Homework 1

周智 2019011350

—

1 说明

使用的是图灵方法.

代码位于 `enigma.py` 和 `enigma.ipynb` 中, notebook 中同时保留有最后一次的运行结果. 顺序运行 notebook 中的所有单元便能看到在最后以正确的配置解密密文的结果.

2 记号约定

环: 可能正确的叫法是圈.

P_i: 处于位置 i 的 Enigma 中 plugboard 之外的所有变换.

S: Enigma 中 plugboard 带来的变换, 注意 $S = S^{-1}$.

$plain_i$: 给定明文的第 i 位.

$cipher_i$: 给定密文的第 i 位.

3 寻找环

逻辑位于 `enigma.py` 中的 `search_ring` 函数.

由 Enigma 的性质注意到 $P_i^{-1} = P_i$, 故环中无论是对于 $SP_iS(plain_i)$ 还是对于 $SP_i^{-1}S(cipher_i)$ 都只需记录下标 i. 由环的定义知只要能找到一条 $plain_i$ 和 $cipher_i$ 之间的变换路径即找到了一个环, 算法采用了简单的递归实现来做到这一点. 规定算法所搜索的为"最简环", 故规定一个环中同一个 i 不应重复出现, 搜索时维护一已访问过的历史记录, 由此有函数中实现的算法. 结果如下:

```
{'p': [[0, 11, 4], [0, 11, 14, 12, 6], [4, 11, 0], [4, 14, 12, 6], [6, 12, 14, 4], [6, 12, 14, 11, 0]], 'b': [[0, 4, 11], [0, 6, 12, 14, 11], [11, 4, 0], [11, 14, 12, 6, 0]], 'l': [[4, 0, 11], [4, 6, 12, 14], [11, 0, 4], [11, 0, 6, 12, 14], [14, 12, 6, 0, 11], [14, 12, 6, 4]], 'x': [[5, 7, 16, 9, 13], [5, 9, 16, 7, 13], [5, 13], [13, 5], [13, 7, 16, 9, 5], [13, 9, 16, 7, 5]], 'o': [[5, 13], [7, 16, 9], [9, 16, 7], [13, 5]], 'j': [[6, 0, 11, 14, 12], [6, 4, 14, 12], [12, 14, 4, 6], [12, 14, 11, 0, 6]], 'a': [[7, 5, 13, 9, 16], [7, 9, 16], [7, 13, 5, 9, 16], [16, 9, 5, 13, 7], [16, 9, 7], [16, 9, 13, 5, 7]], 'e': [[9, 5, 13, 7, 16], [9, 7, 16], [9, 13, 5, 7, 16], [16, 7, 5, 13, 9], [16, 7, 9], [16, 7, 13, 5, 9]], 'f': [[12, 6, 0, 11, 14], [12, 6, 4, 14], [14, 4, 6, 12], [14, 11, 0, 6, 12]]}
```

4 破译

分析

某个密钥 (在initial position 固定为 `AAA` 的情况下为 rotor 选择和 ring setting) 如可能, 则该配置下的 Enigma 必然满足上述的所有环. 故对上述的每个字母 x 对应的某个环 $\{P_i\}$, 都有 $S(x) = P_n \dots P_1 S(x)$, 此时 x 已知而 $S(x)$ 未知, 枚举 26 个字母即得到这个环中 $S(x)$ 的所有可能构成的集合, 集合为空则说明密钥错误; $S(x)$ 的所有可能均应同时满足 x 的所有环, 也即这一搜索结果的交. 最终如某个密钥下的所有环均可成立, 则说明这组密钥可能正确, 加入列表以待分析; 否则说明这组密钥错误, 可以直接排除.

