



西安交通大学
XI'AN JIAOTONG UNIVERSITY

现代密码学 COMP401227

第 1 章：密码学简介

1.5 古典密码

赵俊舟

`junzhou.zhao@xjtu.edu.cn`

2025 年 2 月 25 日

目录

- 1 代换密码
- 2 置换密码
- 3 转轮密码机

目录

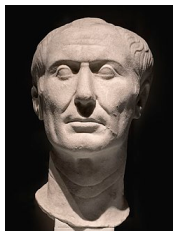
- 1 代换密码
- 2 置换密码
- 3 转轮密码机

代换密码 (Substitution Cipher)

- **代换密码** (也叫**替换密码**) 将一个明文字母替换成其他字母, 实现加密, 通过逆替换实现解密, 是最简单的密码。
- 常见的代换密码包括:
 - 凯撒密码或移位密码
 - Playfair 密码
 - Hill 密码
 - Vigenère 密码
 - 一次一密 (One-Time Pad)
- 分为**单表代换**、**多表代换**等。

凯撒密码 (Caesar Cipher)

- 已知最早的代换密码是由古罗马时期的 Julius Caesar 发明的凯撒密码 (Caesar Cipher)，用于加密作战命令。

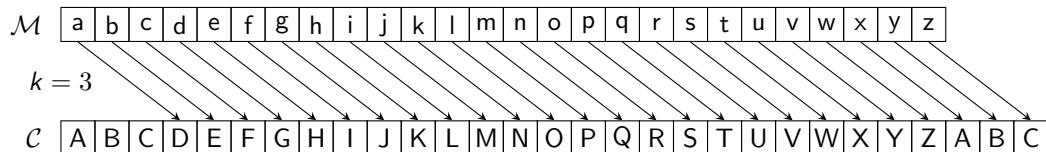


(公元前 100 年 – 公元前 44 年)

There are also letters of his to Cicero, as well as to his intimates on private affairs, and in the latter, if he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out...

The Lives of the Caesars, the Deified Julius (110CE)

凯撒密码



- 将每个明文字母向前循环移 3 位，得到密文字母；对密文进行相反操作得到明文。
- **加密**: $c = (m + 3) \bmod 26$
- **解密**: $m = (c - 3) \bmod 26$

例 (凯撒密码加密)

利用凯撒密码加密 “begin the attack now”，忽略空格，得到密文
EHJLQWKHDWWDFNQRZ

移位密码

- 凯撒密码其实不存在密钥，任何知道凯撒密码算法的人都可以轻易破解密码。
- 移位密码**对凯撒密码进行改进，引入密钥 $k \in \{0, \dots, 25\}$ ，表示循环移位的位数。

$$\text{加密: } c = E(k, m) \triangleq (m + k) \bmod 26$$

$$\text{解密: } m = D(k, c) \triangleq (c - k) \bmod 26$$

- 移位密码的安全性如何？

例 (破解移位密码)

尝试破解密文 EHJLQWKHDWWDFNQRZ

移位密码的安全性

k	尝试解密后
0	ehjqlqwkhdwdfnqrz
1	dgikpvjgcvvcempqy
2	cfhjoui fbuubdl opx
3	begintheattacknow
4	adfhmsgdzsszbjmnv
5	zceglrfcyrryailmu
6	ybdfkqebxqqxzhklt
7	xacejpdawppwygjks
8	wzbdioczvoovxfijr
	...

- 移位密码的密钥空间大小为 26
- 容易对密钥空间进行**穷举攻击**
(brute-force attack)

密码安全原则一：密钥空间应足够大
任何安全的密码体制都应该具有足够大的密钥空间，使穷举攻击不可行。

- 多大的密钥空间才算**足够大**？

对不同密钥空间进行穷举攻击的开销

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/ μ s	Time Required at 10^6 Decryptions/ μ s
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{s} = 35.8 \text{ minutes}$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{s} = 1142 \text{ years}$	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{s} = 5.4 \times 10^{24} \text{ years}$	$5.4 \times 10^{18} \text{ years}$
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36} \text{ years}$	$5.9 \times 10^{30} \text{ years}$
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12} \text{ years}$	$6.4 \times 10^6 \text{ years}$

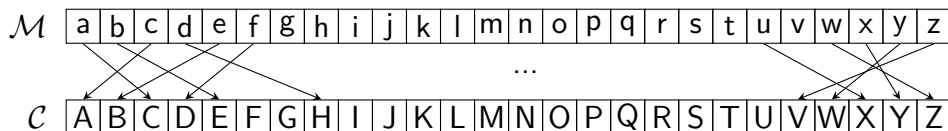
对不同密钥空间进行穷举攻击的开销

Amazon EC2 集群执行不同数量 CPU 时钟需要的花费
(根据 2018 年定价)

<i>clock cycles</i>	<i>approx cost</i>	<i>reference</i>
2^{50}	\$3.50	<i>cup of coffee</i>
2^{55}	\$100	<i>decent tickets to a Portland Trailblazers game</i>
2^{65}	\$130,000	<i>median home price in Oshkosh, WI</i>
2^{75}	\$130 million	<i>budget of one of the Harry Potter movies</i>
2^{85}	\$140 billion	<i>GDP of Hungary</i>
2^{92}	\$20 trillion	<i>GDP of the United States</i>
2^{99}	\$2 quadrillion	<i>all of human economic activity since 300,000 BC⁴</i>
2^{128}	<i>really a lot</i>	<i>a billion human civilizations' worth of effort</i>

单表代换密码

- 为增强移位密码的安全性，设计代换表时可以不仅仅是依次替换，而是允许任意替换，称为**单表代换密码**。
- 单表代换中每个明文字母可以映射到任意一个密文字母，密钥是 26 个字母的任意置换，共有 $26! > 4 \times 10^{26}$ 种可能密钥。



