НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

з лабораторної роботи №*2*

з дисципліни «Криптографічні методи захисту інформації»

на тему:

*Розробка криптосистем на основі шифру Віженера та шифру з використанням двійкового гамування*

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-01 | *ст. викладач Бай Ю. П.* |
| *Боженко А. О.* |  |

Київ — 2022

ЗМІСТ

[ЗАВДАННЯ 2](#_Toc100472356)

[Основні теоретичні відомості з шифру Віженера 3](#_Toc100472357)

[Завдання з шифру Віженера 4](#_Toc100472358)

[Основні теоретичні відомості з шифру двійкового гамування 5](#_Toc100472359)

[Завдання з шифру двійкового гамування 6](#_Toc100472360)

[Список літератури 7](#_Toc100472361)

[Додаток 1 8](#_Toc100472362)

[Додаток 2 9](#_Toc100472363)

# ЗАВДАННЯ

***Мета роботи:*** розробити криптосистеми на основі шифрів Віженера та двійкового гамування.

1. Обрати початковий текст англійською, українською або російською мовою довжиною від 8 до 20 символів, та ключове слово від 3 до 8 символів. Зашифрувати текст, користуючись шифром Віженера. Алфавіт, криптотекст та ключ записати в [Таблицю\_3](https://docs.google.com/spreadsheets/d/1Fqu6vkzaJYX1mQPbSPU4A1CSVkOK_MHggPLZTmA6gzA/edit?usp=sharing) в СВІЙ РЯДОК у стовпчики AF, AG, AH. Перед шифруванням пробіли з початкового тексту необхідно видалити. (*1 бал*).

2. Користуючись наданим ключем, розшифрувати отриманий криптотекст за алгоритмом Віженера (завдання дивитись в [Таблиці\_3](https://docs.google.com/spreadsheets/d/1Fqu6vkzaJYX1mQPbSPU4A1CSVkOK_MHggPLZTmA6gzA/edit?usp=sharing) у рядку НАД СВОЇМ). Відповідь записати в рядок НАД СВОЇМ у стовпчик AI. (*1 бал*)

3. Написати програмний код для шифрування - розшифрування за шифром Віженера. Продемонструвати роботу створеної програми на контрольних прикладах. (*2 бали*)

4. Обрати початковий текст англійською, українською або російською мовою довжиною від 8 до 15 символів (ім'я відомої особистості, відома назва пісні чи групи, комп'ютерна гра, тощо). Записати символи початкового тексту у двійковому вигляді, користуючись кодовою таблицею сp866.

5. Згенерувати ключ - псевдовипадкову гамму, довжина якої дорівнює або більша довжини початкового тексту. Записати ключ у двійковому вигляді в [Таблицю\_4](https://docs.google.com/spreadsheets/d/1TzJmWgV5XCRY0FRKcvfvVlcJ3Yhk3TgyJc02ud0XWRo/edit?usp=sharing) в СВІЙ рядок, стовпчик AG. Значення ключа повинно бути унікальним в стовпчику AG. (*1 бал*)

6. **Шифрування:** для початкового тексту та згенерованого ключа виконати побітово операцію XOR. Отриманий криптотекст у двійковому вигляді записати в [Таблицю\_4](https://docs.google.com/spreadsheets/d/1TzJmWgV5XCRY0FRKcvfvVlcJ3Yhk3TgyJc02ud0XWRo/edit?usp=sharing) в СВІЙ рядок, стовпчик AH. (*1 бал*)

7. **Розшифрування** (завдання дивитись в [Таблиці\_4](https://docs.google.com/spreadsheets/d/1TzJmWgV5XCRY0FRKcvfvVlcJ3Yhk3TgyJc02ud0XWRo/edit?usp=sharing) у рядку НАД СВОЇМ): користуючись наданим ключем, розшифрувати отриманий криптотекст за алгоритмом XOR. Відповідь записати в рядок НАД СВОЇМ: у стовпчик AI - у двійковому вигляді, у стовпчик AJ - у вигляді тексту згідно з таблицею cp866. (2 *бали*)

8. Написати програмний код для шифрування - розшифрування за шифром двійкового гамування. Продемонструвати роботу створеної програми на прикладах. (*1 бал*)

9. Оформити звіт. (*1 бал*)

# Основні теоретичні відомості з шифру Віженера

Шифр Віженера  — [поліалфавітний шифр](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%B0%D0%BB%D1%84%D0%B0%D0%B2%D1%96%D1%82%D0%BD%D0%B8%D0%B9_%D1%88%D0%B8%D1%84%D1%80), який як [ключ](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87) використовує слово.

Якщо пронумерувати літери алфавіту від 0 до 32 (а → 0, б → 1, в → 2, …), то шифрування Віженера можна подати формулою:

Ci = (Pi + Kj) mod 33,

де Kj — j-та літера [ключового](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87) слова, Pi  — і-а літера вихідного слова.

Ключове слово повторюється, поки не отримано [гаму](https://uk.wikipedia.org/w/index.php?title=%D0%93%D0%B0%D0%BC%D0%B0(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D1%96%D1%8F)&action=edit&redlink=1), рівну довжині повідомлення.

