**«Әбілқас Сағынов атындағы Қарағанды техникалық университеті»**

**комерциялық емес акционерлік қоғамы**

*«Ақпараттық техналогия және*

*қауіпсіздік» кафедрасы*

**Зертханалық жұмыс №3**

.

Жетекші: Усербаева Г. М.

Студент­: Турарбеков А. А.

Топ: СИБ-22-3

2024

Тақырыбы: Криптографиялық жүйелер

Жұмыстың мақсаты: Шифрлау жүйелерінің алгоритмдерін зерттеу, осы шифрлау сыныптарының бұзылуға тұрақтылығын талдау.

Тапсырма нұсқасы: 4

Әдістер: 1.1; 2.4.

№4 нұсқа үшін шифрлау мен дешифрлауды жүзеге асыру.

№4 нұсқа үшін DES алгоритмін орындайтын бағдарлама жазу.

1.1. «Цезарь» шифры [қазақ әліпбиіне қарай]:

# Казахский алфавит (42 буквы)

kazakh\_alphabet = "АӘБВГҒДЕЁЖЗИЙКҚЛМНҢОӨПРСТУҰҮФХҺЦЧШЩЪЫІЬЭЮЯабвгғдеёжзийкқлмнңоөпрустуұүфхһцчшщъыіьэюя"

def caesar\_encrypt\_kazakh(text, shift):

encrypted\_text = ""

for char in text:

if char in kazakh\_alphabet:

# Получаем индекс символа в алфавите и сдвигаем его

idx = kazakh\_alphabet.index(char)

new\_idx = (idx + shift) % len(kazakh\_alphabet)

encrypted\_text += kazakh\_alphabet[new\_idx]

else:

encrypted\_text += char # оставляем символы, не входящие в алфавит, без изменений

return encrypted\_text

def caesar\_decrypt\_kazakh(encrypted\_text, shift):

decrypted\_text = ""

for char in encrypted\_text:

if char in kazakh\_alphabet:

# Получаем индекс символа в алфавите и сдвигаем его в обратную сторону

idx = kazakh\_alphabet.index(char)

new\_idx = (idx - shift) % len(kazakh\_alphabet)

decrypted\_text += kazakh\_alphabet[new\_idx]

else:

decrypted\_text += char # оставляем символы, не входящие в алфавит, без изменений

return decrypted\_text

# Тестируем шифрование и дешифрование

text = " Компьютерлік ақпаратты қорғау технологиялары"

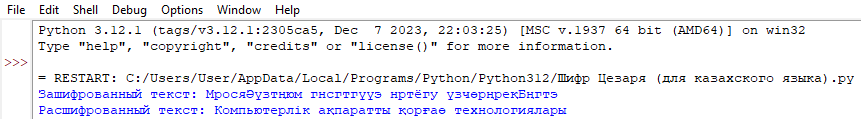
shift = 3 # Указываем ключ (например, 3)

encrypted = caesar\_encrypt\_kazakh(text, shift)

decrypted = caesar\_decrypt\_kazakh(encrypted, shift)

print("Зашифрованный текст:", encrypted)

print("Расшифрованный текст:", decrypted)



«Цезарь шифры» [орыс әліпбиіне қарай]:

def caesar\_encrypt\_russian(text, shift):

encrypted\_text = ""

for char in text:

if 'А' <= char <= 'Я' or 'а' <= char <= 'я':

shift\_base = ord('А') if char.isupper() else ord('а')

encrypted\_text += chr((ord(char) - shift\_base + shift) % 32 + shift\_base)

else:

encrypted\_text += char # не изменяем пробелы и знаки препинания

return encrypted\_text

def caesar\_decrypt\_russian(encrypted\_text, shift):

decrypted\_text = ""

for char in encrypted\_text:

if 'А' <= char <= 'Я' or 'а' <= char <= 'я':

shift\_base = ord('А') if char.isupper() else ord('а')

decrypted\_text += chr((ord(char) - shift\_base - shift) % 32 + shift\_base)

else:

decrypted\_text += char # не изменяем пробелы и знаки препинания

return decrypted\_text

# Тестируем шифрование и дешифрование

text = "Технологии защиты компьютерный информации"

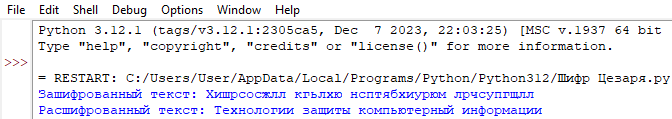
shift = 3 # Указываем ключ (например, 3)

encrypted = caesar\_encrypt\_russian(text, shift)

decrypted = caesar\_decrypt\_russian(encrypted, shift)

print("Зашифрованный текст:", encrypted)

print("Расшифрованный текст:", decrypted)