单表代换密码举例

例

使用代换表：

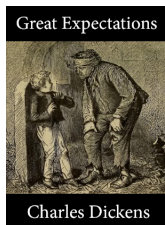
Plaintext: abcdefghijklmnopqrstuvwxyz
Ciphertext: DKVQFIBJWPESCXHTMYAUOLRGZN

对消息 “if we wish to replace letters” 进行加密：

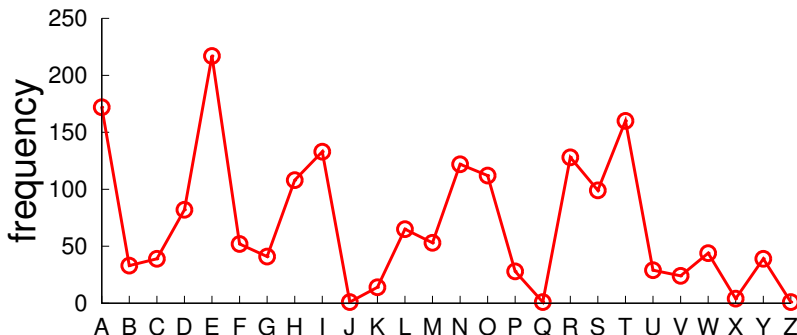
Plaintext: ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

- 单表代换密码有 $26! > 4 \times 10^{26}$ 种可能密钥，似乎已经足够大了，单表代换密码是否真的安全？

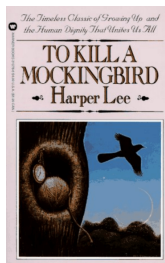
《远大前程》：查尔斯·狄更斯著长篇小说



My father's family name being Pirrip, and my Christian name Philip, my infant tongue could make of both names nothing longer or more explicit than Pip. So, I called myself Pip, and came to be called Pip. I give Pirrip as my father's family name, on the authority of his tombstone and my sister,—Mrs. Joe Gargery, who married the blacksmith. As I never saw my father or my mother, and never saw any likeness of either of them (for their days were long before the days of photographs), my first fancies regarding what they were like were unreasonably derived from their tombstones. The shape of the letters on my father's, gave me an odd idea that he was a square, stout, dark man, with curly black hair. From the character and turn of the inscription, "Also Georgiana Wife of the Above," I drew a childish conclusion that my mother was freckled and sickly. To five little stone lozenges, each about a foot and a half long, which were arranged in a neat row beside their grave, and were sacred to the memory of five little brothers of mine,—who gave up trying to get a living, exceedingly early in that universal struggle,—I am indebted for a belief I religiously entertained that they had all been born on their backs with their hands in their trousers-pockets, and had never taken them out in this state of existence. Ours was the marsh country, down by the river, within, as the river wound, twenty miles of the sea. wilderness beyond the churchyard, intersected with dikes and mounds and gates, with scattered cattle feeding on it, was the marshes; and that the low leaden line beyond was the river; and that the distant savage lair from which the wind was rushing was the sea; and that the small bundle of shivers growing afraid of it all and beginning to cry, was Pip.



《杀死一只知更鸟》：哈珀·李著长篇小说

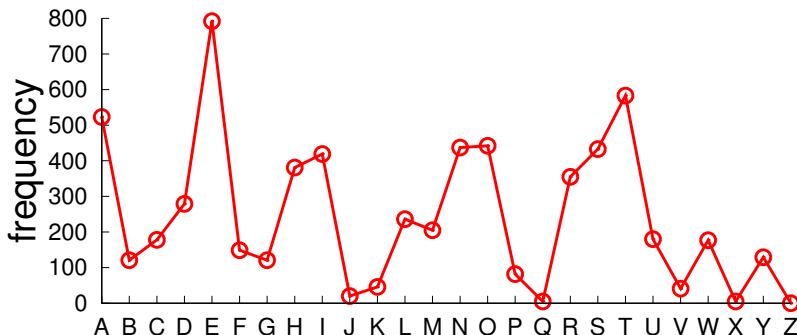


When he was nearly thirteen, my brother Jem got his arm badly broken at the elbow. When it healed, and Jem's fears of never being able to play football were assuaged, he was seldom self-conscious about his injury. His left arm was somewhat shorter than his right; when he stood or walked, the back of his hand was at right angles to his body, his thumb parallel to his thigh. He couldn't have cared less, so long as he could pass and punt.

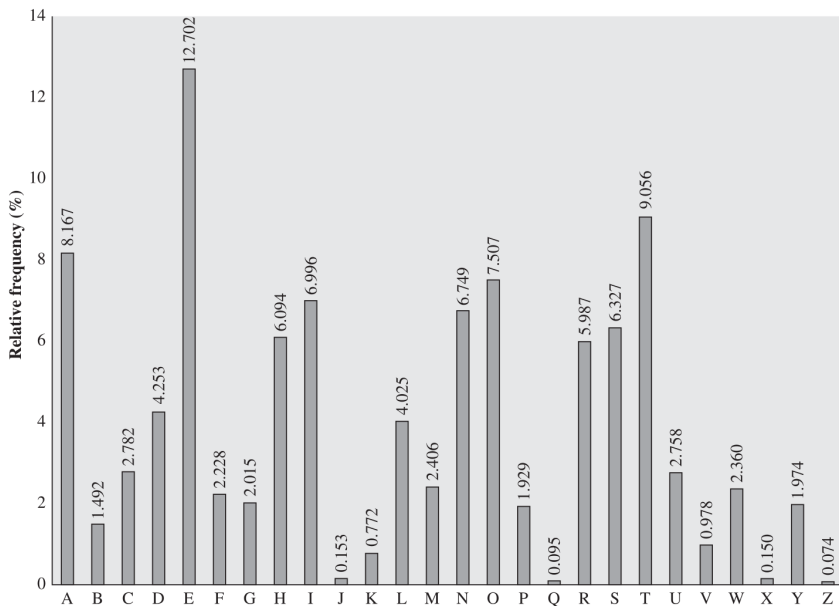
When enough years had gone by to enable us to look back on them, we sometimes discussed the events leading to his accident. I maintain that the Ewells started it all, but Jem, who was four years my senior, said it started long before that. He said it began the summer Dill came to us, when Dill first gave us the idea of making Boo Radley come out.

