

# 上讲主要内容回顾

- 现代密码学与信息安全的关系;
- 现代密码学的主要内容;
- 本课程将讲授的内容;
- 密码学的发展史;

## 第2章 传统密码体制

# 本讲主要内容

- 置换密码（列置换密码和周期置换密码）
- 代换密码（单表代换密码、多表代换密码和维尔姆密码）
- 典型传统密码的分析（统计分析法和明文-密文对分析法）

# 简介

"Communication Theory in Secrecy System"

- 在1949年**Claude Shannon**发表“保密系统的通信理论”之前，密码学算法主要通过**字符**间的**置换**和**代换**实现，一般认为这些密码体制属于传统密码范畴。
- 传统密码体制是指那些比较简单的、大多数采用手工或机械操作对明文进行加密、对密文进行解密的密码体制(**对称**)，其安全性绝大多数与**加解密算法保密性**密切相关。
- 传统密码体制的技术、思想以及破译方法虽然很简单，但是反映了**密码设计和破译的思想**，是学习密码学的**基本入口**，对于理解、设计和分析现代密码仍然具有借鉴的价值。

# 置换密码

- 置换密码(**Permutation Cipher**)又叫换位密码(**Transposition Cipher**), 它根据一定的规则重新排列明文, 以便打破明文的结构性。置换密码的特点是保持明文的所有字符不变, 只是利用置换打乱了明文字符的位置和次序。
- 最常见的置换密码有二种:
  - 列置换密码(明文遵照密钥的规程按列换位并且按列读出序列得到密文);
  - 周期置换密码(将明文 $P$ 按固定长度 $m$ 分组, 然后对每组按  $1, 2, \dots, m$  的某个置换重排位置从而得到密文 $C$ );

## 列置换密码(加密)

- 将明文**P**以设定的固定分组宽度**m**按行写出，即每行有**m**个字符。若明文长度不是**m**的整数倍，则不足部分用双方约定的方式填充，如双方约定用空格代替空缺处字符，不妨设最后得字符矩阵  $[M]_{m \times n}$  ；
- 按  $1, 2, \dots, m$  的某一**置换** $\sigma$  交换列的位置次序得字符矩阵  $[M_P]_{m \times n}$  ；
- 把矩阵  $[M_P]_{m \times n}$  按列  $1, 2, \dots, n$  的顺序依次读出得密文序列**C**；

# 列置换密码(解密)

- 将密文**C**按与加密过程相同的分组宽度**m**按列写得到字符矩阵  $[M_P]_{m \times n}$  。
- 按加密过程用的置换  $\sigma$  的逆置换  $\sigma^{-1}$  交换列的位置次序得字符矩阵  $[M]_{m \times n}$  。
- 把矩阵  $[M]_{m \times n}$  按  $1, 2, \dots, m$  行的顺序依次读出得明文**P**。

## 列置换密码加密(举例)

- 设明文P为 “**Beijing 2008 Olympic Games**”,

密钥  $\sigma = (1\ 4\ 3)(5\ 6)$  , 则加密过程为:

$$[M]_{4 \times 6} = \begin{bmatrix} B & e & i & J & i & n \\ g & 2 & 0 & 0 & 8 & O \\ l & y & m & p & i & c \\ G & a & m & e & s & \end{bmatrix} \xRightarrow{\sigma} [M_P]_{4 \times 6} = \begin{bmatrix} i & e & J & B & n & i \\ 0 & 2 & 0 & g & O & 8 \\ m & y & p & l & c & i \\ m & a & e & G & & s \end{bmatrix}$$



## 列置换密码解密(举例)

$$\delta = (143)(2)(56) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 2 & 1 & 3 & 6 & 5 \end{pmatrix}$$

$$\delta^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 4 & 1 & 6 & 5 \end{pmatrix} = (134)(2)(56)$$

$$\text{显然 } \delta \bullet \delta^{-1} = \varepsilon = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

- 由矩阵  $[M_P]_{4 \times 6}$  得到密文 **C** 为 “**i0mme2yaJ0peBglGnOc i8is**”。

根据加密密钥逆置换  $\sigma^{-1} = (134)(56)$ , 则解密过程如下:

$$[M_P]_{4 \times 6} = \begin{bmatrix} i & e & J & B & n & i \\ 0 & 2 & 0 & g & O & 8 \\ m & y & p & l & c & i \\ m & a & e & G & & s \end{bmatrix} \xRightarrow{\sigma^{-1}} [M]_{4 \times 6} = \begin{bmatrix} B & e & i & J & i & n \\ g & 2 & 0 & 0 & 8 & O \\ l & y & m & p & i & c \\ G & a & m & e & & s \end{bmatrix}$$

# 周期置换密码

周期置换密码是将明文串**P**按固定长度**m**分组，然后对每组中的子串按  $1, 2, \dots, m$  的某个置换重排位置从而得到密文**C**。其中密钥  $\sigma$  包含分组长度信息。解密时同样对密文**C**按长度**m**分组，并按  $\sigma$  的逆置换  $\sigma^{-1}$  把每组子串重新排列位置从而得到明文**P**。

## 周期置换密码(举例)

- 明文:

**“State Key Laboratory of Networking and Switching”;**

加密密钥:  $\sigma = (1\ 5\ 6\ 2\ 3)$

- 明文分为七组:

**(StateK)(eyLabo)(ratory)(ofNetw)(orking)(andSwi)(tching)**

- 加密变换:

**(aKttSe)(Loyaeb)(tyaorr)(Nwfeot)(kgrion)(dinSaw)(hgcitn)**

- 最终密文:

**(aKttSeLoyaebtyaorrNwfeotkgriondinSawhgcitn)**

- 由加密密钥易知解密密钥:  $\sigma^{-1} = (1\ 3\ 2\ 6\ 5)$  , 解密易实现。

# 代换密码 Substitution

- **代换**是传统密码中用到的最基本的处理技巧，在现代密码学中中得到广泛使用。
- 所谓代换，就是将明文中的一个字母由其它字母、数字或符号替代的一种方法。
- 代换密码是指建立一个代换表，加密时将需要加密的明文依次通过查表，替换为相应的字符，明文字符被逐个替换后，生成无任何意义的字符串，即密文。这样的**代换表**，称为**密钥**。

# 代换密码的分类

按照一个明文字母是否总是被一个固定的字符代换进行划分：

凯撒密码caesar

- **单表代换密码**(移位、仿射、替换)

对明文消息中出现的同一个字母，在加密时都使用同一固定的字母来代换，不管它出现在什么地方。

- **多表代换密码**(维吉利亚、Playfair、转轮)

明文消息中出现的同一个字母，在加密时不是完全被同一固定的字母代换，而是根据其出现的位置次序，用不同的字母代换。

# 仿射加密

假定的字符集为 $\text{charset}=\{'a','b',\dots,'z'\}$   
编码为 $\text{coding}[]=\{0,1,\dots,25\}$

- 明文 $P$ =密文 $C=\mathbb{Z}_{26}$ ;
- 加密  $E_k(m)=am+b \bmod n=c$
- 解密  $D_k(c)=a^{-1}(c-b) \bmod n=m$
- 密钥 $K = \{(a,b) : a,b \in \mathbb{Z}_{26}, \text{且} \gcd(a, 26)=1\}$  避免不同的明文字母加密成同样的密文字母
- 举例

假定 $k=(7,3)$ ,  $7^{-1} \bmod 26=15$ , 加密函数为 $E_k(m)=7m+3$ , 则相应的解密函数为 $D_k(c)=15(c-3)=15y-19$ , 其中所有的运算都是在 $\mathbb{Z}_{26}$ 中。容易验证 $D_k(E_k(m))=D_k(7m+3)=15(7m+3)-19=105m+45-19=104m+m=m$ 。

加密明文hot。首先转化这三个字母分别为数字7, 14和19。然后加密得密文串为AGX。 1,3,5,7,9,11,15,17,19,21,23,25

- 因为满足 $a \in \mathbb{Z}_{26}$ , 且 $\gcd(a, 26)=1$ 的只有12整数, 对参数b没有要求。所以仿射密码有 $12 \times 26=312$ 种可能的密钥。

# 移位密码（恺撒密码）

- 明文 26字母
- 密文 26字母
- 密钥空间 $K=\{0, 1, 2, \dots, 25\}$
- 在实际进行加解密运算时，把26个英文字母依次与0, 1, 2, ..., 25对应。
- 加密  $E_k(m)=m+k=c \bmod n$
- 解密  $D_k(c)=c-k=m \bmod n$
- 举例

移位密码的一个典型代表就是恺撒密码， $k=3$ ， $n=26$ 。

明文 meet me after class

密文 PHHW PH DIWHU FODVV

- 由于结构过于简单，密钥空间太小（**26**），很容易被**穷举攻击**方法分析。

# 单表代换密码

加密函数:

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

解密函数:

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

明文: if we wish to replace letters

密文: WI RF RWAJ UH YFTSDVF SFUUFYA

key space

密钥空间:  $26! > 10^{25}$  ( $10^6$ 次/秒100台并行约 $10^9$ 年, 接近宇宙年龄 $10^{10}$ 年)



## 多表代换密码（维吉尼亚）

- 多表密码是利用多个单表代替密码构成的密码体制。它在对明文进行加密的过程中依照密钥的指示轮流使用多个单表代替密码。
- $M=(m_1, m_2, \dots, m_n)$   $K=(k_1, k_2, \dots, k_d)$   $C=(c_1, c_2, \dots, c_n)$
- 加密变换:  $c_{i+td} = E_{k_i}(m_{i+td}) = m_{i+td} + k_i \bmod n$   $i=1\dots d, t=0\dots\text{ceiling}[n/d]$
- 解密变换:  $m_{i+td} = D_{k_i}(c_{i+td}) = c_{i+td} - k_i \bmod n$

w	e	a	r	e	d	i	s	c	o	v	e	r	e	d	s	a	v	e	y	o	u	r	s	e	l	f
d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e
Z	I	C	V	T	W	Q	N	G	R	Z	G	V	T	W	A	V	Z	H	C	Q	Y	G	L	M	G	J

- 密钥空间为  $26^d$ 。

# 多表代换密码(Playfair)

将明文字母按两个字母一组分成若干个单元，然后将这些单元替换为密文字母组合，替换时基于一个 $5 \times 5$ 字母矩阵，该矩阵使用一个密钥来构造，其构造方法如下：从左到右，从上到下依次填入关键词的字母，若关键词中有重复字母，则第二次出现时略过。然后，在母表中剩下的字母按字母顺序依次填入矩阵中，其中字母i和j看作是同一个字符。同时约定如下规则：表中的第一列看作是第五列的右边一列，第一行看作是第五行的下一行。

# Playfair的加解密方法

- 若在同一行，则对应的密文分别是紧靠右端的字母。
- 若在同一列，则对应的密文分别是紧靠下端的字母。
- 若不在同一行，也不在同一列，则对应的密文以对角顶点确定的矩形的另外的两个顶点字母，按同行的原则对应。
- 若相同，则插入一个事先约定好的字母，并用上述方法处理。
- 若明文字母数为奇数，则明文的末端添加一个事先约定好的字母进行填充。

注：解密过程与加密过程基本相似，只是把其中的右边改为左边，把其中的下面改为上面即可。

# Playfair的举例

- 设密钥为“**PLAYFAIR IS A DIGRAM CIPHER**”;

- 字母矩阵:
- |              |          |          |          |          |
|--------------|----------|----------|----------|----------|
| <i>P</i>     | <i>L</i> | <i>A</i> | <i>Y</i> | <i>F</i> |
| <i>I / J</i> | <i>R</i> | <i>S</i> | <i>D</i> | <i>G</i> |
| <i>M</i>     | <i>C</i> | <i>H</i> | <i>E</i> | <i>B</i> |
| <i>K</i>     | <i>N</i> | <i>O</i> | <i>Q</i> | <i>T</i> |
| <i>U</i>     | <i>V</i> | <i>W</i> | <i>X</i> | <i>Z</i> |

keyspace=25!

