Лабораторна робота з криптографії №3

Виконали: Дорош Анастасія та Шатковська Діана ФБ-92

Варіант №1

Мета: засвоєння понять ентропії на символ джерела та його надлишковості, вивчення та порівняння різних моделей джерела відкритого тексту для наближеного визначення ентропії, набуття практичних навичок щодо оцінки ентропії на символ джерела.

Хід роботи

Хід роботи можно умовно розілити на частини, результати роботи над якими були представлені нижче. Спершу було підготовано математичні функції знаходження НСД, розв'язку лінійних порівнянь, а також обрахунок числових значень біграм, що були використані у наступних частинах лабораторної. Далі було реалізовано алгоритм дешифрування тексту за варіантом. Також у виконанні знадобилися функції, що були реалізовані у ході виконання попереднідніх лабораторних робіт, які виявились досить універсальними у використанні. Найскладнішим завданням у виконанні лабораторної стала реалізація функції підбору пари ключів a,b , що також, напевно, є основним кроком у дешифруванні афінного шифру.

Частина 1

Реалізувати підпрограми із необхідними математичними операціями: обчисленням оберненого елементу за модулем із використанням розширеного алгоритму Евкліда, розв’язуванням лінійних порівнянь. При розв’язуванні порівнянь потрібно коректно обробляти випадок із декількома розв’язками, повертаючи їх усі.

```
def gcd(a, b):
  p = [0,1]
  gcd_val = b
  a, b = max(a,b), min(a,b)

  while (b != 0):
    q = a // b
    gcd_val = b
    a, b = b, a % b
    p.append(p[-1]*(-q)+p[-2])
```

```
  return gcd_val, p[ 2]
def linear(a, b, n):
  a = a%n
  b = b%n

  d, a_re = gcd(a, n)

  if d == 1:
    x = (a_re*b) % n

    return [x]
  else:
    if b%d == 0:
      solutions = []
      solutions_loc = []

      a1 = a/d
      b1 = b/d
      n1 = n/d

      solutions_loc = linear(a1, b1, n1)

      for i in range(0, d):
        for x0 in solutions_loc:
          res = x0+(d-i)*n1
          solutions.append(res%n)

      return sorted(solutions)
    else:
      return None
```

Частина 2

За допомогою програми обчислення частот біграм, яка написана в ході виконання комп'ютерного практикуму No1, знайти 5 найчастіших біграм запропонованого шифртексту (за варіантом).

```
# stuff to get the file
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

```
cd /content/drive/MyDrive/5 sem/Crypto/lab3
```

```
/content/drive/MyDrive/5 sem/Crypto/lab3
```

```
def clean_text(txt):
    with open(txt, 'r', encoding='utf-8') as file:
```

```
        text = file.read().lower()

    chars = '.71()-«5d?[“!93286”…—4;»0:],naoti*IVXvxse—'
    for ch in chars:
        text = text.replace(ch, '')

    text = '_'.join([word.strip('\n') for word in text.split()])
    text = ''.join([word.strip('\n') for word in text.split('_')])

    with open('01_clean.txt', 'w', encoding='utf-8') as file:
        file.write(text)

clean_text("01_utf8.txt")
```

```
def open_file(txt):
    with open(txt, 'r', encoding='utf-8') as file:
        text = file.read().lower()
    return text

text = open_file("01_clean.txt")
```

```
from collections import Counter
import operator
from itertools import islice

# from lab1
def count_bi_nointersect(text):
    res = Counter(text[idx: idx + 2] for idx in range(0, (len(text)), 2))
    total_bi = sum(res.values())
    res = {x: round(res[x]/total_bi, 10) for x in res}
    return dict(res)

main_count = count_bi_nointersect(text)

# sort
main_count_sorted = dict(sorted(main_count.items(), key=operator.itemgetter(1),reve

# get the first 5 items
n_items = dict(islice(main_count_sorted.items(), 5))
print(n_items)
```

```
{'рн': 0.0253682488, 'ыч': 0.0167757774, 'нк': 0.0139116203, 'цз': 0.0130932897
```

Частина 3

Перебрати можливі варіанти співставлення частих біграм мови та частих біграм шифртексту (розглядаючи пари біграм із п’яти найчастіших). Для кожного співставлення знайти можливі кандидати на ключ (a,b) шляхом розв’язання системи (1).

```

```

```
alphabet_full = 'абвгдежзийклмнопрстуфхцчшщьыэюя'
alphabet = list(alphabet_full)
top_bi = ('ст', 'но', 'то', 'на', 'ен')

# determine bigram's value
def bi_val(bi):
  val = alphabet.index(bi[0])*len(alphabet) + alphabet.index(bi[1])
  return val

# getting all possible pairs of bigrams cyphered-real
def all_pairs(cyphered, real):
  res = []
  for el1 in cyphered:
    for el2 in real:
      res.append((el2, el1))

  res2 = []
  for p1 in res:
    for p2 in res:
      if p2[0] not in p1 and p2[1] not in p1:
        res2.append((p1,p2))

  return res2

# finding mutual key from a pair of bigrams cyphered-real
def find_key(pair1, pair2):  # (y1, x1) (y2, x2)
  y1, x1 = bi_val(pair1[0]), bi_val(pair1[1])
  y2, x2 = bi_val(pair2[0]), bi_val(pair2[1])

  a = linear(y1-y2,x1-x2,len(alphabet)**2)
  b = []
  if a != None:
    for el in a:
      b.append((x1-(y1*el))%len(alphabet)**2)

    return a[0], b[0]
```

Частина 4

Для кожного кандидата на ключ дешифрувати шифртекст. Якщо шифртекст не є змістовним текстом російською мовою, відкинути цього кандидата.

```
forbidden_lst = ('аъ', 'аь', 'бй', 'бф', 'гщ', 'гъ', 'еъ', 'еь', 'жй', 'жц', 'жщ', 'ж

def decipher(a, b, text):
  plaintext = ''
  n = len(alphabet)
  # get inverse of a
  a_inv = gcd(a, n**2)[1]

  # go through bigrams
  for bi in [text[idx: idx + 2] for idx in range(0, (len(text)), 2)]:
```

```
    bi_idx = bi_val(bi)
    deciph_bi = ((bi_idx - b) * a_inv) % (n**2)

    x2_idx = deciph_bi % n
    x1_idx = (deciph_bi - x2_idx) / n

    x1 = alphabet[int(x1_idx)]
    x2 = alphabet[int(x2_idx)]
    if x1+x2 not in forbidden_lst:
      plaintext += x1 + x2
    else:
      return False

  return plaintext
```

Частина 5

Повторювати дії 3-4 дoти, доки дешифрований текст не буде змістовним.

```
allp = all_pairs(n_items.keys(),top_bi)
for p in allp:
  try:
    a,b = find_key(p[0], p[1])
    res = decipher(a, b, text)
    if res != False:
      print("Keys: ", a, b)
      print(res)
      break
  except TypeError:
    pass
```

```
Keys:  13 151
многограннуюличностьдостоевскогоможнорассматриватьсчетырехсторонкакписателякакневротикакакмысли
```

Висновки

В процесі виконання роботи були засвоєні поняття ентропії на символ джерела та його надлишковості, вивчені різних моделей джерела відкритого тексту для наближеного визначення ентропії, набуття практичних навичок щодо оцінки ентропії на символ джерела.