第 3 章　古典密码算法

密码学发展至今，其发展历史可分为三个阶段：古典密码学、现代密码学及公钥密码学。古典密码学所用的主要有两大基本方法：代替和置换。

（1）置换密码：明文中的字符保持不变，但其顺序发生了变化。

（2）代替密码：将明文中的字符替换为其他字符而得到密文。

古典密码学是密码学的根源，使用的置换与代替方法也是对称密码学中的基本方法，因此了解古典密码学的设计原理和分析方法有助于更好地理解现代密码学技术。本章将介绍几种常见的古典密码算法。

3.1　算法原理

3.1.1　置换密码

置换密码又称换位密码，其特点是对明文中字符的顺序按照一定的规则进行重新排列，而明文中的字符不会发生变化。最简单的置换密码为栅栏密码和矩阵密码。

1．栅栏密码

栅栏密码按照列的顺序将明文（去掉空格）写入 m 行 n 列的数组，按照行的顺序将字符重新组合得到密文，这种方式称为 m 栏栅栏密码。比较常见的是 2 栏栅栏密码（m 取 2）。下面举例说明。

当 $n=9$，$m=2$ 时，假设明文为

the rail fence cipher

加密过程如下：

（1）将明文去掉空格后得到：therailfencecipher。

（2）将明文按照列的顺序写入 2 行 9 列的数组：

t	e	a	l	e	c	c	p	e
h	r	i	f	n	e	i	h	r

（3）按行读取每一行内容，得到栅栏密码的密文为

tealeccpehrifneihr

解密过程如下：

（1）将密文分成 2 组：

tealeccpe

hrifneihr

（2）按照列的顺序将密文进行重新组合：th er ai lf en ce ci ph er。

（3）将组合后的字符拼接起来，根据语义添加相应的空格得到明文：the rail fence cipher。

2．矩阵密码

矩阵密码以一个字符串作为密钥，密钥中的字符各不相同。加密时，将明文消息按行写成矩阵块，之后以密文字符的顺序按列读出矩阵中的字符，得到密文。矩阵密码流程图如图 3-1 所示。

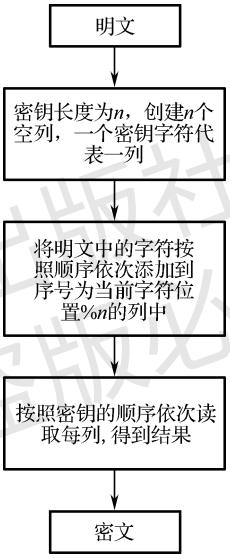

图 3-1　矩阵密码流程图

下面举例说明。

设密钥为 4231，明文为 this is an example。

第 1 步：创建 4 个空列，一个密钥字符代表一列。

4	2	3	1

第 2 步：将明文中的字符按照顺序依次填入列中。

4	2	3	1
t	h	i	s
i	s	a	n
e	x	a	m
p	l	e	

第 3 步：将填入的内容按照密钥的顺序依次读出每列：snm、hsxl、iaae、tiep，得到加密的密文为 snmhsxliaaetiep。

解密是加密的逆运算。首先计算出矩阵的行数，即用明文长度除以密钥长度并向上取整的结果，之后根据密钥顺序按列填充矩阵，最后按行读出，就是解密得到的消息。

3.1.2　代替密码

代替密码的特点是将明文中字符按照一定的规则替换成其他字符。下面分别介绍单表代替密码、仿射密码、维吉尼亚密码、弗纳姆密码及 Hill 密码。

1．单表代替密码

将 26 个英文字母分别替换为另一个字母，通信双方均持有一张固定的表，记录每个字母对应的代替字母。加密时，将明文中的字母按照密码表，用对应的字母进行代替，得到相应的密文。下面给出一个单表代替密码的具体例子，表 3-1 为单表代替密码中的置换表。

表 3-1　置换表

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

明文 word 通过表 3-1 进行代替，所得到的密文为 xwpo。

2．仿射密码

仿射密码将所有字母对应至相应数值，利用加密函数对字母进行加密，将得到的结果转换为相应的字母，从而得到密文。加密函数用以下公式表示：

$$E(x)=(ax+b)\bmod m$$

其中 a 和 m 互质，m 是字母的数量。

以英文字母表中的 26 个字母作为编码系统，将 26 个英文字母表示成 0～25 的数字，m 为字母的数量 26，选择与 26 互质的数字 3，加密函数为 $E(x)=(3x+6)\bmod 26$。以明文 chinese 为例进行加密，表 3-2 为加密时的明、密文及中间数据。