- 明文: **pl ay fa ir ci ph er;**
- 密文: “**LA YF PY RS MR AM CD;**

# 维尔姆密码

美国电话电报公司的**G.Vernam**在1917年为电报通信设计了一种非常方便的密码，后来称之为**Vernam**密码。它将英文字母编成**5比特**二元数字，称之为五单元波多电码。选择随机二元数字序列作为密钥，以 $k = k_1, k_2, k_3 \dots k_i \dots$ ， $k_i \in [0, 1]$ 表示。明文字母变换成二元码后也可表示成二元数字序列 $m = m_1 m_2 m_3 \dots m_i \dots$ ， $m_i \in [0, 1]$ 。加密运算就是将 $k$ 和 $m$ 的相应位逐位相加，即

$$c_i = m_i \oplus k_i \bmod 2, \quad i = 1, 2, 3 \dots$$

译码时，可用同样的密钥纸带对密文数字同步地逐位模2相加，便可恢复出明文的二元码序列，即 $m_i = c_i \oplus k_i \bmod 2$ ， $i = 1, 2, 3 \dots$ 。

如果密钥序列 $k_i$ 能够被独立地随机产生，则**Vernam**密码被称为一次一密(**one-time pad**)，这种密码对于**唯密文攻击**是无条件安全的。如果存在不同的明文使用相同的密钥，则**Vernam**密码是容易破解的。

# 希尔密码(Hill Cipher)

设  $n$  为某一固定的正整数,  $\mathbf{P}$ 、 $\mathbf{C}$  和  $\mathbf{K}$  分别为明文空间、密文空间和密钥空间, 并且  $\mathbf{P}=\mathbf{C}=(\mathbb{Z}_{26})^n$ , 密钥  $k=(k_{ij})_{n \times n}$  是一个  $n \times n$  的非奇异矩阵(行列式  $\det(k) \neq 0$ ), 且满足  $(\det(k), 26)=1$ , 即满足  $\mathbb{Z}_{26}$  上  $\det(k)$  和  $26$  互素, 从而保证了密钥矩阵的逆矩阵存在。对明文序列  $p=(p_1, p_2, \dots, p_n) \in \mathbf{P}$ , 其对应密文记为  $c=(c_1, c_2, \dots, c_n) \in \mathbf{C}$ , 则 Hill 密码的加密函数定义为:

$$(c_1, c_2, \dots, c_n) \equiv (p_1, p_2, \dots, p_n) \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{bmatrix} \pmod{26}$$

写成矩阵简化形式为:  $[c]_{1 \times n} = ([p]_{1 \times n} \times [k]_{n \times n}) \pmod{26}$ 。

行列式

因为方阵  $k=(k_{ij})_{n \times n}$  是  $\mathbb{Z}_{26}$  上的非奇异矩阵, 即满足  $\mathbb{Z}_{26}$  上  $\det(k)$  和  $26$  互素, 所以密钥  $k$  的逆矩阵  $k^{-1}$  必然存在。在 Hill 密码的加密函数等式的两端分别乘以  $k^{-1}$ , 则得到其解密函数的解析式:

$$(p_1, p_2, \dots, p_n) \equiv (c_1, c_2, \dots, c_n) \times \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{bmatrix}^{-1} \pmod{26}$$

写成矩阵简化形式为:  $[p]_{1 \times n} \equiv ([c]_{1 \times n} \times [k]_{n \times n}^{-1}) \pmod{26}$ 。

# 希尔密码的加密方法

- 明文是“**cyber**”，数字化后为 2, 24, 1, 4, 17；

- 密钥：
$$k = \begin{bmatrix} 10 & 5 & 12 & 0 & 0 \\ 3 & 14 & 21 & 0 & 0 \\ 8 & 9 & 11 & 0 & 0 \\ 0 & 0 & 0 & 11 & 8 \\ 0 & 0 & 0 & 3 & 7 \end{bmatrix}$$

- 加密：

$$c = (2 \ 24 \ 1 \ 4 \ 17) \begin{bmatrix} 10 & 5 & 12 & 0 & 0 \\ 3 & 14 & 21 & 0 & 0 \\ 8 & 9 & 11 & 0 & 0 \\ 0 & 0 & 0 & 11 & 8 \\ 0 & 0 & 0 & 3 & 7 \end{bmatrix} = \begin{bmatrix} 100 \\ 355 \\ 539 \\ 95 \\ 151 \end{bmatrix}^T \pmod{26} = \begin{bmatrix} 22 \\ 17 \\ 19 \\ 17 \\ 21 \end{bmatrix}^T \Leftrightarrow \begin{bmatrix} W \\ R \\ T \\ R \\ V \end{bmatrix}^T$$

# 希尔密码的解密密方法

● 解密密钥:

$$k^{-1} = \begin{bmatrix} 21 & 15 & 17 & 0 & 0 \\ 23 & 2 & 16 & 0 & 0 \\ 25 & 4 & 3 & 0 & 0 \\ 0 & 0 & 0 & 7 & 18 \\ 0 & 0 & 0 & 23 & 11 \end{bmatrix}$$