I said if he wanted to take a broad view of the thing, it really began with Andrew Jackson. If General Jackson hadn't run the Creeks up the creek, Simon Finch would never have paddled up the Alabama, and where would we be if he hadn't? We were far too old to settle an argument with a fist-fight, so we consulted Atticus. Our father said we were both right.

Maycomb was an old town, but it was a tired old town when I first knew it. In rainy weather the streets turned to red slop; grass grew on the sidewalks, the courthouse sagged in the square. Somehow, it was hotter then: a black dog suffered on a summer's day; bony mules hitched to Hoover carts flicked flies in the sweltering shade of the live oaks on the square. Men's stiff collars wilted by nine in the morning. Ladies bathed before noon, after their three-o'clock naps, and by nightfall were like soft teacakes with frostings of sweat and sweet talcum.



英文字母的相对使用频率



利用语言的统计特性进行密码分析

- 单表代换密码的密钥空间看似足够大，可以抵御穷举攻击，其实不然，这是因为**语言往往具有统计特性**。
- 人类的语言是有冗余性的，字母使用的频率并不一样：英文字母 E 是使用最频繁，然后是 T, R, N, I, O, A, S 等；有些字母使用得很少，如 Z, J, K, Q, X；双字母也有统计特性，例如 TH 等。
- 这样可以得到**英文字母使用频率分布表**，最早由阿拉伯科学家在公元九世纪发现。
- 单表代换不能掩盖字母出现的频率，只要统计密文中字母出现的频率，与已知的统计值做比较就可以分析出明密文字的对应关系。

单表代换密码攻击举例

例

- 给定密文：

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSX
AIZVUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- 统计相关字母出现的次数，可以猜测 P 和 Z 是 e 和 t，ZW 是 th，这样 ZWP 就是 the。
- 这样反复试验并不断修正错误，最后可得：
it was disclosed yesterday that several informal
but direct contacts have been made with political
representatives of the viet cong in moscow

Playfair 密码

- 两种常用方法用于减少明文结构在密文中的残留度：
 - 多字母代换密码：对明文中的多个字母一起加密；
 - 多表代换密码
- Playfair 密码是最著名的多字母代换密码，由英国科学家 Charles Wheatstone 在 1854 年发明的，以 Lyon Playfair 的名字命名。
- 首次应用于克里米亚战争（1854），最后一次应用是在一战。于 1915 年被德国破解，其变种被德国和英国应用于二战。
- Playfair 密码把明文中的双字母作为一个单元转换成密文的双字母。

Playfair 密码

- Playfair 算法基于一个由**密钥词**构成的 5×5 **密钥矩阵**。
- 先在 5×5 密钥矩阵中填上密钥词，去掉重复字母。
- 再将剩余的字母按字母表的顺序从左至右、从上至下填在矩阵剩下的格子中，I 和 J 当作一个字母。

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

使用密钥词 MONARCHY

Playfair 密码的代换规则

- 将明文拆为字母对，如果字母对的两个字母是相同的，则在其中插入一个填充字母，如 'x'，"balloon" 变成 "ba lx lo on"。
- 对明文的每个字母对，利用密钥矩阵进行代换。
- 代换规则 1**：落在同一行的明文字母对中的字母由其右边的字母来代换，每行中最右的字母用该行最左边的第一个字母来代换，如 "ar" 加密成 "RM"。

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

→ RM

Playfair 密码的代换规则

- **代换规则 2**: 落在同一列的明文字母对中的字母由其下面的字母来代换, 每列中最下面的一个字母用该列最上面的第一个字母来代换, 如 “mu” 加密成 “CM”。
- **代换规则 3**: 其他的每组明文字母对中的字母按如下方式代换: 该字母所在行为密文所在行, 另一字母所在列为密文所在列, 如 “hs” 变换成 “BP”, “ea” 代换为 “IM” 或 “JM”。

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

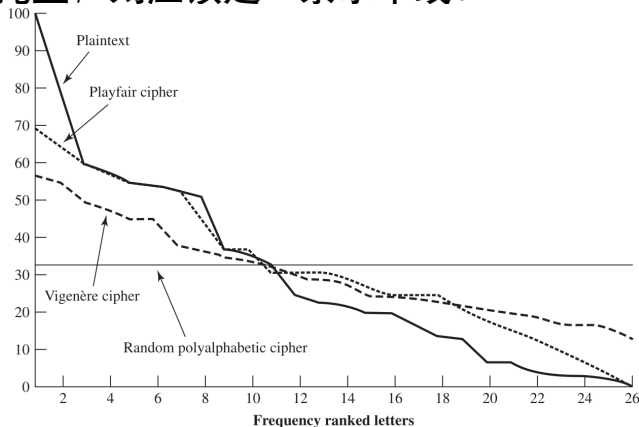
M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair 密码的安全性

- Playfair 密码的安全性比单字母的单表代换密码高。
- 因为有 26 个字母，所以共有 $26 \times 26 = 676$ 个字母对，对字母对进行判断要比对单个字母困难得多。
- 单字母的相对频率比字母对的相对频率有更好的统计规律，利用频率分析字母对就更困难，需要 676 维的频率表。
- Playfair 密码的密文仍然完好地保留了明文语言的大部分结构特征，它仍然是相对容易攻破的，几百个字母的密文就足够分析出规律了。

