Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Основы защиты информации»

Отчёт по практическому занятию №4.1

Студент:Шатерник Г. И.

ФИТ 2 курс 3 группа

Преподаватель: Ржеутская Н.В.

Минск 2023 г.

**Практическое задание №4.1**

**Тема «Криптографическая защита информации»**

Цель: овладение основными криптографическими алгоритмами симметричного шифрования

**Теоретическое введение**

Криптография - наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства) информации.

Изначально криптография изучала методы шифрования информации – обратимого преобразования открытого (исходного) текста на основе секретного алгоритма и/или ключа в шифрованный текст (шифротекст). Традиционная криптография образует раздел симметричных криптосистем, в которых зашифрование и расшифрование проводится с использованием одного и того же секретного ключа.

Помимо этого современная криптография включает в себя асимметричные криптосистемы, системы электронной цифровой подписи, хеш-функции, управление ключами, получение скрытой информации, квантовую криптографию.

Шифрованием (encryption) называют процесс преобразования открытых данных (plaintext) в зашифрованные (шифртекст, ciphertext) или зашифрованных данных в открытые по определенным правилам с применением ключей.

В англоязычной литературе зашифрование / расшифрование – enciphering / deciphering.

Классификация алгоритмов шифрования

1. Симметричные (с секретным, единым ключом, одноключевые, single-key).

1.1. Потоковые:

· с одноразовым или бесконечным ключом (infinite-key cipher);

· с конечным ключом;

· на основе генератора псевдослучайных чисел.

1.2. Блочные:

1.2.1. Шифры перестановки (permutation, P-блоки);

1.2.2. Шифры замены (substitution, S-блоки):

· моноалфавитные;

· полиалфавитные;

2. Асимметричные (с открытым ключом, public-key):

· Диффи-Хеллман DH (Diffie, Hellman);

· Райвест-Шамир-Адлeман RSA (Rivest, Shamir, Adleman);

· Эль-Гамаль (ElGamal).

Симметричные алгоритмы шифрования (или криптография с секретными ключами) основаны на том, что отправитель и получатель информации используют один и тот же ключ. Этот ключ должен храниться в тайне и передаваться способом, исключающим его перехват.

Обмен информацией осуществляется в 3 этапа:

* отправитель передает получателю ключ (в случае сети с несколькими абонентами у каждой пары абонентов должен быть свой ключ, отличный от ключей других пар);
* отправитель, используя ключ, зашифровывает сообщение, которое пересылается получателю;
* получатель получает сообщение и расшифровывает его.

Если для каждого дня и для каждого сеанса связи будет использоваться уникальный ключ, это повысит защищенность системы.

При блочном шифровании информация разбивается на блоки фиксированной длины и шифруется поблочно. Блочные шифры бывают двух основных видов:

· шифры перестановки (transposition, permutation, P-блоки);

· шифры замены (подстановки, substitution, S-блоки).

Шифры перестановок переставляют элементы открытых данных (биты, буквы, символы) в некотором новом порядке. Различают шифры горизонтальной, вертикальной, двойной перестановки, решетки, лабиринты, лозунговые и др.

Шифры замены заменяют элементы открытых данных на другие элементы по определенному правилу. Paзличают шифры простой, сложной, парной замены, буквенно-слоговое шифрование и шифры колонной замены. Шифры замены делятся на две группы:

· моноалфавитные (код Цезаря);

· полиалфавитные (шифр Видженера, цилиндр Джефферсона, диск Уэтстоуна, Enigma).

В моноалфавитных шифрах замены буква исходного текста заменяется на другую, заранее определенную букву. Например в коде Цезаря буква заменяется на букву, отстоящую от нее в латинском алфавите на некоторое число позиций.



Очевидно, что такой шифр взламывается совсем просто. Нужно подсчитать, как часто встречаются буквы в зашифрованном тексте, и сопоставить результат с известной для каждого языка частотой встречаемости букв.

В полиалфавитных подстановках для замены некоторого символа исходного сообщения в каждом случае его появления последовательно используются различные символы из некоторого набора. Понятно, что этот набор не бесконечен, через какое-то количество символов его нужно использовать снова. В этом слабость чисто полиалфавитных шифров.

В современных криптографических системах, как правило, используют оба способа шифрования (замены и перестановки). Такой шифратор называют составным (product cipher). Oн более стойкий, чем шифратор, использующий только замены или перестановки.

В асимметричных алгоритмах шифрования (или криптографии с открытым ключом) для зашифровывания информации используют один ключ (открытый), а для расшифровывания - другой (секретный). Эти ключи различны и не могут быть получены один из другого.