表 3-2　加密时的明、密文及中间数据

明文	c	h	i	n	e	s	e
x	2	7	8	13	4	18	4
y=3x+6	12	27	30	45	18	60	18
y mod 26	12	1	4	19	18	8	18
密文	m	b	e	t	s	i	s

因此，得到的密文为 mbetsis。

3．维吉尼亚密码

维吉尼亚密码是使用一系列凯撒密码组成密码字母表的加密算法，属于多表密码的一种简单形式。其中，凯撒密码是一种代替密码技术，其明文中的所有字母都在字母表中向后（或向前）按照一个固定数目进行偏移后被替换成密文。维吉尼亚密码选择多个密钥对明文中的字母进行偏移。

加密方法为 $C=(P+K) \bmod 26$，其中 C 代表密文，P 代表明文，K 代表密钥。

下面举一个例子来说明，明文为 hello，密钥为 thist，得到的密文如表 3-3 所示。

表 3-3　得到的密文

明文	h	e	l	l	o
密钥	t	h	i	s	t
密文	a	l	t	d	h

将 26 个英文字母表示成 0～25 的数字，明文中的 h 为 7，密文 t 为 19，经过加密之后为 0，得到密文 a。

解密可通过公式 $P=(C-K) \bmod 26$ 进行。

4．弗纳姆密码

弗纳姆（Vernam）密码也称一次一密（One-Time-Pad），密钥长度和明文相同，运算过程基于二进制数形式而非字母，加、解密过程为按位异或，可简要表述如下。

加密：$c_i = p_i \oplus k_i$

解密：$p_i = c_i \oplus k_i$

图 3-2 为弗纳姆密码加密过程。

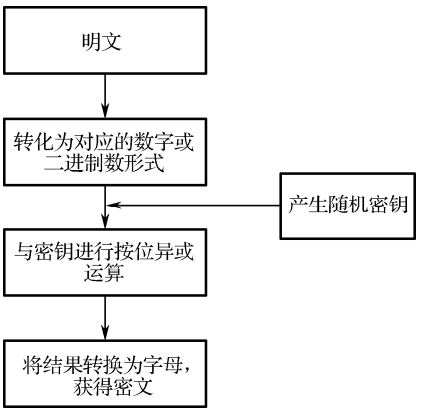

图 3-2 弗纳姆密码加密过程

5．Hill 密码

希尔密码（Hill Cipher）是运用基本矩阵论原理的代替密码技术，由 Lester S. Hill 在 1929 年发明。26 个英文字母可表示为 0～25 的数字，将明文转换为 n 维向量，与一

个 $n\times n$ 阶矩阵相乘后，将得到的结果模 26，即可得到密文对应的数值。

假设对明文 act 加密：a 为 0，c 为 2，t 为 19，对其进行向量化得到 $\boldsymbol{M}=[0,2,19]^{\mathrm{T}}$。选取 3×3 阶矩阵密钥：

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

加密过程如下：

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}=\begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix}\equiv\begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix}\ \mathrm{mod}\ 26$$

得到的密文为 poh。

解密时，必须先算出密钥的逆矩阵，再根据加密的过程做逆运算。

对 m 维 Hill 密码的已知明文攻击介绍如下。

由于 Hill 密码完全采用了线性代数的方法，因此较容易受到攻击，且很难抵抗已知明文攻击。已知明文攻击是一种攻击模式，指的是攻击者在掌握了某段明文和对应的密文的情况下发起的攻击。

对 m 维 Hill 密码的已知明文攻击步骤如下：

（1）假设已知的明/密文对都为 m 维向量，则可知密钥为一个 $m\times m$ 阶矩阵；

（2）根据公式 $\boldsymbol{M}\cdot\boldsymbol{K}=\boldsymbol{C}$，可得 $\boldsymbol{K}=\boldsymbol{M}^{-1}\cdot\boldsymbol{C}$，选取 m 个明/密文对构造明/密文矩阵 $\boldsymbol{M}$ 和 $\boldsymbol{C}$，可通过线性计算获得密钥 $\boldsymbol{K}$；

