Лабораторна робота 3 з Симетричної Криптографії

Варіант: 2

Команда: Бондар, Кістаєв

Група: ФІ-03

Підготовча частина: оголошення констант, функція для зчитування і передобробки тексту

In [ ]:
```
# Constants
A1 = "абвгдежзийклмнопрстуфхцчшщыьэюя"
A2 = "абвгдежзийклмнопрстуфхцчшщьыэюя"
ALPH = A2
m = len(ALPH)
M = m*m

FIVE_MOST_FREQUENT = ['ст', 'но', 'то', 'на', 'ен']

RING = {}
BIGRAM_RING = {}
BIGRAMS = ['aa'] * (M)

# Precompute transformations
for char in ALPH:
    pos = ALPH.index(char)
    RING[char] = pos

for i in ALPH:
    for j in ALPH:
        pos_i = ALPH.index(i)
        pos_j = ALPH.index(j)
        idx = pos_i * m + pos_j
        BIGRAM_RING[i+j] = idx
        BIGRAMS[idx] = i+j

PROBS = [0.0792, 0.0171, 0.0433, 0.0174, 0.0305, 0.0841, 0.0105, 0.0175,
         0.0683, 0.0112, 0.0336, 0.0500, 0.0326, 0.0672, 0.1108, 0.0281,
         0.0445, 0.0533, 0.0618, 0.0280, 0.0019, 0.0089, 0.0036, 0.0147,
         0.0081, 0.0037, 0.0002, 0.0194, 0.0038, 0.0061, 0.0213]
EXPECTED_I = sum([(p ** 2) for p in PROBS])

TEST_TEXT = "test_text.txt"
ENCRYPTED_TEXT = "encrypted_text.txt"
```

In [ ]:
```
# Text preprocessing
def transform_symbol(_c):
    if 'а' <= _c and _c <= 'я':
        return _c
    elif _c <= 'Я' and _c >= 'А':
        return _c.lower()
    elif _c == 'Ё' or _c == 'ё':
        return 'е'
    else:
        return ""

def preprocess_text(_text):
    text_formatted = ""
    # Change symbols according to requirements
    for c in _text:
        text_formatted += transform_symbol(c)
    # Remove consequtive spaces
    text_formatted = ' '.join(text_formatted.split())
    return text_formatted

def read_text(filename):
    f = open(filename, "r", encoding='utf-8')
    text = f.read()
    f.close()
    return preprocess_text(text)
```

Додатковий функціонал для атаки: PAE, розв'язування лінійних порівнянь

In [ ]:
```python
def extended_euclidian(a, b):
    r_0, r_1 = a, b
    u_0, u_1 = 1, 0 # coef at a
    v_0, v_1 = 0, 1 # coef at b

    # Parallel computation of remainders and coefs
    while r_1 != 0:
        q = r_0 // r_1
        r_0, r_1 = r_1, r_0 - q * r_1   # same as r_1, r_0 % r_1
        u_0, u_1 = u_1, u_0 - q * u_1
        v_0, v_1 = v_1, v_0 - q * v_1

    # print(f"gcd({a}, {b}) = {r_0} = {u_0}*{a} + {v_0}*{b}")
    return (r_0, u_0, v_0)
```

In [ ]:
```python
# ax = b (mod n) -> list of possible solutions
def solve_linear_congruence(a, b, n):
    a %= n
    b %= n
    # print(f"{a}x = {b} (mod {n})")
    (d, a_coef, _) = extended_euclidian(a, n) # Check gcd(a, n)
    if d == 1:
        # then a^-1 = a_coef
        return [(a_coef * b) % n]
    elif b % d == 0:
        a_1, b_1, n_1 = a // d, b // d, n // d
        (_, a_1_inv, _) = extended_euclidian(a_1, n_1)
        x_0 = (a_1_inv * b_1) % n_1
        return [x_0 + k * n_1 for k in range(d)]
    else:
        return []
```

Аналіз шифротексту: підрахунок біграм, знаходження n найчастіших біграм у тексті.

При застосуванні алгоритму, використовуватимемо n = 5.

