**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу**

**Кафедра системного проектування**

Лабораторна робота №3

з дисципліни

“Безпека інформаційних систем”

Виконала:

студентка групи ДА-81

Желєзнова Валерія

# Київ – 2021**Мета роботи**

Дослідження криптосистеми шифрування даних RSA за допомогою програмних засобів CryptTool та методики її практичної реалізації.

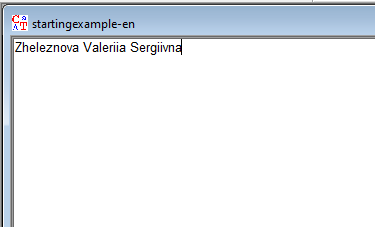
**Завдання**

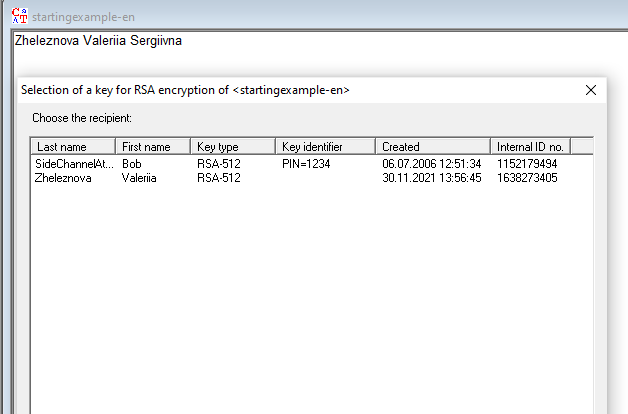
1. Зашифрувати вручну своє прізвище, ім’я та по-батькові, написані великими та малими буквами англійського алфавіту з урахуванням пробілів, за допомогою шифру RSA. Для цього за допомогою пакету CrypTool вибрати параметри криптосистеми RSA з розрахунком, щоб можна було виконати шифрування і дешифрування вручну за допомогою інженерного калькулятора. При кодуванні букв використовувати кодову таблицю ASCII, параметри p і q криптосистеми RSA вибрати в межах від 27 до 28 . Виконати вручну дешифрування отриманого шифротексту.
2. Виконати шифрування і дешифрування свого прізвища, імені та по-батькові за допомогою засобів пакету CrypTool. Зберегти отримані результати шифрування і дешифрування у відповідні файли за допомогою засобів пакету CrypTool. Порівняти результати ручного та автоматичного шифрування і дешифрування.
3. Вибрати варіант текстового файлу (див. Додаток до лабораторної роботи № 1) відповідно до порядкового номера студенту в списку академічної групи. За допомогою засобів пакету CrypTool зашифрувати і дешифрувати обраний файл шифром RSA з параметрами, які використовувалися при виконанні п.2. Порівняти вихідний текст з дешифрованим. Зробити висновки. Результати зберегти.
4. За допомогою засобів аналізу пакету CrypTool провести частотний аналіз вихідного і зашифрованого тексту. Визначити ентропію і максимально можливу ентропію зашифрованого файлу, а також побудувати гістограму розподілу частоти символів в аналізованих файлах. Зробити висновки про можливість криптоаналізу зашифрованого тексту з використанням тільки шифротексту. Дослідити можливості пакета CrypTool для злому шифру RSA.
5. Написати і налагодити програму для шифрування і дешифрування файлів за допомогою шифру RSA використовуючи готові бібліотеки (наприклад  System.Security.Cryptography в середовищі .NET Security Framework). Зашифрувати текстовий файл, вибраний з додатку до лабораторної роботи при виконанні п.3, за допомогою розробленої програми. Порівняти результати шифрування, отримані за допомогою пакета CrypTool, з результатами роботи власної програми. Зробити висновки про коректність роботи програми.
6. Оформити звіт по роботі.

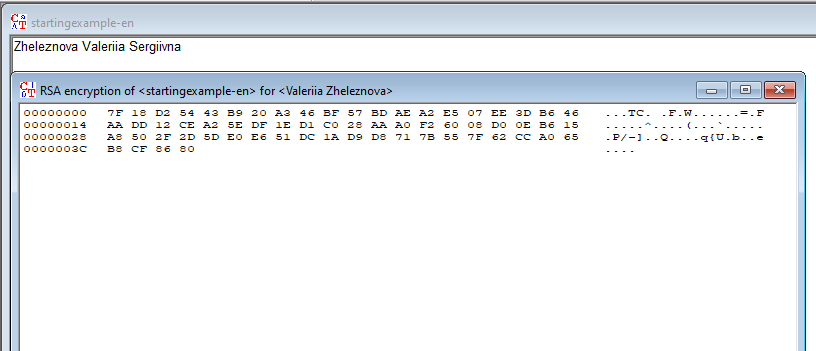
Хід роботи

2. Виконуємо шифрування і дешифрування прізвища, імені та по-батькові за допомогою CrypTool.