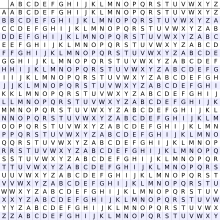
Дешифрування відбувається за наступною формулою:

Ci = (Pi + 33 - Kj) mod 33

Криптоаналіз виконується так:

1. Знаходиться довжина [ключа](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87). [Шифротекст](https://uk.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82) розбивається на групи по номеру літери ключового слова.

2. За допомогою [частотного аналізу](https://uk.wikipedia.org/wiki/%D0%A7%D0%B0%D1%81%D1%82%D0%BE%D1%82%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F)) груп знаходяться літери [ключа](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87).



# Завдання з шифру Віженера

1. Обране речення для шифрування: «WHATAWONDERFULWORLD»

Ключ: «SUNRIZE»

Користуючись формулою для шифрування: Ci = (Pi + Kj) mod 33, де Kj — j-та літера [ключового](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87) слова, Pi  — і-а літера вихідного слова, отримано зашифрований текст: «OBNKIOSFXRINMPOIECL»

1. Отриманий шифротекст: «dmsbdwymmztxd»

Ключ: «kivi»

Користуючись формулою для дешифрування: Ci = (Pi + 33 - Kj) mod 33, де Kj — j-та літера [ключового](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87) слова, Pi  — і-а літера вихідного слова, отримано розшифрований текст: «TEXTTODECRYPT»

# Основні теоретичні відомості з шифру двійкового гамування

Для утворення шифротексту повідомлення об'єднується операцією [XOR](https://uk.wikipedia.org/wiki/XOR) з [ключем](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D1%96%D1%8F)) . При цьому ключ повинен мати три критично важливі властивості:

1. Бути справді випадковим;
2. Збігатися за розміром з заданим відкритим текстом;
3. Застосовуватися тільки один раз;
4. Повинен зберігатися в повній таємниці сторонами, що спілкуються.

Для роботи шифру необхідно утворити ключ, розміром як саме повідомлення, з дійсно [випадкових чисел](https://uk.wikipedia.org/wiki/%D0%92%D0%B8%D0%BF%D0%B0%D0%B4%D0%BA%D0%BE%D0%B2%D1%96_%D1%87%D0%B8%D1%81%D0%BB%D0%B0).

[Ключ](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87) використовує ключове [слово](https://uk.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B2%D0%BE) та може бути записаний формулою

Ci = Pi [XOR](https://uk.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%BB%D1%8E%D1%87%D0%BD%D0%B0_%D0%B4%D0%B8%D0%B7%27%D1%8E%D0%BD%D0%BA%D1%86%D1%96%D1%8F) Kj.

де Kj - j-та літера ключового слова представлена в кодуванні [ASCII](https://uk.wikipedia.org/wiki/ASCII).

Ключове слово повторюється поки не отримано гаму, рівну довжині повідомлення.

Метод [шифрування](https://uk.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) полягає в застосуванні операції XOR до кожної пари бітів ключа та повідомлення, представлених у двійковому виді. Для дешифрування тексту необхідно виконати операцію XOR до пари бітів шифротексу та ключа, представлених у двійковому виді.

# Завдання з шифру двійкового гамування

1. Обраний текст для шифрування: «HIGHHOPES»

Для закодування в двійковий код тексу обрано формат “cp866”.

Двійкове представлення тексу:

01001000 01001001 01000111 01001000 01001000 01001111 01010000 01000101 01010011

Згенерований двійковий гамма-ключ з псевдовипадковими числами:

01111001 01000010 10111101 11110010 00100001 00000110 11110000 10000100 01110111

На основі XOR операції здійснено шифрування:

01001000 01001001 01000111 01001000 01001000 01001111 01010000 01000101 01010011

01111001 01000010 10111101 11110010 00100001 00000110 11110000 10000100 01110111

00110001 00001011 11111010 10111010 01101001 01001001 10100000 11000001 00100100

Зашифрований в двійковому представленні текст: 00110001 00001011 11111010 10111010 01101001 01001001 10100000 11000001 00100100

1. Отриманий двійковий криптотекс: 10011000 00000001 00101011 01111000 01111011 11011101 11010010 00000011 11010000 00110100 01001011 00110101 00010001 01001111

Двійковий ключ: 11111011 01110011 01010010 00001000 00001111 10110010 10110001 01110110 10100010 01000110 00101110 01011011 01110010 00110110

Розшифрований текст: 01100011 01110010 01111001 01110000 01110100 01101111 01100011 01110101 01110010 01110010 01100101 01101110 01100011 01111001

Розшифруймо текст за допомогою операції XOR (двійковий ключ з закодованим повідомленням):

