Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Комп’ютерний практикум №3

з дисципліни «КРИПТОГРАФІЯ»

Тема: «Криптоаналіз афінної біграмної підстановки»

Виконали:

студенти групи ФБ-34

Гузік Андрій
Кувавіна Софія
Нікітчук Дмитрій
Синельник Максим

Київ - 2025

Мета роботи: Набуття навичок частотного аналізу на прикладі розкриття моноалфавітної підстановки; опанування прийомами роботи в модулярній арифметиці.

Постановка задачі:
Реалізувати математичні підпрограми:
- Обчислення оберненого елементу за модулем із використанням розширеного алгоритму Евкліда.
- Розв’язання лінійних порівнянь. Необхідно коректно обробляти випадок із декількома розв’язками, повертаючи їх усі.
Частотний аналіз:
- Знайти 5 найчастіших біграм у запропонованому шифртексті.
Пошук кандидатів на ключ:
- Перебрати можливі варіанти співставлення частих біграм мови та частих біграм шифртексту.
- Для кожного співставлення знайти можливі кандидати на ключ шляхом розв’язання системи лінійних порівнянь.
Дешифрування та фільтрація:
- Для кожного кандидата на ключ дешифрувати шифртекст.
- Якщо дешифрований текст не є змістовним текстом потрібною мовою (використовуючи автоматичний розпізнавач), відкинути цього кандидата.
Результат:
- Повторювати дії доти, доки дешифрований текст не буде змістовним.

Варіанти виконання завдання: 6, 11, 13, 17

Для виконання роботи було розроблено скрипт мовою python. Програма автоматично зчитує файл, проводить частотний аналіз, перебирає можливі ключі, розв'язуючи системи лінійних порівнянь, та оцінює змістовність отриманого тексту.

Скрипт:

```
import collections

import os



ALPHABET = 'абвгдежзийклмнопрстуфхцчшщьыэюя'

M = 31

M_SQ = M * M

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



RARE_BIGRAMS = ["щт","ьо","ыж","юв","яы","аы","бй","гй","дй","еы",

                "шщ","шя","щб","щд","щж","ьы","ыа","ыь","ыы","ыэ"]



def extended_gcd(a, b):

    if a == 0: return b, 0, 1

    g, y, x = extended_gcd(b % a, a)
```

```
    return g, x - (b // a) * y, y

def mod_inverse(a, m):
    g, x, y = extended_gcd(a, m)
    if g != 1: return None
    return (x % m + m) % m

def solve_linear_congruence(a, b, m):
    g, x, y = extended_gcd(a, m)
    if b % g != 0: return []
    x0 = (x * (b // g)) % (m // g)
    return [x0 + i * (m // g) for i in range(g)]

def clean_text(text):
    text = text.lower()
    text = text.replace('ё', 'е').replace('ъ', 'ь')
    return "".join([c for c in text if c in ALPHABET])

def bigram_to_int(bg):
    return ALPHABET.index(bg[0]) * M + ALPHABET.index(bg[1])

def int_to_bigram(val):
    first = ALPHABET[val // M]
    second = ALPHABET[val % M]
    return first + second

def get_top_bigrams_from_text(text, n=5):
    bigrams = [text[i:i+2] for i in range(0, len(text)-1, 2) if len(text[i:i+2])==2]
    return [item[0] for item in collections.Counter(bigrams).most_common(n)]

def decrypt_text(ciphertext, a, b):
```

```
    a_inv = mod_inverse(a, M_SQ)
    if a_inv is None: return ""
    plaintext = []
    for i in range(0, len(ciphertext)-1, 2):
        bg = ciphertext[i:i+2]
        Y = bigram_to_int(bg)
        X = (a_inv * (Y - b)) % M_SQ
        plaintext.append(int_to_bigram(X))
    return "".join(plaintext)

def score_text(text):
    bgs = [text[i:i+2] for i in range(0, len(text), 2)]
    return sum(bg in RARE_BIGRAMS for bg in bgs)

def main():
    filename = input("Введіть назву файлу: ").strip() or "cipher6.txt"

    if not os.path.exists(filename):
        print("Помилка: Файл не знайдено.")
        return

    with open(filename, 'r', encoding='utf-8') as f:
        cipher_clean = clean_text(f.read())

    top_cipher = get_top_bigrams_from_text(cipher_clean, 5)
    print(f"\n5 найчастіших біграм шифртексту: {', '.join(top_cipher)}")

    candidates = set()
    for i in range(len(TOP_LANG)):
        for j in range(len(TOP_LANG)):
            if i == j: continue
```

```
        X1, X2 = bigram_to_int(TOP_LANG[i]), bigram_to_int(TOP_LANG[j])

        for k in range(len(top_cipher)):
            for l in range(len(top_cipher)):
                if k == l: continue
                Y1, Y2 = bigram_to_int(top_cipher[k]), bigram_to_int(top_cipher[l])

                diff_X, diff_Y = (X1 - X2) % M_SQ, (Y1 - Y2) % M_SQ
                possible_as = solve_linear_congruence(diff_X, diff_Y, M_SQ)

                for a in possible_as:
                    if extended_gcd(a, M)[0] == 1:
                        b = (Y1 - a * X1) % M_SQ
                        candidates.add((a, b))

print(f"Знайдено можливих кандидатів на ключ: {len(candidates)}")

results = []
for a, b in candidates:
    decrypted = decrypt_text(cipher_clean, a, b)
    if decrypted:
        s = score_text(decrypted)
        results.append((s, decrypted, a, b))

results.sort(key=lambda x: x[0], reverse=False)

print("\n--- ТОП-5 ВАРІАНТІВ ---")
for i in range(min(5, len(results))):
    score, text, a, b = results[i]
    print(f"{i+1}. Ключ ({a}, {b}) [Штраф: {score}]: {text[:60]}...")
```

```
    if results:
        best_score, best_text, best_a, best_b = results[0]
        print("\n--- ПОВНИЙ РОЗШИФРОВАНИЙ ТЕКСТ (Найкращий варіант) ---")
        print(f"Ключ: a={best_a}, b={best_b}")
        print("-" * 50)
        print(best_text)
    else:
        print("\nНе вдалося розшифрувати текст.")


if __name__ == "__main__":
    main()
```

