信息安全引论作业 01

梁昱桐 2100013116

Peking University

1 求满足方程的最小正整数 x

1.1 $5x \equiv 4 \pmod 3$

简化方程：$2x \equiv 1 \pmod 3$

2 的逆元是 2, 所以方程变为 $x \equiv 2 \pmod 3$

因此 $x = 2$

1.2 $7x \equiv 6 \pmod 5$

简化方程：$2x \equiv 1 \pmod 5$

2 的逆元是 3, 所以方程变为 $x \equiv 3 \pmod 5$

因此 $x = 3$

2 有多少种仿射密码?

仿射密码的形式为：$E(x) = (ax + b) \pmod m$，其中 a 和 m 互质，并且 a 和 b 都在模 m 下取值。

对于给定的模数 m，a 的选择数目为与 m 互质的数的个数，即欧拉函数 $\varphi(m)$。b 的选择数目为 m。

对于 Z_{26}（字母表），$m = 26$：$\varphi(26) = \varphi(2) \cdot \varphi(13) = 1 \cdot 12 = 12$

因此，a 的选择数目为 12，b 的选择数目为 26。

同时在 $a \equiv 1 \pmod{26}$ 时，b 不可以取 0，所以需要减去 1。

总的仿射密码种类数为：$12 \cdot 26 - 1 = 311$

3 求 Z_5 中各个非零元素的乘法逆元

- $1^{-1} \equiv 1 \pmod 5$
- $2^{-1} \equiv 3 \pmod 5$
- $3^{-1} \equiv 2 \pmod 5$
- $4^{-1} \equiv 4 \pmod 5$

4 3.10

4.1 用密钥largest构造一个Playfair矩阵。

1. 去除密钥中的重复字母，得到 "largest"。

2. 将密钥字母按顺序填入矩阵：
3. 剩余的字母按字母顺序填入矩阵，I 和 J 共享一个位置。

最终的Playfair矩阵为：

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

4.2 用密钥occurrence构造一个Playfair矩阵。

对密钥中的冗余字母的处理方法做出合理的假设。

Answer:

1. 去除密钥中的重复字母，得到 "ocuren"。
2. 将密钥字母按顺序填入矩阵：
3. 剩余的字母按字母顺序填入矩阵，I 和 J 共享一个位置。

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

5 3.11

5.1 使用Playfair矩阵加密消息

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

```
Must see you over Cadogan West. Coming at once.
```

Answer:

1. 将消息分割为双字母对：
    - MU
    - ST

- SE
- EY
- OU
- OV
- ER
- CA
- DO
- GA
- NW
- ES
- TC
- OM
- IN
- GA
- TO
- NC
- EX

2. 加密每个双字母对：
    - MU -> UZ
    - ST -> TB
    - SE -> DL
    - EY -> GZ
    - OU -> PN
    - OV -> NW
    - ER -> LG
    - CA -> TG
    - DO -> TU
    - GA -> ER
    - NW -> OV
    - ES -> LD
    - TC -> BD
    - OM -> UH
    - IN -> FP

- GA -> ER
- TO -> HW
- NC -> QS
- EX -> RZ

3. 将加密后的双字母对连接起来，得到密文：
    - UZTBDLGZPNNWLGTGTUEROVLDBDUHFPERHWQSRZ

5.2 用习题3.10(a)中的Playfair矩阵重做习题3.11(a)

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

1. 将消息分割为双字母对：
    - MU
    - ST
    - SE
    - EY
    - OU
    - OV
    - ER
    - CA
    - DO
    - GA
    - NW
    - ES
    - TC
    - OM
    - IN
    - GA
    - TO
    - NC
    - EX
2. 加密每个双字母对：

- MU -> UZ
- ST -> TB
- SE -> DL
- EY -> GZ
- OU -> PN
- OV -> NW
- ER -> LG
- CA -> TG
- DO -> TU
- GA -> ER
- NW -> OV
- ES -> LD
- TC -> BD
- OM -> UH
- IN -> FP
- GA -> ER
- TO -> HW
- NC -> QS
- EX -> RZ

3. 将加密后的双字母对连接起来，得到密文：
    - UZTBDLGZPNNWLGTGTUEROVLDBDUHFPERHWQSRZ

5.3 对这个习题的结果你如何解释?

