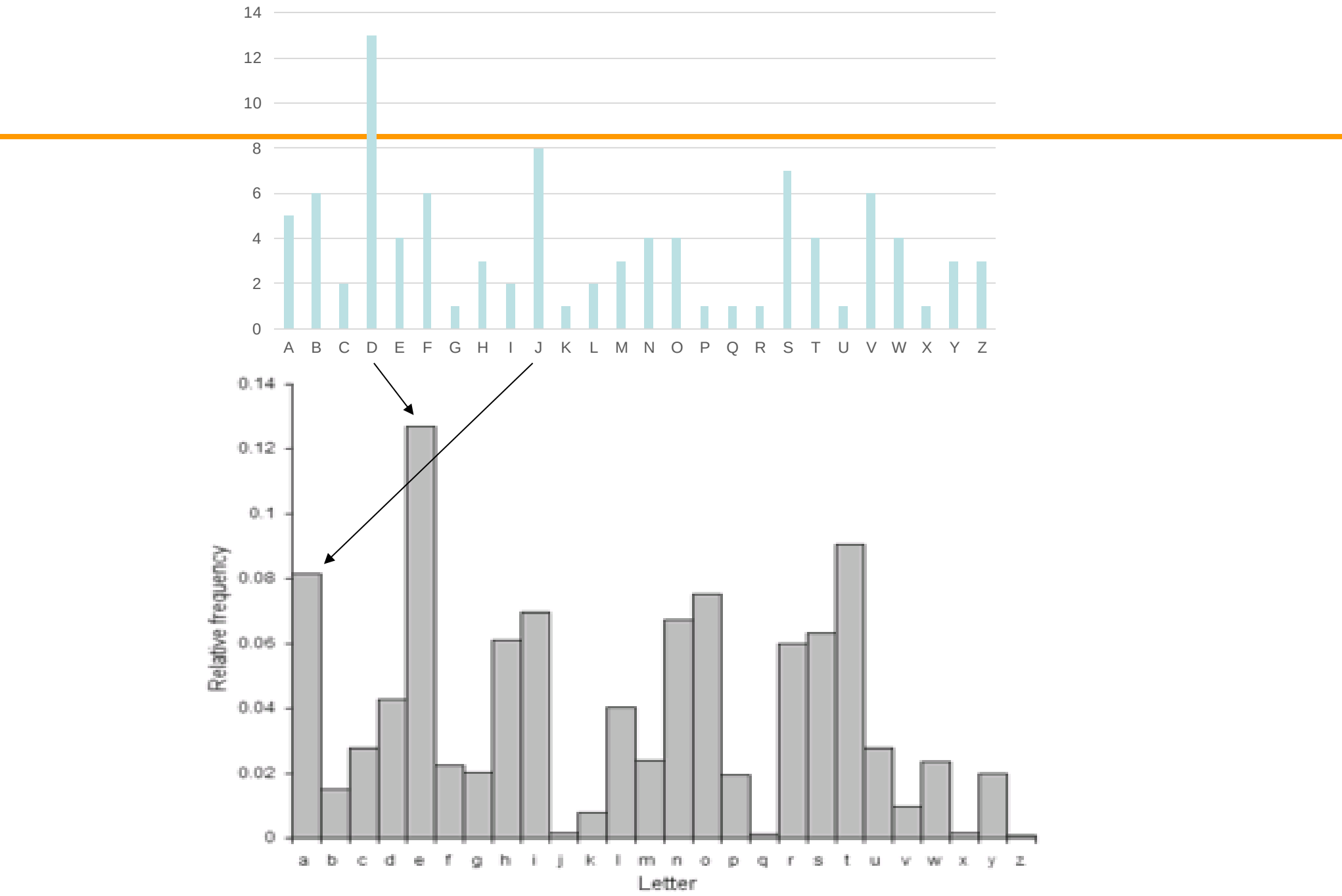
No exact letter frequency distribution underlies a given language, since all writers write slightly differently. However, most languages have a characteristic distribution which is strongly apparent in longer texts.

Encryption and decryption example 1

- Question
  - ASI JL DUJZTED SA J EJZD JVV NBTODI, VDD FOD AHB VBFD: OFFT://YYY.AHB.MSK/TJMD2/CDN08/NSCD_122908.OFZE, PSQD GSW MWDVV? LS, JNFWJE AHB RWBX.

Count Letter Frequency : https://www.browserling.com/tools/letter-frequency




14
12
10
8
6
4
2
0
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
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Relative frequency
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
Letter

Swapping Cipher Alphabets

Sometime in the 1460s, Alberti was wandering through the gardens of the Vatican, when he bumped into his friend Leonardo Dato, the pontifical secretary, who began chatting to him about some of the finer points of cryptography. This casual conversation prompted Alberti to write an essay on the subject, outlining what he believed to be a new form of cipher. Up until this point, a substitution cipher involved establishing a single cipher alphabet for encrypting each message. However, Alberti proposed using two or more cipher alphabets, switching between them during encipherment, thereby confusing potential cryptanalysts.

Here we have two possible cipher alphabets, and we can encrypt a message by alternating between them. The first letter of the plaintext message is encrypted using Ciphertext Alphabet 1, the second letter of the message is encrypted using Ciphertext Alphabet 2. We encrypt the third letter of the message by returning to Ciphertext Alphabet 1, the fourth letter is encrypted using Ciphertext Alphabet 2, and so on. To try out this cipher, type your message into the box labelled Plaintext, then click the 'Encipher Plaintext' button.

Plaintext Alphabet	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
Ciphertext Alphabet 1	S	T	G	F	Z	V	M	D	P	I	K	Y	H	B	C	J	W	X	E	O	U	R	A	L	N	Q
Ciphertext Alphabet 2	H	B	C	L	Z	W	X	N	M	K	U	S	T	A	O	P	G	Y	E	V	R	Q	I	F	D	J

Randomize Cipher Alphabets

Clear Cipher Alphabet

Letter Encrypt

Plaintext: abcabcxyzxyz

Ciphertext: SBGHTCLDQFNJ

莱昂·巴蒂斯塔·阿尔伯蒂（**Leon Battista Alberti**，1404－1472）是文艺复兴时期在意大利的建筑师、建筑理论家、作家、诗人、哲学家、密码学家。

Polyalphabetic substitution: Vigenère cipher

被独立发明了三次1467,1508,1553

Plaintext: **ATTACKATDAWN**

Key: **LEMONLEMONLE**

Ciphertext: **LXFOPVEFRNHR**

$$C_i \equiv P_i + K_i \pmod{26}$$

$$P_i \equiv C_i - K_i \pmod{26}$$

Ki=11,4,12,14,13,11,4,12,14,13…

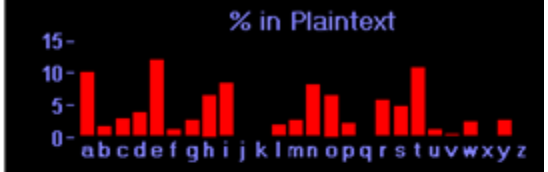
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T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

frequency analysis of Vigenère cipher

If we encrypt the same piece of text using the monoalphabetic substitution cipher and the Vigenère cipher, we can see why the latter cipher is so much stronger than the former. Let us use a short text about Vigenère to see the difference. Start by encrypting it with the monoalphabetic cipher.

Plaintext

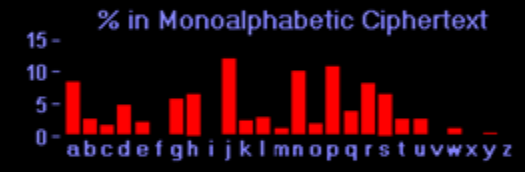
Aged twenty six, Vigenere was sent to Rome on a diplomatic mission. It was here that he became acquainted with the writings of Alberti, Trithemius and Porta, and his interest in cryptography was ignited. For many years, cryptography was nothing more than a tool that helped him in his diplomatic work, but, at the age of thirty nine, Vigenere decided that he had amassed enough money to be able to abandon his career and concentrate on a life of study. It was only then that he began research into a new cipher.



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

Monoalphabetic Ciphertext

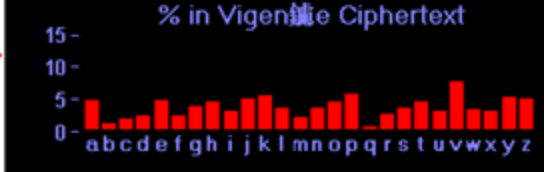
NUJQPKJRPTDAXYAUJRJGJKNDDJRPPHGHBJHRNQAEOHBNPALBADDAHRAPKNDSJG
JPSNPSJCJLNBJNLFMNARPJQKAPSPSJKGAPARUDHWNOCJGPAPGAPSJBAMDNRQEH
GPNNRQSADARPJGJDPARLGTEPHUGNESTKNDAURAPJQWHGBNRTTJNGDLGTEPHU
GNESTKNDRHPSARUBHGJPSNRNPHHOPSNPSJOEJQSABARSADQAEOHBNPALKHGZC
MPNPPSJNUJHWPSAGPTRARJYAUJRJGJQJLAQJQPSNPSJSNQNBNDDJQJRHMUSBHR
JTPHCJNCOJPHNCNRQHRSADLNGJJGNRQLHRLJRPGNPJHRNOAWJHWDPMQTAPKND
HROTPSJRPSNPSJCJUNRGJDJNGLSARPHNRJKLAESJG



As you can see, the frequency distribution is now much flatter. The peaks are less obvious, because each letter has been encrypted in 8 different ways, because the keyword is 8 letters long. The peak that was at E has been shared among 8 other letters. A flatter frequency distribution means a much stronger cipher.

Vigenère Ciphertext

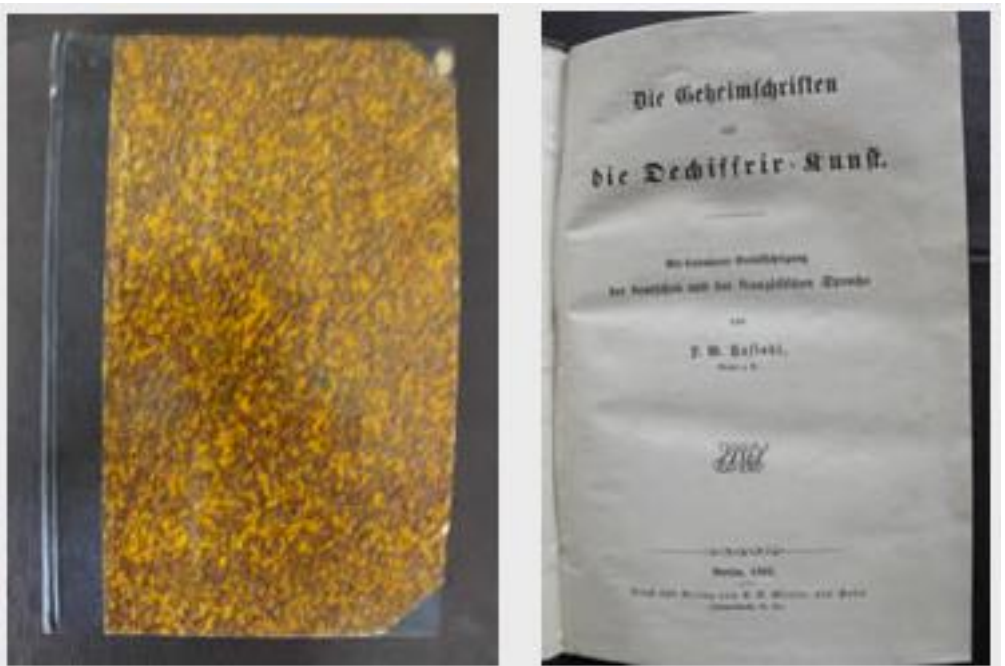
CNEUEAWIVFSZIZABGUEIPASNULNKESJJOLOELHAKNVMREMUHKZSZZRAOYHSYPVW
OJHTYPFWXCTERNUMVKUTVOAAOJAHVHVAOKUGJZJSGDLRKTXJDVOEDTYKVPKPFC
XSVPKHZDMFOGYEJEMFXTFPKZKJVROYNLWABPPTVOJGMOHNPJISMUJRPAXGBTHPY
JASNPVTYTRYHQYEKSEFVVVOCELSOJLLGPHZDOPNYTWVDRSODLXAXYVRBMYLVVA
HVLKWJHAHZCXQIKUEMTKWIGYEUPGAYGKTYLXZZJHDRXEKNGKEEZYYCOVNVJXGWG
HBCPXGVDHNUZRZDUJAIPIJVPKCFYGWIVYAKPSFVNPFVZJKOWKYZEASNQULPELWIV
OAKSITZIHNIPWWVTJHZYXGVPLWTTTZZT



Vigenere 算法曾里坚不可摧，持续了300+年

Breaking **Vigenère cipher**

- 弗里德里希–威廉–卡西斯基（Friedrich Wilhelm Kasiski）, 1805–1881, 德国步兵军官、密码学家和考古学家。
- 1863 年，出版“Secret writing and the Art of Deciphering”（Die Geheimschriften und die Dechiffrir-Kunst）



Breaking **Vigenère cipher**

- **Charles Babbage**,1791-1871 (查理斯.巴贝奇)
- **英国数学家、哲学家、发明家、机械工程师（可编程计算机概念的**创始人）
- 1846 , Babbage **破解了 Vigenère cipher**
- 克里米亚战争期间，作为军事秘密，直到1985年才被承认.

Augusta Ada King, Countess of Lovelace (*née* **Byron**; 10 December 1815 – 27 November 1852)

http://en.wikipedia.org/wiki/Charles_Babbage

如果你知道了密钥长度...

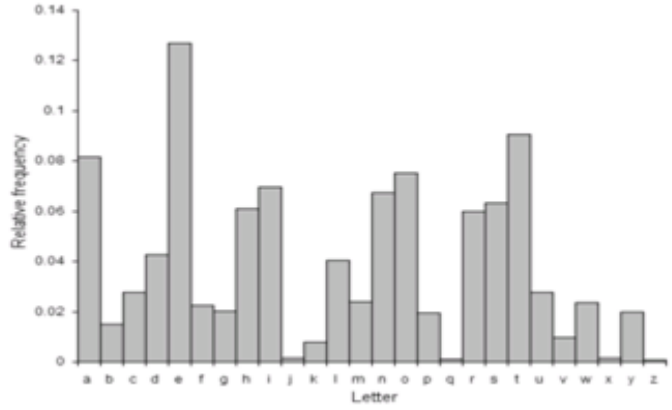
Plaintext: ATTAC KATDA WNONE ATTAC K……

Key: LEMON LEMON LEMON LEMON……

Ciphertext: LXFOP VEFRN HR . . . LXFOP . . . .