（3）通过其他的明/密文对验证密钥的正确性；

（4）当 m 未知的时候，需多次猜测 m 的值进行验证。

对 m 维 Hill 密码的已知明文攻击流程图如图 3-3 所示。

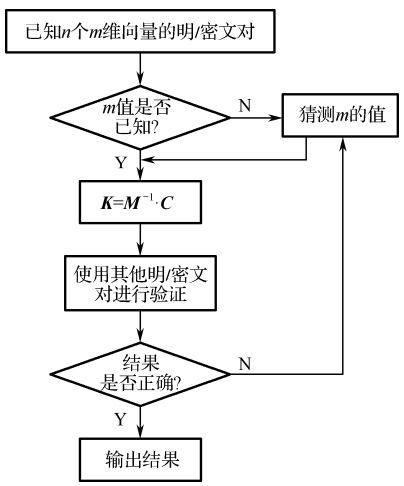

图 3-3　对 m 维 Hill 密码的已知明文攻击流程图

3.2 算法实现

3.2.1 栅栏密码实现

本节介绍如何实现算法，算法实现使用 Java 语言。

栅栏密码算法在实现时可以被定义为一个单独的类 RailfenceCipher，RailfenceCipher 类中包含的各种数据如表 3-4 所示。

表 3-4 RailfenceCipher 类中包含的各种数据

变 量 名	类 型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
row	int	行数
column	int	最大列数

除了数据定义，RailfenceCipher 类中还包含各种运算函数，如表 3-5 所示。

表 3-5 RailfenceCipher 类中包含的各种运算函数

函 数 名	类 型	中 文 解 释
encrypt	String[]	加密函数
decrypt	String[]	解密函数

通过上述的结构分析，得到 RailfenceCipher 类的声明代码：

```
public class RailfenceCipher{
    public String[] encrypt();
    public String[] decrypt();
}
```

下面给出加密函数的具体描述。算法加密时输入行数 row 和明文 plaintext，输出加密后的密文 ciphertext。加密函数伪代码如下：

```
public String[] encrypt(int row, String[] plaintext) {
    若明文不能正常分组，则用“*”填充;
    计算最大列数 column 为 plaintext.length/row 并向上取整;
    构造二维数组 ans，行数为 row，列数为 column;
    逐字符提取明文 plaintext，将提取出的字符按照从上到下、从左到右的顺序依次填入 temp;
    将 temp 按行填入 ciphertext;
    返回密文 ciphertext;
}
```

解密函数与加密函数类似，解密函数伪代码如下：

```
public String[] decrypt(int row, String[] ciphertext) {
    计算最大列数 column 为 plaintext.length/row 并向上取整;
    声明一个行数为 row，列数为 column 的二维数组 temp[][]，初始化时用“*”填充;
    计算最大列数 column;
    逐字符提取密文 ciphertext，将其按照从左到右、从上到下的顺序填入 temp;
    将 temp 按列填入 plaintext;
    返回明文 plaintext;
}
```

3.2.2　矩阵密码实现

矩阵密码是栅栏密码的一种改进，其在栅栏密码的基础上增加密钥序列，调整列的顺序。加密时，将明文按行写入二维数组，读入密钥序列后，以每一位密钥序列的值为下标，以该位的下标为值构造顺序表，输出时即可按照密钥序列来打乱列的次序。解密时则相反，将密文按列写入二维数组，按照密钥序列次序按行输出即可。与栅栏密码的处理方法相同，对于长度不能整除二维数组列数的情况，不用字母进行填充，输出时直接输出空格。

矩阵密码算法类中包含的各种数据如表 3-6 所示。

表 3-6　矩阵密码算法类中包含的各种数据

变　量　名	类　　型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
key	int[]	密钥

除了数据定义，矩阵密码算法类中还包含各种运算函数，如表 3-7 所示。

表 3-7　矩阵密码算法类中包含的各种运算函数

函　数　名	类　　型	中 文 解 释
encrypt	String[]	加密函数
decrypt	String[]	解密函数

首先实现加密函数，加密函数伪代码如下：

```
public String[] encrypt (String[] plaintext, int[] key) {
    若明文不能正常分组，则用“*”填充;
    计算最大行数 row 为 plaintext.length/key.length 并向上取整;
    构造二维数组 ans，行数为 row，列数为 key.length;
    逐字符提取明文 plaintext，将其按照从左到右、从上到下的顺序填入数组;
```

```
        根据密钥序列按密钥字符顺序读取每列数据;
        返回密文 ciphertext;
    }
```