字母出现的相对频率

- “明文”曲线画出 7 万个字母的频率分布，对文中出现的每个字母计数，结果除以字母 e 的出现次数，按降序排序。
- 加密后的曲线体现了加密后字母频率分布被掩盖的程度，如果完全被掩盖，则应该是一条水平线。



Hill 密码

- 1929 年美国数学家 Lester Hill 发明 Hill 密码。为每个字母指定一个数值，将 m 个连续明文替换成 m 个密文，由 m 个线性方程决定，例如 $m = 3$ 时

$$c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$$

$$c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$$

$$c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$$

- 用矩阵表示为

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \bmod 26$$

- 加密**: $C = E(K, P) = KP \bmod 26$
- 解密**: $P = D(K, C) = K^{-1}C \bmod 26 = P$

Hill 密码加密举例

例 (Hill 密码加密)

- 明文为 paymoremoney, 加密密钥为

$$K = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$$

- 明文前三个字母用向量 $[15, 0, 24]^T$ 表示, 则

$$C = K \cdot \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} \bmod 26 = \begin{bmatrix} 375 \\ 819 \\ 486 \end{bmatrix} \bmod 26 = \begin{bmatrix} 11 \\ 13 \\ 18 \end{bmatrix} = \text{LNS}$$

- 照此转换剩下字母, 可得密文 LNSHDLEWMTRW。

Hill 密码解密举例

例 (Hill 密码解密)

- 解密需要用到矩阵 K 的逆 K^{-1}

- 由 $K = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$ 可以得到 $K^{-1} = \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix}$

- $KK^{-1} = I$ 可以验证如下

$$\begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \cdot \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix} = \begin{bmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{bmatrix} \bmod 26 = I$$

多表代换密码

- **多表代换密码**：使用多个代换表对明文消息进行多重单表代换加密。
- 因为需要猜测更多的字母表，并且频率分布特性也变得平坦了，所以使得密码破译更加困难。
- 多表代换的特点：
 - 采用相关的单表代换规则；
 - 密钥决定给定变换的具体规则。
- 最简单的多表代换密码为**维吉尼亚密码** (Vigenère Cipher)。

Vigenère 密码

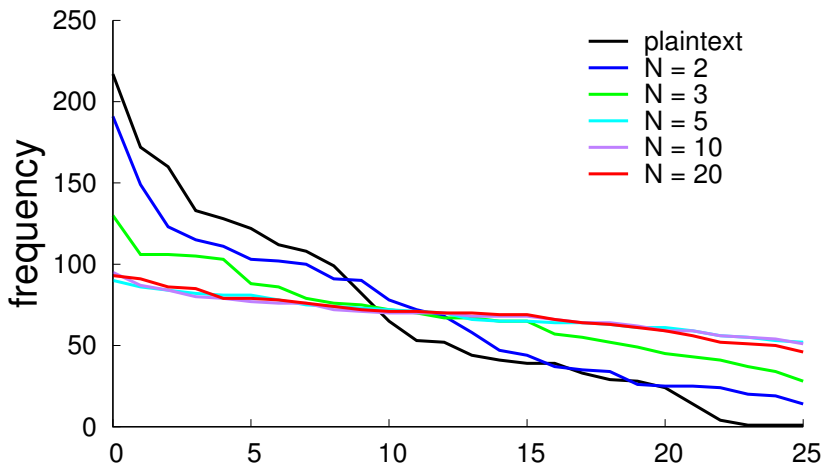
- Vigenère 密码的**代换规则集**由 26 个凯撒密码的代换表组成，每个代换表对明文字母移位 $0 \sim 25$ 次。
- 密钥词中的**密钥字母**用来代换明文字母 a ，故移位 3 次的凯撒密码由密钥字母 d 代表。
- 加密一条消息需要与消息一样长的密钥，通过**重复密钥**实现。
- **加密**：给定密钥字母 x 和明文字母 y ，密文字母是位于 x 行和 y 列的那个字母。
- **解密**：密钥字母决定行，行里密文字母所在列的顶部字母就是明文字母。

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

29 / 55

Vigenère 密码的安全性

- 每一个明文字母可以有多个密文字母对应，这样字母使用的频率特性减弱了，但是没有完全消失。

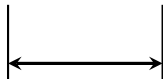


横轴为 26 个字母， N 为密钥词长度

Vigenère 密码的安全性

- 破译的关键是判定密钥词的长度，可以通过发现重复序列来判断。

key: deceptivedeceptivedeceptive
plaintext: weare**red**discover**red**saveyourself
ciphertext: ZIC**VTW**QNGRZG**VTW**AVZHCQYGLMGJ



相距整数个密钥词长度

💡 最终措施是选择与明文毫无统计关系且和它一样长的密钥。

一次一密 (One-Time Pad)

- 一次一密 (One-Time Pad, 简称 OTP):
 - 每个消息使用与之等长且随机的密钥来加密。
 - 一个密钥只对一个消息加解密，之后弃之不用。
- OTP 满足理论安全，是不可攻破的。
- 1882 年 Frank Miller 首次描述了 OTP，之后又被其他人再次发明。Gilbert Vernam 在 1919 年申请了基于异或运算的 OTP 的专利。
- 在 1900 年以前的密码学研究中，只有两个想法对现代密码学有用，其中之一为柯克霍夫准则，另外一个则为一次一密。

一次一密 (One-Time Pad)