- 如果密钥长度为 5,我们可知 1st, 6th, 11th ,...位置上的字符是由维吉尼亚表中的同一行字符替换而成

- 这不还是一个单表替换吗？
- 凯撒密码？ 上频率分析？



怎么求密钥长度？

观察密文中重复的字符串

Key KINGKINGKINGKINGKING

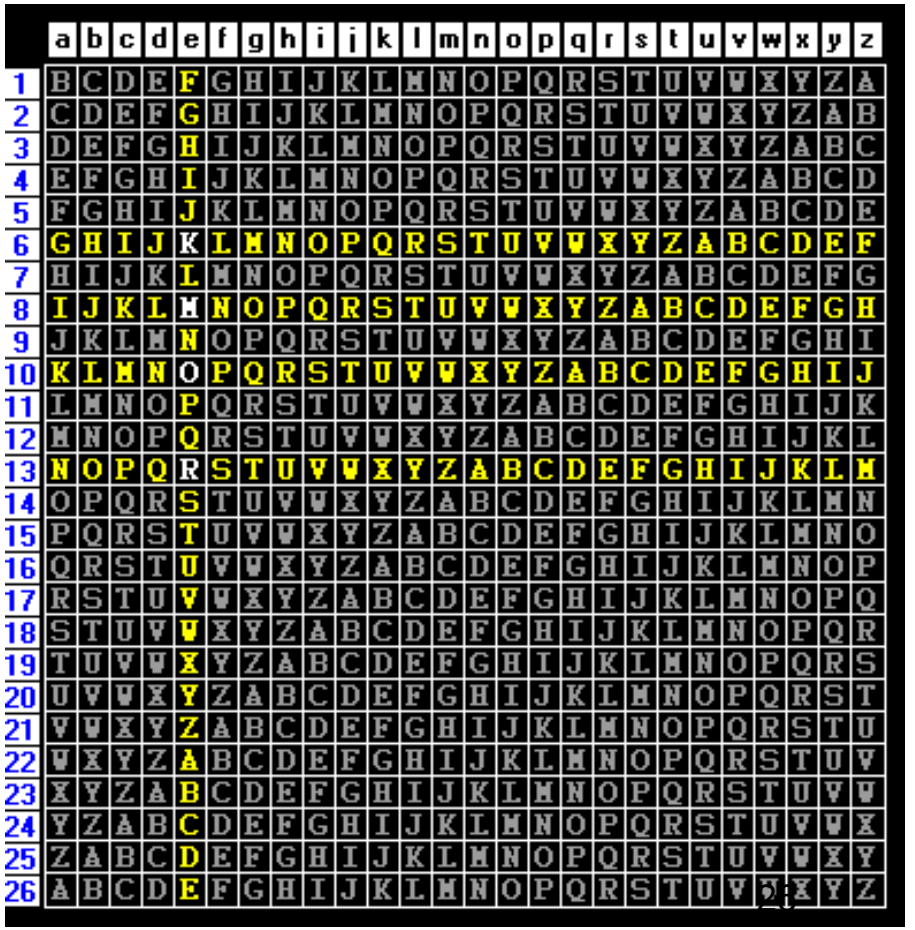
Plaintex thesunandthemaninthemoon

10 18

ciphertext dpryevntnbukwiaoxbukwwbt

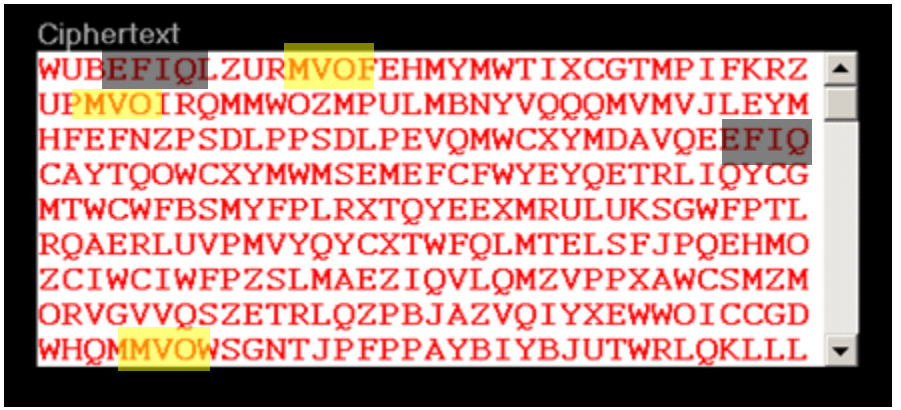
重复出现的字符串可能是由 **相同的明文**产生的

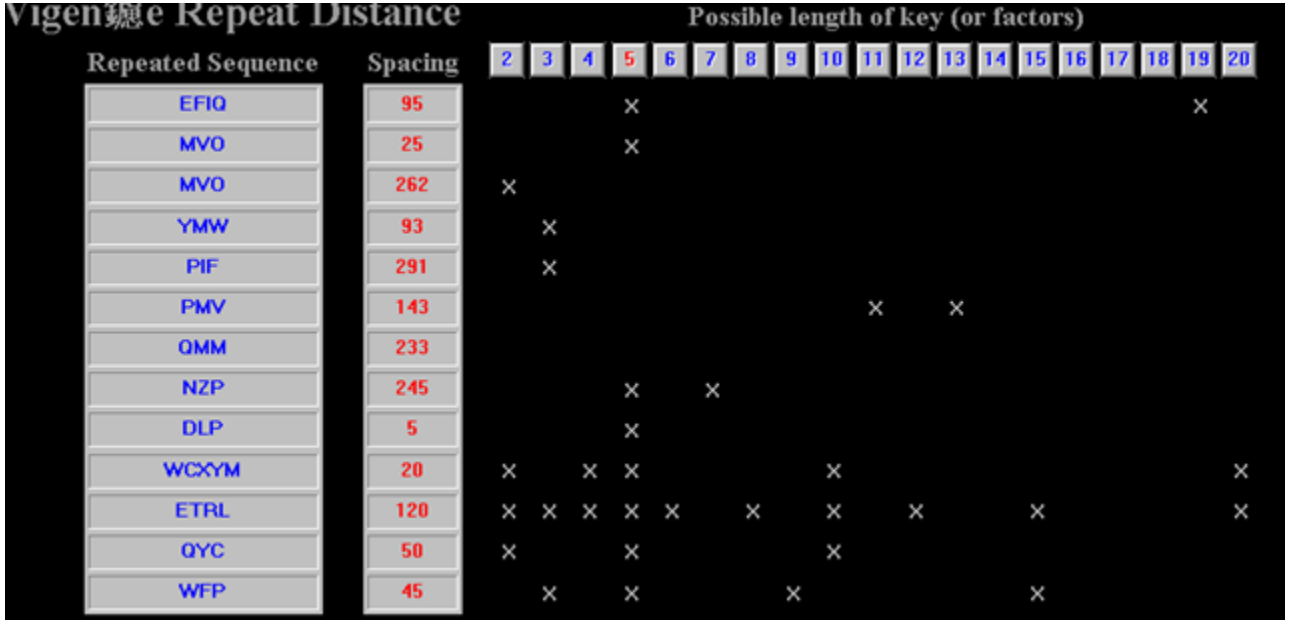
他们的**距离是密**钥长度的整数倍



寻找重复的字符串序列

- 寻找重复出现的字符串序列，计算他们之间的距离（间隔）
  - EFIQ： 95
  - MVO：25，262
- **距离或是密**钥长度的整数倍
- **把距离**进行因数分解
- 寻找公因数
- 出现最多的公因数**可能就是密**钥长度





古典密码之换位密（Transposition)

- Transposition(换位密码)

明文：Can you understand

输入方向 ⟶

C	A	N	Y
O	U	U	N
D	E	R	S
T	A	N	D

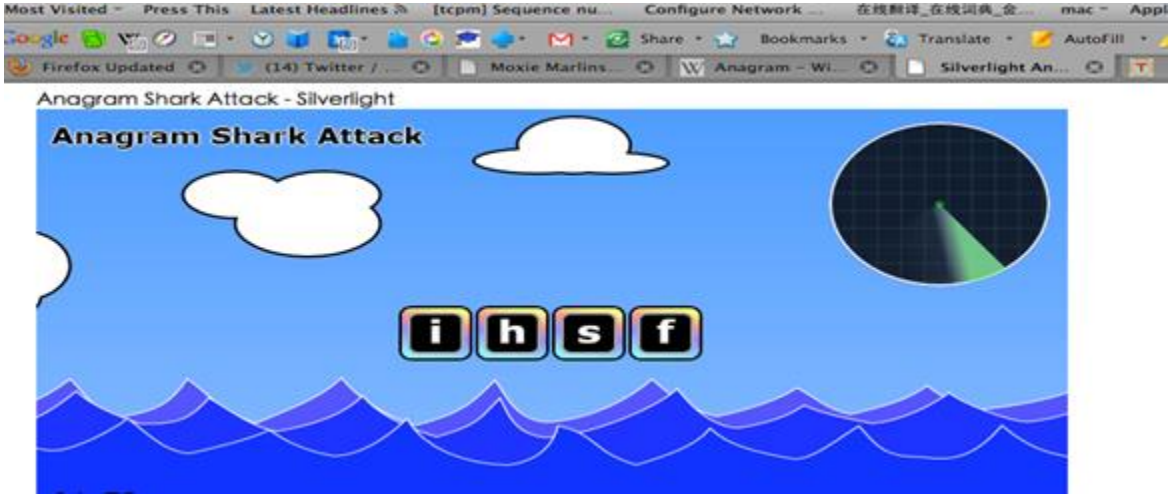
输出方向 ⟶

密文：codtaueanurnynsd

Anagram（变位词）

- An **anagram** is a type of word play, the result of rearranging the letters of a word or phrase to produce a new word or phrase, using all the original letters exactly once

游戏：http://www.pay4foss.org/jumpstation/AnagramSharkAttack/

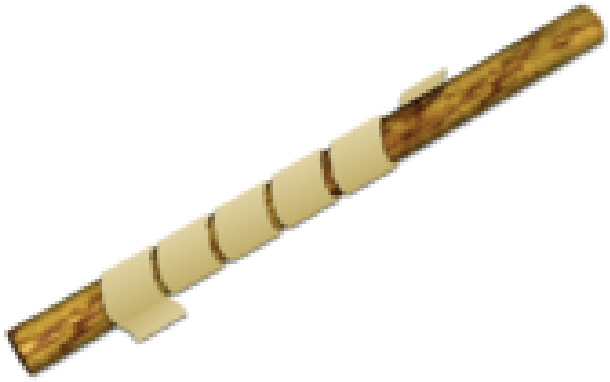


相同字母异序词

Anagram Examples

- *Eleven plus two = Twelve plus one*,
- *A decimal point = I'm a dot in place*,
- *Astronomers = Moon starers.*

More Transpositions, again…



6	3	2	4	1	5
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	F	L	E
E	A	T	O	N	C
E	Q	K	J	E	U

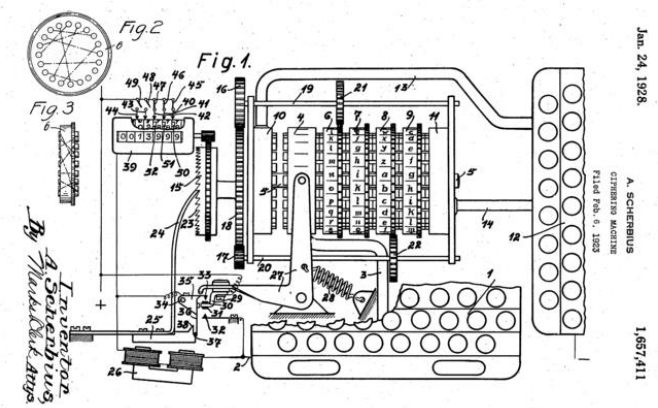
WE ARE DISCOVERED. FLEE AT ONCE.

EVLNE ACDTK ESEAQ ROFOJ DEECU WIREE

World War II: ENIGMA

1919 年发明
ENIGMA加密机

谢尔比乌斯**Arthur Scherbius** (20 October 1878–13 May 1929) , German electrical engineer who patented an invention for a mechanical cipher machine, later sold as the Enigma machine.



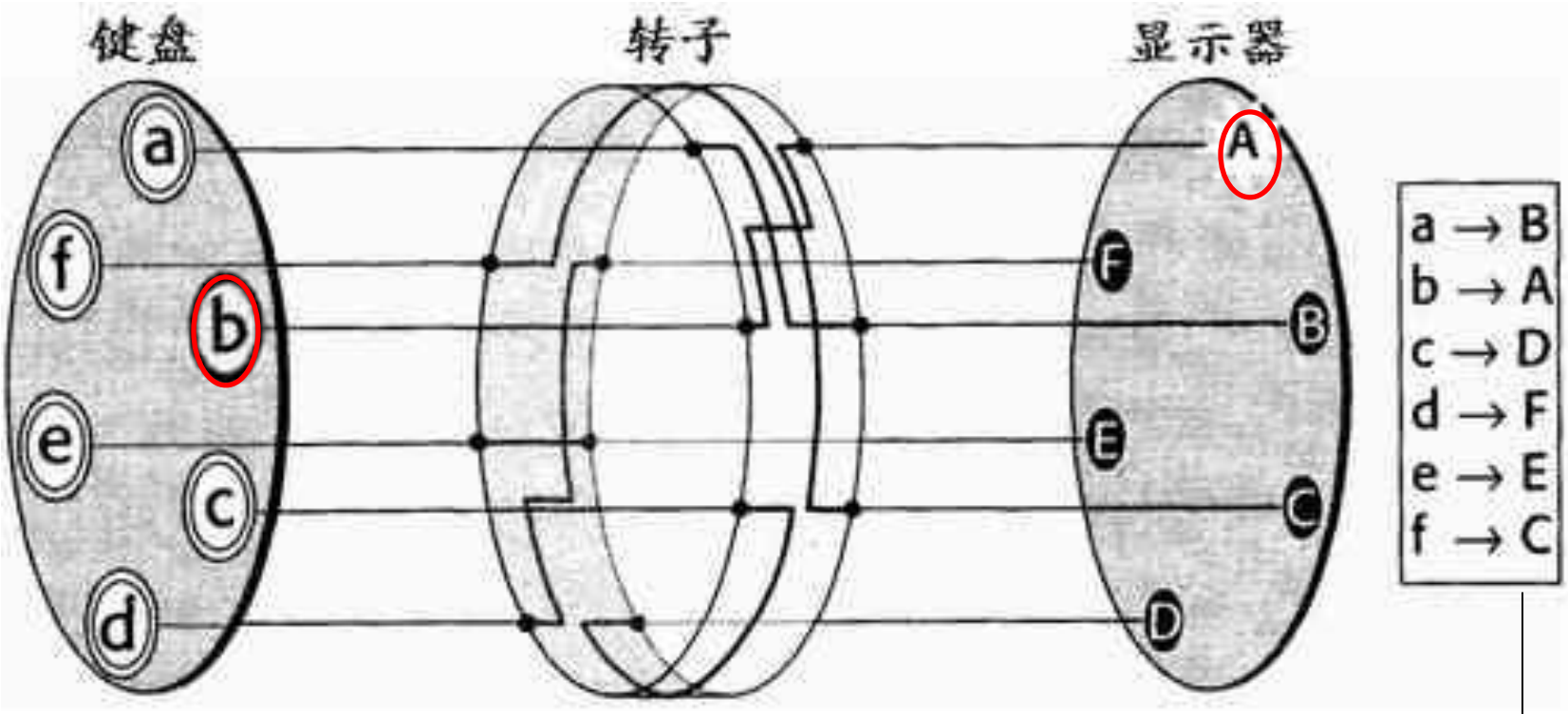
U.S. Paten0, 1928

Enigma in the WW II

- “闪电战”的提出者，德国装甲部队之父，纳粹德国的海因茨·古德里安 (Heinz Guderian)将军在指挥车上。

- 在照片的左下方我们可以看见一台ENIGMA。

How does Enigma work ?



- mono-alphabetic substitution cipher
- 如果Rotor不转，则只是一个单表替换

A-B-A

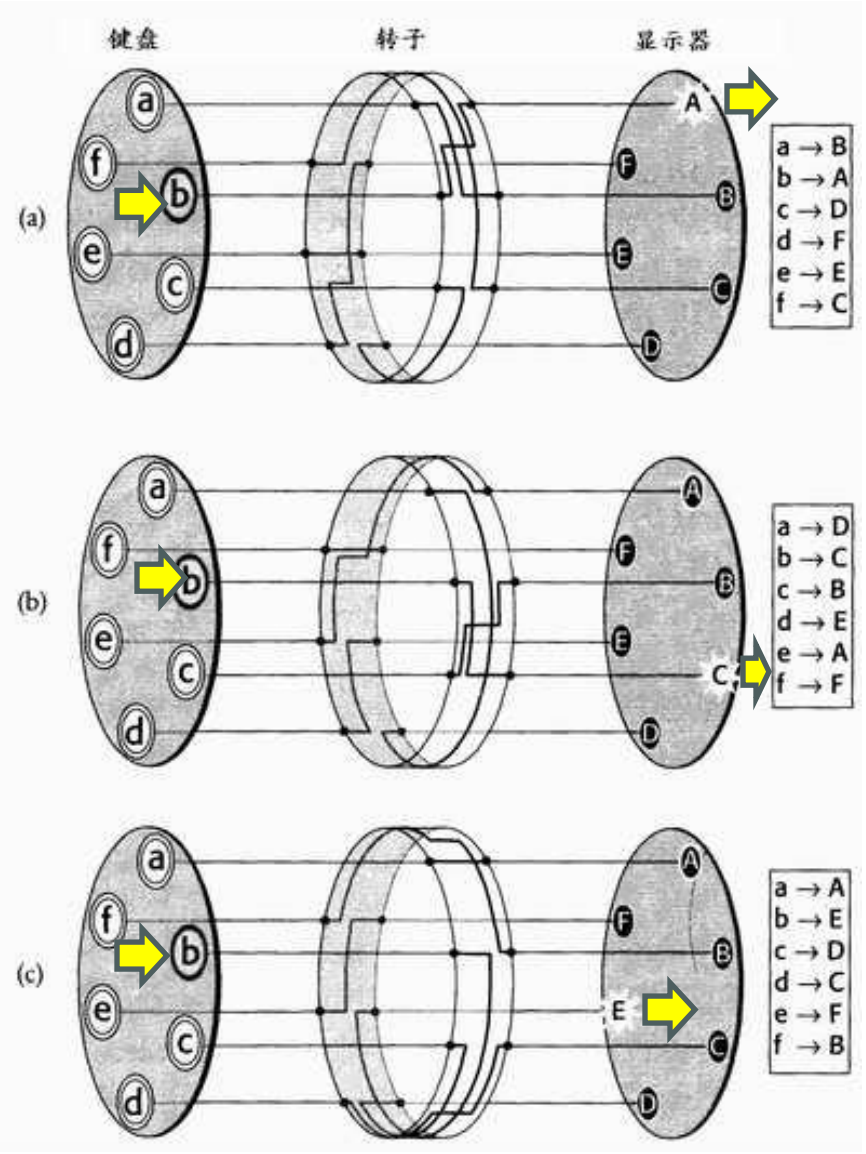
C-D-F-C

E-E

How does Enigma work ?