解密函数与加密函数类似，输入密钥 key 和密文 ciphertext，计算最大行数 row 并构造二维数组，逐字符提取密文，将其按照从左到右、从上到下的顺序填入数组，根据密钥序列按行输出即可。具体算法读者可自行实现。

3.2.3 单表代替密码实现

单表代替密码算法在实现时可以被定义为一个单独的类 MonoSubCipher，MonoSubCipher 类中包含的各种数据如表 3-8 所示。

表 3-8 MonoSubCipher 类中包含的各种数据

变 量 名	类 型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
keyBox	String[]	置换表
invKeyBox	String[]	反置换表

除了数据定义，MonoSubCipher 类中还包含各种运算函数，如表 3-9 所示。

表 3-9 MonoSubCipher 类中包含的各种运算函数

函 数 名	类 型	中 文 解 释
encrypt	String[]	加密函数
decrypt	String[]	解密函数

通过上述的结构分析，得到 MonoSubCipher 类的声明代码：

```
public class MonoSubCipher {
    public String[] encrypt();
    public String[] decrypt();
}
```

加密函数伪代码如下：

```
public String[] encrypt(String[] plaintext, String[] keyBox){
    对明文每个字符按照置换表进行替换;
    返回密文 ciphertext;
}
```

解密函数伪代码如下：

```
public String[] decrypt (String[] ciphertext, String[] invKeyBox){
    对密文每个字符按照反置换表进行替换;
    返回明文 plaintext;
}
```

3.2.4　仿射密码实现

仿射密码算法在实现时可以被定义为一个单独的类 Affine，Affine 类中包含的各种数据如表 3-10 所示。

表 3-10　Affine 类中包含的各种数据

变 量 名	类　型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
a	int	密钥
b	int	密钥

除了数据定义，Affine 类中还包含各种运算函数，如表 3-11 所示。

表 3-11　Affine 类中包含的各种运算函数

函 数 名	类　型	中 文 解 释
encrypt	String[]	加密函数
decrypt	String[]	解密函数

通过上述的结构分析，得到 Affine 类的声明代码：

```
public class Affine {
    public String[] encrypt();
    public String[] decrypt();
}
```

加密函数伪代码如下：

```
public String[] encrypt(String[] plaintext, int a, int b) {
    对明文进行编码;
    对明文的每个字符进行如下计算：
        密文字符 = (a*明文字符 + b) mod 26 ;
    返回密文 ciphertext;
}
```

解密函数伪代码如下：

```
public String[] decrypt(String[] ciphertext, int a, int b) {
    对密文进行编码;
    计算 a 在模 m 意义下的逆元 a^-1;
    对密文的每个字符进行如下计算：
        明文字符 = (a^-1*(密文字符 – b)) mod 26;
    返回明文 plaintext;
}
```

3.2.5 维吉尼亚密码实现

维吉尼亚密码是一种简单形式的多表代替密码技术。维吉尼亚密码算法在实现时可以被定义为一个单独的类 Vigenere，Vigenere 类中包含的各种数据如表 3-12 所示。

表 3-12 Vigenere 类中包含的各种数据

变 量 名	类 型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
key	String[]	密钥

除了数据定义，Vigenere 类中还包含各种运算函数，如表 3-13 所示。

表 3-13 Vigenere 类中包含的各种运算函数

函 数 名	类 型	中 文 解 释
encrypt	String[]	加密函数
decrypt	String[]	解密函数

通过上述的结构分析，得到 Vigenere 类的声明代码：

```
public class Vigenere{
    public String[] encrypt();
    public String[] decrypt();
}
```

加密函数伪代码如下：

```
public String[] encrypt(String[] plaintext, String[] key) {
    对 plaintext 和 key 进行编码;
    对 plaintext 的每个字符进行如下计算：
        密文字符 = (明文字符 + 密钥字符) mod 26;
    返回密文 ciphertext;
}
```

解密函数伪代码如下：

```
public String[] decrypt(String[] ciphertext, String[] key) {
    对 ciphertext 进行编码;
    对 ciphertext 的每个字符进行如下计算：
        明文字符 = (密文字符 - 密钥字符) mod 26;
    返回明文 plaintext;
}
```

3.2.6　弗纳姆密码实现

弗纳姆密码算法的明文和密文都可以定义为等长的二进制字符串，明文和密文直接通过与密钥进行异或运算互相转换，在实现时可以被定义为一个单独的类 Vernam，Vernam 类中包含的各种数据如表 3-14 所示。