实现

`enigma.ipynb` 中的 `test_ring`, `test_all_rings` 实现了对环的检查. 变换 P_i 通过模拟特定配置下的 Enigma 并将其设置在特定位置实现. 对 Enigma 的模拟实现于 `enigma.py` 中的 `Enigma` 类. 对总大小为 6×26^3 的密钥空间进行枚举, 通过上述方法搜索出可能的密钥, 同时得到一个 plugboard 的子集. 结果如下:

```
[((0, 2, 1), 'gye', {'p': 'y', 'b': 'w', 'l': 'l', 'x': 'x', 'o': 'k', 'j': 'j', 'a': 'a', 'e': 'e', 'f': 'i'}), ((0, 2, 1), 'hll', {'p': 'w', 'b': 'y', 'l': 'q', 'x': 'c', 'o': 'o', 'j': 'n', 'a': 'r', 'e': 'k', 'f': 'd'}), ((0, 2, 1), 'onh', {'p': 'd', 'b': 'r', 'l': 'v', 'x': 'm', 'o': 'g', 'j': 'c', 'a': 'd', 'e': 'u', 'f': 'm'}), ((2, 1, 0), 'zwl', {'p': 'q', 'b': 'z', 'l': 'g', 'x': 'd', 'o': 'z', 'j': 'c', 'a': 'm', 'e': 'n', 'f': 'w'})]
```

筛选

此时的密钥空间已经小到能手动处理. 将密钥和搜索出的 plugboard 一并应用于 Enigma, 解密密文按顺序得到:

```
plain : bhuilopalopbjxfce
enigma: bhqilopalopbjxfce

plain : bhuilopalopbjxfce
enigma: bommlopadopbjxfje

plain : bhuilopalopbjxfce
enigma: bkadloaayozbjffve

plain : bhuilopalopbjxfce
enigma: oipglopafoqojxfoe
```

发现第一个密钥最接近, 猜测还有一个 plugboard 连接为 `Q/U`, 设置后得到正确结果. 由此得到正确的密钥与完整的 Enigma 配置: plugboard 为 `P/Y, B/W, O/K, F/I, Q/U`, 转子顺序为 `I-III-II`, ring setting 为 `G-Y-E`.

事后能够看出如果原文的形式较为特殊导致没有环或环的数量不多, 则会使得密钥空间大大上升.

二

1.6

$26 = 2 \times 13$, 显然有 $k_1 = 0, k_2 = 13$.

1.7

$n = m\ \phi(m)$, 由欧拉函数定义有如下计算:

$30 = 2 \times 3 \times 5, \phi(30) = (2-1)(3-1)(5-1) = 8$, 密钥空间大小为 240.

$100 = 2^2 \times 5^2, \phi(100) = (4-2)(25-5) = 40$, 密钥空间大小为 4000.

$1225 = 5^2 \times 7^2, \phi(1225) = (25-5)(49-7) = 840$, 密钥空间大小为 1029000.

1.21

a

代码及运行结果位于 `a.ipynb`.

其密文为

> *EMGLOSUDCGDNCUSWYSFHNSFCYKDPUMLWGYICOXYSIPJCKQPKUGKMGOLICGINCGACKSNISACYKZSCKXECJCKSHYS*
> *XCGOIDPKZCNKSHICGIWYGKKGKGOLDSILKGOIUSIGLEDSPWZUGFZCCNDGYYSFUSZCNXEOJNCGYEOWEUPXEZGACGN*
> *FGLKNSACIGOIYCKXCJUCIUZCFZCCNDGYYSFEUEKUZCSOCFZCCNCIACZEJNCSHFZEJZEGMXCYHCJUMGKUCY*

对其进行词频分析得到

```
Counter({'C': 37, 'G': 24, 'S': 20, 'K': 18, 'Y': 15, 'I': 15, 'U': 14, 'N': 13, 'Z': 13, 'E': 12,
'O': 10, 'F': 9, 'D': 8, 'L': 7, 'X': 7, 'J': 7, 'P': 6, 'M': 5, 'W': 5, 'H': 5, 'A': 5, 'Q': 1})
```