● 解密:

$$p \equiv (22 \ 17 \ 19 \ 17 \ 21) \begin{bmatrix} 21 & 15 & 17 & 0 & 0 \\ 23 & 2 & 16 & 0 & 0 \\ 25 & 4 & 3 & 0 & 0 \\ 0 & 0 & 0 & 7 & 18 \\ 0 & 0 & 0 & 23 & 11 \end{bmatrix} = \begin{bmatrix} 1328 \\ 440 \\ 703 \\ 602 \\ 537 \end{bmatrix}^T \pmod{26} = \begin{bmatrix} 2 \\ 24 \\ 1 \\ 4 \\ 17 \end{bmatrix}^T \Leftrightarrow \begin{bmatrix} c \\ y \\ b \\ e \\ r \end{bmatrix}^T$$



# 转轮密码机

<http://users.telenet.be/d.rijmenants/en/enigmasim.htm>

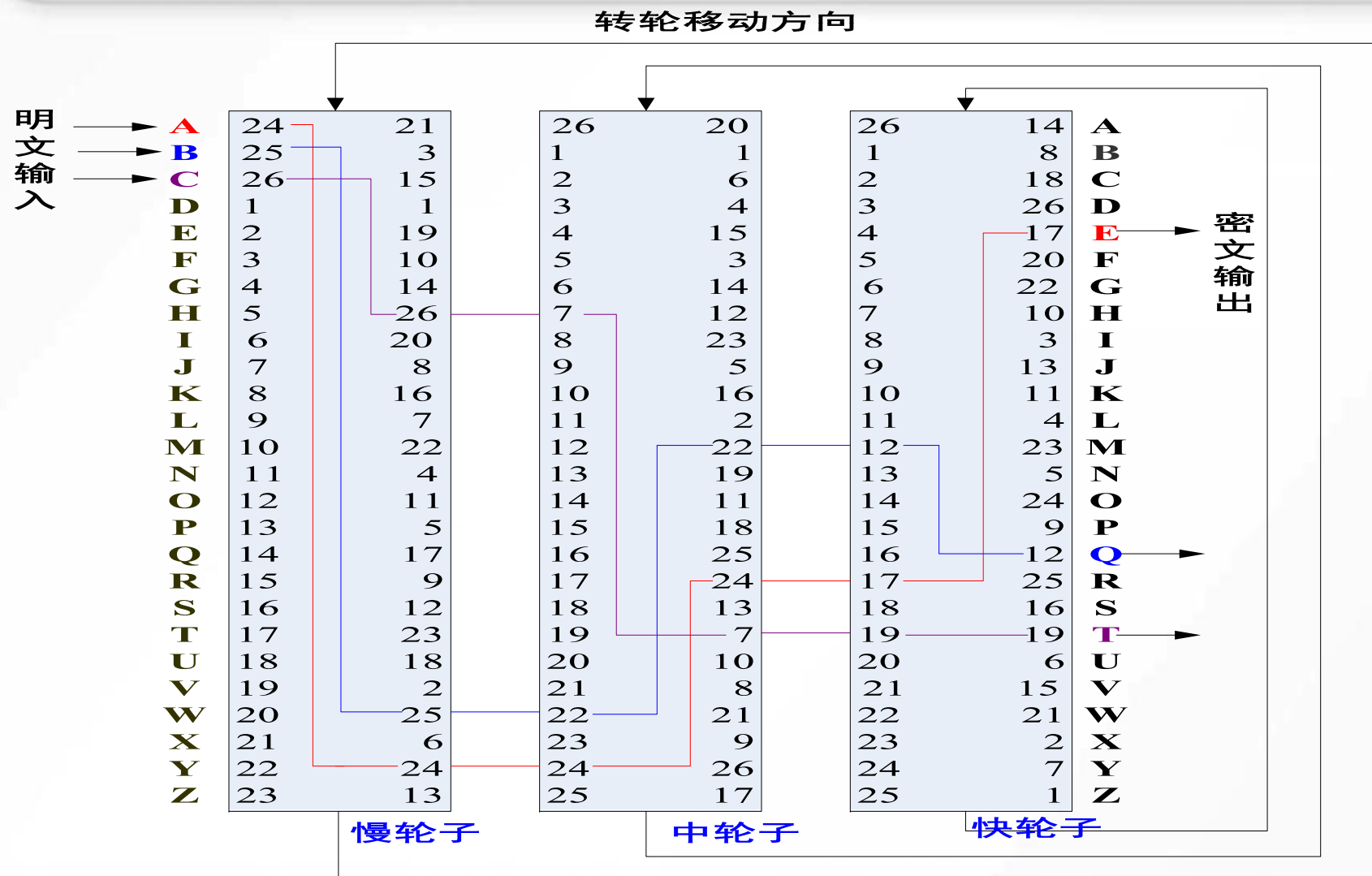
<http://www.enigmaworldcodegroup.com/>

从**19**世纪**20**年代，人们开始发明各种机械加密设备用来处理数据的加解密。起初普遍使用的是转轮机和转轮加密算法。

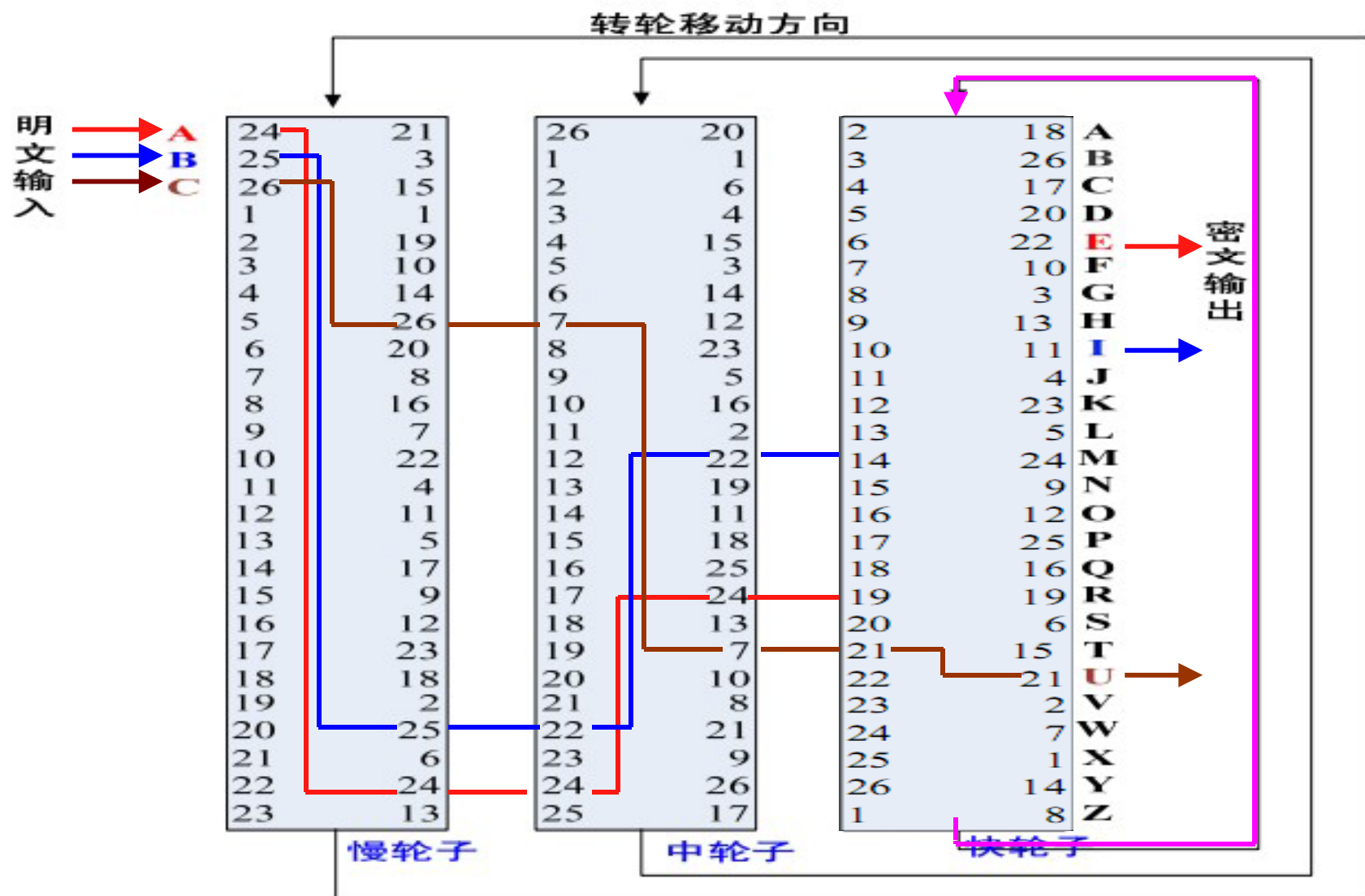
转轮密码机是由一个用于输入的键盘和一组转轮组成，每个轮转上有**26**个字母的任意组合。转轮之间由齿轮进行连接，当一个轮转动时，可以将一个字母转化成为另一个字母。



# 转轮加密算法



# 转轮加密算法



# 转轮密码机的分析

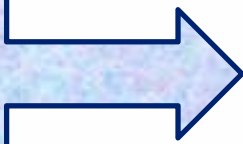
事实上，转轮密码机中的每个转轮都有可能在转动，其规律是：当快转子转动**26**次以后，中转子就转动一个位置；而当中转子转动**26**次以后，慢转子就转动一个位置。因此，在加解（或解密）**26x26x26**个字母以后，所有转轮都恢复到初始状态。也就是说，一个有**3**个转轮的转轮密码机是一个周期长度为**26x26x26(17576)**的多表代换密码。

转轮密码机的使用大大提高了密码加解密速度，在二战期间有广泛的应用。

# 转轮密码机的思考

转轮密码机是近代密码发展史中里程碑的事件。

实用的密码  
设备应具备



- 安全
- 易实现