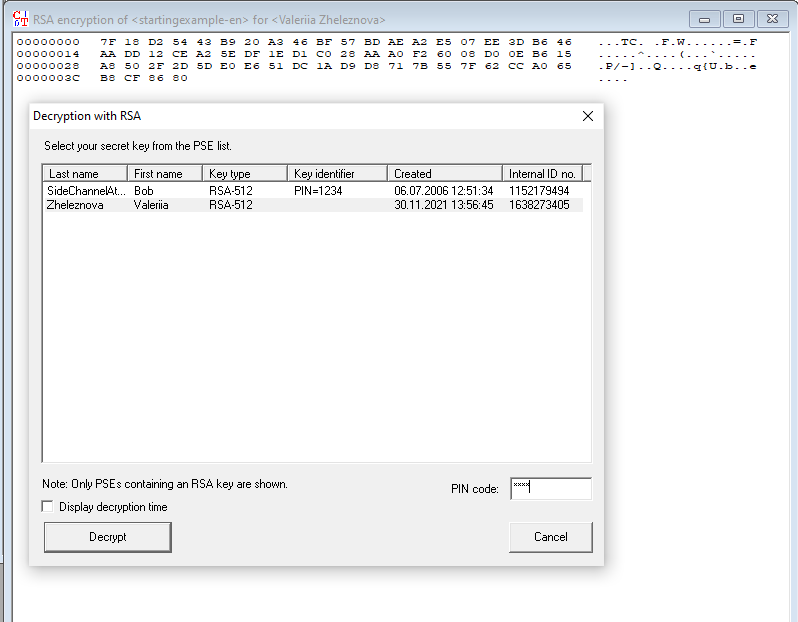
Шифрування:

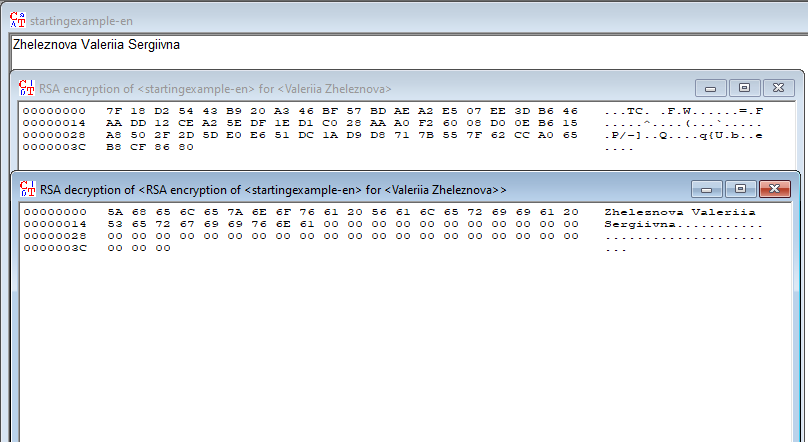






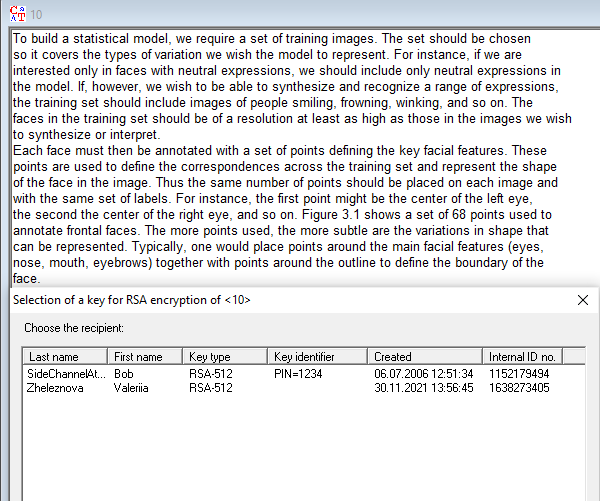
Дешифрування:

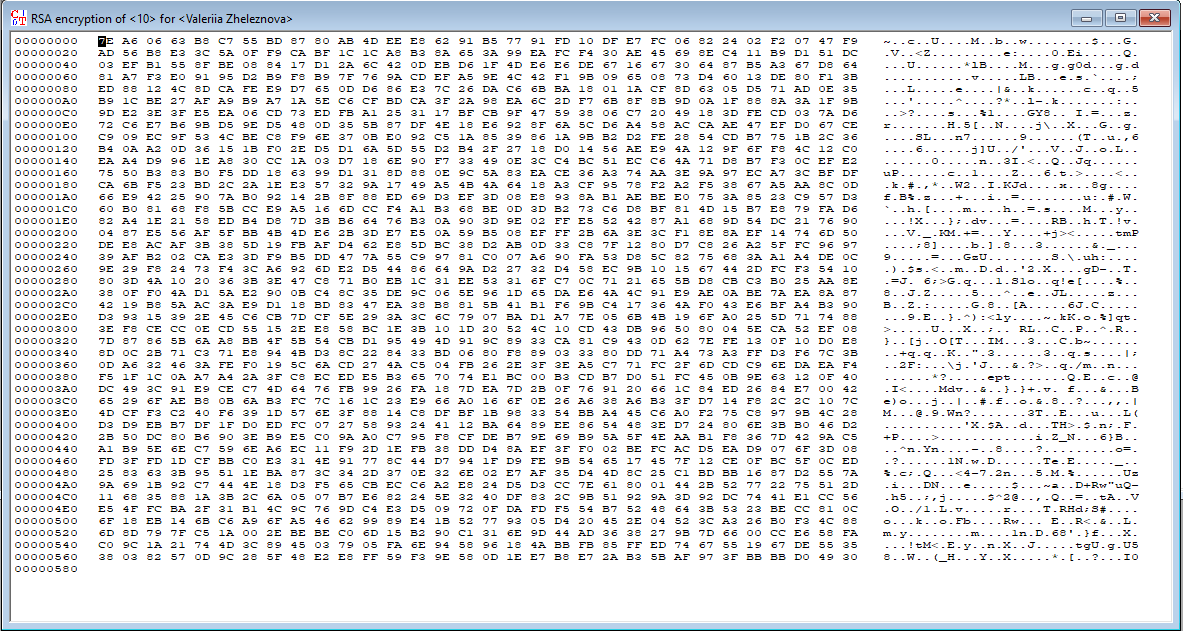




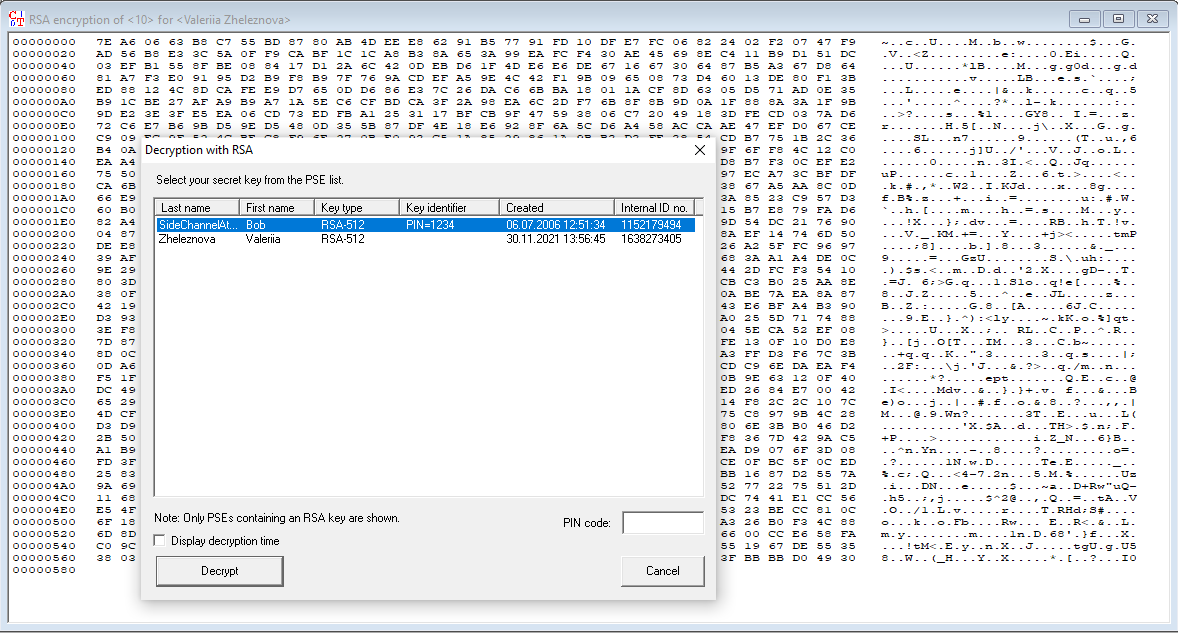
3. Зашифрувати і дешифрувати текст з файлу, згідно варіанту, використовуючи CrypTool.