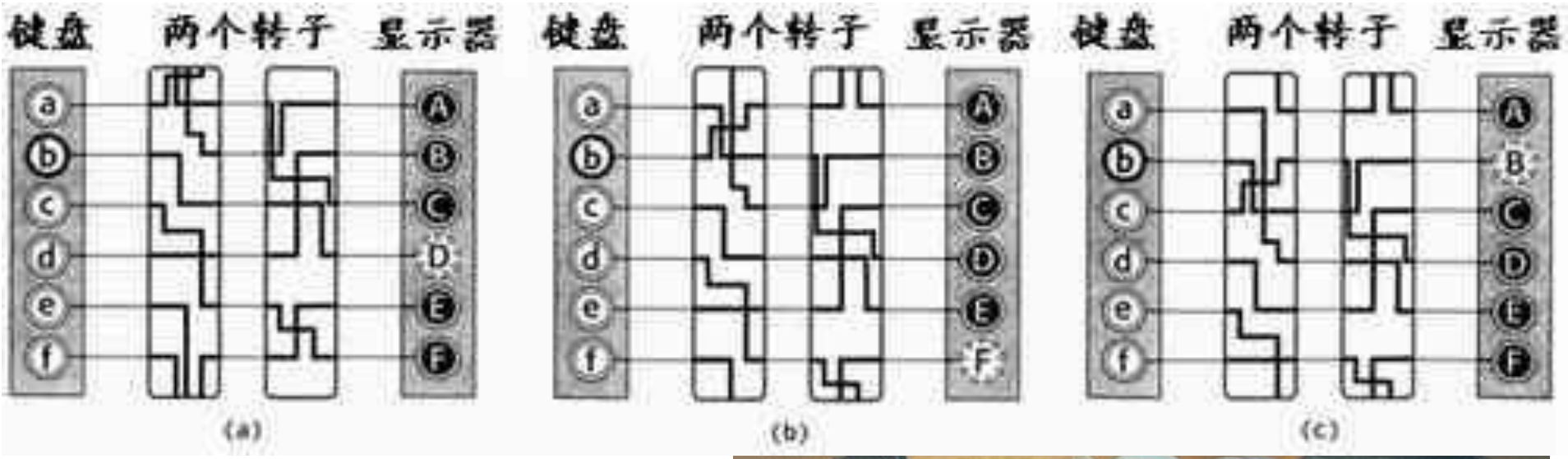
- **Polyalphabetic substitution** ：
  - 当键盘 上一个键被按下时，相应的密文在显示器上显示，然后转子的方向就 自动地转动一个字母的位置（在示意图中就是转动1/6圈，而在实际中 转动1/26圈）。
  - 有26 个替换表的Vigenere加密

Brute force attack: 26

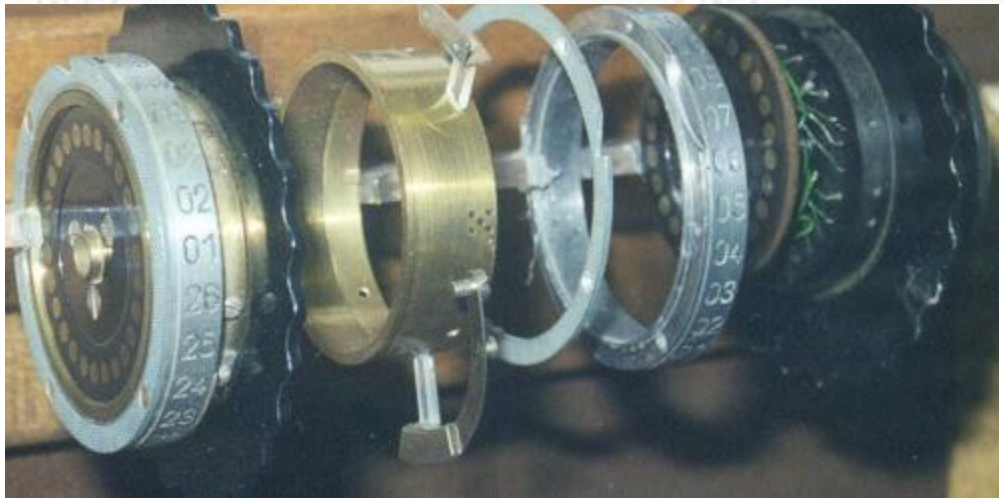


Enigma工作原理

- 转子转动一圈重复的问题：增加多个转子

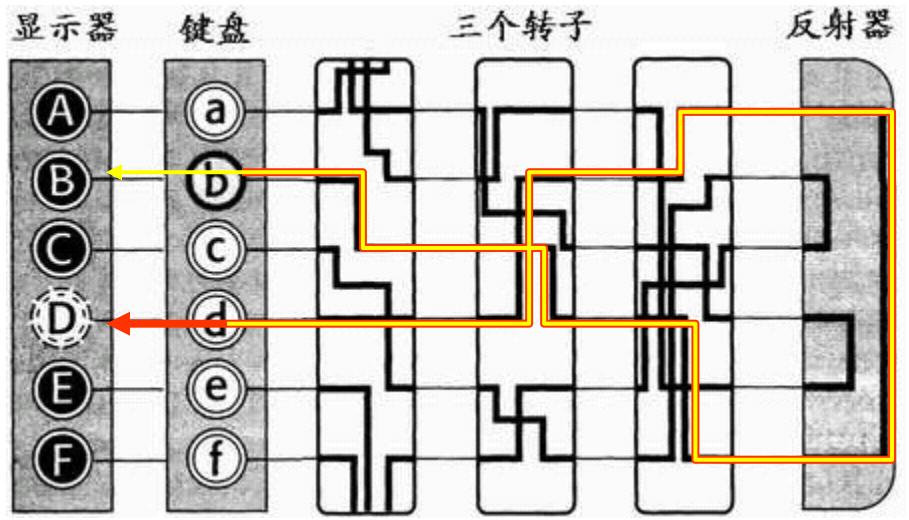


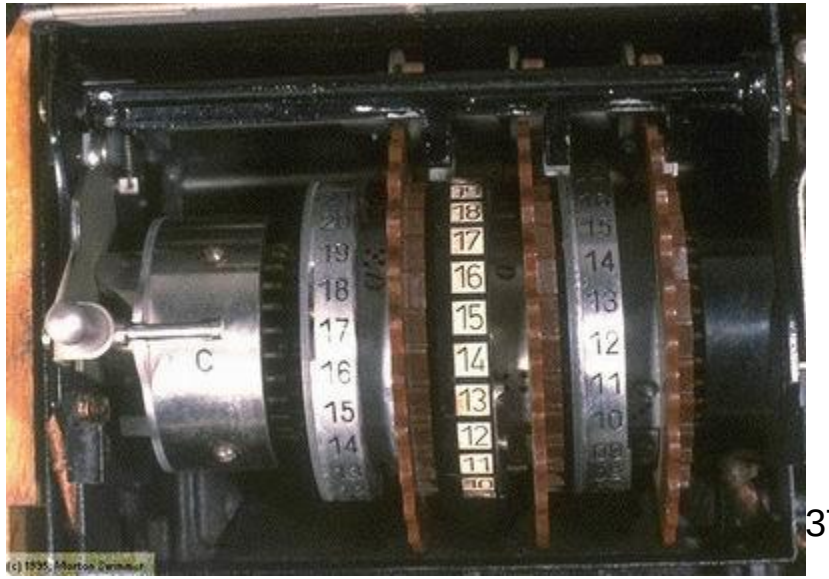
- ENIGMA里有三个转子不重复的方向个数达到26*26*26 =17576个。
- 密钥长度17576的多表替换



Reflector：加密与解密过程相同

- 反射器和转子一样，把某一个字母连在另一个字母上，但是它并不转动。
- 反射器可以使译码的过程和编码的过程完全一样。
- 加密机和解密机相同，相同的密钥
- 发射器从不把输入映射成自身





Plugboard（连接板）

- 从26个字母中取出6对，交换，比如
  A/J, S/O, T/D, B/W，K/F, U/Y

Select 12 letters from 26:

C(26,12) = 26!/ 14!* 12!

Swap 12 letters:

(12!/(6! *6!)) * 6! / 2

$$\frac{26!}{14! * 12!} * \frac{12!}{6! * 2} = 26! / (14! * 6! * 2)$$

= 150,738,274,937,250 (1.5E14)

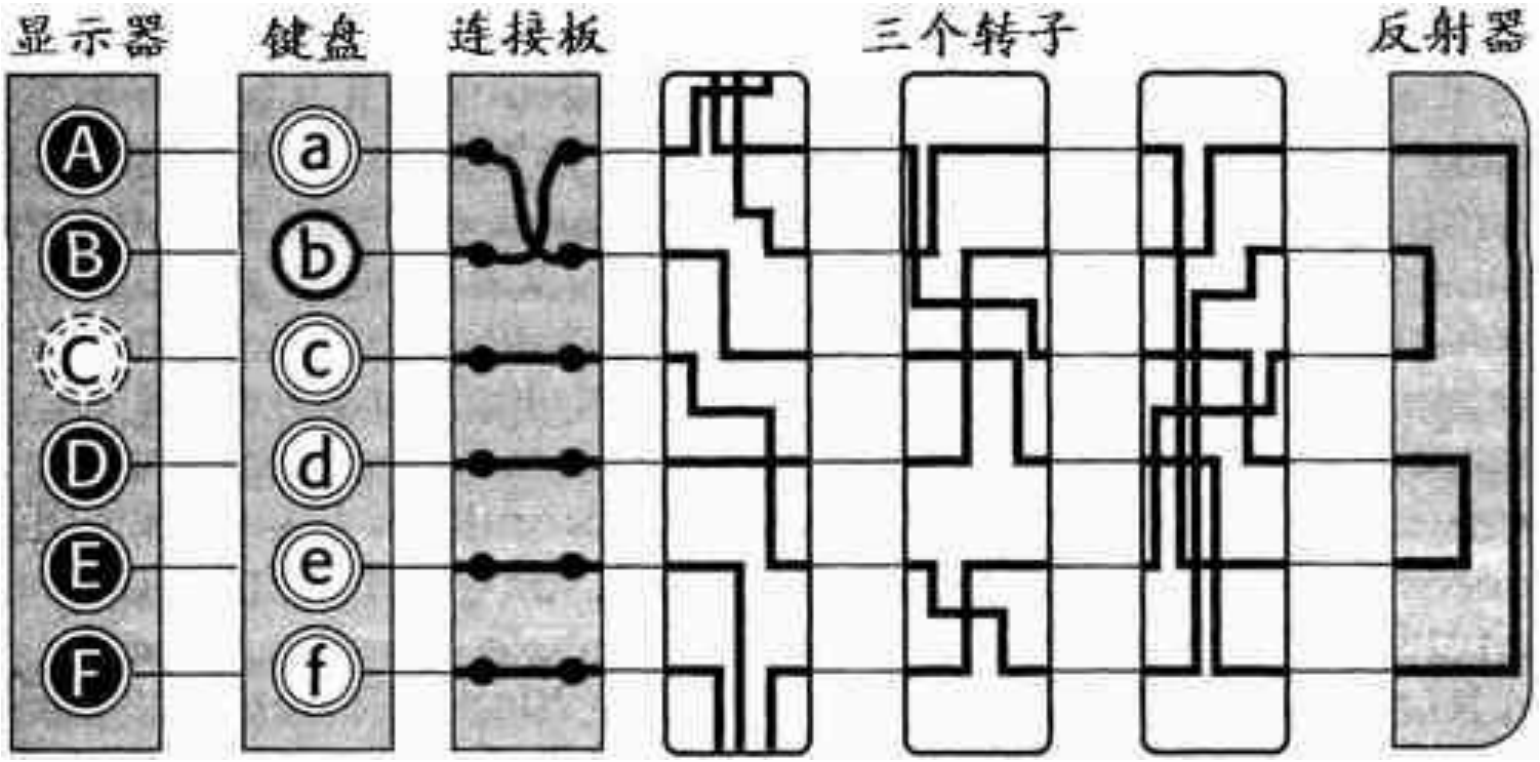
5. For the plugboards, we have a similar situation as with the reflector plate. There are 26 holes in the board, each hole representing a letter in the alphabet. Then, we can have up to 13 wires, with one end of each wire plugging into a hole and with the other end of each wire plugging into another hole. Depending on how many wires we use, call this number p, we would have different amounts of possible combinations. First, we have $\binom{26}{2p}$ letters that we need to choose to be plugging into. Then, for the first cable, we plug one end into a random hole, and we're left with $(2p-1)$ holes to plug the other end into. For the second cable, we plug one end into a hole, and we're left with $(2p-3)$ holes to plug the other end into. So on with all the other cables and this gives us $\binom{26}{2p} \cdot (2p-1)!!$ possible combinations for any given p from 0 to 13. The total number of possible combinations for each p is shown below in Figure 3. If p were unknown, the total possible combinations would be the sum of the possible combinations for each p.

p	combinations	p	combinations
0	1	7	1,305,093,289,500
1	325	8	10,767,019,638,375
2	44,850	9	53,835,098,191,875
3	3,453,450	10	150,738,274,937,250
4	164,038,875	11	205,552,193,096,250
5	5,019,589,575	12	102,776,096,548,125
6	100,391,791,500	13	7,905,853,580,625

Figure 3: The number of combinations given *p* plugboard settings[2].

https://courses.csail.mit.edu/6.857/2018/project/lyndat-nayoung-ssrusso-Enigma.pdf

Plugboard（连接板）



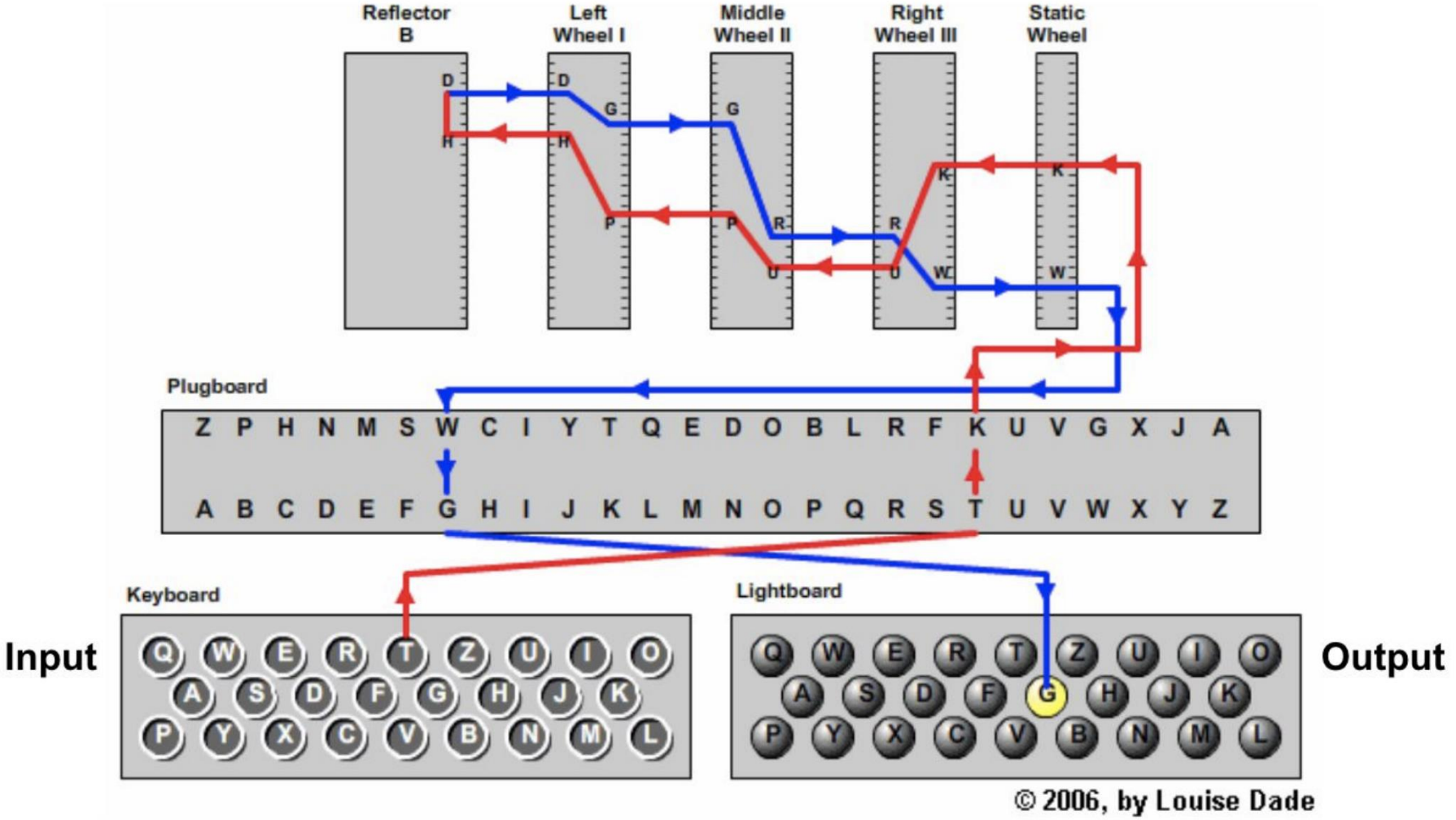
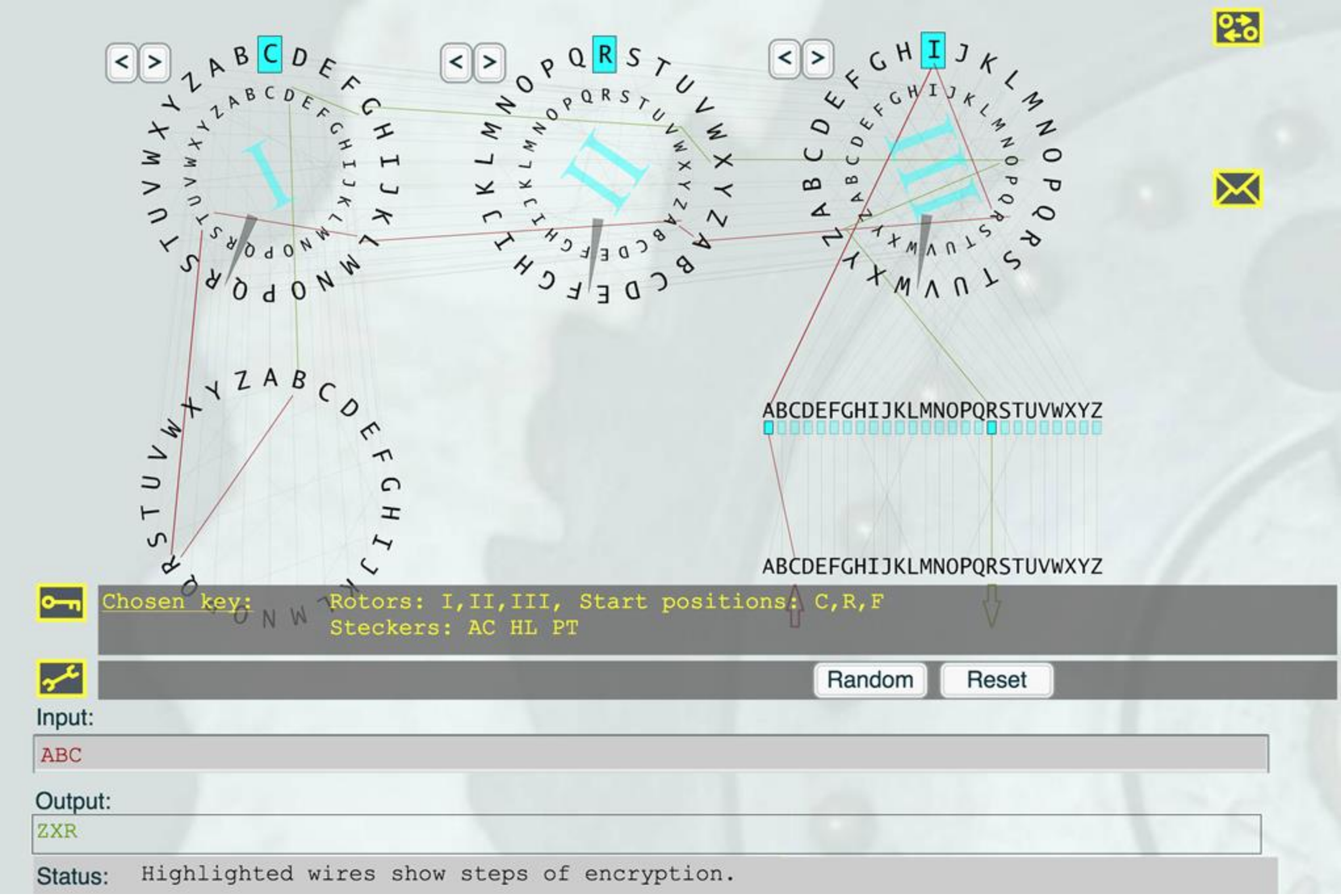