- 性能
- 使用方便

# 传统密码体制分析

- 单表代换密码分析
- 多表代换密码分析
- **Hill**密码分析

# 明文中各个字母出现的统计概率

Letter	Frequency	Letter	Frequency	Letter	Frequency
e	12.31%	l	4.03%	b	1.62%
t	9.59%	d	3.65%	g	1.61%
a	8.05%	c	3.20%	v	0.93%
o	7.94%	u	3.10%	k	0.52%
n	7.19%	p	2.29%	q	0.20%
i	7.18%	f	2.28%	x	0.20%
s	6.59%	m	2.25%	j	0.10%
r	6.03%	w	2.03%	z	0.09%
h	5.14%	y	1.88%		

# 字符出现频率分类

- **E**，有概率大约**0.120**。
- **T, A, O, I, N, S, H, R**，每个有概率在**0.06~0.09**间。
- **D, L**，每个有概率大约**0.04**。
- **C, U, M, W, F, G, Y, P, B**，每个有概率在**0.015~0.023**之间。
- **V, K, J, X, Q, Z**，每个概率少于**0.01**。
- 当考虑位置特性时，字母**A, I**和**H**一般不作为单词的结尾，而**E, N**和**R**出现在起始位置比出现在结束位置的概率更小，字母**T, O**和**S**出现在单词前后位置的概率基本相同。
- 应该强调的是，这些表并不包含结论性的信息。字母的分布大大依赖于明文文本的类型：诗歌，标语，科技等等，所以有些出入也是正常的。



## 字母组合概率(递减)

- **双字母组合:** TH, HE, IN, ER, AN, RE, DE, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG和AS。
- **三字母组合:** THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS和ETH。

Word Frequency		Word Frequency		Word Frequency	
the	6.421%	a	2.092%	i	0.945%
of	4.028%	in	1.778%	it	0.930%
and	3.150%	that	1.244%	for	0.770%
to	2.367%	is	1.034%	as	0.764%