In [ ]:
```python
# Bigrams without intersection (ex: [1, 2], [3, 4])
def count_bigrams_wo_i(_text):
    b_count = {}
    i = 1
    while i < len(_text):
        bg = _text[i - 1] + _text[i]
        if bg not in b_count:
            b_count[bg] = 1
        else:
            b_count[bg] = b_count[bg] + 1
        i = i + 2
    # Sort by amount reversed
    return dict(sorted(b_count.items(), key=lambda pair: -pair[1]))

def get_best_n_bigrams(_text, n):
    bigrams_count = count_bigrams_wo_i(_text)
    return list(bigrams_count)[:n:]
```

Отримання ключа, дешифрування тексту за допомогою цього ключа

In [ ]:
```python
def get_keys(best_5):
    keys = []
    for i_1 in range(5):
        for j_1 in range(5):
            for i_2 in range(5):
                for j_2 in range(5):
                    if i_1 == i_2 or j_1 == j_2:
                        continue

                    X_1 = BIGRAM_RING[FIVE_MOST_FREQUENT[i_1]]
```

```
                    Y_1 = BIGRAM_RING[best_5[j_1]]
                    X_2 = BIGRAM_RING[FIVE_MOST_FREQUENT[i_2]]
                    Y_2 = BIGRAM_RING[best_5[j_2]]

                    A = solve_linear_congruence(X_1 - X_2, Y_1 - Y_2, M)
                    keys += [(a, (Y_1 - a * X_1) % M) for a in A]
    return keys
```

In [ ]:
```
def decrypt_bigram(encr, a_inv, b):
    Y = BIGRAM_RING[encr]
    return BIGRAMS[a_inv * (Y - b) % M]

def try_decrypt_text(_text, key):
    (a, b) = key
    (d, a_inv, _) = extended_euclidian(a, M)
    if d != 1:
        return "invalid key!"

    res = ""
    for i in range(1, len(_text), 2):
        encr = _text[i - 1] + _text[i]
        res += decrypt_bigram(encr, a_inv, b)

    return res
```

Перевірка тексту на змістовність

Ми скористались критеріями найчастіших та найрідших символів, а також критерієм індексу відповідності.

In [ ]:
```
def count_chars(_text):
    c_count = {}
    for c in _text:
        if c not in c_count:
            c_count[c] = 1
        else:
            c_count[c] = c_count[c] + 1

    return list(sorted(c_count.items(), key=lambda pair: pair[1]))

# Function to evaluate index of coincidence
def coincidence_index(text):
    sum = 0
    for c in ALPH:
        occurances = text.count(c)
        sum += occurances * (occurances - 1)
    return sum / (len(text) * (len(text) - 1))
```

In [ ]:
```
def rate_text(_text, complexity=5) -> int:
    char_freqs = count_chars(_text)
    k = 4
    rating = 0.0

    # Check top k freq
    top_n_sym = [char_freqs[-i - 1][0] for i in range(k)]
    if top_n_sym.count('o') != 0:
        rating += 1
    if top_n_sym.count('e') != 0:
        rating += 1
    if top_n_sym.count('a') != 0:
        rating += 1

    if complexity <= 1:
        return rating

    # Check bot k freq
    bot_n_sym = [char_freqs[i][0] for i in range(k)]
    if bot_n_sym.count('ф') != 0:
        rating += 1
    if bot_n_sym.count('ц') != 0:
        rating += 1
    if bot_n_sym.count('щ') != 0:
        rating += 1

    if complexity <= 2:
        return rating
```

```
    # Check index of coincidence
    delta_idx = abs(coincidence_index(_text) - EXPECTED_I) * 200
    rating -= delta_idx

    if complexity <= 3:
        return rating

    return rating
```

Побудова атаки

Експериментальним шляхом ми з'ясували, що тестовий текст над алфавітом А1, а текст задачі над А2.