Figure 2: The internal wirings of the Enigma[2].

https://courses.csail.mit.edu/6.857/2018/project/lyndat-nayoung-ssrusso-Enigma.pdf



http://enigmaco.de/enigma/enigma.html

Mathematical description

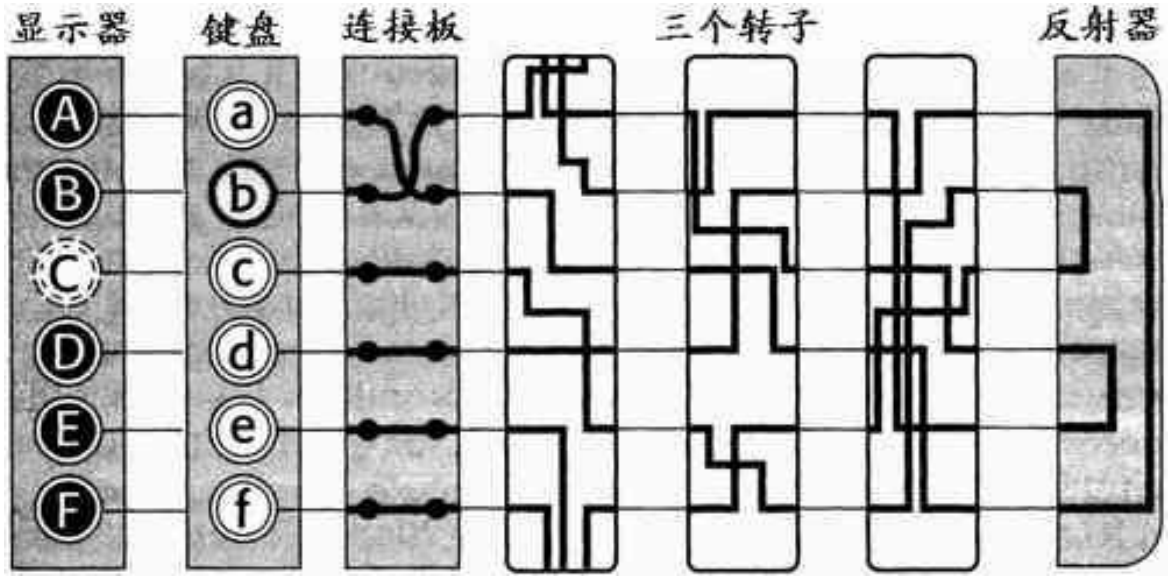
- Permutations
  - P : plugboard transformation,
  - U : reflector,
  - L,M,R : the left, middle and right rotors,
- Encryption E :

$$E = P\,R\,M\,L\,U\,L^{-1}\,M^{-1}\,R^{-1}\,P^{-1}$$

https://www.codesandciphers.org.uk/enigma/

安全性：多少个密钥？

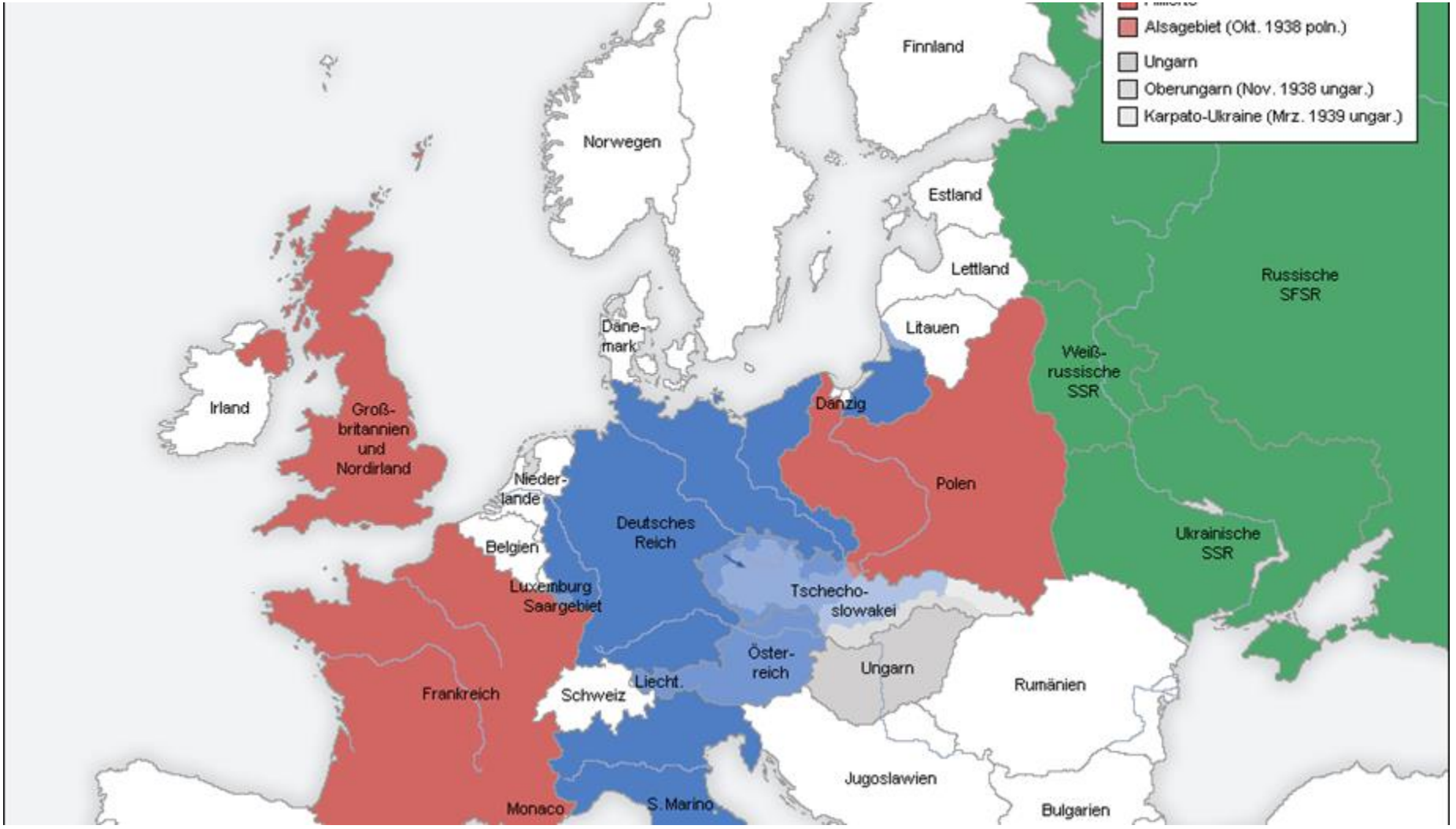
- 三个转轮顺序: 3!
- 三个转轮的初始位置：26*26*26=17576



- 连接板上两两交换6对字母的可能性数:1.5E14
- Total： 17576 * 6 * 1.5E14 ~ 1.58 E 19

- Enigma仍然可看作是一个多表替换，但多少字符之后可能出现重复？
- Enigma曾在13年时间里（1926-1939）被认为坚不可摧

二战前，恐怖笼罩之下的波兰



破解 Enigma 的第一功臣：德国间谍

Hans-Thilo Schmidt

9 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

Hans–Thilo Schmidt (13 May 1888 – 19 September 1943) codenamed **Asché** or **Source D**, was a German spy who sold secrets about the Enigma machine to the French during World War II. The materials he provided facilitated Polish mathematician Marian Rejewski's reconstruction of the wiring in the Enigma's rotors and reflector; thereafter the Poles were able to read a large proportion of Enigma–enciphered traffic. He was the younger brother of Wehrmacht general Rudolf Schmidt.

Hans–Thilo Schmidt

Born	13 May 1888
Died	19 September 1943 (aged 55) Berlin, Germany
Cause of death	Execution or Suicide
Nationality	German
Occupation	Civil servant
Espionage activity	
Allegiance	France
Codename	Asché
Codename	Source D

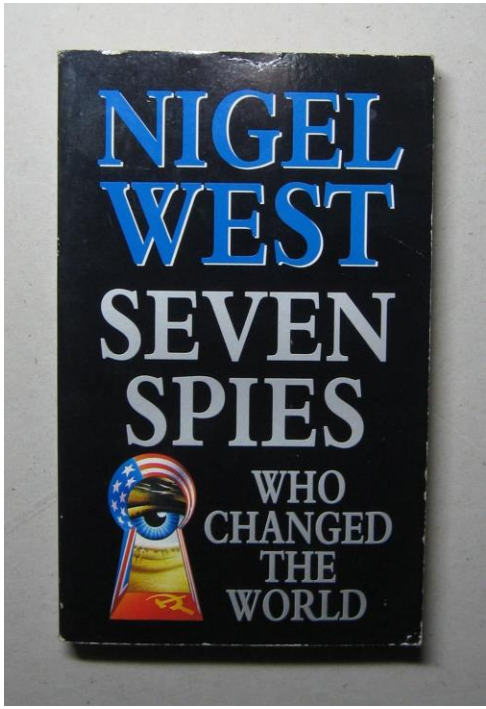
Selling Enigma secrets [edit]

Rodolphe Lemoine [fr], born Rudolf Stallmann (1871–1946), served as his French point of contact as "Rex".

A former officer, Schmidt had been forced to leave the army having suffered from gas during the First World War.[1] His brother, Rudolf Schmidt, secured him a civilian post at the German Armed Forces' cryptographic headquarters, the Cipher Office.[1] Shortly after the military version of the Enigma machine was introduced, he contacted French intelligence and offered to supply information about the new machine. His offer was accepted by Captain Gustave Bertrand of French Intelligence, and he received from the French the codename *Asché*, and was assigned a contact, the French agent codenamed *Rex*.

For the next several years, until he left his position in Germany, he met with French agents at various European cities and supplied them copies of the Enigma machine's instruction

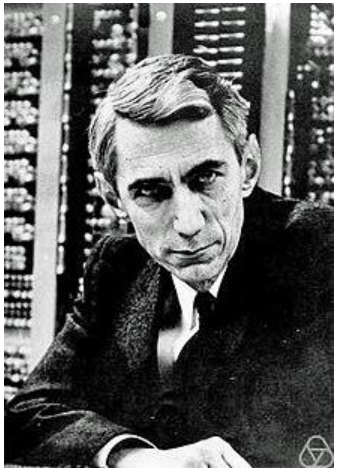
汉斯-提罗·施密特，
世界七大间谍之一



Kerckhoff's principle,柯克霍夫原则

- A cryptosystem should be secure even if **everything** about the system, except the secret key, **is public knowledge,** 1883
- **Shannon's maxim:** The enemy knows the system: one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them 1949

Auguste Kerckhoffs (1835 –1903)
Dutch linguist and cryptographer



Claude Elwood Shannon (1916 –2001)

假设你的敌人知道所有的细节

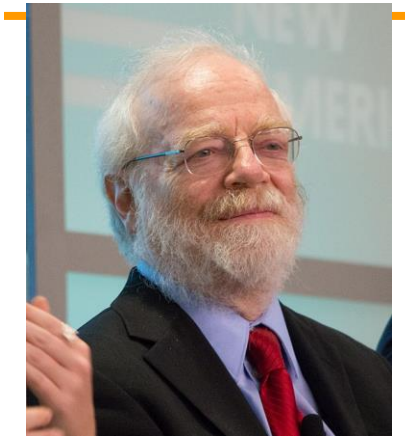
"Steven M. Bellovin" <smb@cs.columbia.edu>
Sat, 6 Jun 2009 22:21:01 -0400

The subject of security through obscurity comes up frequently. I think a lot of the debate happens because people misunderstand the issue.

It helps, I think, to go back to Kerckhoffs' second principle, translated as "The system must not require secrecy and can be stolen by the enemy without causing trouble", per http://petitcolas.net/fabien/kerckhoffs/). Kerckhoffs said neither "publish everything" nor "keep everything secret"; rather, he said that the system should still be secure *even if the enemy has a copy*.

In other words — design your system assuming that your opponents know it in detail. (A former official at NSA's National Computer Security Center told me that the standard assumption there was that serial number 1 of any new device was delivered to the Kremlin.) After that, though, there's nothing wrong with trying to keep it secret — it's another hurdle factor the enemy has to overcome. (One obstacle the British ran into when attacking the German Engima system was simple: they didn't know the unkeyed mapping between keyboard keys and the input to the rotor array.) But — *don't rely on secrecy*.

http://catless.ncl.ac.uk/Risks/25.71.html#subj19



Steven M. Bellovin

professor in Columbia University

Fellow at AT&T Labs Research

Cracking the Enigma

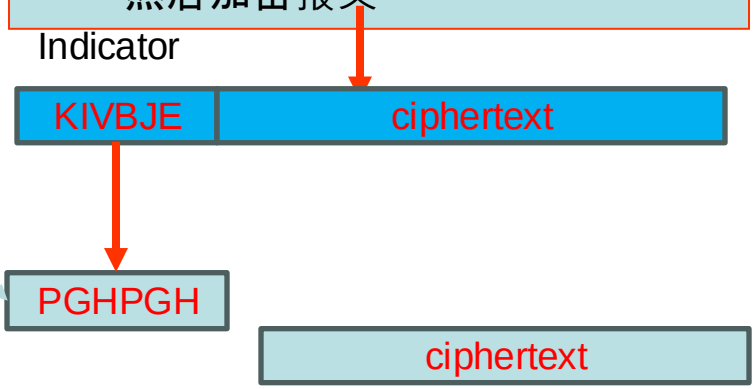
- 最大的弱点不在于Enigma本身，而在于如何使用
- 密钥的管理、分发、管理是密码算法的致命弱点
  - 每条报文开头重复的密钥：PGHPGH □ KIVBJE

Enigma的密钥管理

- 密钥
  - 连接板交换6对字符（后来10对），如：A/L ，P/R, T/D, B/W, K/F, O/Y
  - 转子的顺序，比如： 2,3,1
  - 转子的初始位置，如：Q-C-W