# 十个最常用中文单字频率

政 治	文 艺	新 闻	科 技	综 合
字 频率	字 频率	字 频率	字 频率	字 频率
的 0.0536	的 0.0324	的 0.0375	的 0.0320	的 0.0384
是 0.0165	一 0.0218	一 0.0132	一 0.0097	一 0.0125
一 0.0136	了 0.0196	了 0.0120	在 0.0092	是 0.0098
在 0.0115	不 0.0165	和 0.0086	用 0.0079	在 0.0095
这 0.0109	是 0.0141	在 0.0086	有 0.0073	了 0.0082
主 0.0108	说 0.0130	人 0.0083	是 0.0070	不 0.0081
不 0.0101	他 0.0130	大 0.0083	不 0.0069	和 0.0075
和 0.0098	这 0.0119	主 0.0083	中 0.0066	有 0.0069
人 0.0087	着 0.0107	是 0.0078	大 0.0064	的 0.0069
们 0.0087	个 0.0097	们 0.0065	时 0.0063	这 0.0064

# 多表代换密码分析

- 决定密钥字的长度

卡斯基(Kasiski)测试法

重合指数法

1918年William F.Friedman  
的专题论文《重合指数及其  
在密码学中的应用》是1949  
年之前最有影响的密码文献。

- 确定密钥

# 卡斯基(Kasiski)测试法

基本原理：若用给定的 $k$ 个密钥表周期地对明文字母加密，则当明文中有两个相同字母组在明文序列中间隔的字母数为 $k$ 的倍数时，**这两个明文字母组对应的密文字母组必相同**。但反过来，若密文中出现两个相同的字母组，它们所对应的明文字母组未必相同，但相同的可能性很大。如果我们将密文中相同的字母组找出来，并对其相同字母数综合研究，找出它们的相同字母数的最大公因子，就有可能提取出有关密钥字的长度 $k$ 的信息。

## 举例(密文)

**CHR**EEVOAHMAERATBIAXXWTNXBEEOPHBSBQMQ  
EQERBWRVXUOAKXAOSXXWEAHBWGJMMQMKNKGR  
FVGXWTRZXWIAKLXFPSKAUTEMNDCMGTSXMXBTUI  
ADNGMGPSRELXNJELXVRVPRTULHDNQWTWDTYG  
BPHXTFALJHASVBFXNGLL**CHR**ZBWELEKMSJIKNBH  
WRJGNMGJSGLXFEYPHAGNRBIEQJTAMRVLCRREM  
NDGLXRRIMGNSNRW**CHR**QHAEYEVTAQEBBIPEEWE  
VKAKOEWADREMXMTBHH**CHR**TKDNVRZ**CHR**CLQOH  
PWQAIIXNRMGWOIIFKEE

## 估算密钥字的长度

可以看出，密文片段**CHR**在密文中出现了**5**次，每次出现的开始位置分别为**1, 166, 236, 276, 286**。第一次出现的距离分别为**165, 235, 275, 285**。容易计算： **$\gcd(165, 235, 275, 285) = 5$** 。

因此，密钥字的长度很可能是**5**。

# 重合指数法的引入

- 如果我们考虑一个来自**26**个字母表的完全随机文本，则每个字母有同样的概率发生，等于**1/26**。假定我们另有第二个随机文本，把它放在第一个下面，然后我们计算有多大的机会找到上下两个字母相等。因为每个字母都是一个随机字符，找到两个都是的概率是 **$(1/26)^2$** 。显然，对于其他字母而言这个概率是不变的，所以找到两个同样字母的总的概率是：  
 **$26(1/26)^2=1/26=0.0385$** 。
- 对英语文本，与随机文本不同，我们发现字母发生的概率是不同的。设字母**A、B、...Y、Z**出现的期望概率分别为 **$p_0、p_1、...p_{24}、p_{25}$** ，此时找到两个等同字母发生的概率为： $\sum_{i=0}^{25} p_i^2 = 0.065$ 。这个值比随机文本的情况大得多。我们把它称之为重合指数。

# 重合指数法的含义及计算方法

- 设某种语言由个 $n$ 字母组成，每个字母 $i$ 发生的概率为  $p_i, 1 \leq i \leq n$ ，则重合指数就是指两个随机字母相同的概率，记为： $IC = \sum_{i=1}^n p_i^2$ 。
- 在单表代换情况下，明文与密文的**IC**值相同。
- 由于现实世界中密文的长度有限，故从密文计算的重合指数总是不同于理论值，所以一般用**IC**的无偏估计值  $IC' = \sum_{i=1}^n \frac{x_i(x_i - 1)}{L(L - 1)}$  来近似计算**IC**， $IC'$ 公式中  $x_i$  是密文符号 $i$ 出现的次数， $L$ 指的是密文长度， $n$ 表示某门语言包含的字母数，如该语言是英文字母，则 **$n=26$** 。



# 利用重合指数估算密钥字长度

密钥字长度	重合指数				
i=1	0.045				
i=2	0.046	0.041			
i=3	0.043	0.050	0.047		
i=4	0.042	0.039	0.046	0.040	
i=5	0.063	0.068	0.069	0.061	0.072

## 交互重合指数的含义

设 $\mathbf{x}=\mathbf{x}_1\mathbf{x}_2\cdots\mathbf{x}_n$ 和 $\mathbf{y}=\mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_{n'}$ 是两个长度分别为 $n$ 和 $n'$ 的字母串， $\mathbf{x}$ 和 $\mathbf{y}$ 的交互重合指数定义为 $\mathbf{x}$ 中的一个随机元素与 $\mathbf{y}$ 中的一个随机元素相同的概率，记为 $MI_c(\mathbf{x},\mathbf{y})$ 。假如英文字母A,B,C, ...,Z在 $\mathbf{x}$ 和 $\mathbf{y}$ 中出现次数分别为 $f_0,f_1,f_2, \dots, f_{25}$ 和 $f'_0,f'_1,f'_2, \dots, f'_{25}$ ，