表 3-14　Vernam 类中包含的各种数据

变 量 名	类　型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
key	String[]	密钥

除了数据定义，Vernam 类中还包含各种运算函数，如表 3-15 所示。

表 3-15　Vernam 类中包含的各种运算函数

函 数 名	类　型	中 文 解 释
encrypt	String[]	加密函数
decrypt	String[]	解密函数

通过上述的结构分析，得到 Vernam 类的声明代码：

```
public class Vernam{
    public String[] encrypt();
    public String[] decrypt();
}
```

弗纳姆密码算法的加密和解密过程比较简单，在加密和解密时分别循环执行下述异或操作即可。

加密函数伪代码如下：

```
public String[] encrypt(String[] plaintext, String[] key){
    对 plaintext 和 key 进行编码;
    对 plaintext 和 key 按位异或得到 ciphertext;
    返回密文 ciphertext;
}
```

解密函数伪代码如下：

```
public String[] decrypt(String[] ciphertext, String[] key){
    对 ciphertext 和 key 进行编码;
    对 ciphertext 和 key 按位异或得到 plaintext;
    返回明文 plaintext;
}
```

3.2.7 Hill 密码实现

Hill 密码是一种基于基本矩阵理论的代替密码，在本次算法实现中，用明文和密文向量左乘密钥矩阵。Hill 密码算法在实现时可以被定义为一个单独的类 Hill，Hill 类中包含的各种数据如表 3-16 所示。

表 3-16 Hill 类中包含的各种数据

变 量 名	类 型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
key	String[][]	密钥矩阵
invkey	String[][]	密钥的逆矩阵

除了数据定义，Hill 类中还包含各种运算函数，如表 3-17 所示。

表 3-17 Hill 类中包含的各种运算函数

函 数 名	类 型	中 文 解 释
mulitMatrix	String[][]	计算矩阵乘法
getInv	String[][]	计算逆矩阵
encrypt	String[]	加密函数
decrypt	String[]	解密函数

通过上述的结构分析，得到 Hill 类的声明代码：

```
public class Hill{
    public String[][] mulitMatrix();
    public String[][] getInv();
    public String[] encrypt();
    public String[] decrypt();
}
```

加密函数伪代码如下：

```
public String[] encrypt(String[] plaintext, String[][] key){
```

```
        对 plaintext 进行分组，编码为多组 n×n 矩阵形式;
        按顺序与 key 进行模 26 矩阵乘法;
        将结果编码为 ciphertext;
        返回密文 ciphertext;
    }
```

解密函数伪代码如下：

```
public String[] decrypt(String[] ciphertext, String[][] key){
    调用 getInv，输入 key，计算 invkey;
    对 ciphertext 进行分组，编码为多组 n×n 矩阵形式;
    按顺序与 invkey 进行模 26 矩阵乘法;
    将结果编码为 plaintext;
    返回明文 plaintext;
}
```

对 *m* 维 Hill 密码的已知明文攻击算法实现介绍如下。

对 *m* 维 Hill 密码的已知明文攻击在实现时可以被定义为一个单独的类 AttackHill，AttackHill 类中包含的各种数据如表 3-18 所示。

表 3-18　AttackHill 类中包含的各种数据

变　量　名	类　　型	中 文 解 释
plaintext	String[]	输入的明文
ciphertext	String[]	输出的密文
n	int	密钥维数
M	String[][]	明文可逆矩阵
C	String[][]	M 对应的密文矩阵
key	String[][]	密钥

除了数据定义，AttackHill 类中还包含各种运算函数，如表 3-19 所示。

表 3-19　AttackHill 类中包含的各种运算函数

函　数　名	类　　型	中 文 解 释
decrypt	String[][]	解密函数
choose	String[][][]	获取可逆矩阵

通过上述的结构分析，得到 AttackHill 类的声明代码：

```
public class AttackHill{
    public String[][] decrypt();
    public String[][][] choose();
}
```

获取可逆矩阵的伪代码如下：

```
public String[][][] choose(String[] plaintext, String[] ciphertext, int n){
    从所有明文中选择可逆的明文矩阵 M 和对应的密文矩阵 C;
    计算 M^-1 ;
    返回 M^-1 和 C;
}
```

解密函数伪代码如下：

```
public String[] decrypt(String[] plaintext, String[] ciphertext, int n){
    调用 choose 函数，获取 M^-1 和 C;
    进行如下计算：
        key = (M^-1*C) ;
    返回 key;
}
```