«Цезарь» шифры [ағылшын әліпбиіне қарай]:

def caesar\_encrypt(text, shift):

encrypted\_text = ""

for char in text:

if char.isalpha():

shift\_base = 65 if char.isupper() else 97

encrypted\_text += chr((ord(char) - shift\_base + shift) % 26 + shift\_base)

else:

encrypted\_text += char

return encrypted\_text

def caesar\_decrypt(encrypted\_text, shift):

decrypted\_text = ""

for char in encrypted\_text:

if char.isalpha():

shift\_base = 65 if char.isupper() else 97

decrypted\_text += chr((ord(char) - shift\_base - shift) % 26 + shift\_base)

else:

decrypted\_text += char

return decrypted\_text

# Пример использования:

text = "Computer information protection technologies"

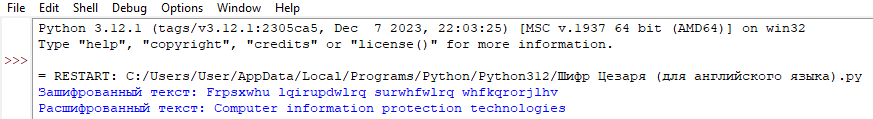
shift = 3

encrypted = caesar\_encrypt(text, shift)

decrypted = caesar\_decrypt(encrypted, shift)

print("Зашифрованный текст:", encrypted)

print("Расшифрованный текст:", decrypted)



2.4 «Кездейсоқ орын ауыстыру» әдісі

import random

def random\_shift\_encrypt(text):

shifts = [random.randint(1, 25) for \_ in text] # случайные смещения от 1 до 25

encrypted\_text = ""

for i, char in enumerate(text):

if char.isalpha():

shift\_base = 65 if char.isupper() else 97

encrypted\_text += chr((ord(char) - shift\_base + shifts[i]) % 26 + shift\_base)

else:

encrypted\_text += char

return encrypted\_text, shifts

def random\_shift\_decrypt(encrypted\_text, shifts):

decrypted\_text = ""

for i, char in enumerate(encrypted\_text):

if char.isalpha():

shift\_base = 65 if char.isupper() else 97

decrypted\_text += chr((ord(char) - shift\_base - shifts[i]) % 26 + shift\_base)

else:

decrypted\_text += char

return decrypted\_text

# Пример использования:

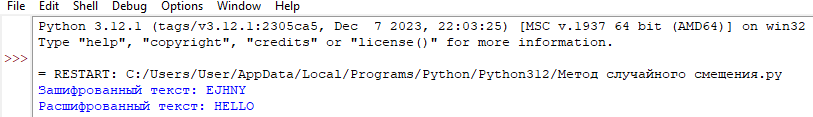
text = "HELLO"

encrypted, shifts = random\_shift\_encrypt(text)

decrypted = random\_shift\_decrypt(encrypted, shifts)

print("Зашифрованный текст:", encrypted)

print("Расшифрованный текст:", decrypted)



DES алгоритмі

from Crypto.Cipher import DES

from Crypto.Util.Padding import pad, unpad

def des\_encrypt(text, key):

des = DES.new(key, DES.MODE\_ECB)

padded\_text = pad(text.encode(), DES.block\_size) # добавляем padding

encrypted\_text = des.encrypt(padded\_text)

return encrypted\_text

def des\_decrypt(encrypted\_text, key):

des = DES.new(key, DES.MODE\_ECB)

decrypted\_padded\_text = des.decrypt(encrypted\_text)

decrypted\_text = unpad(decrypted\_padded\_text, DES.block\_size)

return decrypted\_text.decode()

# Пример использования:

text = "Apple"

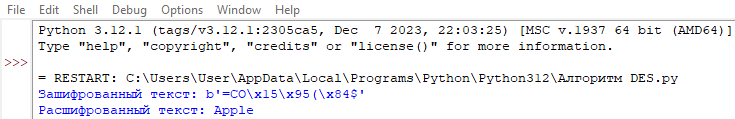
key = b"8bytek#d" # DES ключ должен быть 8 байт

encrypted = des\_encrypt(text, key)

decrypted = des\_decrypt(encrypted, key)

print("Зашифрованный текст:", encrypted)

print("Расшифрованный текст:", decrypted)



Бақылау сұрақтары:

1. Криптография не істейді?

2. Криптографиялық жүйенің түсінігі және схемасы.

3. "Тікелей алмастыру" әдісінің мәні.

4. "Көпфавиттік алмастыру" әдісінің артықшылықтары мен

кемшіліктері.

5. Кодтау мен шифрлаудың айырмашылығы неде?

6. Вернам шифрының мәні.

7. Блокты шифрлау ағыннан несімен ерекшеленеді?

1) Криптография ақпаратты шифрлау әдістерін, асимметриялық криптожүйелерді, электрондық цифрлық қолтаңба жүйелерін, хэш функцияларын, кілттерді басқаруды, жасырын ақпаратты алуды, сондай-ақ кванттық криптографияны қамтиды.

2) Криптографиялық жүйе – криптографиялық алгоритмдер мен кілттердің көмегімен деректерді оқылмайтын пішінге түрлендіру арқылы қорғауға арналған әдістер мен құралдардың жиынтығы. Криптографиялық жүйенің негізгі мақсаты коммерциялық ақпараттың құпиялылығын, тұтастығын, түпнұсқалығын және ақауларға төзімділігін қамтамасыз ету болып табылады.