Зчитуємо текст і знаходимо топ-5 біграм за частотою

In [ ]:
```
text = read_text(ENCRYPTED_TEXT)
best_5 = get_best_n_bigrams(text, 5)

best_5
```

Out[ ]:
```
['йа', 'юа', 'чш', 'юд', 'рщ']
```

Маючи топ-5 біграм, знаходимо можливі ключі і сортуємо їх за оцінкою змістовності ВТ

In [ ]:
```
# Get keys and remove duplicates
keys = list(dict.fromkeys(get_keys(best_5)))
keys_rated = []
for key in keys:
    open_text = try_decrypt_text(text, key)
    if open_text == "invalid key!":
        continue
    keys_rated.append((key, rate_text(open_text)))
sorted_by_rate = sorted(keys_rated, key=lambda entry: -entry[1])

print("Top 10 by rate:")
for (key, rate) in sorted_by_rate[:10]:
    print(f"{rate} : {key}")
```

```
Top 10 by rate:
5.761652298790493 : (27, 211)
3.204202418051068 : (895, 552)
3.0969305455384237 : (554, 521)
2.1501026298198402 : (926, 118)
2.0375574927503606 : (244, 56)
2.007691935769523 : (647, 180)
1.965017554078102 : (151, 397)
1.894984043621128 : (709, 273)
1.861863385061997 : (399, 769)
1.8282381857582308 : (895, 211)
```

Текст, отриманий при використанні найбільш ймовірного ключа

In [ ]:
```
result_text = try_decrypt_text(text, sorted_by_rate[0][0])
result_text
```