上述伪代码中关于求逆的算法和矩阵乘法在 Hill 密码实验部分已经给出，这里不再赘述。

3.3 算法测试

3.3.1 栅栏密码测试

可自定义行数 n，当 $n=3$ 或 $n=2$ 时，测试数据如表 3-20 所示。

表 3-20 栅栏密码测试数据

行　数	明　文	密　文
n=3	whateverisworthdoingisworthdoingwell	wtesrdnsrdneherwtogwtoglaviohiiohiwl
n=2	healthismoreimportantthanwealth	hatimriprathnelhelhsoemotntawat

3.3.2 矩阵密码测试

以密钥 key = 6234517 或 key = 7345261 为例，测试数据如表 3-21 所示。

表 3-21 矩阵密码测试数据

密　钥	明　文	密　文
6234517	tobeornottobethatisaquestion	reaootaebttseoitobsitohuntqn
7345261	ohtheworldhasbeenmadebyfools	obeseaaohleytdnfhhmowsdloreb

3.3.3　单表代替密码测试

单表代替密码置换表如表 3-22 所示，单表代替密码测试数据如表 3-23 所示。

表 3-22　单表代替密码置换表

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

表 3-23　单表代替密码测试数据

明　　文	密　　文
doyouwannatodance	wbobmkqggqjbwqgzs
pysibrjgbxxhbrig	youcanreallydance

3.3.4　仿射密码测试

编程实现仿射密码，明文为 p，密文为 c，且 $c=(kp+b) \bmod 26$。若参数 k 不合法，则报错。表 3-24 给出了仿射密码测试数据。

表 3-24　仿射密码测试数据

k，b	明　　文	密　　文
k=3，b=1	iambuaaer	zblejbbna
k=5，b=5	beijinghuanyingni	kztytsjobfsytsjst
k=7，b=10	cryptography	yzwlneazklhw
k=9，b=13	seeyoutomorrow	txxvjlcjrjkkjd
k=15，b=20	thisisciphertext	tvkekeyklvcptcbt
k=2，b=1	abcdef	报错

3.3.5　维吉尼亚密码测试

要求实现加密与解密功能，字符编号：a为0，b为l，…，z为25，模数为 26。表 3-25 给出了 3 组测试数据。

表 3-25　维吉尼亚密码测试数据

密　　钥	明　　文	密　　文
deceptive	wearediscoveredsaveyourself	zicvtwqngrzgvtwavzhcqyglmgj
chinese	zhonghuaminzuweidafuxing	bowakzyctqadmagplnjmbkuo
music	chenxingyinyueting	obwvzuhygkzsmmvuhy

3.3.6 弗纳姆密码测试

要求实现加密与解密功能。表 3-26 给出了 3 组十六进制测试数据，可直接按位异或。

表 3-26 弗纳姆密码测试数据

密　钥	明　文	密　文
0xf1571c94	0x01234567	0xf07459f3
0x3475bd76fa040b73	0x1b5e8b0f1bc78d23	0x2f2b3679e1c38650
0x2b24424b9fed596659842a4d0b007c61	0x41b267bc5905f0a3cd691b3ddaee149d	0x6a9625f7c6e8a9c594ed3170d1ee68fc

3.3.7 Hill 密码测试

明文左乘矩阵即可生成所需的密文，表 3-27 给出了具有不同阶数密钥的 3 组测试数据。

表 3-27 Hill 密码测试数据

密　钥	明　文	密　文
[5,8;17,3]	loveyourself	haryuazdcakz
[6,24,1;13,16,10;20,17,15]	ysezymxvv	qweasdzxc
[25,0,0,0;25,9,0,0;0,6,21,6;20,7,6,1]	thisisnotciphertext	wdqouatotpylfldrrorh

对 m 维 Hill 密码的已知明文攻击算法测试介绍如下。

字符编号：a为0，b为l，…，z为25，模数为 26。表 3-28 给出了 2 组测试数据。

表 3-28 m 维 Hill 密码的已知明文攻击测试数据

明　文	密　文	m	明文分组 X	密文分组 K	X 的逆	密钥 K
youarepretty	kqoimjvdbokn	2	repr	mjvd	[7,6;3,7]	[2,3;1,22]
youaresocute	ywwpcwsogfuk	3	无解	无解	null	无解

3.4 思考题

（1）对于 m 维 Hill 密码，请使用 m 表示其破解效率。

（2）请指出古典密码体制的缺陷与不足。

（3）请指出一次一密的两个问题。