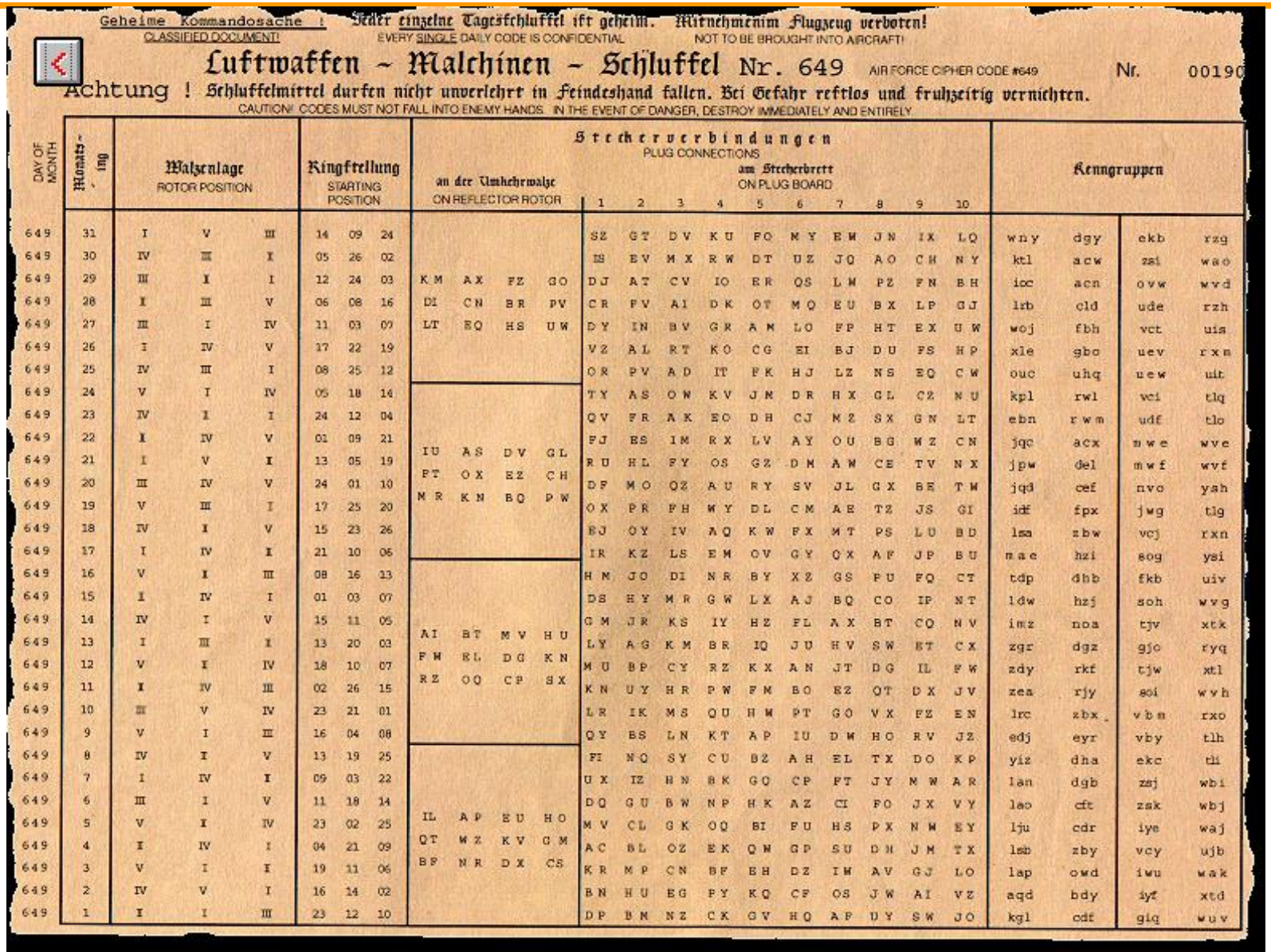
- 每月发送当月每天的密钥 Kd
- 每条密文使用不同的密钥 Ks
- 每条电文的开头是加密的临时密钥，由三个字母重复两次加密而成，称为indicator

- Sender：使用当天的密钥(QCW)加密一个Indicator，如输入PGHPGH，假设输出KIVBJE ；
- 把KIVBJE写在报文的开头，然后调整转子的初始方向位PGH，然后加密报文



- Receiver: 用当天的密钥(QCW)解密报文开头的6字符，得到PGHPGH ；
- 开头，然后调整转子的初始方向位PGH，然后解密报文

Code book



Geheime Kommandosache ! Jeder einzelne Tagesschlüssel ist geheim. Mitnehmen im Flugzeug verboten!
CLASSIFIED DOCUMENT! EVERY SINGLE DAILY CODE IS CONFIDENTIAL NOT TO BE BROUGHT INTO AIRCRAFT!

Luftwaffen – Maschinen – Schlüssel Nr. 649 AIR FORCE CIPHER CODE #649 Nr. 00190

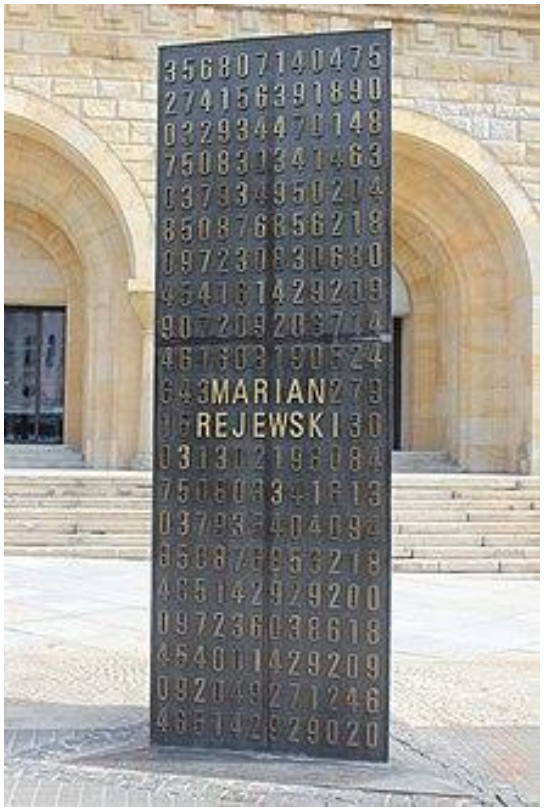
Achtung ! Schlüsselmittel dürfen nicht unverlehrt in Feindeshand fallen. Bei Gefahr restlos und fruhzeitig vernichten.
CAUTION! CODES MUST NOT FALL INTO ENEMY HANDS. IN THE EVENT OF DANGER, DESTROY IMMEDIATELY AND ENTIRELY.

DAY OF MONTH	Monats-tag	Walzenlage ROTOR POSITION			Ringstellung STARTING POSITION			Steckerverbindungen PLUG CONNECTIONS an der Umkehrwalze ON REFLECTOR ROTOR				am Steckerbrett ON PLUG BOARD 1	2	3	4	5	6	7	8	9	10	Kenngruppen			
649	31	I	V	III	14	09	24						SZ	GT	DV	KU	FO	NY	EW	JN	IX	LQ	wny	dgy	ekb	rzg
649	30	IV	III	I	05	26	02						BE	EV	MX	RW	DT	UZ	JQ	AO	CH	NY	ktl	acw	zsi	wao
649	29	III	I	I	12	24	03	KM	AX	FZ	GO		DJ	AT	CV	IQ	ER	QS	LW	PZ	FN	BH	icc	acn	ovw	wvd
649	28	I	III	V	06	08	16	DI	CN	BR	PV		CR	FV	AI	DK	OT	MQ	EU	BX	LP	GJ	lrb	cld	ude	rzh
649	27	III	I	IV	11	03	07	LT	EQ	HS	UW		DY	IN	BV	GR	AM	LO	FP	HT	EX	UW	woj	fbh	vct	uis
649	26	I	IV	V	17	22	19						VZ	AL	RT	KO	CG	EI	BJ	DU	FS	HP	xle	gbo	uev	rxn
649	25	IV	III	I	08	25	12						OR	PV	AD	IT	FK	HJ	LZ	NS	EQ	CW	ouc	uhq	uew	uit
649	24	V	I	IV	05	18	14						TY	AS	ON	KV	JM	DR	HX	GL	CZ	NU	kpl	rwl	vci	tlq
649	23	IV	I	I	24	12	04						QV	FR	AK	EO	DH	CJ	MZ	SX	GN	LT	ebn	rwm	udf	tlo
649	22	I	IV	V	01	09	21						FJ	BS	IM	RX	LV	AY	OU	BG	WZ	CN	jqc	acx	mwe	wve
649	21	I	V	I	13	05	19	IU	AS	DV	GL		RU	HL	FY	OS	GZ	DM	AW	CE	TV	NX	jpw	del	mwf	wvf
649	20	III	IV	V	24	01	10	PT	OX	EZ	CH		DF	MO	QZ	AU	RY	SV	JL	GX	BE	TW	jqd	cef	nvo	ysh
649	19	V	III	I	17	25	20	MR	KN	BQ	PW		OX	PR	FH	WY	DL	CM	AE	TZ	JS	GI	idf	fpx	jwg	tlg
649	18	IV	I	V	15	23	26						BJ	OY	IV	AQ	KW	FX	MT	PS	LU	BD	lsa	zbw	vcj	rxn
649	17	I	IV	I	21	10	06						IR	KZ	LS	EM	OV	GY	QX	AF	JP	BU	mae	hzi	sog	ysi
649	16	V	I	III	08	16	13						HM	JO	DI	NR	BY	XZ	GS	FU	FQ	CT	tdp	dhb	fkb	uiv
649	15	I	IV	I	01	03	07						DS	HY	MR	GW	LX	AJ	BQ	CO	IP	NT	ldw	hzj	soh	wvg
649	14	IV	I	V	15	11	05	AI	BT	MV	HU		GM	JR	KS	IY	HZ	FL	AX	BT	CQ	NV	imz	noa	tjv	xtk
649	13	I	III	I	13	20	03	FW	EL	DG	KN		LY	AG	KM	BR	IQ	JU	HV	SW	ET	CX	zgr	dgz	gjo	ryq
649	12	V	I	IV	18	10	07	RZ	OQ	CP	SX		MU	BP	CY	RZ	KX	AN	JT	DG	IL	FW	zdy	rkf	tjw	xtl
649	11	I	IV	III	02	26	15						KN	UY	HR	PW	FM	BO	EZ	QT	DX	JV	zea	rjy	soi	wvh
649	10	III	V	IV	23	21	01						LR	IK	MS	QU	HW	PT	GO	VX	FZ	EN	lrc	zbx	vbm	rxo
649	9	V	I	III	16	04	08						QY	BS	LN	KT	AP	IU	DW	HO	RV	JZ	edj	eyr	vby	tlh
649	8	IV	I	V	13	19	25						FI	NQ	SY	CU	BZ	AH	EL	TX	DO	KP	yiz	dha	ekc	thi
649	7	I	IV	I	09	03	22						UX	IZ	HN	BK	GQ	CP	FT	JY	MW	AR	lan	dgb	zsj	wbi
649	6	III	I	V	11	18	14						DQ	GU	BW	NP	HK	AZ	CI	FO	JX	VY	lao	cft	zsk	wbj
649	5	V	I	IV	23	02	25	IL	AP	EU	HO		MV	CL	GK	OQ	BI	FU	HS	PX	NW	EY	lju	cdr	iye	waj
649	4	I	IV	I	04	21	09	QT	WZ	KV	GM		AC	BL	OZ	EK	QN	GP	SU	DH	JM	TX	lsb	zby	vcy	ujb
649	3	V	I	I	19	11	06	BF	NR	DX	CS		KR	MP	CN	BF	EH	DZ	IW	AV	GJ	LO	lap	owd	iwu	wak
649	2	IV	V	I	16	14	02						BN	HU	EG	PY	KQ	CF	OS	JW	AI	VZ	aqd	bdy	iyf	xtd
649	1	I	I	III	23	12	10						DP	BM	NZ	CK	GV	HQ	AP	UY	SW	JO	kgl	cdf	giq	wuv

马里安·雷杰夫斯基

Marian Rejewski

波兰数学家（1905-1980）



Cracking Enigma：Jobs by Pole

	1	2	3	4	5	6
M1：	L	O	K	R	G	M
M2:	M	V	T	X	Z	E
M3:	J	K	T	M	P	E
M4:	D	V	Y	P	Z	X

同一天的密文前六个字符（indicator), 是三个字符重复
L-R, M-X, J-M, D-P 对应的明文相同

The first letter ：ABCDEFGHIJKLMNOPQRSTUVWXYZ
The fourth letter ：____P_____M_RX_____

The first letter ：ABCDEF GHIJKLMNOPQR STUVWXYZ
The fourth letter ：FQHPLWOGBMVRXUYCZI TNJEA SDK

The circles ：

A→F→W→A	3个字母的循环圈
B→Q→Z→K→V→E→L→R→I→B	9个字母的循环圈
C→H→G→O→Y→D→P→C	7个字母的循环圈
J→M→X→S→T→N→U→J	7个字母的循环圈

同样可以找到第二和第五个、第三和第六个字母的循环圈。

The Fingerprint of this circle : 4 { 3,7,7,9}

波兰人的破解工作

- **Marian Rejewski：**循环圈的数目和每个圈的长度与plugboard没有关系
- 只取决于转轮、顺序和初始位置

Fingerprint: 4 {3,7,7,9}

A→F→W→A	3
B→Q→Z→K→V→E→L→R→I→B	9
C→H→S→O→Y→D→P→C	7
J→M→X→G→T→N→U→J	7

1st letter：	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
4th letter：	F	Q	H	P	L	W	T	S	B	M	V	R	X	U	Y	C	Z	I	O	N	J	E	A	G	D	K

- Now, we switch S-G

Fingerprint: 4 {3,7,7,9}

A→F→W→A	3
B→Q→Z→K→V→E→L→R→I→B	9
C→H→G→O→Y→D→P→C	7
J→M→X→S→T→N→U→J	7

1st letter：	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
4th letter：	F	Q	H	P	L	W	O	G	B	M	V	R	X	U	Y	C	Z	I	T	N	J	E	A	S	D	K

Enigma的破译：波兰人的工作(Bomba)

- plugboard任何连线变化都不影响循环圈的个数和每个循环圈上字母的个数；
- 尝试所有3 个rotors的初始位置，可得所有密钥的“fingerprint”，即循环圈个数、每个循环圈长度:
  - 需要操作的次数：26 * 26 * 26 ＝17576
  - 考虑3个转子交换顺序3!，用6台Enigma协作运算
- 建立这样一个档案（book of fingerprints）用了一年时间
- 根据密文的循环圈查找指纹库，可得到部分秘钥
- Plugboard 的字母替换怎么解决？
- 回到单表替换

LIEHLETHIR Heil Hitler

From Poland to British

- In1938, Germans improved security
  - by providing Enigma with 2 more rotors: **5 * 4 * 3 = 60**
  - increased the number of plugboard cables from 6 to 10
- 波兰人没有足够的经费造出这么多Enigma机器
- Hans-Thilo Schmidt stopped passing on information and codebooks
- In late April 1939, Germany repudiated its non-aggression pact with Poland.
- On 24 July 1939, Pole showed their achievement to his French and British counterparts.
- 1 September 1939, Hitler invaded Poland

https://vc.airvectors.net/ttcode_08.html

Government Communications Headquarters (GCHQ)

GCHQ is an intelligence and security organization, working to keep Britain safe and secure in the challenging environment of modern communications

GCHQ

WHO WE ARE | WHAT WE DO | HOW WE WORK | CAREERS | PRESS & MEDIA

Search | Go »

You are here > Home > History

Beginnings

Britain's Signals Intelligence effort essentially dates from the beginning of World War I. A number of radio intercept stations were then created, and an increasing number of cryptanalysts, linguists and radio traffic analysts enjoyed considerable success in decrypting messages sent by Germany and its allies and in disseminating this

WWII: Bletchley Park

Bletchley Park, a country house in Buckinghamshire, was bought by SIS in 1938 as a site to which the Government Code & Cypher School and SIS could be evacuated when war came.

It was widely expected that London would be the target

Post War

After the war, the Government Code & Cipher School changed its name officially to GCHQ and moved its headquarters to Eastcote in Middlesex (1946), and later to Cheltenham in Gloucestershire (1950s).

The Central Training School stayed at Bletchley Park until

布莱切利公园(Bletchley Park)

- Government Code and Cypher School moved to Bletchley Park in 1939
  - crossword enthusiasts
  - chess mavens
  - Linguists
  - mathematicians
  - computer scientists

Bletchley Park, 50 miles (80km) north-west of London

http://www.bletchleypark.org.uk/

Vulnerabilities in the usages of Enigma

- 过于简单的密钥：比如DFG
- 重复使用的密钥：比如， CIL (cillie)， -> silly
- 已知明文攻击 (Know plaintext)，候选单词（crib）
  - weather report
  - Heil Hitler
  - Nothing to report
  - Seeding, to get cribs
- A letter can never be encrypted to itself.

Cribs

When attempting to crack a coded message (ciphertext), it helps if you can somehow guess some part of the original message (plaintext). This is known as a crib. Finding a crib will help you to deduce the key, which in turn will help you to decipher the rest of the message.

Plaintext + Key = Ciphertext

You can think of codebreaking in terms of solving the equation above. In general, this equation is tough to solve because it has two unknowns. All you know is the ciphertext that you have intercepted, and you have no idea of the key and the plaintext, the latter being your ultimate goal.

However, if you can guess a bit of the plaintext, then you have an equation with only one unknown. Solving an equation with only one unknown is much easier. Once you have deduced the key, you can decipher the complete message, and any other messages encrypted using the same key.

Finding cribs is difficult, but the task was made easier for codebreakers at Bletchley Park because of the way that the Enigma machine works. Let's use an example in English to illustrate how a crib was found for Enigma messages.

Codebreakers might suspect that a message sent at 6.15 in the morning contains a reference to the time, so the challenge is to match the phrase 憐erosixonefive' with the correct bit of ciphertext. Crucially, the Enigma cipher can never encrypt a letter as itself. So the plaintext can be slid along and compared to the ciphertext until no letters match. Any comparison that satisfies this requirement might be a crib.

Slide the following guessed plaintext along the ciphertext using the arrows.

See if you can find the two correct potential positions for the crib.

Cribs

When attempting to crack a coded message (ciphertext), it helps if you can somehow guess some part of the original message (plaintext). This is known as a crib. Finding a crib will help you to deduce the key, which in turn will help you to decipher the rest of the message.

Plaintext + Key = Ciphertext

You can think of codebreaking in terms of solving the equation above. In general, this equation is tough to solve because it has two unknowns. All you know is the ciphertext that you have intercepted, and you have no idea of the key and the plaintext, the latter being your ultimate goal.

However, if you can guess a bit of the plaintext, then you have an equation with only one unknown. Solving an equation with only one unknown is much easier. Once you have deduced the key, you can decipher the complete message, and any other messages encrypted using the same key.

Finding cribs is difficult, but the task was made easier for codebreakers at Bletchley Park because of the way that the Enigma machine works. Let's use an example in English to illustrate how a crib was found for Enigma messages.

Codebreakers might suspect that a message sent at 6.15 in the morning contains a reference to the time, so the challenge is to match the phrase 'erosixonefive' with the correct bit of ciphertext. Crucially, the Enigma cipher can never encrypt a letter as itself. So the plaintext can be slid along and compared to the ciphertext until no letters match. Any comparison that satisfies this requirement might be a crib.

