实验指导书1 古典密码学

by 李蔚洋

第一次实验较为简单，主要是帮助大家理解古典密码体制，进行简单的加密、解密算法实现。

替代密码（凯撒密码）

Caesar密码，典型的单表古典密码体制。

其加密过程如下：

$$E(m) = (m + k)\ mod\ n$$

其中m为明文字母在字母表中的位置数，n为字母表中的字母个数（26），k为密钥，E（m）为密文字母在字母表中对应的位置数。

可逆推解密过程：

$$D(c) = (c - k)\ mod\ n$$

整个加密、解密的思路都很简单，在具体实现时，我们需要用到我们的老朋友ASCII码来对字符进行处理，这里给出一个表供大家参考：

ASCII可显示字符（共95个）

二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形
0010 0000	32	20	(space)	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(	0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29	)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	–	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s
0011 0100	52	34	4	0101 0100	84	54	T	0111 0100	116	74	t
0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u
0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	v
0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	w
0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	x
0011 1001	57	39	9	0101 1001	89	59	Y	0111 1001	121	79	y
0011 1010	58	3A	:	0101 1010	90	5A	Z	0111 1010	122	7A	z
0011 1011	59	3B	;	0101 1011	91	5B	[	0111 1011	123	7B	{
0011 1100	60	3C	<	0101 1100	92	5C	\	0111 1100	124	7C	|
0011 1101	61	3D	=	0101 1101	93	5D	]	0111 1101	125	7D	}
0011 1110	62	3E	>	0101 1110	94	5E	^	0111 1110	126	7E	~
0011 1111	63	3F	?	0101 1111	95	5F	_				

在设计过程中，我们需要设计两个函数：encrypt() & decrypt()

为了方便起见，我们可以规定输入的字符串全为小写，或者全部转换为小写进行处理。(其实大小写分开处理也很方便，有兴趣的同学可以把这个情况考虑进去)

在密钥的选择上，很明显，k在[0,25]上的取值是有效的，因此在给完k后，我们可以对k做一次模26运算。

encrypt()函数用于加密字符串，输入密钥和明文，输出密文。

我们依次对字符串中的每个字符进行处理：

1. 若字符为' ',则跳过不处理。
2. 若字符不为空，我们要对其进行加密，加密方式参考公式。特别地，我们要判断m+k的ASCII码是否溢出（即数值大于'z'的数值122），若溢出，我们做-26的处理。（当然如果你不直接用ASCII码，你也可以直接取模26）
3. 所有字符处理完成后，输出字符串。

decrypt()函数和encrypt()函数大同小异，只用注意加密解密的差异和溢出的判断不同。

置换密码（矩阵换位法）

直接看例子：

> 明文为attack begins at five，密钥为cipher，将明文按照每行6个字母的形成排在矩阵中，形成如下形式：
>
> a t t a c k
>
> b e g i n s
>
> a t f i v e
>
> 根据密钥cipher中各字母在字母表中出现的先后顺序，给定一个置换：
>
> $$f=\begin{pmatrix}1&2&3&4&5&6\\1&4&5&3&2&6\end{pmatrix}$$
>
> 根据上面的置换，将原有矩阵中字母按照第1，4，5，3，2，6列的顺序排列，则有下面的形式：
>
> a a c t t k
>
> b i n g e s
>
> a i v f t e
>
> 从而得到密文：aba tgf tet cnv aii kse
>
> 其解密的过程是根据密钥的字母数作为列数，将密文按照列、行的顺序写出，再根据由密钥给出的矩阵置换产生新的矩阵，从而恢复明文。

同样地，在设计过程中，我们需要设计两个函数：encrypt() & decrypt()

encrypt()函数用于加密字符串，输入密钥和明文，输出密文。

密钥是长度为n的字符串，我们用它生成置换，密钥cipher中各字母在字母表中出现的先后顺序给定置换。

对于输入的明文，我们按照一排n个生成矩阵，如果最后一排不足，我们规定生成‘abcd...’进行补全。

然后再对于矩阵的每一行做替换，生成密文矩阵，将密文按照列、行的顺序写出。

decrypt()函数和encrypt()函数的思路也非常相似，就是矩阵行列互换，还有进行逆置换。