猜测 `C` 为 `e`, `GSK` 应分别为 `taoinshr` 中的几个. 又已知 `F` 为 `w`, 对其进行长度为 1-4 的词频分析发现 `FZCC`, `ZCCN`, `FZC`, `ZCC`, `CCN` 有着完全一样的出现频率, 推测为单一词根 `FZCCN -> wZeeN`. 进行更长的子串统计发现 `FZCCNDGYYSF -> wZeeNDGYYSw` 重复出现, 应为完整单词. 借助词库得知 `FZCCN -> wheel` 也即 `Z=h, N=l`, 由上述词汇确定为 `wheel`, `DGYYSw -> barrow`. 此时代换完的密文串为:

> *EMaLOoUbeableUoWrowHlowerKbPUMLWarIeOXroIPJeKQPKUaKMaOLIeaIleaAeKolIoAerKhoeKXEeJeKoHroXeaOIbP*
> *KhelKoHIeaIWraKKaKaOLboILKaOIUoIaLEboPWhUawheelbarrowUohelXEOJlearEOWEUPXEhaAealwaLKloAeIaOIreK*
> *XeJUeIUhewheelbarrowEUEKUheoOewheeleIAehEJleoHwhEJhEaMXerHeJUMaKUer*

已有的代换为 `C=e, F=w, D=b, G=a, Y=r, S=o, N=l, Z=h`. 观察开头发现 `beableUoWrowHlower` 其中前三个词猜测应为 `be able to`, 从而 `U=t`; 观察发现 `aOI` 出现的次数和 `OI` 相同且很高, 而在更长的子串中基本不出现, 推测其为独立词汇 `and`, 从而 `O=n, I=d`. 从 `wheelbarrow` 推测文章在讲花园或种植相关的主题, 从而推测 `Wrow -> grow`, `Hlower -> flower`, 此时 `Warden -> garden` 印证了这一推测, 有 `W=g, H=f`. 从 `GSK` 对应的推测来看猜测 `K=s`, `graKK -> grass` 印证了此猜测. 此时代换完的密文串为:

> *EMaLnotbeabletogrowflowersbPtMLgardenXrodPJesQPstasManLdeadleaAesoldoAershoesXEeJesofroXeandbPshelsofdeadgrassasanLbodLsandtodaLEboPghtawheelbarrowtohelXEnJlearEngEtPXEhaAealwaLsloAedandresXeJtedthewheelbarrowEtEstheonewheeledAehEJleofwhEJhEaMXerfeJtMaster*

注意到尾段, 猜测其分词为 `the wheelbarrow EtEs the one wheeled`, 此处如猜测 `EtEs` 为 `it is` 则十分合理, 故 `E=i`. 此时猜测开头为 `I MaL not be able to grow flowers bPt ML garden...`, 猜测 `ML -> my, MaL -> may`, 语义上猜测 `bPt -> but`, 从而有 `M=m, L=y, P=u`. 回到尾段, 猜测有分词 `it is the one wheeled AehiJle of whiJh i am XerfeJt master`, 猜测 `AehiJle -> vehicle, XerfeJt -> Xerfect`, 有 `A=v, J=c`. 从此时的 `Xroduces`, `Xieces`, `roXe` 猜测 `X=p`. 由此只剩下 `Qust`, 猜测为 `just`, 有 `Q=j`. 全明文如下:

> *imaynotbeabletogrowflowersbutmygardenproducesjustasmanydeadleavesoldovershoespiecesofropeandbushelsofdeadgrassasanybodysandtodayiboughtawheelbarrowtohelpinclearingitupihavealwayslovedandrespectedthewheelbarrowitistheonewheeledvehicleofwhichiamperfectmaster*

b

代码及运行结果位于 `b.ipynb`.

