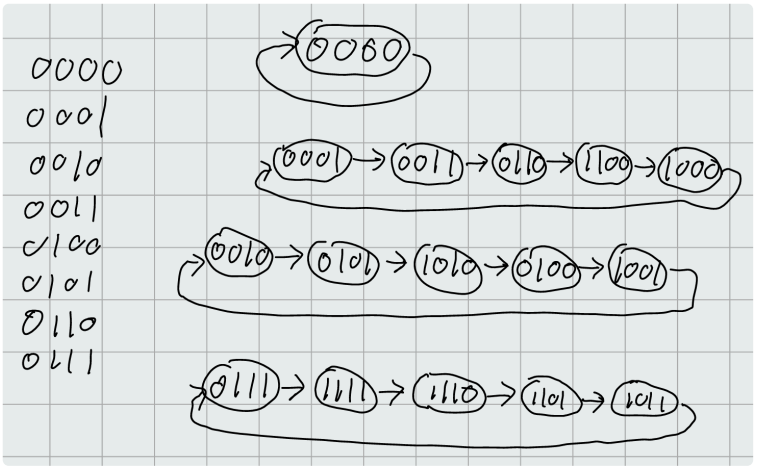
T1

初始向量	周期
0000	1
0001、0011、0110、1100、1000	5
0010、0101、1010、0100、1001	5
0111、1000、1011、1101、1110、1111	5

流程图如下



T2

可能的密码体制是：

1. 移位密码
2. 代换密码
3. 放射密码
4. 维吉尼亚密码
5. 希尔密码
6. 置换密码
7. 流密码

密文如下：

```
BNVSNSIHQCEELSSKKYERIFJKXUMBGYKAMQLJTYAVFBKVT
DVBPVVRJYYLAOKYMPQSCGDLFSRLLPROYGESEBUUALRWXM
MASAZLGLEDFJBZAVVPXWICGJXASCBYEHOSNMULKCEAHTQ
OKMFLEBKFXLRRFDTZXCIWBJSICBGAWDVYDHAVFJXZIBKC
GJIWEAHTTOEWTUHKRQVVRGZBXYIREMMASCSPBNLHJMBLR
FFJELHWEYLWISTFVVYFJCMHYUYRUFSFMGESIGRLWALSWM
NUHSIMYYITCCQPZSICEHBCCMZFEGVJYOCDEMMPGHVAAUM
ELCMOEHVLTIPSUYILVGFLMVWDVYDBTHFRAYISYSGKVSUU
HYHGGCKTMBLRX
```

首先我们分析词频：

L: 23 (占 6.17%)
S: 23 (占 6.17%)
Y: 22 (占 5.90%)
M: 21 (占 5.63%)
V: 21 (占 5.63%)
E: 20 (占 5.36%)
C: 18 (占 4.83%)
A: 17 (占 4.56%)
B: 17 (占 4.56%)
F: 17 (占 4.56%)
G: 16 (占 4.29%)
H: 16 (占 4.29%)
I: 16 (占 4.29%)
R: 15 (占 4.02%)
K: 13 (占 3.49%)
J: 12 (占 3.22%)
T: 12 (占 3.22%)
U: 12 (占 3.22%)
W: 11 (占 2.95%)
D: 9 (占 2.41%)
X: 9 (占 2.41%)
P: 8 (占 2.14%)
O: 7 (占 1.88%)
Z: 7 (占 1.88%)
Q: 6 (占 1.61%)
N: 5 (占 1.34%)

1. 可以发现前五个字母的词频差不多，推测该密码不会将字母一一映射到另一个字母，而是一个字母可能映射到多个字母。从而我们可以排除移位密码、代换密码、仿射密码
2. 由于置换密码不会改变整体的词频，所以它虽然不是一一映射，也可以通过词频排除。
3. 希尔密码和流密码容易受到已知明文攻击，但唯密文攻击似乎较难，且书上没有给它们的唯密文攻击的方法，于是我们可以优先尝试剩下的维吉尼亚密码。