Slide the following guessed plaintext along the ciphertext using the arrows.

See if you can find the two correct potential positions for the crib.

Z E R O S I X O N E F I V E

Z X I N P T Z U T P X E P S F V S X B I E A L C R

Cribs

When attempting to crack a coded message (ciphertext), it helps if you can somehow guess some part of the original message (plaintext). This is known as a crib. Finding a crib will help you to deduce the key, which in turn will help you to decipher the rest of the message.

Plaintext + Key = Ciphertext

You can think of codebreaking in terms of solving the equation above. In general, this equation is tough to solve because it has two unknowns. All you know is the ciphertext that you have intercepted, and you have no idea of the key and the plaintext, the latter being your ultimate goal.

However, if you can guess a bit of the plaintext, then you have an equation with only one unknown. Solving an equation with only one unknown is much easier. Once you have deduced the key, you can decipher the complete message, and any other messages encrypted using the same key.

Finding cribs is difficult, but the task was made easier for codebreakers at Bletchley Park because of the way that the Enigma machine works. Let's use an example in English to illustrate how a crib was found for Enigma messages.

Codebreakers might suspect that a message sent at 6.15 in the morning contains a reference to the time, so the challenge is to match the phrase 'erosixonefive' with the correct bit of ciphertext. Crucially, the Enigma cipher can never encrypt a letter as itself. So the plaintext can be slid along and compared to the ciphertext until no letters match. Any comparison that satisfies this requirement might be a crib.

Slide the following guessed plaintext along the ciphertext using the arrows.

See if you can find the two correct potential positions for the crib.

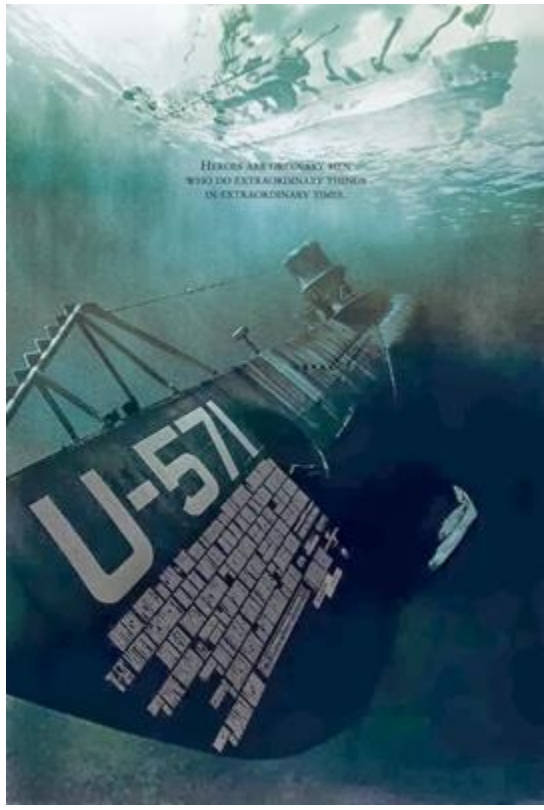
< Z E R O S I X O N E F I V E >

Z X I N P T Z U T P X E P S F V S X B I E A L C R

使用中的漏洞

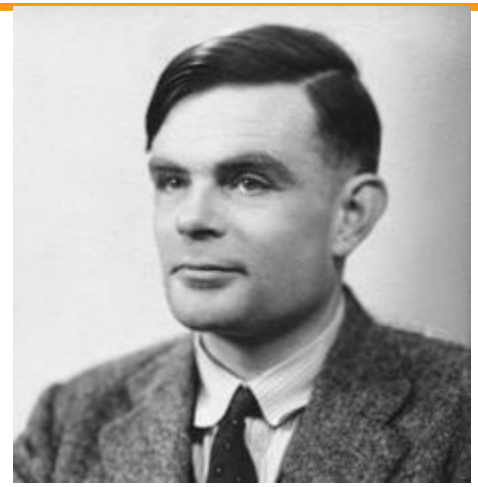
- 过分严格的规定
  - 转子的位置不得重复：
    - 如对3个rotor位置, 不是6， 而是 2
  - Plugboard上的相邻的字符不得相邻（如A-B）

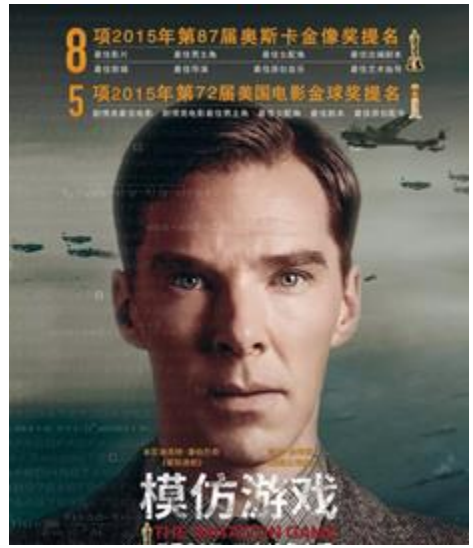
- 其他手段：Steal the codebook
  - 电影：007, U571



Contributions by Alan Turing

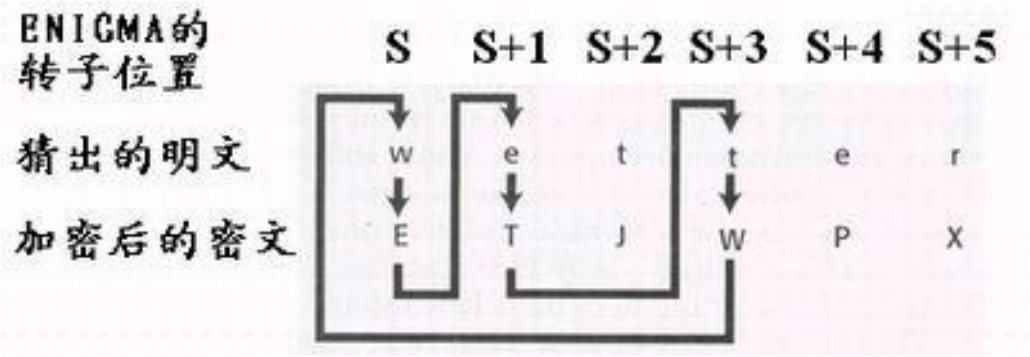
- Alan Turing(1912-1954)
  - 1931年进入剑桥大学，提出图灵机模型
  - 1933年受聘于GCCS，1939进入Bletchley Park
- 波兰的方法依赖密钥的重复，如果德军改变密钥规则怎么办？
- 图灵的方法在候选单词的明文和密文之间找到字母循环圈的指纹





Circles in the crib and its ciphertext

- 已知明文（Known plaintext）
- 在Crib 和对应的ciphertext中 寻找最短的 circles
  - Crib:　w e t t　e　r
  - cipher: E T J　W　P　X
  - Circle:　w -> e -> t -> w



- S:　w->E
  - Enigma 的状态改变到S+1
- S+1: e->T
  - Enigma 的状态改变到S+2
  - T->J，　改变到S+3
- S+3：　t->w
  - Enigma的状态改变到S+4

Turing's Bombe

Turing website www.AlanTuring.net

Although the achievements of Bletchley Park were a team effort, Alan Turing stands out as probably the greatest British codebreaker of the Second World War.

Before the outbreak of the war, he was a mathematician at Cambridge University, where he laid the foundations for the theory of computers, many years before a programmable machine could be built. After joining Bletchley Park, he turned his brilliant mind to the problem of cracking codes. Turing's most important idea was the 'bombe', a mechanised way of cracking Enigma based on exploiting cribs. What follows is a greatly oversimplified explanation of Turing's brilliant idea.

Alan Turing
(copyright Dermot Turing)

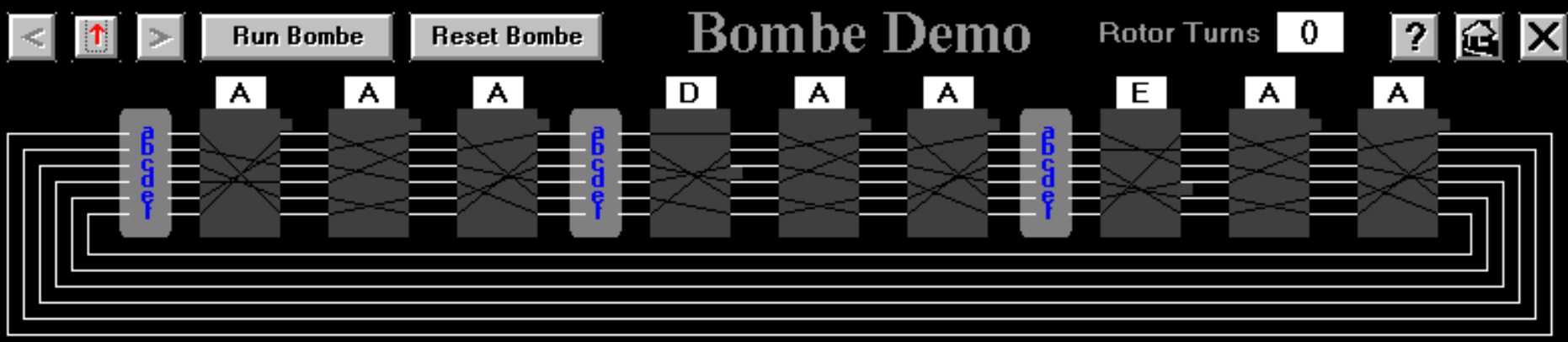
Imagine that we are dealing with a 6-letter Enigma and that we have intercepted the Enigma enciphered message below. We suspect that the message really says BADCAFE. This puts a severe constraint on the key, because the key must turn BADCAFE into CFAABDD, and most Enigma settings will not do this.

However, the number of possible keys is still too large to check. Turing was able to reduce the problem by concentrating on those letters that formed a loop, e.g., the C in the ciphertext can be joined to the C in the plaintext; this is encrypted as A, which can be joined to the next A; this is encrypted as B, which can be joined to the starting B, which is encrypted into the C that we started with, so the loop is complete. **Click here to see the loop.**

CRIB

Plaintext	B	A	D	C	A	F	E
	↓	↓	↓	↓	↓	↓	↓
Ciphertext	C	F	A	A	B	D	D

To see why this loop helps crack Enigma, have a look at the bombe demo. >



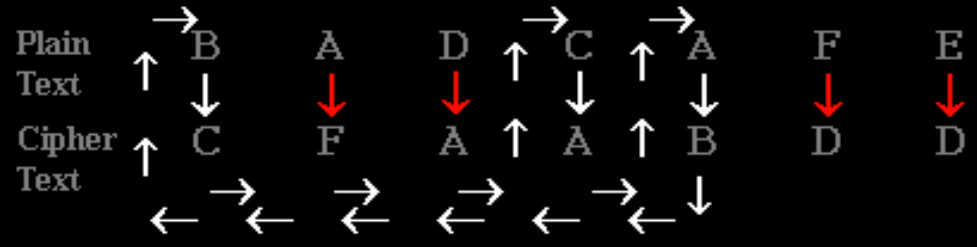
The loop containing 3 pairs of letters leads us to the following scenario. Take three Enigma machines that will correspond to the three pairings. The first is set to some basic setting, the second is set so that its rotor is three places further on (because it represents the letter pairing three places further on), and the third is set so that its rotor is four places further on (because it represents the letter pairing four places further on). The three Enigmas can then be directly wired to each other, because the loop tells us that the output of the first is the same as the input of the second and so on. Bletchley constructed larger blocks of Enigmas (known as bombes) that could be wired in this way.

In the vastly simplified diagram above, there is no plugboard to be seen, because Turing realised that we can ignore the plugboard and the massive key space that it would otherwise contribute to the problem. The plugboard can be ignored because at the output of the first Enigma it might swap A with P, but at the input of the next Enigma it would swap P back with A ?the plugboards effectively cancel each other out.

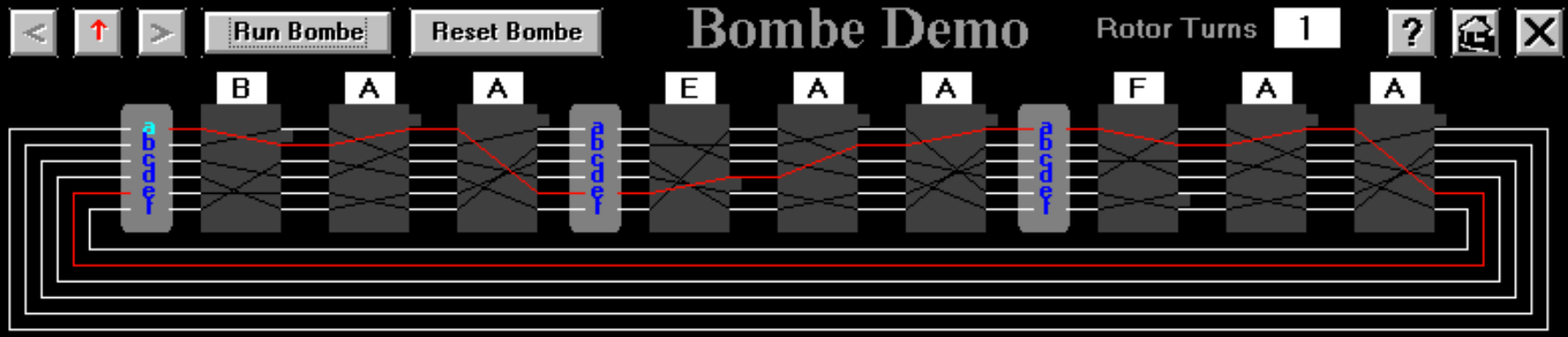
With this set up, Turing's bombe could check for consistency. Was there a loop in the bombe that reflected the original loop in the crib? Once each of the six inputs are tested, every Enigma moves forward one rotor position, keeping in step, and the six inputs are tested again. This process is repeated until a complete circuit is found. You can see this by clicking the button at the top of the page and running the

B-> . . C -> A

A complete circuit might represent the rotor position part of the key that was used to encrypt the message, and the remainder of the key can be deduced relatively easily. There may be several complete circuits, but the number of false consistencies can be greatly reduced by using longer cribs with more complex loops.



Click here to see a picture of a bombe.

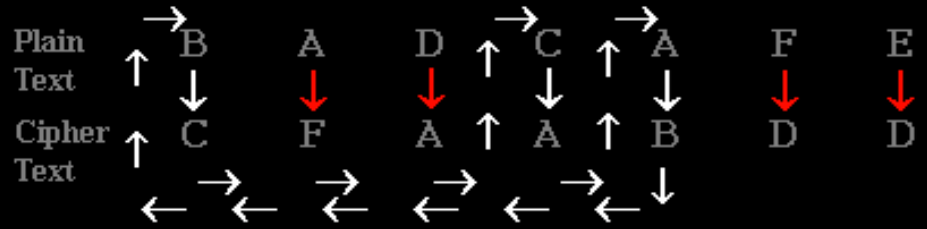


The loop containing 3 pairs of letters leads us to the following scenario. Take three Enigma machines that will correspond to the three pairings. The first is set to some basic setting, the second is set so that its rotor is three places further on (because it represents the letter pairing three places further on), and the third is set so that its rotor is four places further on (because it represents the letter pairing four places further on). The three Enigmas can then be directly wired to each other, because the loop tells us that the output of the first is the same as the input of the second and so on. Bletchley constructed larger blocks of Enigmas (known as bombes) that could be wired in this way.

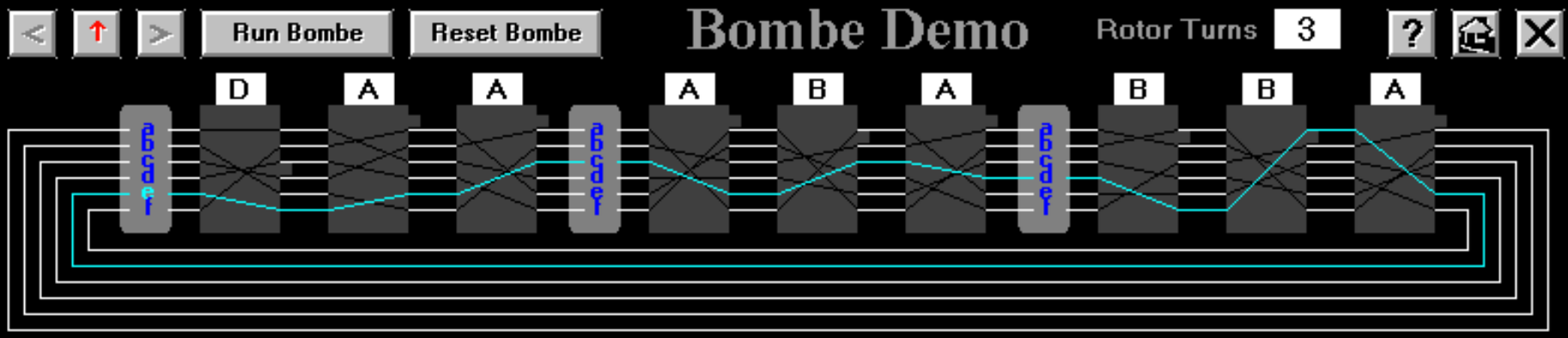
In the vastly simplified diagram above, there is no plugboard to be seen, because Turing realised that we can ignore the plugboard and the massive key space that it would otherwise contribute to the problem. The plugboard can be ignored because at the output of the first Enigma it might swap A with P, but at the input of the next Enigma it would swap P back with A ?the plugboards effectively cancel each other out.

With this set up, Turing's bombe could check for consistency. Was there a loop in the bombe that reflected the original loop in the crib? Once each of the six inputs are tested, every Enigma moves forward one rotor position, keeping in step, and the six inputs are tested again. This process is repeated until a complete circuit is found. You can see this by clicking the button at the top of the page and running the bombe demo.