以 3 和 4 的子串长度分别进行 Kasiski 测试, 发现 `2, 3, 6` 最有可能是密钥长度, `11, 22` 其次. 对这些长度使用重合指数法验证, 有:

```
[0.038461538461538464, 0.04726261762189906] when m = 2, avg = 0.042862078041718765
[0.055941845764854614, 0.04858429858429859, 0.04826254826254826] when m = 3, avg = 0.05092956420390049
[0.06265664160401002, 0.08506493506493507, 0.04935064935064935, 0.06493506493506493,
0.04285714285714286, 0.07337662337662337] when m = 6, avg = 0.06304017619807094
[0.034408602150537634, 0.05591397849462366, 0.043010752688172046, 0.034408602150537634,
0.030107526881720432, 0.03655913978494624, 0.03870967741935484, 0.059770114942528735,
0.06206896551724138, 0.03218390804597701, 0.03908045977011494] when m = 11, avg = 0.04238379344052314
[0.058333333333333334, 0.1, 0.041666666666666664, 0.05, 0.016666666666666666, 0.058333333333333334,
0.041666666666666664, 0.12380952380952381, 0.01904761904761905, 0.0380952380952381,
0.047619047619047616, 0.01904761904761905, 0.047619047619047616, 0.0761904761904762,
0.02857142857142857, 0.01904761904761905, 0.047619047619047616, 0.0380952380952381,
0.009523809523809525, 0.10476190476190476, 0.06666666666666667, 0.05714285714285714] when m = 22, avg
= 0.050432900432900434
```

能够看出来 `m = 6` 时均值最接近 0.065, 各项的值之间也较接近, 可较可靠地猜测密钥长度为 6. 计算所有 y_i 的 M_g, 有:

```
substring y_0: substitution: c; max M_g: 0.06463157894736843
substring y_1: substitution: r; max M_g: 0.07019642857142858
substring y_2: substitution: y; max M_g: 0.058732142857142865
substring y_3: substitution: p; max M_g: 0.06600000000000002
substring y_4: substitution: t; max M_g: 0.055785714285714286
substring y_5: substitution: o; max M_g: 0.07042857142857144
```

可猜测密钥为 `CRYPTO`, 解密得到明文为:

> *ilearnedhowtocalculatetheamountofpaperneededforaroomwheniwasatschoolyoumultiplythesquarefootageofthewallsbythecubiccontentsofthefloorandceilingcombinedanddoubleityouthenallowhalfthetotalforopeningssuchaswindowsanddoordthenyouallowtheotherhalfformatchingthepatternthenyoudoublethewholethingagaintogiveamarginoferrorandthenyouorderthepaper*

c

代码及运行结果位于 `c.ipynb`.

对其进行词频统计得到:

```
Counter({'C': 32, 'B': 21, 'K': 20, 'P': 20, 'I': 16, 'E': 13, 'A': 13, 'R': 12, 'F': 10, 'D': 9, 'J': 6, 'U': 6, 'Q': 4, 'Z': 4, 'V': 4, 'O': 2, 'X': 2, 'H': 1, 'N': 1, 'Y': 1, 'S': 1})
```

猜测 `C=e, B=t`, 得方程组 ($\mathrm{mod}\ 26$ 意义下):

$$\begin{cases} 4a + b \equiv 2 \\ 19a + b = 1 \end{cases} \tag{4}$$

解得 $a=19, b=4$ 为合法密钥, 对应的解密函数为 $d(y)=11(y-4) \mod 26=11y+8 \mod 26$, 可得明文为:

> *ocanadaterredenosaieuxtonfrontestceintdefleuronsglorieuxcartonbrassaitporterlepeeilsaitporterlacroixtonhistoireestuneepopeedesplusbrillantsexploitsettavaleurdefoitrempeeprotegeranosfoyersetnosdroits*

虽然能看出有意义的音节但明显不是英文, 询问同学得知是法语.

d

代码及运行结果位于 `d.ipynb`.