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

是

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

向下平移 4 行, 向右平移 1 列得到的结果, Playfair 加密过程保证了, 密钥矩阵的平移是不会影响加密结果的, 所以两种矩阵的加密结果是相同的.

6 加密

密文为 c，明文为 m，26 个字母编号为 $0 \sim 25$，加密算法为 $c = (7m + 11) \pmod{26}$，当明文为 `hello` 时，对应的密文是什么?

Answer:

H -> 7 -> 7 * 7 + 11 = 60 -> 60 % 26 = 8 -> I
E -> 4 -> 4 * 7 + 11 = 39 -> 39 % 26 = 13 -> N
L -> 11 -> 11 * 7 + 11 = 88 -> 88 % 26 = 10 -> K
L -> 11 -> 11 * 7 + 11 = 88 -> 88 % 26 = 10 -> K
O -> 14 -> 14 * 7 + 11 = 109 -> 109 % 26 = 5 -> F

综上，密文为 `INKKF`

7 逆置换

设 π 为集合 $\{1, \ldots, 8\}$ 上的置换:

x	1	2	3	4	5	6	7	8
$\pi(x)$	4	1	6	2	7	3	8	5

求出逆置换 π^{-1}

Answer:

x	1	2	3	4	5	6	7	8
$\pi^{-1}(x)$	2	4	6	1	8	3	5	7

8 穷尽密钥搜索法

使用穷尽密钥搜索法，破译如下列用移位密码加密的密文

BEEAKFYDJXUQYHYJIQRYHTYJIQFBQDUYJIIKFUHCQD

Answer:

代码:

```c
#include <stdio.h>
#include <string.h>

void decrypt(char *text, int shift)
{
    int length = strlen(text);
    char decrypted[length + 1];
```

```
    for (int i = 0; i < length; i++)
    {
        char c = text[i];
        if (c >= 'A' && c <= 'Z')
        {
            decrypted[i] = (c - 'A' - shift + 26) % 26 + 'A';
        }
        else
        {
            decrypted[i] = c;
        }
    }
    decrypted[length] = '\0';
    printf("Shift %d: %s\n", shift, decrypted);
}

int main()
{
    char ciphertext[] = "BEEAKFYDJXUQYHYJIQRYHTYJIQFBQDUYJIIKFUHCQD";
    for (int shift = 1; shift < 26; shift++)
    {
        decrypt(ciphertext, shift);
    }
    return 0;
}
```