Схема обмена информацией такова:

· получатель вычисляет открытый и секретный ключи, секретный ключ хранит в тайне, открытый же делает доступным (сообщает отправителю, группе пользователей сети, публикует);

· отправитель, используя открытый ключ получателя, зашифровывает сообщение, которое пересылается получателю;

· получатель получает сообщение и расшифровывает его, используя свой секретный ключ.

**Шифрование с помощью аффинной системы подстановок Цезаря**

В данном методе используется ключ шифрования в виде пары целых чисел (A, K). Число A задает переход при шифровании вперед на A\*J букв, а число K – дополнительное смещение по алфавиту на K букв.

**Шифр Трисемуса**

В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием «Полиграфия». В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. Каждая буква открытого сообщения заменяется буквой, расположенной под ней в том же столбце. Если буква находится в последней строке таблицы, то для ее шифрования берут самую верхнюю букву столбца.

**Шифр Плейфейра**

Шифр Плейфера использует матрицу 5х5 (для латинского алфавита, для кириллического алфавита необходимо увеличить размер матрицы до 4х8 или 6х6), содержащую ключевое слово или фразу. Для создания матрицы и использования шифра достаточно запомнить ключевое слово и четыре простых правила. Чтобы составить ключевую матрицу, в первую очередь нужно заполнить пустые ячейки матрицы буквами ключевого слова (не записывая повторяющиеся символы), потом заполнить оставшиеся ячейки матрицы символами алфавита, не встречающимися в ключевом слове, по порядку (в английских текстах обычно опускается символ «Q», чтобы уменьшить алфавит, в других версиях «I» и «J» объединяются в одну ячейку). Ключевое слово может быть записано в верхней строке матрицы слева направо, либо по спирали из левого верхнего угла к центру. Ключевое слово, дополненное алфавитом, составляет матрицу 5х5 и является ключом шифра.

Для того чтобы зашифровать сообщение, необходимо разбить его на биграммы (группы из двух символов), например «Hello World» становится «HE LL OW OR LD», и отыскать эти биграммы в таблице. Два символа биграммы соответствуют углам прямоугольника в ключевой матрице. Определяем положения углов этого прямоугольника относительно друг друга. Затем, руководствуясь следующими 4 правилами, зашифровываем пары символов исходного текста:

1. Если два символа биграммы совпадают (или если остался один символ), добавляем после первого символа «Х», зашифровываем новую пару символов и продолжаем. В некоторых вариантах шифра Плейфера вместо «Х» используется «Q».

2. Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

3. Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящиеся непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

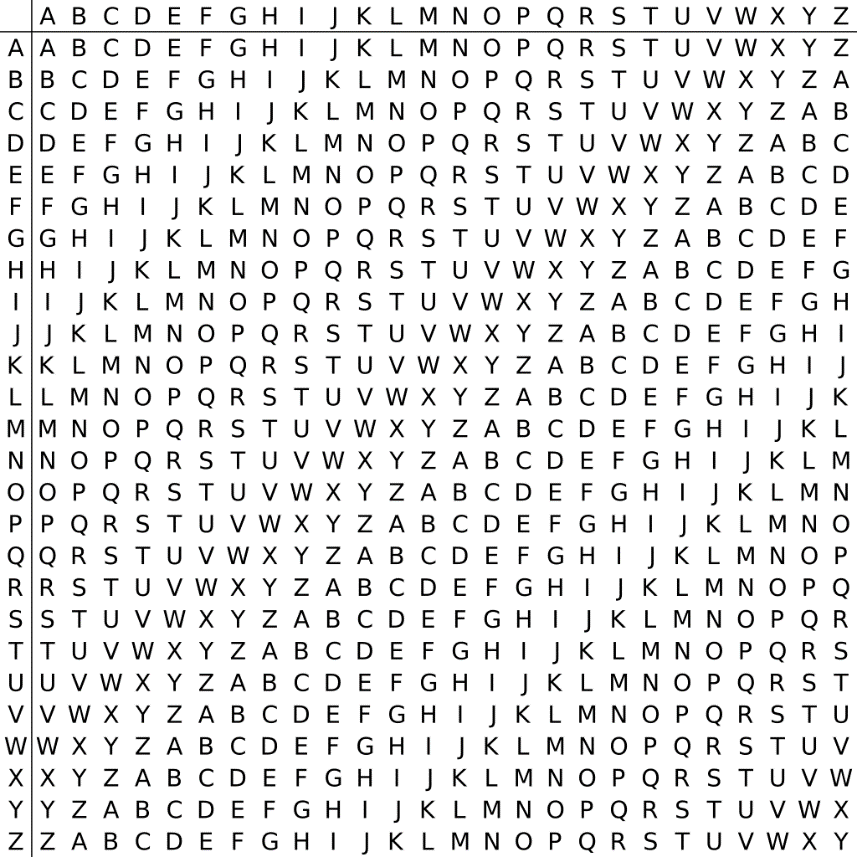
4. Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Для расшифровки необходимо использовать инверсию этих четырёх правил, откидывая символы «Х» (или «Q»), если они не несут смысла в исходном сообщении.