10011000 00000001 00101011 01111000 01111011 11011101 11010010

11111011 01110011 01010010 00001000 00001111 10110010 10110001

01100011 01110010 01111001 01110000 01110100 01101111 01100011

00000011 11010000 00110100 01001011 00110101 00010001 01001111

01110110 10100010 01000110 00101110 01011011 01110010 00110110

01110101 01110010 01110010 01100101 01101110 01100011 01111001

У текстовому вигляді (розшифроване повідомлення): cryptocurrency

# Список літератури

1. Тарнавський Ю.А. Технології захисту інформації [Електронний ресурс] / Ю. А. Тарнавський. – Київ: КПІ ім. Ігоря Сікорського, 2018. – 162 с.
2. <https://uk.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%92%D1%96%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B0>
3. <https://uk.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_XOR>

# Додаток 1

Текст програми, що реалізує криптосистему Віженера

def eng\_alpha():

    start = ord('A')

    # 26 in the following formula is length of english alphabet

    return [chr(el) for el in range(start, start + 26)]

def ukr\_alpha():

    alpha\_string = "а, б, в, г, ґ, д, е, є, ж, з, и, і, ї, й, к, л, м, н, о, п, р, с, т, у, ф, х, ц, ч, ш, щ, ь, ю, я".upper()

    return [el for el in alpha\_string if el != ',' and el != ' ']

def ru\_alpha():

    alpha\_string = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"

    return [el for el in alpha\_string]

def table(alpha):

    for i in range(len(alpha)):

        for j in range(len(alpha)):

            if j == len(alpha) - 1:

                print(alpha[(i + j) % len(alpha)])

            else:

                print(alpha[(i + j) % len(alpha)] + " ", end="")

def vigenere\_encode(text, key\_word, alpha):

    encoded\_text = []

    for position in range(len(text)):

        key\_letter = key\_word[position % len(key\_word)]

        encoded\_letter = alpha[(alpha.index(text[position]) + alpha.index(key\_letter)) % len(alpha)]

        encoded\_text.append(encoded\_letter)

    return ''.join(encoded\_text)

def vigenere\_decode(text, key\_word, alpha):

    decoded\_text = []

    for position in range(len(text)):

        key\_letter = key\_word[position % len(key\_word)]

        decoded\_letter = alpha[(alpha.index(text[position]) - alpha.index(key\_letter)) % len(alpha)]

        decoded\_text.append(decoded\_letter)

    return ''.join(decoded\_text)

alphabet = eng\_alpha()

print(alphabet)

string = "WHATAWONDERFULWORLD"

key = "SUNRISE"

encoded = vigenere\_encode(string, key, alphabet)

print(encoded)

# print(vigenere\_decode("dmsbdwymmztxd".upper(), "kivi".upper(), alphabet))

# table(alphabet)

# Додаток 2

Текст програми, що реалізує криптосистему з використанням двійкового гамування

import random

ENCODING = 'CP866'

def key\_in\_bytes(rand\_seed, text):

  seed = random.seed(rand\_seed)

  return bytes(random.randint(0, 255) for char in text)

def string\_repres\_of\_bytes(byte\_object):

  return ' '.join(["{0:08b}".format(x) for x in byte\_object])

def encode\_vernam(text, rand\_seed, encoding\_f, logging = True):

  bytes\_text = text.encode(encoding\_f)

  bytes\_key = key\_in\_bytes(rand\_seed, text)

  bytes\_ciphered\_text = bytes(i ^ j for i, j in zip(bytes\_text, bytes\_key))

  if (logging):

    print(f"str\_bytes\_text = {string\_repres\_of\_bytes(bytes\_text)}")

    print(f"str\_bytes\_key = {string\_repres\_of\_bytes(bytes\_key)}")

    print(f"str\_bytes\_ciphered\_text = {string\_repres\_of\_bytes(bytes\_ciphered\_text)}")

  return [bytes\_ciphered\_text, bytes\_key]

def decode\_vernam(ciphered\_text, cipher\_key, encoding\_f, logging = True):

  decoded\_text = bytes(i ^ j for i, j in zip(ciphered\_text, cipher\_key))

  if (logging):

    print(f"str\_bytes\_deciphered\_text: {string\_repres\_of\_bytes(decoded\_text)}")

  return decoded\_text.decode(encoding\_f)

def str\_bytes\_to\_bytes(str):

  return bytes(int(i, 2) for i in str.split())

# Enciphering

# -----------

# message = "HIGHHOPES"

# ciphered = encode\_vernam(message, 3, ENCODING)

# ciphered\_t = ciphered[0]

# key = ciphered[1]

# deciphered = decode\_vernam(ciphered\_t, key, ENCODING)

# Deciphering

# -----------

key\_str = "1111001 1000010 10111101 11110010 100001 110 11110000 10000100 1110111"

enciphered\_str = "110001 1011 11111010 10111010 1101001 1001001 10100000 11000001 100100"

print('-' \* len("01001000 01001001 01000111 01001000 01001000 01001111 01010000 01000101 01010011"))

key = str\_bytes\_to\_bytes(key\_str)

enciph = str\_bytes\_to\_bytes(enciphered\_str)

enciphered = decode\_vernam(enciph, key, ENCODING)

print(enciphered)