则

$$MI_c(x, y) = \frac{\sum_{i=0}^{25} f_i f'_i}{nn'}$$

# 交互重合指数的应用

假设已经确定密钥字的长度 $m$ ，密文子串 $y_i$ 中的各个密文字母都是由同一个加法密码得到的。设密钥为 $k=k_1k_2\dots k_m$ ，可估算 $MI_c(y_i, y_j)$ 的值。显然， $y_i$ 中的一个随机元素与 $y_j$ 中的一个随机元素同时为第 $h$ 个英文字母的概率为 $p_{h-k_i}p_{h-k_j}$ ， $0 \leq h \leq 25$ ，这里下标运算为模26运算。因此有

$$MI_c(y_i, y_j) \approx \sum_{h=0}^{25} p_{h-k_i} p_{h-k_j} = \sum_{h=0}^{25} p_h p_{h+k_i-k_j}$$

注：这个估计值仅依赖于 $(k_i-k_j) \bmod 26$ ，称 $(k_i-k_j) \bmod 26$ 为 $y_i$ 和 $y_j$ 的相对位移。

## 举例(密文)

**CHREE**  
VOAHM  
AERAT  
BIAXX  
WTNXB  
EEOPH  
BSBQM  
QEQER  
BWRVX  
UOAKX  
AOSXX  
WEAHB  
WGJMM  
QMNKG  
RFVGX

WTRZX  
WIAKL  
XFPSK  
AUTEM  
NDCMG  
TSXMX  
BTUIA  
DNGMG  
PSREL  
XNJEL  
XVRVP  
RTULH  
DNQWT  
WDTYG  
BPHXT

FALJH  
ASVBF  
XNGLL  
**CHRZB**  
WELEK  
MSJIK  
NBHWR  
JGNMG  
JSGLX  
FEYPH  
AGNRB  
IEQJT  
AMRVL  
CRREM  
NDGLX

RRIMG  
NSNRW  
**CHRQH**  
AEYEV  
TAQEB  
BIPEE  
WEVKA  
KOEWA  
DREMX  
MTBHH  
**CHRTK**  
DNVRZ  
**CHRCL**  
QOHPW  
QAIIW

XNRMG  
WOIIF  
KEE

# 密文子串的交互重合指数

i	j									
1	2	0.028	0.027	0.028	0.034	0.039	0.037	0.026	0.025	0.052
		0.068	0.44	0.026	0.037	0.043	0.037	0.043	0.037	0.028
		0.041	0.041	0.034	0.037	0.051	0.045	0.042	0.036	
1	3	0.039	0.033	0.040	0.034	0.028	0.053	0.048	0.033	0.029
		0.056	0.050	0.045	0.039	0.040	0.036	0.037	0.032	0.027
		0.037	0.036	0.031	0.037	0.055	0.029	0.024	0.037	
1	4	0.034	0.043	0.025	0.027	0.038	0.049	0.040	0.032	0.029
		0.034	0.039	0.044	0.044	0.034	0.039	0.045	0.044	0.037
		0.055	0.047	0.032	0.027	0.039	0.037	0.039	0.035	
1	5	0.043	0.033	0.028	0.046	0.043	0.044	0.039	0.031	0.026
		0.030	0.036	0.040	0.041	0.024	0.019	0.048	0.070	0.044
		0.028	0.038	0.044	0.043	0.047	0.033	0.026	0.046	
2	3	0.046	0.048	0.041	0.032	0.036	0.035	0.036	0.030	0.024
		0.039	0.034	0.029	0.040	0.067	0.041	0.033	0.037	0.045
		0.033	0.033	0.027	0.033	0.045	0.052	0.042	0.030	

# 密文子串的交互重合指数（续）

i	j									
2	4	0.046	0.034	0.043	0.044	0.034	0.031	0.040	0.045	0.040
		0.048	0.44	0.33	0.024	0.028	0.042	0.039	0.026	0.034
		0.050	0.035	0.032	0.040	0.056	0.043	0.028	0.028	
2	5	0.033	0.033	0.036	0.046	0.026	0.018	0.043	0.080	0.050
		0.029	0.031	0.045	0.039	0.037	0.027	0.26	0.031	0.039
		0.041	0.037	0.41	0.046	0.045	0.043	0.035	0.030	
3	4	0.038	0.036	0.040	0.033	0.036	0.060	0.035	0.041	0.029
		0.058	0.035	0.035	0.034	0.053	0.030	0.032	0.035	0.036
		0.036	0.028	0.046	0.032	0.051	0.032	0.034	0.030	
3	5	0.035	0.034	0.034	0.036	0.030	0.043	0.043	0.050	0.025
		0.041	0.051	0.050	0.035	0.032	0.033	0.033	0.052	0.031
		0.027	0.030	0.072	0.035	0.034	0.032	0.043	0.027	
4	5	0.052	0.038	0.033	0.038	0.041	0.043	0.037	0.048	0.028
		0.028	0.036	0.061	0.033	0.033	0.032	0.052	0.034	0.027
		0.039	0.043	0.033	0.027	0.030	0.039	0.048	0.035	

## 确定密钥字

$$k_1 - k_2 = 9 \bmod 26$$

$$k_1 - k_5 = 16 \bmod 26$$

$$k_2 - k_3 = 13 \bmod 26$$

$$k_2 - k_5 = 7 \bmod 26$$

$$k_3 - k_5 = 20 \bmod 26$$

$$k_4 - k_5 = 11 \bmod 26$$

因此,  $k = (k_1, k_1 + 17, k_1 + 4, k_1 + 21, k_1 + 10)$

通过尝试  $k_1$  的每一个可能的取值, 不难确定  
密钥字为 JANET,  $k = (9, 0, 13, 4, 19)$ .