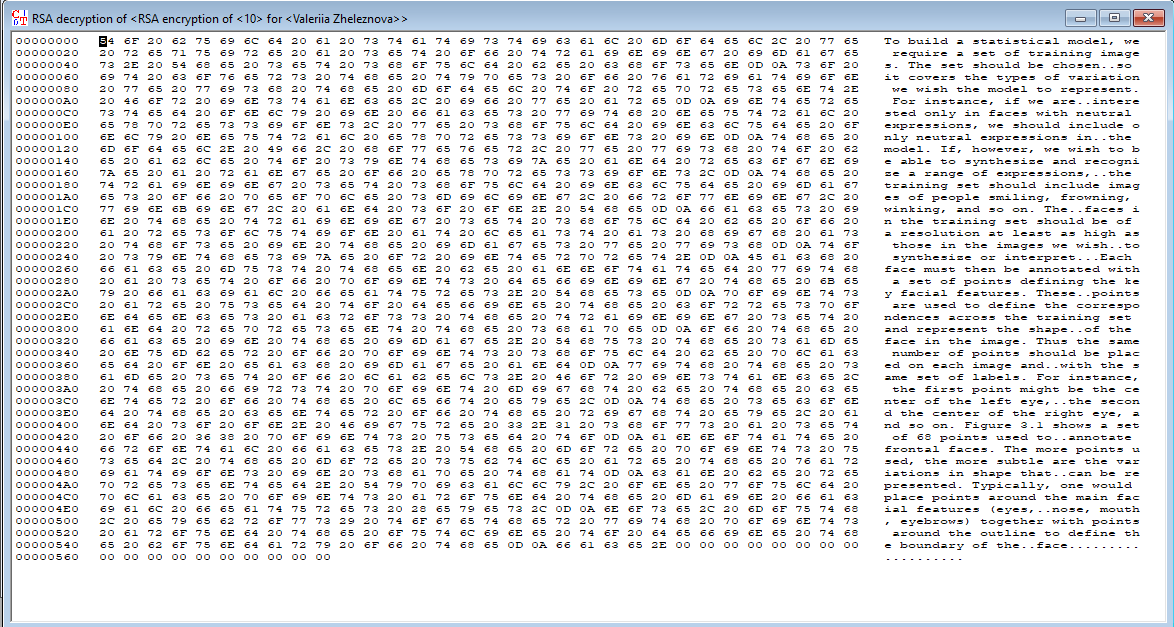
Шифрування:





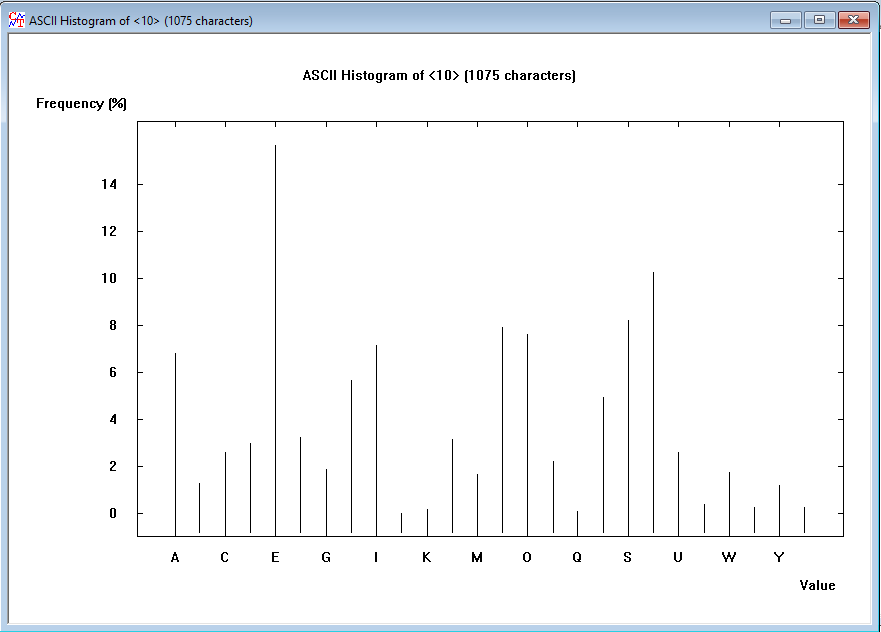
Дешифрування:



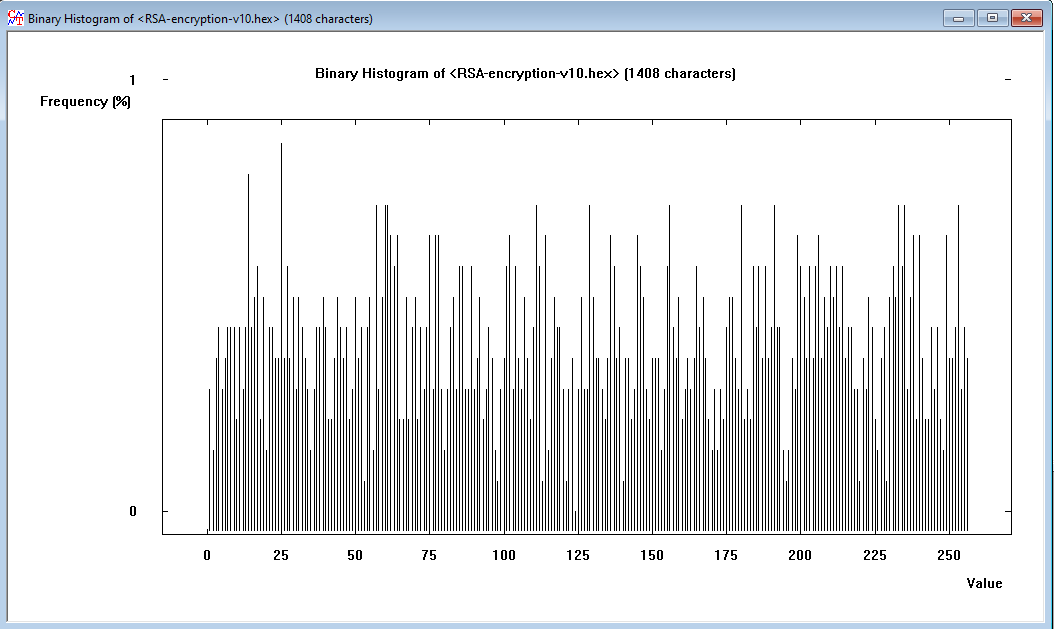


4. За допомогою засобів аналізу пакету CrypTool провести частотний аналіз вихідного і зашифрованого тексту. Визначити ентропію і максимально можливу ентропію зашифрованого файлу, а також побудувати гістограму розподілу частоти символів в аналізованих файлах.

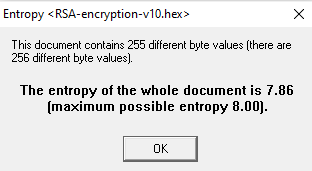
Гістограма розподілу частоти символів початкового тексту:



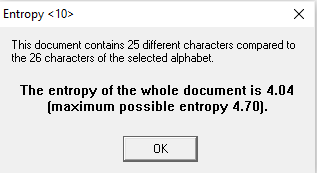
Гістограма розподілу частоти символів зашифрованого тексту:



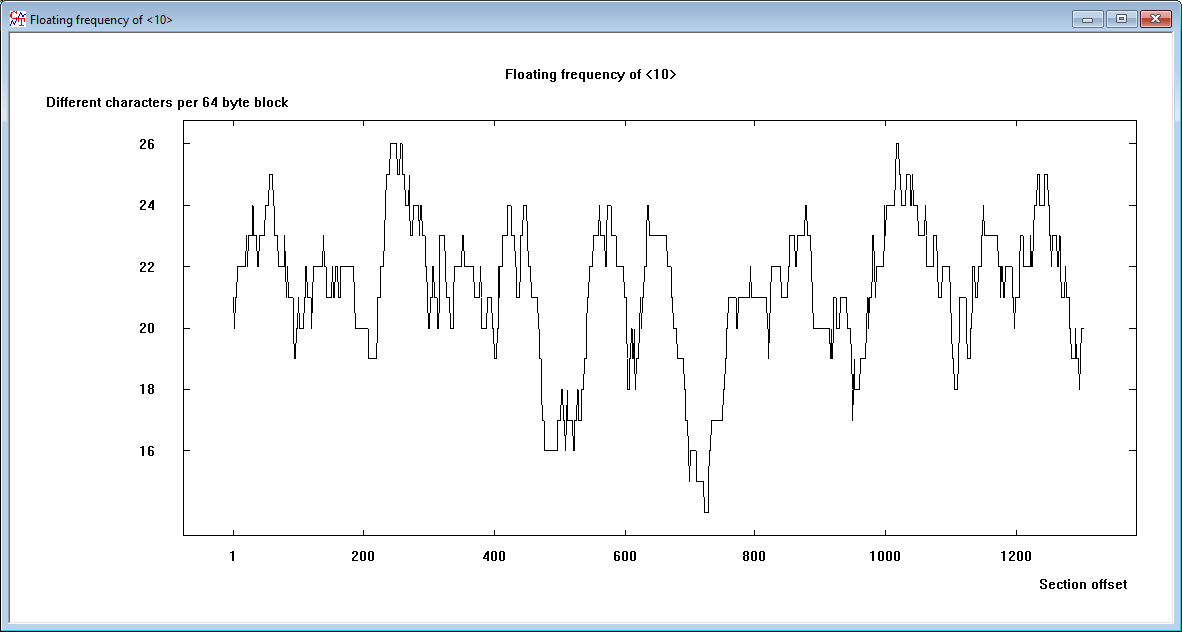
Ентропія та максимально можлива ентропія зашифрованого тексту:



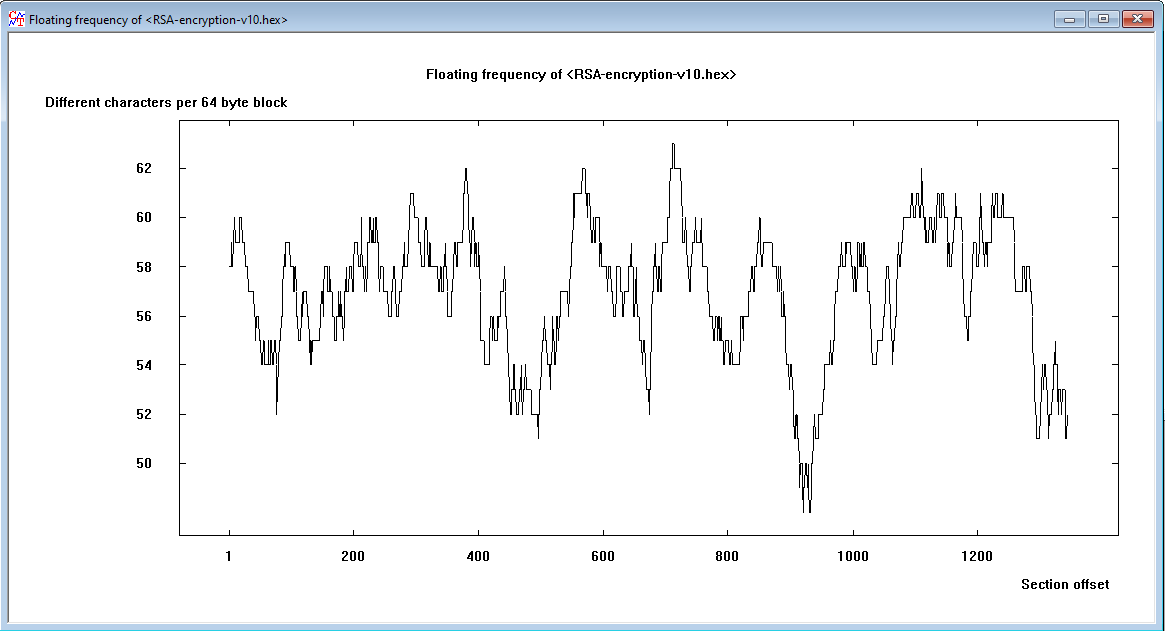
Ентропія та максимально можлива ентропія початкового тексту:



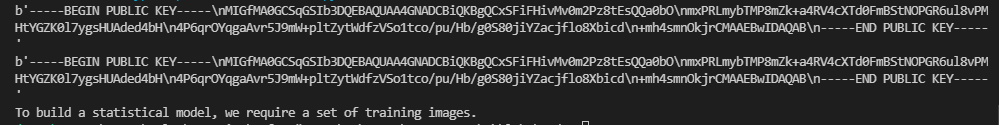
Частотний аналіз початкового тексту:

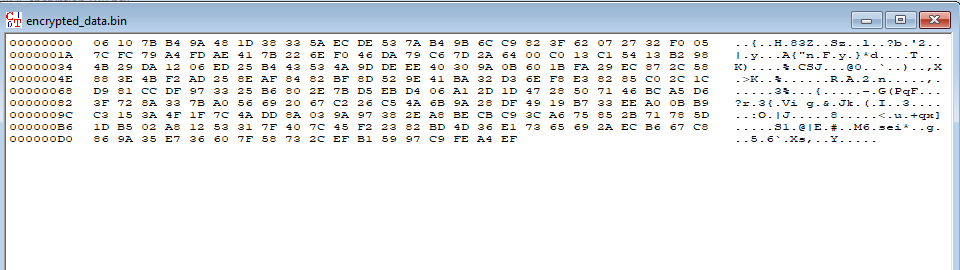


Частотний аналіз зашифрованого тексту:



5. Результат роботи програми:





Лістинг програми:

from Cryptodome.PublicKey import RSA

from Cryptodome.Random import get\_random\_bytes

from Cryptodome.Cipher import AES, PKCS1\_OAEP

secret\_code = "secret"

key = RSA.generate(1024)

encrypted\_key = key.export\_key(passphrase=secret\_code, pkcs=8,

                              protection="scryptAndAES128-CBC")

file\_key\_out = open("rsa\_key.bin", "wb")

file\_key\_out.write(encrypted\_key)

file\_key\_out.close()

print(key.publickey().export\_key())

encoded\_key = open("rsa\_key.bin", "rb").read()

key = RSA.import\_key(encoded\_key, passphrase=secret\_code)

print(key.publickey().export\_key())

key = RSA.generate(1024)

private\_key = key.export\_key()

file\_out = open("private.pem", "wb")

file\_out.write(private\_key)

file\_out.close()

public\_key = key.publickey().export\_key()

file\_out = open("receiver.pem", "wb")

file\_out.write(public\_key)

file\_out.close()

data = "To build a statistical model, we require a set of training images.".encode("utf-8")

file\_out\_encrypted = open("encrypted\_data.bin", "wb")

recipient\_key = RSA.import\_key(open("receiver.pem").read())

session\_key = get\_random\_bytes(16)

# Encrypt the session key with the public RSA key

cipher\_rsa = PKCS1\_OAEP.new(recipient\_key)

enc\_session\_key = cipher\_rsa.encrypt(session\_key)

# Encrypt the data with the AES session key

cipher\_aes = AES.new(session\_key, AES.MODE\_EAX)

ciphertext, tag = cipher\_aes.encrypt\_and\_digest(data)

[ file\_out\_encrypted.write(x) for x in (enc\_session\_key, cipher\_aes.nonce, tag, ciphertext) ]

file\_out\_encrypted.close()

file\_in = open("encrypted\_data.bin", "rb")

private\_key = RSA.import\_key(open("private.pem").read())

enc\_session\_key, nonce, tag, ciphertext = \

   [ file\_in.read(x) for x in (private\_key.size\_in\_bytes(), 16, 16, -1) ]

# Decrypt the session key with the private RSA key

cipher\_rsa = PKCS1\_OAEP.new(private\_key)

session\_key = cipher\_rsa.decrypt(enc\_session\_key)

# Decrypt the data with the AES session key

cipher\_aes = AES.new(session\_key, AES.MODE\_EAX, nonce)

data = cipher\_aes.decrypt\_and\_verify(ciphertext, tag)

print("DECRYPTED:\n\n")

print(data.decode("utf-8"))

Висновки:

У ході виконання лабораторної роботи було вивчено асиметричне шифрування RSA. Виконано шифрування/дешифрування тексту за допомогою CrypTool, а також проведено аналіз початкового та зашифрованого текстів.

Було написано програмний код, який реалізує шифрування та дешифрування RSA.