- OTP 运算基于二进制数据而非字母。

- 加密:

$$c = m \oplus k$$

其中 \oplus 表示按位进行异或运算。

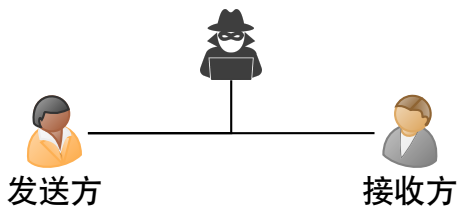
- 解密:

$$m = c \oplus k$$

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

一次一密的安全性

- 假设 Alice 和 Bob 使用 OTP 进行通信，从攻击者的视角观察到一个密文等同于得到下面算法的一个输出结果：



$\text{OTP}(m \in \{0, 1\}^\lambda):$

$k \leftarrow_R \{0, 1\}^\lambda$

$c = k \oplus m$

return c

- 一个好的加密算法不应由密文得到关于明文的**任何信息**。
- 由于密钥 k 是每次随机产生的，因此 OTP 输出的密文在攻击者看来是**随机**的。

一次一密的安全性

- 例如给定两个不同输入 $m_1 = 010$ 和 $m_2 = 111$, 输出分布为

$m_1 = 010$			$m_2 = 111$		
Pr	k	$output\ c = k \oplus 010$	Pr	k	$output\ c = k \oplus 111$
$1/8$	000	010	$1/8$	000	111
$1/8$	001	011	$1/8$	001	110
$1/8$	010	000	$1/8$	010	101
$1/8$	011	001	$1/8$	011	100
$1/8$	100	110	$1/8$	100	011
$1/8$	101	111	$1/8$	101	010
$1/8$	110	100	$1/8$	110	001
$1/8$	111	101	$1/8$	111	000

- OTP 得到 $\{0, 1\}^3$ 中每种可能输出的概率都为 $1/8$ 。
- 密码分析者不能由密文区分 m_1 和 m_2 , 因此**满足理论安全**。

一次一密的安全性

定理 (OTP 满足理论安全)

OTP 对于 $\forall m \in \{0, 1\}^\lambda$ 产生的输出都是 $\{0, 1\}^\lambda$ 上的均匀分布。

证明.

给定 $m, c \in \{0, 1\}^\lambda$, 考虑 $\text{OTP}(m)$ 产生输出 c 的概率。注意到

$$c = k \oplus m \Leftrightarrow k = m \oplus c$$

因此

$$P(\text{OTP}(m) = c) = P(k = m \oplus c)$$

- 也就是说, 只有当 $k = m \oplus c$ 时, $\text{OTP}(m)$ 的输出才为 c 。
- 由于 k 是 $\{0, 1\}^\lambda$ 上的均匀分布, 因此这个概率是 $1/2^\lambda$ 。
- 所以对于所有 m 和 c , $\text{OTP}(m)$ 的输出为 c 的概率都是 $1/2^\lambda$, 即服从均匀分布。

一次一密的局限性

- 产生大规模随机密钥有实际困难。
- 密钥的分配和保护无法保证。

目录

- 1 代换密码
- 2 置换密码
- 3 转轮密码机

置换密码 (Transposition Ciphers)

- 置换，亦称 transposition 或者 permutation。
- 置换密码通过改变明文字母的相对位置实现加密，并不替换明文字母，即明文内容形式不变。
- 通过重新安排明文字母的位置来隐藏明文内容信息，而不是用其他字母来代换明文字母。
- 这种方法是很容易破译的，因为密文拥有与明文一样的字母频率统计特性。

置换密码

- 一维变换：矩阵转置

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

明文：can you understand

密文：codtaueanurnynsd

- 二维变换：图形转置

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

明文：can you understand

密文：dnsuaruteodynnac

栅栏技术 (Rail Fence cipher)

按照对角线的顺序写出明文，按行的顺序读出作为密文。

例 (栅栏技术)

例如，加密 meet me after the toga party:

```
m e m a t r h t g p r y  
e t e f e t e o a a t
```

可以得到密文 MEMATRHTGPRYETEFETEOAAT

行置换密码 (Row Transposition Ciphers)

- 一个更复杂的方案是把消息一行一行地写成矩形块，然后按列读出，但是把列的次序打乱，列的次序就是算法的密钥。
- 可以采用多步置换来得到相对较高的安全性。

例

Key: 4 3 1 2 5 6 7

Plaintext: a t t a c k p

o s t p o n e


d u n t i l t

w o a m x y z

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

乘积密码 (Product Ciphers)

- 单纯的代换密码或者置换密码是不安全的，因为语言具有统计特性。
- 可以考虑连续使用若干个这样的密码使其难以破解，但是
 - 两次代换只能生成更复杂的代换，即**多次代换等价于一次代换**；
 - 两次置换只能生成更复杂的置换，即**多次置换等价于一次置换**。
- 如果在一次代换之后进行一次置换，可以生成一种新的更难破解的密码，这就是**乘积密码**。

 乘积密码是从古典密码通往现代密码的桥梁。

目录

- 1 代换密码
- 2 置换密码
- 3 转轮密码机**

转轮密码机 (Rotor Machines)

- 在现代密码系统出现之前，转轮密码机是最为广泛使用的多重加密器，尤其是在第二次世界大战中。
- 其中德国 Enigma 密码机是密码学界划时代的丰碑。
- 1918 年，德国发明家亚瑟·谢尔比乌斯 (Arthur Scherbius) 发明了一种能够自动编码的机器。谢尔比乌斯给自己所发明的电气编码机械取名 “Enigma”，意为 “哑谜”。
- Enigma 是一种用于加密与解密文件的密码机。确切地说，Enigma 是一系列相似的转子机械的统称，包括了一系列不同的型号。
- 不同型号的 Enigma 密码机大都包含四个共同部分：键盘、转子、显示器和接线板。