Скрипт працює за таким алгоритмом роботи:

Програма зчитує шифротекст і розбиває його на блокові біграми без перетинів. Далі обчислюється частота кожної пари та виділяється топ-5 найчастіших біграм шифру.

Наступним кроком запускається перебір варіантів: кожна з 5 найчастіших біграм мови по черзі співставляється з кожною з 5 найчастіших біграм шифру.

Для кожної такої пари пар формується система лінійних порівнянь:

$$\begin{cases} Y^{*} \equiv aX^{*} + b \pmod{m^2} \\ Y^{**} \equiv aX^{**} + b \pmod{m^2} \end{cases}, \qquad (1)$$

Програма розв'язує цю систему відносно невідомих a та b. Для знаходження a використовується розширений алгоритм Евкліда для розв'язання лінійного порівняння.

Знайдені пари (a, b) перевіряються на коректність: число a має бути взаємно простим із 31 (найбільший спільний дільник має дорівнювати 1), інакше дешифрування неможливе.

Усі коректні пари (a, b) зберігаються у списку кандидатів.

Після цього програма проходить по списку знайдених кандидатів. І для кожного ключа (a, b) виконується пробне розшифрування всього тексту за формулою:

$$X_i = a^{-1}(Y_i - b) \bmod m^2$$

Отриманий текст одразу передається на аналіз функції оцінювачу score_text, яка розбиває текст на біграми і підраховує кількість входжень біграм зі списку заборонених. Рейтинг тексту дорівнює кількості знайдених заборонених біграм (чим менше це число, тим краще).

Усі варіанти сортуються за зростанням рейтингу. Варіант із найменшою кількістю заборонених біграм визнається змістовним текстом і виводиться як результат розшифрування.

Опис роботи автоматичного розпізнавача російської мови:

Для відбору правильного тексту використано критерій заборонених біграм. У російській мові існують пари літер, які ніколи не стоять поруч, наприклад: ьы, щт, ыж. Програма перевіряє кожен варіант розшифрованого тексту і підраховує кількість таких неможливих пар. У результаті текст із найменшою кількістю заборонених біграм вважається змістовним і правильним.