# 明文

**The** almond tree was in tentative blossom. **The** days were longer, often ending with magnificent evenings of corrugated pink skies. **The** hunting season was over, with hounds and guns put away for six months. **The** vineyards were busy again as the well-organized farmers treated **their** vines and the more lackadaisical neighbors hurried to do **the** pruning **they** should have done in November.



# 明文-密文对分析法

所谓明文-密文对分析法是指攻击者不仅获得若干密文，而且还得到这些密文对应的明文，通过若干明文-密文对以分析出密钥的方法。如对于希尔密码而言，其抵抗频率分析攻击能力非常强，若仅知若干密文是很难破译明文的，但如果知道比密钥长度多的明文-密文对则破译就变得相对容易。

# 希尔密码的分析

假设攻击者已经确定正在使用的希尔密码的密钥长度为  $m$ ，并且知道至少有  $m$  个不同的明文-密文对：

$$P_i = (p_{1i}, p_{2i}, \dots, p_{mi}), c_i = (c_{1i}, c_{2i}, \dots, c_{mi}), \quad j = 1, 2, \dots, m$$

则由 Hill 密码的定义可得：

$$[c]_{m \times m} = ([p]_{m \times m} \times [k]_{m \times m}) \bmod 26 \Rightarrow [k]_{m \times m} = ([p]_{m \times m}^{-1} \times [c]_{m \times m}) \bmod 26$$

即

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix}^{-1} \times \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mm} \end{bmatrix} \pmod{26}$$

上述结论成立的前提条件是  $[p]^{-1}$  存在，这就要求明文  $p$  是一个非奇异方阵，即要求  $\det(k) \neq 0$ ，且满足  $(\det(k), 26) = 1$ ，即满足  $\mathbb{Z}_{26}$  上  $\det(k)$  和 26 互素，所以利用明文-密文对方法分析 Hill 矩阵时至少要知道与密钥长度相等的明文-密文对。如果明文矩阵不可逆，则尝试其它已知的明文-密文对重新组成明文矩阵再分析，直到找到满足可逆要求明文矩阵为止。

# 希尔密码举例

设明文“cryptology information sets”用 $m=5$ 的希尔密码加密得到5个密文序列如下：

crypt	ology	infor	matio	nsets
DWVOT	ZMHII	DHIXX	MXPAG	IPGDS

同时知道明文“information security center”用 $m=5$ 的希尔密码加密同样得到5个密文序列：

infor	matio	nsecu	rityc	Enter
DHIXX	MXPAG	IPGEA	IEJKY	XJKRV

首先测试前5个明文组成的矩阵是否可逆，如果可逆，则可以求出密钥矩阵，然后再利用后5个明文-密文验证得出的密钥矩阵的正确性。

把前5个明文-密文对数字化得明文矩阵 $p$ 和密文矩阵 $c$ 分别为：

$$p = \begin{bmatrix} 2 & 17 & 24 & 15 & 19 \\ 14 & 11 & 14 & 6 & 24 \\ 8 & 13 & 5 & 14 & 17 \\ 12 & 0 & 19 & 8 & 14 \\ 13 & 18 & 4 & 19 & 18 \end{bmatrix}, \quad c = \begin{bmatrix} 3 & 22 & 21 & 14 & 19 \\ 25 & 12 & 7 & 8 & 8 \\ 3 & 7 & 8 & 23 & 23 \\ 12 & 23 & 15 & 0 & 6 \\ 8 & 15 & 6 & 3 & 18 \end{bmatrix}$$

## 希尔密码举例(续)

由明文矩阵  $p$  易得其行列式的值为  $|p| = \det(p) = (-338697) \bmod 26 = 5 \neq 0$ 。可得：

$$p^{-1} = 21p^* = \begin{bmatrix} 441 & 462 & 483 & 189 & 105 \\ 294 & 315 & 504 & 294 & 294 \\ 84 & 252 & 42 & 483 & 252 \\ 357 & 42 & 105 & 231 & 168 \\ 84 & 525 & 483 & 189 & 252 \end{bmatrix} (\bmod 26) = \begin{bmatrix} 25 & 20 & 15 & 7 & 1 \\ 8 & 3 & 10 & 8 & 8 \\ 6 & 18 & 16 & 15 & 18 \\ 19 & 16 & 1 & 23 & 12 \\ 6 & 5 & 15 & 7 & 18 \end{bmatrix} \leftarrow$$

从而得 Hill 密码的密钥矩阵为：

$$k = p^{-1} \times c = \begin{bmatrix} 712 & 1071 & 896 & 858 & 1040 \\ 289 & 586 & 437 & 390 & 598 \\ 840 & 1075 & 713 & 650 & 1040 \\ 832 & 1326 & 936 & 453 & 866 \\ 416 & 728 & 494 & 523 & 865 \end{bmatrix} (\bmod 26) = \begin{bmatrix} 10 & 5 & 12 & 0 & 0 \\ 3 & 14 & 21 & 0 & 0 \\ 8 & 9 & 11 & 0 & 0 \\ 0 & 0 & 0 & 11 & 8 \\ 0 & 0 & 0 & 3 & 7 \end{bmatrix} \leftarrow$$

至此 Hill 密码已经解密，可通过后 5 个明文-密文对验证结论的正确性。

# 传统密码的一些启发

- 代换或置换交叉使用;

多次使用一种代换或置换不增加安全性。

- 加解密码算法;

算法保密增加安全性，但实际应用往往难于保证。

- 密钥空间;

应能抵御穷举攻击。

- 密文的数量;

常更新密钥可减少密文的数量。

- 明文密文对;

做好已用信息的保密。

# 小结

- 置换密码（列置换密码和周期置换密码）
- 代换密码（单表代换密码、多表代换密码和轮转密码机）
- 典型传统密码的分析（统计分析法和明文-密文对分析法）

# 谢谢！