Enigma 密码机



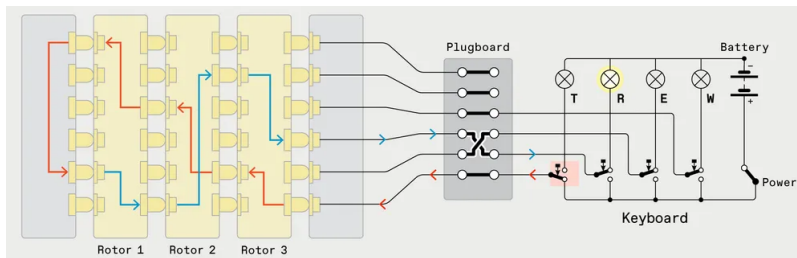
亚瑟 · 谢尔比乌斯



Enigma 密码机

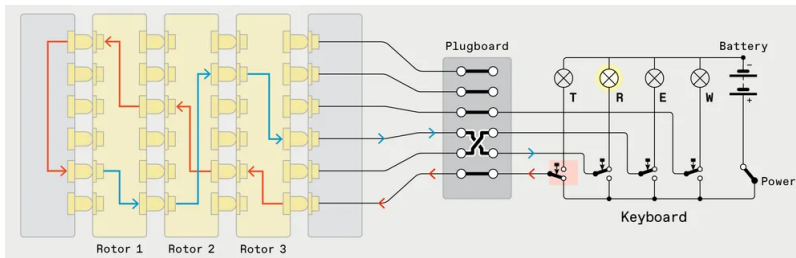
Enigma 密码机原理

- 转轮密码机实现了一个非常复杂、变化多端的多表代换密码。
- 每个转轮有 26 个输入和 26 个输出，每个输入仅与一个输出相连，一个转轮就定义了一个单表代换。
- 每按下一个键，转轮旋转一个位置，内部连线相应改变，改变下一次代替。
- 一个转轮转 26 个位置后回到初始状态，因此可以形成 26 种单表代换。



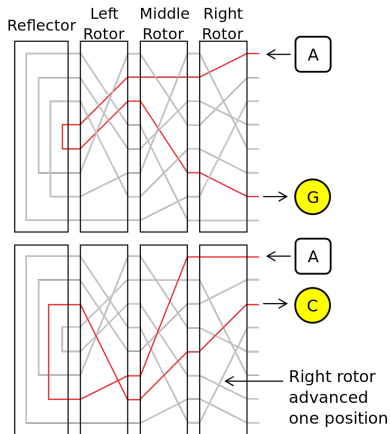
Enigma 密码机原理

- 为使代换更复杂，可以把多个转轮连接起来。Enigma 密码机有 3 个转轮，从提供的 5 个转轮中随机选择。
- 三个转轮以不同的速度移动，3 个转轮的机器的周期是 26^3 。
- 为进一步阻止密码分析，有些转轮机在每个转轮上还有不同的起始位置号。
- 尾部有一块连接板，可以连接 6 ~ 10 个插板，用来改变字母对之间的映射关系。



Enigma 密码机原理

- 最左边的转轮与反射板相接触，反射板使最左边转轮的不同端子随机相连，作用是使电路闭合。
- 由于反射板的存在，一个明文字母不会映射为它自己。



Enigma 密码机的使用

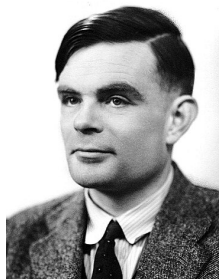
- 使用时，发信人首先调节三个转子的初始位置，转子的初始位置就是密钥，是收发双方预先约定好的。
- 然后键入明文，并把灯泡闪亮的字母依次记下来，最后把记录下的字母按顺序用电报发送出去。
- 收信方收到电文后，也使用一台 Enigma，按照原来的约定，把转子的位置调整到和发信方相同的初始位置上，然后依次键入收到的密文，灯泡闪亮的字母就是明文。
- 使用 Enigma 密码机解密和加密的过程完全一样，这就是反射板的作用，同时反射板的一个副作用就是一个字母永远也不会被加密成它自己，因为反射板中一个字母总是被连接到另一个不同的字母。