将之前的各方法都试验一下, 发现词频统计没有给出一个合理的分布, Kasiski 和重合指数法指示密钥长度可能为 6, 计算 M_g 有:

```
substring y_0: substitution: t; max M_g: 0.06098412698412699
substring y_1: substitution: h; max M_g: 0.06906451612903225
substring y_2: substitution: e; max M_g: 0.06109677419354839
substring y_3: substitution: o; max M_g: 0.06824193548387096
substring y_4: substitution: r; max M_g: 0.06319354838709677
substring y_5: substitution: y; max M_g: 0.06454838709677418
```

是一个合理的有意义的密钥 `THEORY`, 解密得到明文:

> *igrewupamongslowtalkersmeninparticularwhodroppedwordsafewatatimelikebeansinahillandwhenigottominneapoliswherepeopletookalakewobegoncommatomeantheendofastoryicouldntspeakawholesentenceincompanyandwasconsiderednottoobriahtsoienrolledinaspeechcouqsetaughtbyorvillesandthefounderofreflexiverelaxologyaselfhypnotictechniquethatenabledapersontospeakuptothreehundredwordsperminute*

1.24

求密钥也即求出 L 和 $b, m=3$, 共有 12 个未知数, 共可列出 18 个方程, 故列了解出来就好. 计算时将 L 的行和 b 中对应的位置组合成一个线性方程组, 有 ($\mathrm{mod} 26$ 意义下):

$$
\begin{cases}
\begin{bmatrix} 0 & 3 & 8 & 1 \\ 18 & 15 & 11 & 1 \\ 0 & 24 & 4 & 1 \\ 3 & 4 & 16 & 1 \end{bmatrix} \begin{bmatrix} l_{11} \\ l_{21} \\ l_{31} \\ b_1 \end{bmatrix} = \begin{bmatrix} 3 \\ 12 \\ 14 \\ 23 \end{bmatrix} \\
\begin{bmatrix} 0 & 3 & 8 & 1 \\ 18 & 15 & 11 & 1 \\ 0 & 24 & 4 & 1 \\ 3 & 4 & 16 & 1 \end{bmatrix} \begin{bmatrix} l_{12} \\ l_{22} \\ l_{32} \\ b_2 \end{bmatrix} = \begin{bmatrix} 18 \\ 18 \\ 15 \\ 11 \end{bmatrix} \\
\begin{bmatrix} 0 & 3 & 8 & 1 \\ 18 & 15 & 11 & 1 \\ 0 & 24 & 4 & 1 \\ 3 & 4 & 16 & 1 \end{bmatrix} \begin{bmatrix} l_{13} \\ l_{23} \\ l_{33} \\ b_3 \end{bmatrix} = \begin{bmatrix} 17 \\ 8 \\ 11 \\ 9 \end{bmatrix}
\end{cases} \tag{5}
$$

解得:

$$
L=\begin{bmatrix} 3 & 6 & 4 \\ 5 & 15 & 18 \\ 17 & 8 & 5 \end{bmatrix}, \vec{b}=\begin{bmatrix} 8 \\ 13 \\ 1 \end{bmatrix} \tag{6}
$$

1.26

1

将密文写成 $n \times m$ 的形式后依次取各列得明文.

2

将密文长度 42 进行质因数分解, 考虑以其因数作为 m/n, 直接对密文进行解密得到:

> *MARRYQECOARYDOEURGENGYMAUITNTRHOWSYOARDROW*
> *MMRIETAODYUREOYRUQNOHYSEAGRGAAYTCRRWOORDNW*
> *MREADUEYUNHSARAYCRORNMITOYRORQOYEGGATRWODW*
> *MUCOEDYYTYOEAIOWURMQRDANRTASROAEHORGRNRYGW*
> *MCEYTOAOUMRARARAHRRRGUODYYEIWRQDNTSOEOGNYW*
> *MOYYAWMDRSAORYUEYOIUQATRERNGCDTEORRNAOHGRW*

没有观察到有意义的结果，考虑到在第一块中得到了 `MARRY` 这一有意义的单词，怀疑是分块加密. 因第一块的存在，`n=2`，只需实验块的数量. 遍历，发现在 `n=2, m=3`，块数量为 7 时，得到了有意义的结果为明文:

> *MARYMARYQUITECONTRARYHOWDOESYOURGARDENGROW*