A complete circuit might represent the rotor position part of the key that was used to encrypt the message, and the remainder of the key can be deduced relatively easily. There may be several complete circuits, but the number of false consistencies can be greatly reduced by using longer cribs with more complex loops.



Click here to see a picture of a bombe.

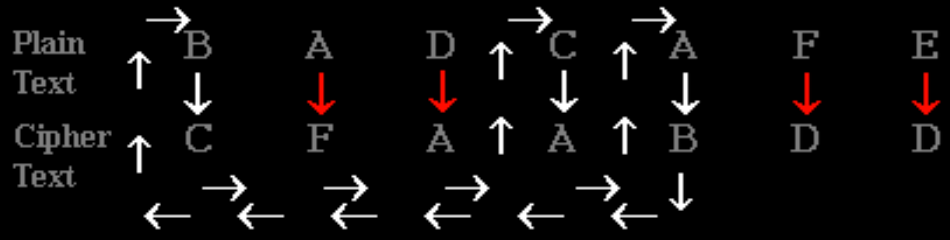


The loop containing 3 pairs of letters leads us to the following scenario. Take three Enigma machines that will correspond to the three pairings. The first is set to some basic setting, the second is set so that its rotor is three places further on (because it represents the letter pairing three places further on), and the third is set so that its rotor is four places further on (because it represents the letter pairing four places further on). The three Enigmas can then be directly wired to each other, because the loop tells us that the output of the first is the same as the input of the second and so on. Bletchley constructed larger blocks of Enigmas (known as bombes) that could be wired in this way.

In the vastly simplified diagram above, there is no plugboard to be seen, because Turing realised that we can ignore the plugboard and the massive key space that it would otherwise contribute to the problem. The plugboard can be ignored because at the output of the first Enigma it might swap A with P, but at the input of the next Enigma it would swap P back with A ?the plugboards effectively cancel each other out.

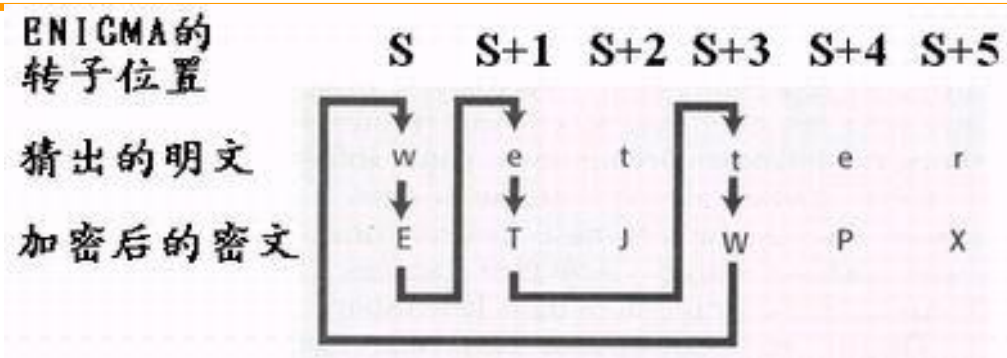
With this set up, Turing's bombe could check for consistency. Was there a loop in the bombe that reflected the original loop in the crib? Once each of the six inputs are tested, every Enigma moves forward one rotor position, keeping in step, and the six inputs are tested again. This process is repeated until a complete circuit is found. You can see this by clicking the button at the top of the page and running the bombe demo.

A complete circuit might represent the rotor position part of the key that was used to encrypt the message, and the remainder of the key can be deduced relatively easily. There may be several complete circuits, but the number of false consistencies can be greatly reduced by using longer cribs with more complex loops.

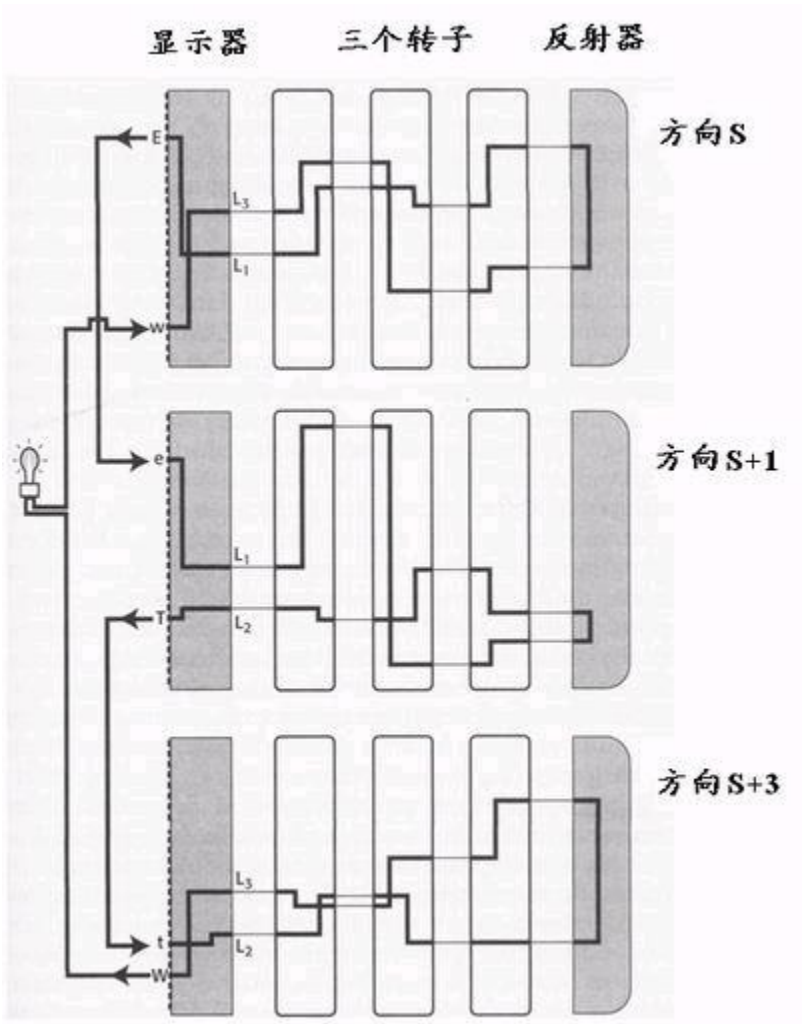


Click here to see a picture of a bombe.

Connect multiple Enigmas in a circle



- According to
  - The number(3) of the circle
  - The position of the letter in the plaintext(S, S+1, S+3)
- 3 Enigma were connected:
  - Same rotors and orders
  - The orientation of the rotors were set to S, S+1, S+3
- If all the status were right, the lamp bulb should be lightened on



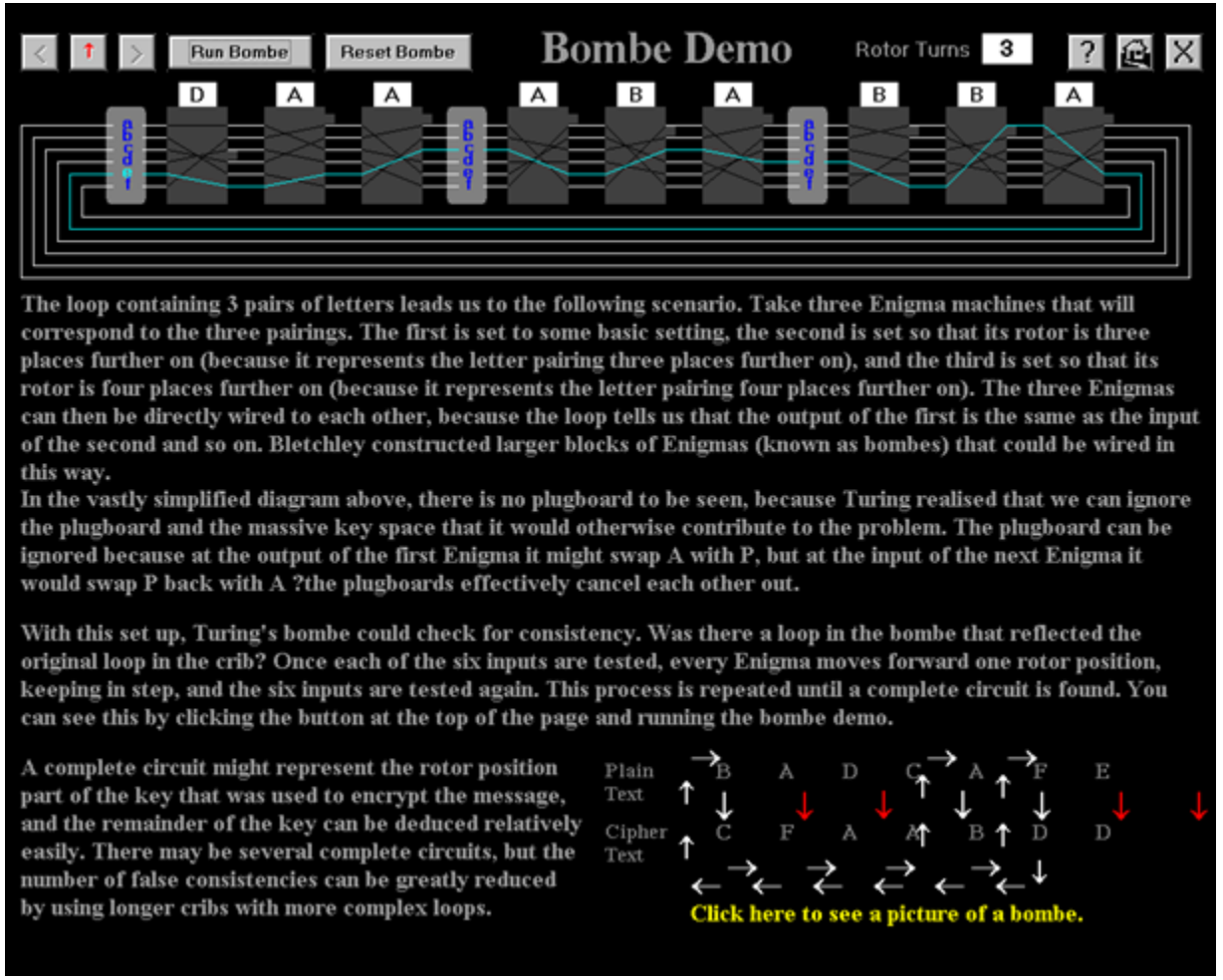
穷举转轮的初始状态，需多少次？

需要多少次测试？

26 X 26 X 26 = 17576 ?

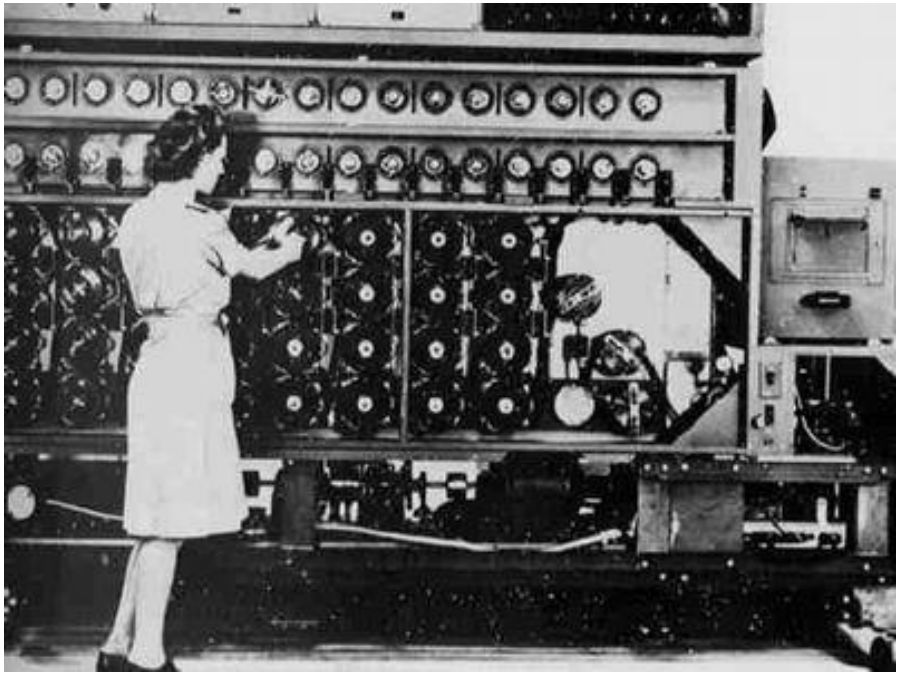
还是

17576 X 17576 X 17576 ?



bombe: 多台Enigma并行工作

- 德国增强了设计
  - Select 3 from 5 rotors
  - 那么造 60台
- 60 Parallel
  - 12 letters in the circle
  - 12 enigma in a circle
  - 60 * 12 = 720 台
- Given cribs, the key could be found in weeks , later in hours



http://www.bletchleypark.org.uk/

Bletchley Park after WWII

- WWII -> Cold War
- 1974 '*The Ultra Secret*' by Winterbotham
- 1991面临拆迁
- 1992 , Bletchley Park Trust成立
- 1999, Trust 获得250 年的租期
- 2008, investment from English Heritage

http://www.bletchleypark.org.uk/

计算机诞生之后的密码算法

异或运算XOR

$$0 \oplus 0 = 0$$

$$1 \oplus 1 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$x \oplus 0 = x$$

$$x \oplus x = 0$$

$$x \oplus 1 = x^{-1}$$

$$x \oplus x^{-1} = 1$$

XOR, 一种简单而有效的加密算法?

- Algorithm:

```
for (i=0; i<length_of_P; i++)
    C[i] = P[i]  XOR K[ i MOD length_of_key]
```

```
>>> p
['A', 'T', 'T', 'A', 'C', 'K', 'A', 'T', 'D', 'A', 'W', 'N']
>>> c=[ord(p[i]) ^ ord(k[i%len(k)]) for i in range(0,len(p)) ]
>>> c
[13, 17, 25, 14, 13, 7, 4, 25, 11, 15, 27, 11]
```

```
>>> d=[chr(c[i] ^ ord(k[i%len(k)])) for i in range(0,len(c)) ]
>>> d
['A', 'T', 'T', 'A', 'C', 'K', 'A', 'T', 'D', 'A', 'W', 'N']
```

- 已知明文、密文，怎样求得密钥？ $K=C \oplus P$

```
>>> kk=[chr(c[i] ^ ord(p[i])) for i in range(0,len(c)) ]
>>> kk
['L', 'E', 'M', 'O', 'N', 'L', 'E', 'M', 'O', 'N', 'L', 'E']
```

XOR, 一种简单而有效的加密算法?

- Algorithm:

```
FOR (i=0; i<length_of_P; i++)
    C[i] = P[i]  XOR K[ i MOD length_of_key]
```

```
>>> p
['A', 'T', 'T', 'A', 'C', 'K', 'A', 'T', 'D', 'A', 'W', 'N']
>>> c=[ord(p[i]) ^ ord(k[i%len(k)]) for i in range(0,len(p)) ]
>>> c
[13, 17, 25, 14, 13, 7, 4, 25, 11, 15, 27, 11]
```

```
>>> d=[chr(c[i] ^ ord(k[i%len(k)])) for i in range(0,len(c)) ]
>>> d
['A', 'T', 'T', 'A', 'C', 'K', 'A', 'T', 'D', 'A', 'W', 'N']
```

只知道密文，如何求得明文和密钥？

- 仅有密文，怎样求得密钥？ $K=C \oplus ?$

Plaintext: ATTAC KATDA WNONE ATTAC K……

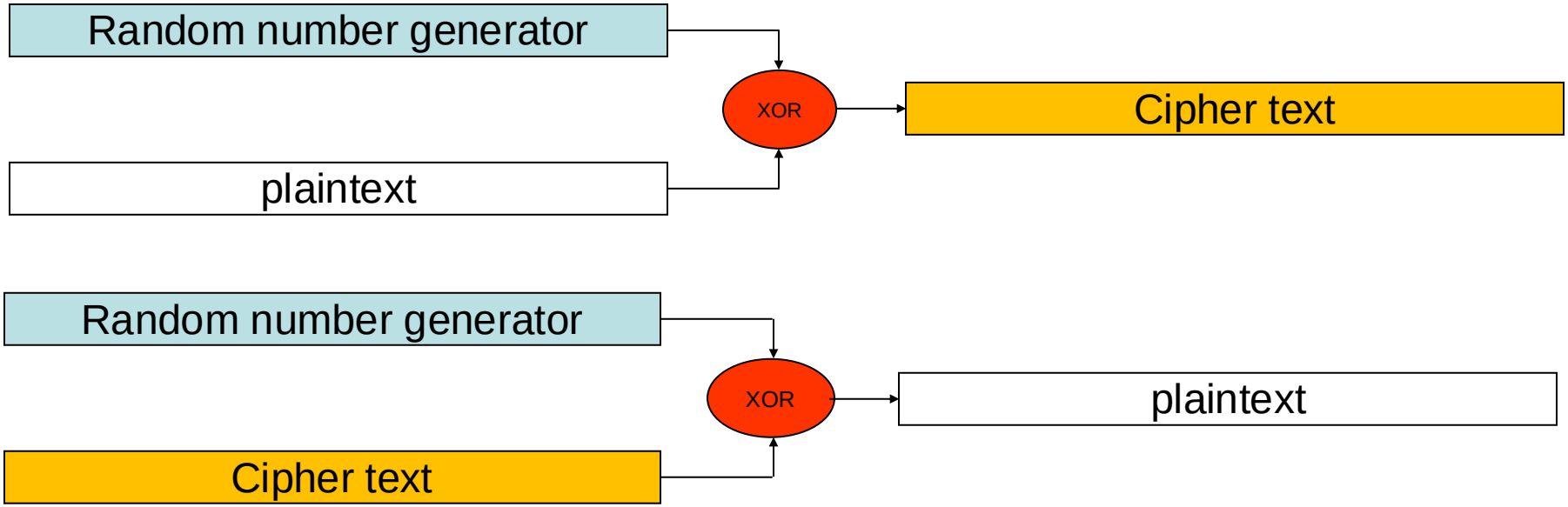
Key: LEMON LEMON LEMON LEMON……

XOR

Ciphertext: 11172514 13070425 1115 . . . 11172514……

- 这只是一个类似Vigener的多表替换
- 不用计算机，甚至都可以攻破它
- 如果密钥密钥很长很长，会怎么样呢？
  - 比如，无限长

Stream Cipher(流密码，序列密码)