Результати варіанту 6:

Відрізок шифротексту:
ывлеюгзебщпещхщуйэвиывиюфгувхцубхщыюнюжлепэшфмиьхдощбуднзегдщебоцвшуюгьпцвэщу
вкмзеиэбчиюндхщюасдбмонхегщгдэщжезьщемвощфысьмайыегыйя

```
5 найчастіших біграм шифртексту: ще, хе, чв, ле, цв
Знайдено можливих кандидатів на ключ: 348

--- ТОП-5 ВАРІАНТІВ ---
1. Ключ (441, 310) [Штраф: 4]: утробылотихоегородокутанныйтьмоймирнонежилсявпостелипришлоле...
2. Ключ (534, 930) [Штраф: 13]: дтроцылощихочгцредюкдтынгыщттмщйуимнкнджылуяжпьснетичрсшложе...
3. Ключ (348, 341) [Штраф: 16]: атноиыиоиитопггрфдыкатбнфыхтамайвиснпнгжслмяьпясфебидрышионе...
4. Ключ (595, 538) [Штраф: 17]: шоыяяйряцчзящмшцяиыпшофгнйвоязсупчтгьгшаяльэаьгтшдучфцгхряхд...
5. Ключ (242, 675) [Штраф: 19]: зошюиудюпенюпрвжэфпнзоощщуюосхзйцерщищиэясдаивккхштежжмздюшш...

--- ПОВНИЙ РОЗШИФРОВАНИЙ ТЕКСТ (Найкращий варіант) ---
Ключ: a=441, b=310
--------------------------------------------------
утробылотихоегородокутанныйтьмоймирнонежилсявпостелипришлолетоиветербыллетнийтеплоедыханиемиранесп
```

Результати варіанту 11:

Відрізок шифротексту:
оквкпкящсройюфчвфбчллфэйлзщоыифуххгьижфбчбжройэжиавкхбоаэлбзьздблфюжвпыхожеуфыхь
фисццоисццикшхгтчьюбрйэунемкщхлфэсццикоэйсыфляьэсблавххуоаебвщвцззабюжэйзэсюфцхцчь
двкьбивцкьхбвхщзфийамсьэхьжофшнйсбгежоэзхбнннкндхбххьюэкублфйлщзхкгсебэуяффдзэсццоь

```
5 найчастіших біграм шифртексту: нк, юж, хб, шь, мк
Знайдено можливих кандидатів на ключ: 220

--- ТОП-5 ВАРІАНТІВ ---
1. Ключ (703, 956) [Штраф: 5]: хорошосэрбиллнехотясунулденьгивкарманвотчтобиллвыпростопосее...
2. Ключ (845, 564) [Штраф: 14]: зрхрлрмоийймсщыпслкфзщгмбднббиохоюьтцаслюльйймйавзхрхльзяфчд...
3. Ключ (635, 373) [Штраф: 18]: бнщнянйоябхптшуюпащлчшшпеучепсщьоцммахпайахбхпцхэвщннарвплщу...
4. Ключ (36, 370) [Штраф: 23]: ихтхвхчшрруыччнзлнепзчэыюихэпвьююсутыолншнфруыаобутхеныужпыи...
5. Ключ (951, 336) [Штраф: 24]: шоуоыощэхбхлзньхотчспнблжезььиикскркащвотчтубхлчвжпуостшпзсзе...

--- ПОВНИЙ РОЗШИФРОВАНИЙ ТЕКСТ (Найкращий варіант) ---
Ключ: a=703, b=956
--------------------------------------------------
хорошосэрбиллнехотясунулденьгивкарманвотчтобиллвыпростопосеетеэтуновуютравукогданибудьвдругойразкактолькояпом
тдатьконцыужбудьтеувереныподождусказалбиллсамнезнаюкаквамобяснитьнодляменяжужжаньеэтойкосилкисамаяпрекраснаям
куяпошелковрагувыславныйюношаивсепонимаетеяуверенизвасполучитсяблестящийиумныйрепортерсказалдедушкапомогаяему
```

Результати варіанту 13:

Відрізок шифротексту:

дюэорэдюэорэтнтфоэлкшэунскынайцбюовыюежэмюшыафткьэапжнечсюкэфэгябаейпблрщкбсяфий
пкдчаясюлюхэитрэшуафюэмпсьфэпбщзскынафвфюэбэеыыпыфьркэяфщюхэнфкймюмфыфтфрюсю
ьэфээбжооафэьякюфььчтвлэйзцндюткяхмфяюкпнюузнонмынюбжалюхэзящпызхбврэвьдоанфхэкю

```
5 найчастіших біграм шифртексту: аф, яф, дю, ап, нф
Знайдено можливих кандидатів на ключ: 306

--- ТОП-5 ВАРІАНТІВ ---
1. Ключ (99, 60) [Штраф: 3]: раннееранneеутропервыеотсветызаринакрышезаокномвселистьянаде...
2. Ключ (316, 866) [Штраф: 12]: цайнаецайнаедтрокеявцеятавхтязирднпкжыуенаюкноьвметивтэяуаяе...
3. Ключ (812, 618) [Штраф: 14]: наснуенаснуейтлоюеьвкедтывьтчзирмнактызедаокиоцваеюизтбякате...
4. Ключ (347, 432) [Штраф: 14]: оаднсеоаднсештроыебвиеуттвктьзтрянькйыееeайкноэвюеуицтпяларе...
5. Ключ (257, 452) [Штраф: 22]: чпргсичпргсикыцогибжлисымжлыуякечгпбсвйиспобфоужщигкуырдхпжи...

--- ПОВНИЙ РОЗШИФРОВАНИЙ ТЕКСТ (Найкращий варіант) ---
Ключ: a=99, b=60
--------------------------------------------------
раннееранneеутропервыеотсветызаринакрышезаокномвселистьянадеревьяхвздрагиваютотзываясьнамалейшеедуновение
ыхколесахяркооранжевыйкакмандариннанемэполетымерцающеймедиизолотойкантпроводовижелтыйзвонокгромкозвякает
```

Результати варіанту 17:

Відрізок шифротексту:

ккщлпжатвкофааощпкррнькбхнььшсрцдфтжщзляжахеунелцвдьсмунэкшщжмпеунзмздввелбмярьяф
всщпктсыщмыьбхеикфкцвэвцяюфудмппьткьяйммящщуздмввгкгмдсрцикопюьрбпебыткнькбэыюен
вскылмщевдчзвжяоасфтмнщмаймибикуьлублхывмнщмаймибикопмфушпебыяпждлщшчмяюьмхщ

```
5 найчастіших біграм шифртексту: вк, нв, ья, юв, пк
Знайдено можливих кандидатів на ключ: 262

--- ТОП-5 ВАРІАНТІВ ---
1. Ключ (470, 312) [Штраф: 12]: борисзаэтовремясвоейслужбыблагодарязаботаманнымихайловнысобс...
2. Ключ (512, 313) [Штраф: 21]: гоюрюжнятоггцщонноцьипзынфгпгэязогджоиучгщодйффрыюхпйтйфиогн...
3. Ключ (47, 96) [Штраф: 22]: гоыржжэятооглщпннокьзпщызфвпфэжзщгмжфикччщшдгфтрмюфпетгфиодн...
4. Ключ (77, 637) [Штраф: 32]: эоумыцюкономщщгшптотврньбзихнчаухбщхцэфленгщшлиэмцягнщкличодп...
5. Ключ (729, 158) [Штраф: 38]: гобрсжжятовгчщэннойьщплыцффпшэдзнгчжеисчдщядтфчргюзпптгфиосн...

--- ПОВНИЙ РОЗШИФРОВАНИЙ ТЕКСТ (Найкращий варіант) ---
Ключ: a=470, b=312
--------------------------------------------------
борисзаэтовремясвоейслужбыблагодарязаботаманнымихайловнысобственнымвкусамисвойствамсвоегосдержанногох
июитолькочтовозвратилсяоттудакурьеромонвполнеусвоилсебетупонравившуюсяемувольмюценеписаннуюсубординац
стоянствоанужнобылотолькоуменьеобращатьсястемикоторыевознаграждаютзаслужбуиончастосамудивлялсясвоимбь
```

Висновки:

У ході виконання лабораторної роботи було успішно вирішено задачу автоматизованого криптоаналізу афінного шифру біграмної заміни та відновлено зміст чотирьох зашифрованих повідомлень.

В результаті було експериментально підтверджено, що афінний шифр біграмної заміни, хоч і приховує статистику окремих символів, залишається вразливим до частотного аналізу біграм. Для успішного злому виявилося достатнім використати інформацію лише про 5 найчастіших біграм мови (ст, но, то, на, ен).

Реалізація алгоритмів теорії чисел (розширений алгоритм Евкліда, пошук оберненого елемента, розв'язання лінійних порівнянь) дозволила звести задачу криптоаналізу до розв'язання системи лінійних рівнянь у кільці лишків за модулем m^2

Для автоматичного відбору правильного ключа серед сотень кандидатів було використано метод мінімізації заборонених біграм. Цей підхід показав високу ефективність і дозволив безпомилково ідентифікувати змістовний текст, відсіявши варіанти з текстовим шумом.