使用 Kasiski 测试法尝试找到密钥长度：

字串	出现次数	首字母位置	差距
AHT	2	[131, 185]	[54]
ASC	2	[115, 211]	[96]
AVF	2	[38, 170]	[132]
BLR	2	[222, 369]	[147]
CGJ	2	[111, 179]	[68]
DVY	2	[165, 339]	[174]
EAH	2	[130, 184]	[54]
EMM	2	[208, 304]	[96]
GES	2	[77, 257]	[180]
MAS	2	[90, 210]	[120]
MBL	2	[221, 368]	[147]
MMA	2	[89, 209]	[120]
SIC	2	[158, 285]	[127]
VVR	2	[49, 198]	[149]
VYD	2	[166, 340]	[174]
WDV	2	[164, 338]	[174]
DVYD	2	[165, 339]	[174]
EAHT	2	[130, 184]	[54]
MBLR	2	[221, 368]	[147]
MMAS	2	[89, 209]	[120]
WDVY	2	[164, 338]	[174]
WDVYD	2	[164, 338]	[174]

经过计算可以知道差距的最大公因数是 1，但是我们可以发现，似乎大部分数都可以被 3 整除，于是排除掉那些不能被 3 整除的。因为有可能确实有两个不同的字符串被

加密成相同的字符串，所以这么做是 ok 的。
进一步观察发现，除了 147，其他都是 6 的倍数，所以 m=6 也是有可能的。

计算重合指数

m = 3 时，字串的重合指数
0.0442, 0.0478,0.0483
m =6 时，字串的重合指数
0.0512, 0.0613,0.0549,0.0708,0.0555,0.0698,
同时尝试了其他 m 的值，最后认为 m=6 时，I_c最接近 0.065

得到密钥

假设 m = 6,
使用 $M_g = \sum_{i=0}^{25} \frac{p_i f_{i+g}}{n'}$，计算出字串的 $M_g(y_i)$ 的值。
从而可以得到密钥为 $K = (19, 7, 4, 14, 17, 24)$

解密

解密后得到

```
IGREWUPAMONGSLOWTALKERSMENINPARTICULARWHODROPPEDWORDSAFEWATATIMELIKEBE
ANSINAHILLANDWHENIGOTTOMINNEAPOLISWHEREPEOPLETOOKALAKEWOBEGONCOMMATOME
ANTHEENDOFASTORYICOULDNTSPEAKAWHOLESENTENCEINCOMPANYANDWASCONSIDEREDNO
TTOOBRIGHTSOIENROLLEDINASPEECHCOURSETAUGHTBYORVILLESANDTHEFOUNDEROFREF
LEXIVERELAXOLOGYASELFHYPNOTICTECHNIQUETHATENABLEDAPERSONTOSPEAKUPTOTHR
EEHUNDREDWORDSPERMINUTE
```

分词后得到

```
I GREW UP AMONG SLOW TALKERS MEN IN PARTICULAR WHO DROPPED WORDS A FEW
AT A TIME LIKE BEANS IN A HILL AND WHEN I GOT TO MINNEAPOLIS WHERE
PEOPLE TOOK A LAKE WOBEGON COMMA TO MEANT THE END OF A STORY I
COULDN'T SPEAK A WHOLE SENTENCE IN COMPANY AND WAS CONSIDERED NOT TOO
BRIGHT SO I ENROLLED IN A SPEECH COURSE TAUGHT BY ORVILLE SAND THE
FOUNDER OF REFLEXIVE RELAXOLOGY A SELF HYPNOTIC TECHNIQUE THAT ENABLED
A PERSON TO SPEAK UP TO THREE HUNDRED WORDS PER MINUTE.
```

翻译后的句子大意：
我在慢语速的讲者中长大，特别是那些每次说话都把话断成一小段一小段的人，就像豆子掉在山坡上一样。等我到了明尼阿波利斯，那里的每个人都以一种方式讲述故事，几乎每次讲话都停顿一下，而我自己在别人面前几乎说不出完整的句子，被认为不太聪明。所以我报名参加了由 Orville Sand 教授的演讲课程，他是反射放松学（Reflexive Relaxology）这一自我催眠技巧的创始人，这个技巧让人能够每分钟讲出三百个单词。

发现是有意义的。

代码见`https://github.com/CraftOldWang/Cryption`