- **如果接收双方的使用相同的随机数序列，且足**够长，那么这个**算法将是不可破解的**

- **如果可以 安全**传输这个随机数序列，为何不直接传明文信息呢？

量子通信/量子密钥分发

- **BB84协议**

如果存在监听者，可被察觉

经典信道（如Internet）

量子通信链路（光线）

Alice的随机比特	0	1	1	0	1	0	0	1
Alice随机选择的基	+	+	×	+	×	×	×	+
Alice所传光子的偏振态	↑	→	↘	↑	↘	↗	↗	→
Bob随机选择测量的基	+	×	×	×	+	×	+	+
Bob测量的光子的偏振态	↑	↗	↘	↗	→	↗	→	→
在公共信道中对比基								
共有的密钥	0		1			0		1

量子信道只用于协商出一个随机数，数据传输仍然依赖经典信道

https://en.wikipedia.org/wiki/BB84

Stream Cipher

- 给定Seed或Secret, 产生伪随机数PRNG
- C= P XOR PRNG(secret)
- P= C XOR PRNG(secret)

RC4 Overview

- RC4,by Ron Rivest ,1987, RSA patent, Disclosed in 1994; Arcfour,
- Array: S[256], K[256]; 8bits Registers: i, j ; Less than 100 lines C code

Key-scheduling algorithm (KSA)

```
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod keylength]) mod 256
    swap(&S[i],&S[j])
endfor
```

PRGA

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap(&S[i],&S[j])
    byte_cipher := S[(S[i] + S[j]) mod 256]
    result_ciphered := byte_cipher XOR byte_message
endwhile
```

reuse the random sequence

C1 XOR C2 = (P1 XOR **R**) XOR (P2 XOR **R**)

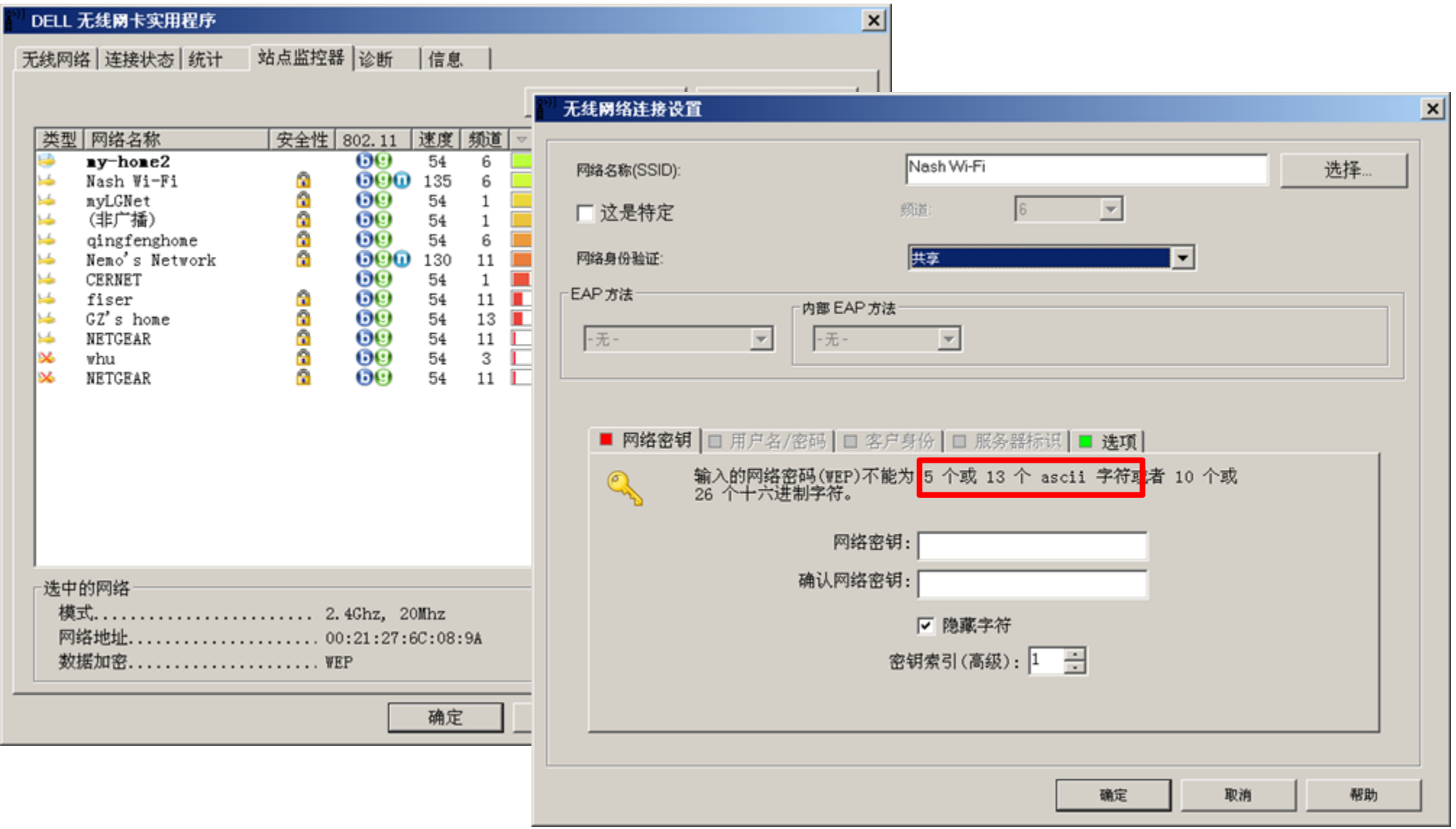
= P1 XOR P2

– If we know P1, then we can get P2

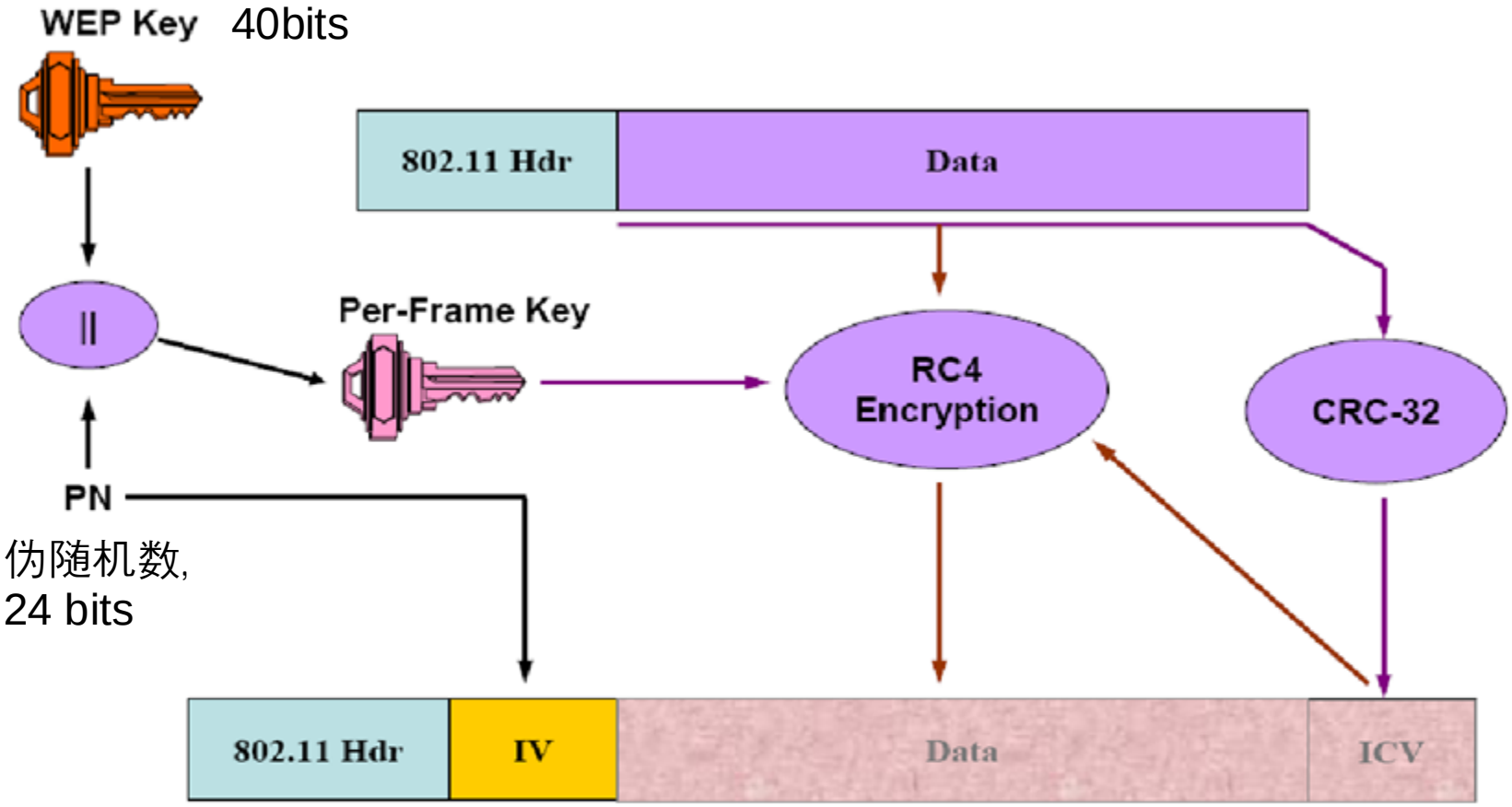
– Or, we can recover P1, P2 only from P1 XOR P2

- E. Dawson and L. Nielsen. Automated cryptanalysis of XOR plaintext strings. *Cryptologia*, (2):165–181, Apr. 1996.

WEP (Wired Equivalent Privacy)



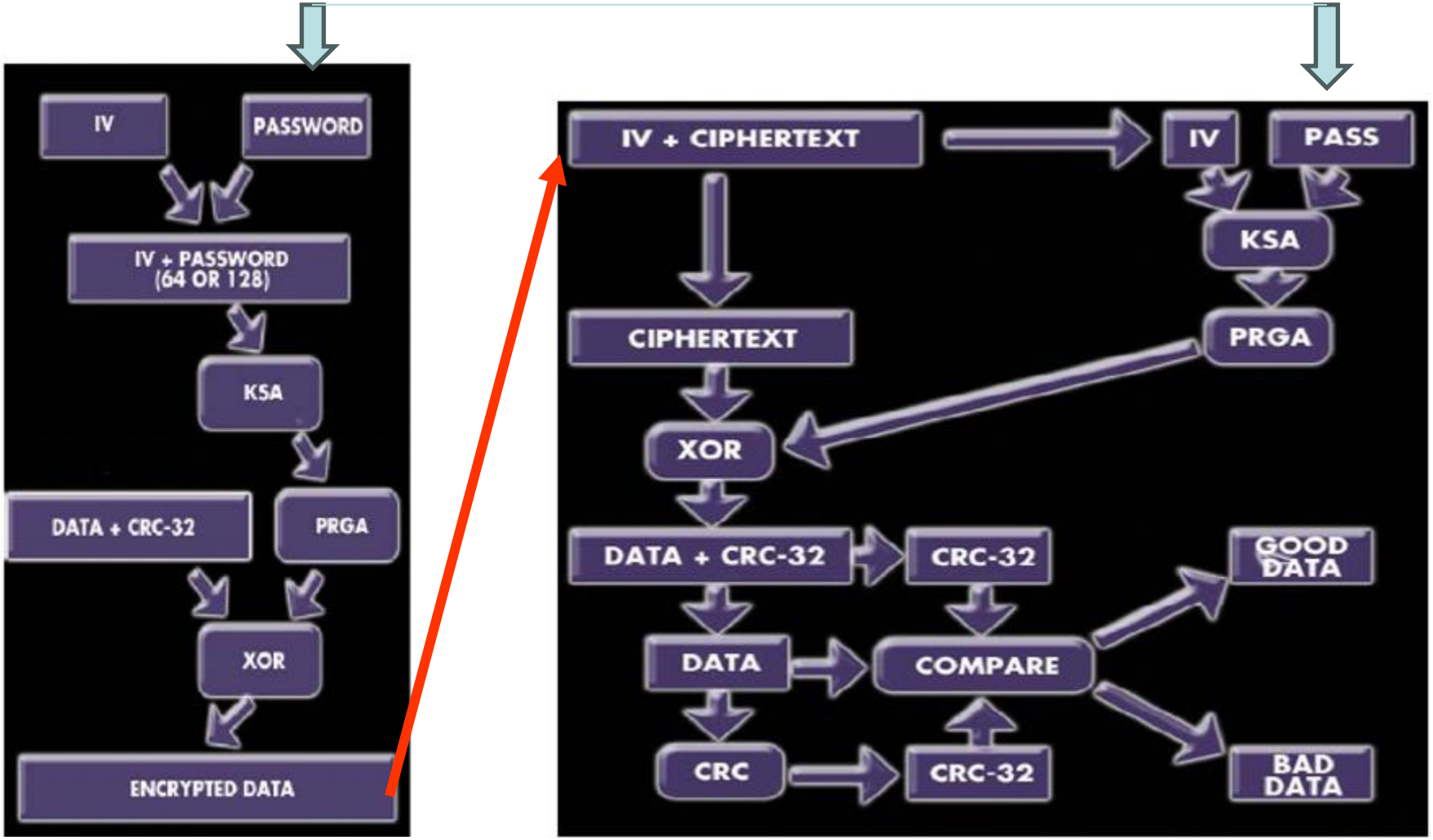
WEP Frame



Jesse Walker, A History of 802.11 Security

WEP Encryption and Decryption

IV: 24 bits, password 40bits



Attacks on WEP

- Key size is not the only problem
- **KEYSTREAM REUSE**
  - IV : 24 bits ， 如果两次加密使用相同的IV，结果？
  - 两个Packet重用IV，Collision
    - 需要多少Packets？ 2^{24} ？
    - Birthday attack:
      - 1% chance of collision after 582 encrypted frames;
      - 10% after 1,881
      - 50% after 4,823; 99% after 12,430
  - 发12,430个IP Packet，需要多长时间？

Attacks on WEP

- Walker, “Unsafe at any Key Size” ,October 2000
  - http://www.dis.org/wl/pdf/unsafe.pdf
- Nikita Borisov,Ian Goldberg, David Wagner. Intercepting Mobile Communications: The Insecurity of 802.11
  - http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf
- Weaknesses in the Key Scheduling Algorithm of RC4
  - http://course.ccert.edu.cn/upload/Dhx/Weaknesses-in-the-Key-Scheduling-Algorithm-of-RC4.pdf
- Cracking WEP
  - http://course.ccert.edu.cn/upload/Dhx/Cracking-WEP.pdf

Crack Tools

- WEPCrack, an open source tool
  - http://wepcrack.sourceforge.net/
- **AirSnort, Aircrack-ng**
  - http://airsnort.sourceforge.net/
  - http://www.aircrack-ng.org/doku.php
- **Kali Linux**

回顾

100-40 B.C. Caesar cipher

800 A.D. Frequency analysis

Vigenere ciper, 1463

1800 , Babbage

Friedrich Wilhelm Kasiski

1920 Enigma

1940 Turing

1987 RC4

2001 attacks against RC4

2015 prohibit RC4 in TLS

Lessons we should learned

- 攻击与防御，永无休止的军备竞赛（arm race）
- 演进，还是革命？
- 密码技术的革命和进步始于对旧的密码体制的攻击
  - 证伪：并非证明技术无效，而是找出它有效的边界
  - 对密码技术不懈的攻击，导致密码算法越来越安全
  - 如果不允许密码算法的破解会怎样？
- 密钥的管理通常加密体制中最薄弱的环节
  - 重复使用，随机数
- 加密体制的安全不能建立在算法保密的基础上
  - 开放的环境；算法开放，密钥保密

Reference

- Codes, Ciphers, & Codebreaking
  - http://vc.airvectors.net/ttcode.html

- **The Code Book** on CD-ROM
  - https://simonsingh.net/books/the-code-book/the-book/

Simon Singh

The Code Book

When I wrote my first book, Fermat's Last Theorem, I made a passing reference to the mathematics of cryptography. Although I did not know it at the time, this was the start of a major interest in the history and science of codes and code breaking, which has resulted in a 400-page book on the subject, an adaptation of the book for teenagers, a 5-part TV series, numerous talks and lectures, the purchase of an Enigma cipher machine and the development of an interactive crypto CD-ROM.

You can find information about TV series and the free CD-ROM version of the book (and lots more) in the Cryptography section of the site. In this section, however, you will find details about my book on cryptography (The Code Book).

I have not included any material about the teenage version of The Code Book, but you can find it at Amazon.co.uk and Amazon.com.

SEARCH

Search

ABOUT ME

Find out more about my latest book...

I am an author, journalist and TV produ-

The Code Book on CD-ROM
https://simonsingh.net/books/the-code-book/the-book/