结果:

```
Shift 1: ADDZJEXCIWTPXGXIHPQXGSXIHPEAPCTXIHHJETGBPC
Shift 2: ZCCYIDWBHVSOWFWHGOPWFRWHGODZOBSWHGGIDSFAOB
Shift 3: YBBXHCVAGURNVEVGFNOVEQVGFNCYNARVGFFHCREZNA
Shift 4: XAAWGBUZFTQMUDUFEMNUDPUFEMBXMZQUFEEGBQDYMZ
Shift 5: WZZVFATYESPLTCTEDLMTCOTEDLAWLYPTEDDFAPCXLY
Shift 6: VYYUEZSXDROKSBSDCKLSBNSDCKZVKXOSDCCEZOBWKX
Shift 7: UXXTDYRWCQNJRARCBJKRAMRCBJYUJWNRCBBDYNAVJW
Shift 8: TWWSCXQVBPMIQZQBAIJQZLQBAIXTIVMQBAACXMZUIV
Shift 9: SVVRBWPUAOLHPYPAZHIPYKPAZHWSHULPAZZBWLYTHU
Shift 10: RUUQAVOTZNKGOXOZYGHOXJOZYGVRGTKOZYYAVKXSGT
Shift 11: QTTPZUNSYMJFNWNYXFGNWINYXFUQFSJNYXXZUJWRFS
Shift 12: PSSOYTMRXLIEMVMXWEFMVHMXWETPERIMXWWYTIVQER
Shift 13: ORRNXSLQWKHDLULWVDELUGLWVDSODQHLWVVXSHUPDQ
Shift 14: NQQMWRKPVJGCKTKVUCDKTFKVUCRNCPGKVUUWRGTOCP
Shift 15: MPPLVQJOUIFBJSJUTBCJSEJUTBQMBOFJUTTVQFSNBO
```

```
Shift 16: LOOKUPINTHEAIRITSABIRDITSAPLANEITSSUPERMAN
Shift 17: KNNJTOHMSGDZHQHSRZAHQCHSRZOKZMDHSRRTODQLZM
Shift 18: JMMISNGLRFCYGPGRQYZGPBGRQYNJYLCGRQQSNCPKYL
Shift 19: ILLHRMFKQEBXFOFQPXYFOAFQPXMIXKBFQPPRMBOJXK
Shift 20: HKKGQLEJPDAWENEPOWXENZEPOWLHWJAEPOOQLANIWJ
Shift 21: GJJFPKDIOCZVDMDONVWDMYDONVKGVIZDONNPKZMHVI
Shift 22: FIIEOJCHNBYUCLCNMUVCLXCNMUJFUHYCNMMOJYLGUH
Shift 23: EHHDNIBGMAXTBKBMLTUBKWBMLTIETGXBMLLNIXKFTG
Shift 24: DGGCMHAFLZWSAJALKSTAJVALKSHDSFWALKKMHWJESF
Shift 25: CFFBLGZEKYVRZIZKJRSZIUZKJRGCREVZKJJLGVIDRE
```

最可能的明文是 `LOOK UP IN THE AIR IT'S A BIRD IT'S A PLANE IT'S SUPERMAN`，对应密钥为 16

9 3.5

编写一个程序，实现 2×2 Hill 密码的加解密算法

Answer:

代码:

```python
import numpy as np
import string

def mod_inverse_matrix(matrix, modulus):
    det = int(np.round(np.linalg.det(matrix)))
    det_inv = pow(det, -1, modulus)
    matrix_mod_inv = det_inv * np.round(det * np.linalg.inv(matrix)).astype(int) % modulus
    return matrix_mod_inv

def char_to_num(char):
    return ord(char) - ord('A')

def num_to_char(num):
    return chr(num + ord('A'))

def preprocess_text(text):
    return ''.join(filter(lambda c: c in string.ascii_letters, text)).upper()

def hill_encrypt(plaintext, key_matrix):
    n = key_matrix.shape[0]
    plaintext = preprocess_text(plaintext)
    if len(plaintext) % n != 0:
        plaintext += 'X' * (n - len(plaintext) % n)

    ciphertext = ""
```

```
    for i in range(0, len(plaintext), n):
        block = np.array([char_to_num(c) for c in plaintext[i:i+n]])
        encrypted_block = key_matrix.dot(block) % 26
        ciphertext += ''.join(num_to_char(num) for num in encrypted_block)

    return ciphertext

def hill_decrypt(ciphertext, key_matrix):
    n = key_matrix.shape[0]
    inverse_key_matrix = mod_inverse_matrix(key_matrix, 26)

    plaintext = ""
    for i in range(0, len(ciphertext), n):
        block = np.array([char_to_num(c) for c in ciphertext[i:i+n]])
        decrypted_block = inverse_key_matrix.dot(block) % 26
        plaintext += ''.join(num_to_char(num) for num in decrypted_block)

    return plaintext

key_matrix = np.array([[3, 3], [2, 5]])

plaintext = "Thou to live, thou art alive."
ciphertext = hill_encrypt(plaintext, key_matrix)
decryptedtext = hill_decrypt(ciphertext, key_matrix)

print(f"Plaintext: {plaintext}")
print(f"Ciphertext: {ciphertext}")
print(f"Decrypted Text: {decryptedtext}")
```

结果:

```
Plaintext: Thou to live, thou art alive.
Ciphertext: AVYYVEFKXKAVYYZHFMFKXK
Decrypted Text: THOUTOLIVETHOUARTALIVE
```