Enigma 密码机的设置

- Enigma 密码机每天都需要根据一份收发方共享的设置单进行设置，例如转轮选择与排列、转轮位置、连接板字母对等。

Kenngruppenheft Nr. 7												Teil A											
No.	Kenn- buch	Gruppe	No.	Kenn- buch	Gruppe	No.	Kenn- buch	Gruppe	No.	Kenn- buch	Gruppe	No.	Kenn- buch	Gruppe	No.	Kenn- buch	Gruppe	No.	Kenn- buch	Gruppe	No.	Kenn- buch	Gruppe
1	DD	ARCK	101	DD	WQID	201	DD	WQID	301	DD	WQID	401	DD	WQID	501	DD	WQID	601	DD	WQID	701	DD	WQID
2	DD	ARCK	102	DD	WQID	202	DD	WQID	302	DD	WQID	402	DD	WQID	502	DD	WQID	602	DD	WQID	702	DD	WQID
3	DD	ARCK	103	DD	WQID	203	DD	WQID	303	DD	WQID	403	DD	WQID	503	DD	WQID	603	DD	WQID	703	DD	WQID
4	DD	ARCK	104	DD	WQID	204	DD	WQID	304	DD	WQID	404	DD	WQID	504	DD	WQID	604	DD	WQID	704	DD	WQID
5	DD	ARCK	105	DD	WQID	205	DD	WQID	305	DD	WQID	405	DD	WQID	505	DD	WQID	605	DD	WQID	705	DD	WQID
6	DD	ARCK	106	DD	WQID	206	DD	WQID	306	DD	WQID	406	DD	WQID	506	DD	WQID	606	DD	WQID	706	DD	WQID
7	DD	ARCK	107	DD	WQID	207	DD	WQID	307	DD	WQID	407	DD	WQID	507	DD	WQID	607	DD	WQID	707	DD	WQID
8	DD	ARCK	108	DD	WQID	208	DD	WQID	308	DD	WQID	408	DD	WQID	508	DD	WQID	608	DD	WQID	708	DD	WQID
9	DD	ARCK	109	DD	WQID	209	DD	WQID	309	DD	WQID	409	DD	WQID	509	DD	WQID	609	DD	WQID	709	DD	WQID
10	DD	ARCK	110	DD	WQID	210	DD	WQID	310	DD	WQID	410	DD	WQID	510	DD	WQID	610	DD	WQID	710	DD	WQID
11	DD	ARCK	111	DD	WQID	211	DD	WQID	311	DD	WQID	411	DD	WQID	511	DD	WQID	611	DD	WQID	711	DD	WQID
12	DD	ARCK	112	DD	WQID	212	DD	WQID	312	DD	WQID	412	DD	WQID	512	DD	WQID	612	DD	WQID	712	DD	WQID
13	DD	ARCK	113	DD	WQID	213	DD	WQID	313	DD	WQID	413	DD	WQID	513	DD	WQID	613	DD	WQID	713	DD	WQID
14	DD	ARCK	114	DD	WQID	214	DD	WQID	314	DD	WQID	414	DD	WQID	514	DD	WQID	614	DD	WQID	714	DD	WQID
15	DD	ARCK	115	DD	WQID	215	DD	WQID	315	DD	WQID	415	DD	WQID	515	DD	WQID	615	DD	WQID	715	DD	WQID
16	DD	ARCK	116	DD	WQID	216	DD	WQID	316	DD	WQID	416	DD	WQID	516	DD	WQID	616	DD	WQID	716	DD	WQID
17	DD	ARCK	117	DD	WQID	217	DD	WQID	317	DD	WQID	417	DD	WQID	517	DD	WQID	617	DD	WQID	717	DD	WQID
18	DD	ARCK	118	DD	WQID	218	DD	WQID	318	DD	WQID	418	DD	WQID	518	DD	WQID	618	DD	WQID	718	DD	WQID
19	DD	ARCK	119	DD	WQID	219	DD	WQID	319	DD	WQID	419	DD	WQID	519	DD	WQID	619	DD	WQID	719	DD	WQID
20	DD	ARCK	120	DD	WQID	220	DD	WQID	320	DD	WQID	420	DD	WQID	520	DD	WQID	620	DD	WQID	720	DD	WQID
21	DD	ARCK	121	DD	WQID	221	DD	WQID	321	DD	WQID	421	DD	WQID	521	DD	WQID	621	DD	WQID	721	DD	WQID
22	DD	ARCK	122	DD	WQID	222	DD	WQID	322	DD	WQID	422	DD	WQID	522	DD	WQID	622	DD	WQID	722	DD	WQID
23	DD	ARCK	123	DD	WQID	223	DD	WQID	323	DD	WQID	423	DD	WQID	523	DD	WQID	623	DD	WQID	723	DD	WQID
24	DD	ARCK	124	DD	WQID	224	DD	WQID	324	DD	WQID	424	DD	WQID	524	DD	WQID	624	DD	WQID	724	DD	WQID
25	DD	ARCK	125	DD	WQID	225	DD	WQID	325	DD	WQID	425	DD	WQID	525	DD	WQID	625	DD	WQID	725	DD	WQID
26	DD	ARCK	126	DD	WQID	226	DD	WQID	326	DD	WQID	426	DD	WQID	526	DD	WQID	626	DD	WQID	726	DD	WQID
27	DD	ARCK	127	DD	WQID	227	DD	WQID	327	DD	WQID	427	DD	WQID	527	DD	WQID	627	DD	WQID	727	DD	WQID
28	DD	ARCK	128	DD	WQID	228	DD	WQID	328	DD	WQID	428	DD	WQID	528	DD	WQID	628	DD	WQID	728	DD	WQID
29	DD	ARCK	129	DD	WQID	229	DD	WQID	329	DD	WQID	429	DD	WQID	529	DD	WQID	629	DD	WQID	729	DD	WQID
30	DD	ARCK	130	DD	WQID	230	DD	WQID	330	DD	WQID	430	DD	WQID	530	DD	WQID	630	DD	WQID	730	DD	WQID
31	DD	ARCK	131	DD	WQID	231	DD	WQID	331	DD	WQID	431	DD	WQID	531	DD	WQID	631	DD	WQID	731	DD	WQID
32	DD	ARCK	132	DD	WQID	232	DD	WQID	332	DD	WQID	432	DD	WQID	532	DD	WQID	632	DD	WQID	732	DD	WQID
33	DD	ARCK	133	DD	WQID	233	DD	WQID	333	DD	WQID	433	DD	WQID	533	DD	WQID	633	DD	WQID	733	DD	WQID