Out[ ]: 'однакоэтакартинаскакойбысторонымыеенирассматривалирасплываетсявнечтонеопределенноеприпадкипроявляющиесярезкос
прикусываниемусиливающиесядоопасногодляжизниприводящегоктяжкомусамокалечениюмогутвсежевнекоторыхслучаяхнедости
гатьтакойсилыослабляясьдократкихсостоянийабсансадобыстропроходящихголовокруженийимогуттакжесменятьсякраткимипе
риодамикогдабольнойсовершаетчуждыеегоприродепоступкикакбынаходясьвовластибессознательногообуславливаясьвобщемк
акбыстранноэтониказалосьчистотелесныминричинамиэтисостоянямогутпервоначальновозникатьпоричинамчистодушевными
спугилимогутвдальнейшемнаходитьсявзависимостиотдушевныхволненийкакнихарактернодляогромногобольшинстваслучаевин
теллектуальноеснижениеноизвестенпокрайнеймереодинслучайкогдаэтотнедугненарушилвысшейинтеллектуальнойдеятельнос
тигельмгольцдругиеслучаивотношениикоторыхутверждалосьтожесамоененадежныилиподлежатсомнениюкакислучайсамогодост
оевскоголицастрадающиеэпилепсиеймогутпроизводитьвпечатлениетупостинедоразвитоститкакэтаболезньчастосопряжена
сярковыраженнымидиотизмомикрупнейшимимозговымидефектаминеявляяськонечнообязательнойсоставнойчастьюкартиныболез
ниноэтиприпадкисовсемисвоимивидоизмененямибываютиудругихлицулицсполнымдушевнымразвитиемискореесосверхобычнаяв
большинствеслучаевнедостаточноуправляемойимиаффективностьюнеудивительночтопритакихобстоятельствахневозможноуст
ановитьсовокупностьклиническоюаффектаэпилепсииточтопроявляетсяводнородностиуказанныхсимптомовтребуетповидимому
функционального опониманиякакеслибымеханизманормального говысвобожденияпервичныхпозывовбылподготовленорганическимех
анизмкоторыйиспользуетсяприналичиивесьмаразныхусловийкакпринарушениимозговойдеятельностипритяжкомзаболеваниитк
анейилитоксическомзаболеваниитакипринедостаточномконтроледушевнойэкономиикризисномфункционированиидушевнойэнер
гиизаэтимразделениемнадвавидамычувствуемндентичностьмеханизмалежащеговосновевысвобожденияпервичныхпозывовэтотм
еханизмнедалекиотсексуальныхпроцессовпорождаемыхвсвоейосноветоксическиужедревнейшиеврачиназываликоитусмалойэпи
лепсиейивиделивполовомактесмягченииеиадаптациювысвобожденияэпилептическогоотводараздраженияэпилептическаяреакци
якаковымименемможноназватьвсеэтовместевзятоенесомненнотакжепоступаетивраспоряжениеневрозасущностькотороговтомч
тобыликвидироватьсоматическимассыраздраженияскоторыминеврознеможетсправитьсяпсихическиэпилептическийприпадокст
ановитсятакимобразомсимптомомистериииюадаптируетсяивидоизменяетсяподобнотомукакэтопроисходитпринормальномтече
ниисексуальногопроцессатакимобразоммысполнымправомразличаеморганическуюиаффективнуюэпилепсиюпрактическоезначен
иеэтогоследующеестрадающийпервойпораженболезньюмозгастрадающийвторойневротиквпервомслучаедушевнаяжизньподверже
нанарушениюизвневовторомслучаенарушениеявляетсявыражениемсамойдушевнойжизнивесьмавероятночтоэпилепсиядостоевск
огоотносится ковторомувидуточнодоказатьэтонельзятаккаквтакомслучаенужнобылобывключитьвцелокупностьегодушевнойжи
зниначалоприпадковипоследующиевидоизмененияэтихприпадковадляэтогоунаснедостаточноданныхописаниясамихприпадковн
ичегонедаютсведенияосоотношенияхмеждуприпадкамиипереживанияминеполныичастопротиворечивывсеговероятнеепредполож
ениечтоприпадкиначалисьудостоевскогоужевдетствечтоонивначалехарактеризовалисьболееслабымисимптомамиитолькопосл
епотрясшегоегопереживаниянавосемнадцатомгодужизниубийстваотцапринялиформуэпилепсиибылобывесьмауместноеслибыопр
авдалосьточтоониполностьюпрекратилисьвовремяотбыванияимкаторгивсибириноэтомупротиворечатдругиеуказанияочевидна
ясвязьмеждуотцеубийствомвбратьяхкарамазовыхисудьбойотцадостоевскогобросилась вглазанеодномубиографудостоевского
ипослужилаимуказаниемнаизвестноесовременноепсихологическоенаправлениепсихоанализтаккакподразумеваетсяименноонс
клоненвидетьвэтомсобытиитягчайшуютравмуивреакциидостоевскогонаэтоключевойпунктегоневрозаеслияначнуобосновывать
этуустановкупсихоаналитическиопасаюсьчтоокажусьнепонятнымдлявсехтехкомунезнакомыучениеивыраженияпсихоанализаун
асодиннадежныйисходныйпункт намизвестенсмыслпервыхприпадковдостоевскоговегоюношескиегодызадолгодопоявленияэпиле
псииуэтихприпадковбылоподобиесмертиониназывалисьстрахомсмертиивыражалисьвсостоянииелетаргическогоснаэтаболезньн
аходилананеговначалекогдаонбылещемальчикомкаквнезапнаябезотчетнаяподавленностьчувствокаконпозжерассказывалсвое
мудругусоловьевутакоекакбудтобыемупредстоялосейчасжеумеретьивсамомделенаступалосостояниесовершенноподобноедейс
твительнойсмертиегобратандрейрассказывалчтофедоружевмолодыегодыпередтемкакзаснутьоставлялзапискичтобоитсяночью
заснутьсмертоподобнымсномипроситпоэтомучтобыегопохоронилитолькочерезпятьднейдостоевскийзарулеткойвведениеснами
звестнысмыслинамерениетакихприпадковсмертиониозначаютотождествлениесумершимчеловекомкоторыйдействительноумерил
исчеловекомживымещенокоторомумыжелаемсмертивторойслучайболеезначителенприпадокуказанномслучаеравноценнаказа
ниюмыпожелалисмертидругомутеперьмысталисамиэтимдругимисамиумерлитутпсихоаналитическоеучениеутверждаетчтоэтотдр
угойдлямальчикаобычноотецименуемыйистериейприпадокявляетсятакимобразомсамонаказаниемзапожеланиесмертиненавист
номуотцуа'