**Шифр Вижинера**

В [шифре Цезаря](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80%D1%8F) каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

Таблица 5.4 – Таблица шифрозамен для шифра Вижинера



# **Задание для выполнения**

Зашифровать сообщение с использованием шифра Цезаря, Трисемуса, Плейфейра и Вижинера и полученного секретного ключа (по номеру варианта и ключевому слову «Защита»). В качестве сообщения использовать свою Фамилию Имя Отчество.

**Шифр Цезаря**

Необходимо зашифровать сообщение «Шатерник Глеб Игоревич» с помощью шифра Цезаря и ключа k=21 вместе со словом “Защита”.

code="защита"

arr ="";

out\_arr =""

input\_text="Шатерник Глеб Игоревич"

min\_let = 'а'

max\_let = 'я'

caes\_shift = 21 %(32-len(code))-1

delta = ord(max\_let)-(ord)(min\_let)+1

for i in range(delta):

    buff = chr(ord(min\_let)+i)

    if(not (code.find(buff)+1)):

        arr+=buff

buff = arr[caes\_shift]

arr=arr.replace(buff,buff+code)

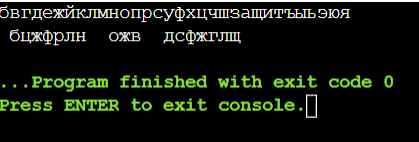
print(arr)

for decripted\_char in input\_text:

    if('я'>=decripted\_char>='а'):

        print(arr[ord(decripted\_char)-ord(min\_let)],end='')

    else: print(" ",end="")



**Шифр Трисемуса**

Необходимо зашифровать сообщение «Шатерник Глеб Игоревич» с помощью шифра Трисемуса и ключевого слова «Защита».

code="минск"

arr = code;

out\_arr =""

input\_text="шатерник глеб игоревич"

min\_let = 'а'

max\_let = 'я'

delta = ord(max\_let)-(ord)(min\_let)+1

for i in range(delta):

buff = chr(ord(min\_let)+i)

if(not (code.find(buff)+1)):

arr+=buff

print(arr)

arr2 = [0] \* 4

for i in range(4):

arr2[i] = [0] \* 8

for cnt in range(4):

for i in range(8):

arr2[cnt][i] = arr[i+8\*cnt]

print(arr2[0])

print(arr2[1])

print(arr2[2])

print(arr2[3])

for cripted\_let in input\_text:

if(cripted\_let !=' '):

position = arr.index(cripted\_let)#arr.find((cripted\_let))

ypos = position/8

xpos= (position)%8

ypos=ypos.\_\_floor\_\_()

xpos=xpos.\_\_floor\_\_()

#print(xpos, end=" ")

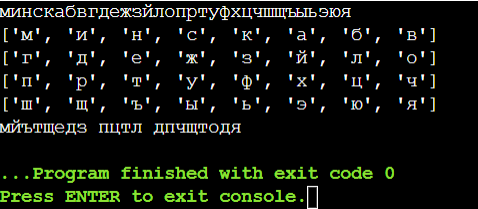
#print(ypos)

ypos+=1

if(ypos==4):ypos=0

print(arr2[ypos][xpos],end='')

else:print(" ", end="")

****

**Шифр Плейфейра**

Необходимо зашифровать сообщение «Шатерник Глеб Игоревич» с помощью шифра Плейфейра и ключевого слова «Защита». Промежуточная биграмма имеет «ша те рн ик гл еб иг ор ев ич».

code="защита"

arr = code;

out\_arr =""

input\_text="шатерник глеб игоревич"

min\_let = 'а'

max\_let = 'я'

delta = ord(max\_let)-(ord)(min\_let)+1

for i in range(delta):

buff = chr(ord(min\_let)+i)

if(not (code.find(buff)+1)):

arr+=buff

print(arr)

arr2 = [0] \* 4

for i in range(4):

arr2[i] = [0] \* 8

start\_alp = [0] \* 4

for cnt in range(4):

for i in range(8):

arr2[cnt][i] = arr[i+8\*cnt]

print(arr2[0])

print(arr2[1])

print(arr2[2])

print(arr2[3])

print(" ")

bigramm=""

for decripted\_let in input\_text:

if(decripted\_let !=' '):

bigramm+=decripted\_let

if(len(bigramm)>=2):

print(bigramm,end=" ")

bigramm=""

deltaxy2=[]

bigramm=""

print("")

for decripted\_let in input\_text:

if(decripted\_let !=' '):

position = arr.find((decripted\_let))

ypos = position/8

xpos= (position)%8

ypos=ypos.\_\_floor\_\_()

xpos=xpos.\_\_floor\_\_()

deltaxy2.append(xpos)

deltaxy2.append(ypos)

bigramm+=decripted\_let

if(len(bigramm)>=2):

if deltaxy2[0] == deltaxy2[2]:

deltaxy2[0] = (deltaxy2[0] - 1) % 8

deltaxy2[2] = (deltaxy2[2] - 1) % 8

elif deltaxy2[1] == deltaxy2[3]:

deltaxy2[3] = (deltaxy2[3] - 1) % 4

deltaxy2[1] = (deltaxy2[1] - 1) % 4

else:

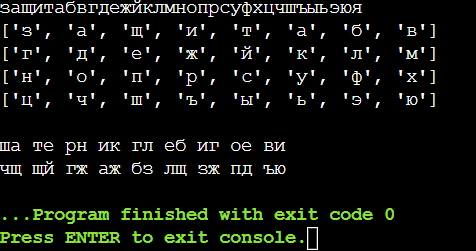
deltaxy2[1], deltaxy2[3] = deltaxy2[3], deltaxy2[1]

print(arr2[deltaxy2[3]][deltaxy2[2]],end='')

print(arr2[deltaxy2[1]][deltaxy2[0]],end=' ');

bigramm="";

deltaxy2=[]



**Шифр Вижинера**

Необходимо зашифровать сообщение «Шатерник Глеб Игоревчи» с помощью шифра Вижинера и ключевого слова «Защита». Промежуточный ключ имеет вид «защитазащитазащитазащитазащ» (человек, посылающий сообщение, записывает ключевое слово («защита») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста).

code="защита"

arr = "";

out\_arr =""

input\_text="шатерник глеб игоревич"

min\_let = 'а'

max\_let = 'я'

delta = ord(max\_let)-(ord)(min\_let)+1

for i in range(delta):

buff = chr(ord(min\_let)+i)

arr+=buff

arr = arr.replace(chr(ord(min\_let)+5),"её")

print(arr)

deltaxy2=[]

i=0

print("/////")

for decripted\_let in input\_text:

if(decripted\_let!=" "):

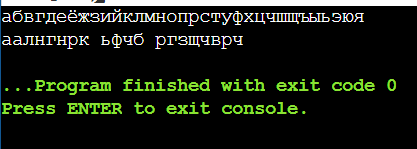
encripted=(arr.index(decripted\_let)+arr.index(code[i%len(code)]))%len(arr)

print(arr[encripted],end="")

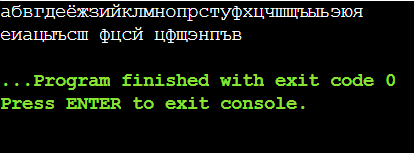
i+=1

else: print(end=" ")

Для слова защита



Для слова минск



# **Индивидуальное задание**

Расшифровать следующие сообщения.

Таблица 7.6-Вариант выполнения задания

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта | Сообщение | Способ | Ключ |
| 21 | зчгы очхей, й зчгы гйък щчрейв | Расшифровать с помощью шифра Трисемуса | Минск |

code="минск"

arr = code;

out\_arr =""

input\_text="зчгы очхей, й зчгы гйък щчрейв"

min\_let = 'а'

max\_let = 'я'

delta = ord(max\_let)-(ord)(min\_let)+1

for i in range(delta):

buff = chr(ord(min\_let)+i)

if(not (code.find(buff)+1)):

arr+=buff

print(arr)

arr2 = [0] \* 4

for i in range(4):

arr2[i] = [0] \* 8

for cnt in range(4):

for i in range(8):

arr2[cnt][i] = arr[i+8\*cnt]

print(arr2[0])

print(arr2[1])

print(arr2[2])

print(arr2[3])

for cripted\_let in input\_text:

if(cripted\_let !=' '):

position = arr.find((cripted\_let))

ypos = position/8

xpos= (position)%8

ypos=ypos.\_\_floor\_\_()

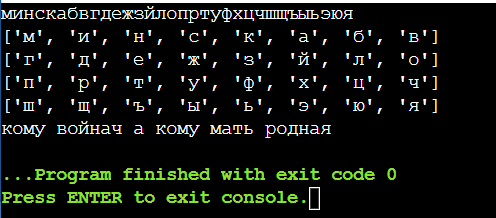
xpos=xpos.\_\_floor\_\_()

ypos-=1

if(ypos==-1):ypos=3

print(arr2[ypos][xpos],end='')

else:print(" ", end="")



Результат: кому войнач а кому мать родная.

**Вывод:** в ходе лабораторной работы были изучены основные криптографические алгоритмы симметричного шифрования.