34	DD	ARCK	134	DD	WQID	234	DD	WQID	334	DD	WQID	434	DD	WQID	534	DD	WQID	634	DD	WQID	734	DD	WQID
35	DD	ARCK	135	DD	WQID	235	DD	WQID	335	DD	WQID	435	DD	WQID	535	DD	WQID	635	DD	WQID	735	DD	WQID
36	DD	ARCK	136	DD	WQID	236	DD	WQID	336	DD	WQID	436	DD	WQID	536	DD	WQID	636	DD	WQID	736	DD	WQID
37	DD	ARCK	137	DD	WQID	237	DD	WQID	337	DD	WQID	437	DD	WQID	537	DD	WQID	637	DD	WQID	737	DD	WQID
38	DD	ARCK	138	DD	WQID	238	DD	WQID	338	DD	WQID	438	DD	WQID	538	DD	WQID	638	DD	WQID	738	DD	WQID
39	DD	ARCK	139	DD	WQID	239	DD	WQID	339	DD	WQID	439	DD	WQID	539	DD	WQID	639	DD	WQID	739	DD	WQID
40	DD	ARCK	140	DD	WQID	240	DD	WQID	340	DD	WQID	440	DD	WQID	540	DD	WQID	640	DD	WQID	740	DD	WQID
41	DD	ARCK	141	DD	WQID	241	DD	WQID	341	DD	WQID	441	DD	WQID	541	DD	WQID	641	DD	WQID	741	DD	WQID
42	DD	ARCK	142	DD	WQID	242	DD	WQID	342	DD	WQID	442	DD	WQID	542	DD	WQID	642	DD	WQID	742	DD	WQID
43	DD	ARCK	143	DD	WQID	243	DD	WQID	343	DD	WQID	443	DD	WQID	543	DD	WQID	643	DD	WQID	743	DD	WQID
44	DD	ARCK	144	DD	WQID	244	DD	WQID	344	DD	WQID	444	DD	WQID	544	DD	WQID	644	DD	WQID	744	DD	WQID
45	DD	ARCK	145	DD	WQID	245	DD	WQID	345	DD	WQID	445	DD	WQID	545	DD	WQID	645	DD	WQID	745	DD	WQID
46	DD	ARCK	146	DD	WQID	246	DD	WQID	346	DD	WQID	446	DD	WQID	546	DD	WQID	646	DD	WQID	746	DD	WQID
47	DD	ARCK	147	DD	WQID	247	DD	WQID	347	DD	WQID	447	DD	WQID	547	DD	WQID	647	DD	WQID	747	DD	WQID
48	DD	ARCK	148	DD	WQID	248	DD	WQID	348	DD	WQID	448	DD	WQID	548	DD	WQID	648	DD	WQID	748	DD	WQID
49	DD	ARCK	149	DD	WQID	249	DD	WQID	349	DD	WQID	449	DD	WQID	549	DD	WQID	649	DD	WQID	749	DD	WQID
50	DD	ARCK	150	DD	WQID	250	DD	WQID	350	DD	WQID	450	DD	WQID	550	DD	WQID	650	DD	WQID	750	DD	WQID
51	DD	ARCK	151	DD	WQID	251	DD	WQID	351	DD	WQID	451	DD	WQID	551	DD	WQID	651	DD	WQID	751	DD	WQID
52	DD	ARCK	152	DD	WQID	252	DD	WQID	352	DD	WQID	452	DD	WQID	552	DD	WQID	652	DD	WQID	752	DD	WQID
53	DD	ARCK	153	DD	WQID	253	DD	WQID	353	DD	WQID	453	DD	WQID	553	DD	WQID	653	DD	WQID	753	DD	WQID
54	DD	ARCK	154	DD	WQID	254	DD	WQID	354	DD	WQID	454	DD	WQID	554	DD	WQID	654	DD	WQID	754	DD	WQID
55	DD	ARCK	155	DD	WQID	255	DD	WQID	355	DD	WQID	455	DD	WQID	555	DD	WQID	655	DD	WQID	755	DD	WQID
56	DD	ARCK	156	DD	WQID	256	DD	WQID	356	DD	WQID	456	DD	WQID	556	DD	WQID	656	DD	WQID	756	DD	WQID
57	DD	ARCK	157	DD	WQID	257	DD	WQID	357	DD	WQID	457	DD	WQID	557	DD	WQID	657	DD	WQID	757	DD	WQID
58	DD	ARCK	158	DD	WQID	258	DD	WQID	358	DD	WQID	458	DD	WQID	558	DD	WQID	658	DD	WQID	758	DD	WQID
59	DD	ARCK	159	DD	WQID	259	DD	WQID	359	DD	WQID	459	DD	WQID	559	DD	WQID	659	DD	WQID	759	DD	WQID
60	DD	ARCK	160	DD	WQID	260	DD	WQID	360	DD	WQID	460	DD	WQID	560	DD	WQID	660	DD	WQID	760	DD	WQID
61	DD	ARCK	161	DD	WQID	261	DD	WQID	361	DD	WQID	461	DD	WQID	561	DD	WQID	661	DD	WQID	761	DD	WQID
62	DD	ARCK	162	DD	WQID	262	DD	WQID	362	DD	WQID	462	DD	WQID	562	DD	WQID	662	DD	WQID	762	DD	WQID
63	DD	ARCK	163	DD	WQID	263	DD	WQID	363	DD	WQID	463	DD	WQID	563	DD	WQID	663	DD	WQID	763	DD	WQID
64	DD	ARCK	164	DD	WQID	264	DD	WQID	364	DD	WQID	464	DD	WQID	564	DD	WQID	664	DD	WQID	764	DD	WQID
65	DD	ARCK	165	DD	WQID	265	DD	WQID	365	DD	WQID	465	DD	WQID	565	DD	WQID	665	DD	WQID	765	DD	WQID
66	DD	ARCK	166	DD	WQID	266	DD	WQID	366	DD	WQID	466	DD	WQID	566	DD	WQID	666	DD	WQID	766	DD	WQID
67	DD	ARCK	167	DD	WQID	267	DD	WQID	367	DD	WQID	467	DD	WQID	567	DD	WQID	6					

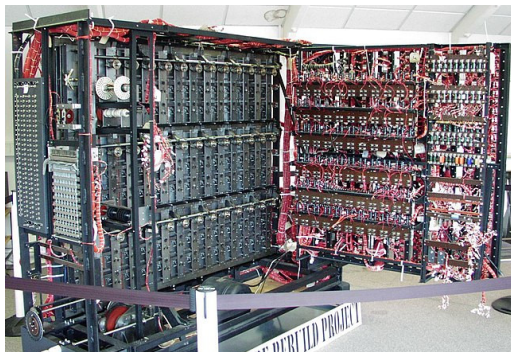
Enigma 密码机的破译



艾伦·图灵



布莱切利
公园



“炸弹” 破
译机

小结

- 1 代换密码
- 2 置换密码
- 3 转轮密码机