Криптографиялық жүйенің негізгі ұғымдары:

\* Ашық мәтін – шифрлануы тиіс бастапқы ақпарат.

\* Шифрлау – шифрлау кілтінің көмегімен ашық мәтінді шифрленген мәтінге (шифрлық мәтінге) түрлендіру процесі.

\* Шифрленген мәтін (шифрленген мәтін) – шифрлаудың нәтижесі, ол шифрды шешу процесінсіз оқылмайтын ақпарат түрі болып табылады.

\* Шифрды шешу – шифрды шешу кілтінің көмегімен шифрленген мәтінді ашық мәтінге қайта түрлендіру процесі.

\* Кілт – деректерді шифрлау және шифрын ашу үшін қолданылатын құпия параметр. Кілтті білмей, мәтіннің шифрын ашу мүмкін емес.

\* Шифрлау алгоритмі – мәліметтерді шифрлау және дешифрлеу процесін жүзеге асыратын математикалық әдіс.

Криптографиялық жүйе схемасы:

Криптографиялық жүйе үш негізгі кезеңді қамтиды: шифрлау, деректерді беру және шифрды шешу. Жалпы схема келесідей көрінеді:

\* Шифрлау: ашық мәтін → шифрлау алгоритмі + шифрлау кілті → шифрлау мәтіні.

\* Мәліметтерді жіберу: байланыс арнасы арқылы шифрленген мәтін және ол ұсталып қалса да, оны кілтсіз оқу мүмкін емес.

\* Шифрды шешу: шифрлық мәтін → шифрды шешу алгоритмі + шифрды шешу кілті → ашық мәтін.

Криптографиялық құралдарда жиі қолданылатын кілттердің екі түрі бар:

\* Симметриялық шифрлау (шифрлау және шифрды шешу үшін бірдей кілт).

\* Асимметриялық шифрлау (екі кілт пайдаланылады: шифрлау үшін ашық және шифрды шешу үшін жеке).

3) Тікелей ауыстыру әдісі – ашық мәтіннің әрбір әрпі немесе таңбасы басқа таңбамен немесе алдын ала анықталған жиыннан әріппен ауыстырылатын әдістердің жалпы атауы. Бұл жағдайда әрбір таңба дербес шифрланады және сәйкестік позициялардың белгіленген саны бойынша жылжу ғана емес, кез келген нәрсе болуы мүмкін. Мысалы, кейбір алдын ала белгіленген ауыстыру кестесіне сәйкес «А» әрпін «Q», «B» әрпін «L» және т.б. ауыстыруға болады.

4) Көпфавиттік алмастыру (көпалфавитті шифрлау деп те аталады) – мәтіндегі әр символды әртүрлі алфавиттерді пайдаланып бірнеше рет ауыстыратын криптографиялық әдіс. Бұл әдістің ең танымал түрі – Виженер шифры, онда бірқатар әріп ауыстыру ережелері қолданылады, және шифрлау процесі бірнеше кілт арқылы жүзеге асады.

Артықшылықтары:

\* Қауіпсіздігі жоғары: Жиілік талдауын қолдану арқылы шифрды бұзу қиын, себебі бірдей әріп әртүрлі орындарда әртүрлі әріптермен алмастырылады. Бұл криптоанализ жасау мүмкіндігін азайтады.

\* Кілттің икемділігі: Кілт ұзындығын арттыру арқылы шифрдың беріктігін күшейтуге болады. Ұзын кілттер шифрды талдауды күрделендіреді.

\* Қарапайым құрылым: Бұл әдісті түсіну және қолмен шифрлау оңай. Көптеген жағдайларда көпфавиттік алмастыру басқа күрделі шифрлармен салыстырғанда оңайырақ жүзеге асады.

\* Аралас шифрлау: Басқа әдістермен, мысалы, монофавиттік шифрлармен біріктіру арқылы қауіпсіздік деңгейін арттыруға болады.

Кемшіліктері:

\* Кілтті басқарудың қиындығы: Кілт өте құпия сақталуы тиіс, және ұзын кілттерді есте сақтау қиын болуы мүмкін.

\* Криптоанализге әлсіздігі: Егер кілт қысқа болса немесе белгілі сөз, сөз тіркестері қолданылса, криптоанализ әдістері арқылы (мысалы, Касиски талдауы немесе Виженер шифрын бұзудың басқа әдістері) шифрды бұзу мүмкін болады.

\* Мәтін ұзындығының кілт ұзындығымен қатынасы: Егер шифрланған мәтіннің ұзындығы кілт ұзындығынан әлдеқайда ұзын болса, онда кейбір шифрланған бөліктер кілттің қайталануына байланысты әлсіреуі мүмкін.

\* Шифрлау жылдамдығы: Қолданылатын алфавиттердің саны артқан сайын шифрлау және дешифрлау уақыты да артады, бұл әсіресе ұзын мәтіндерді шифрлауда қолайсыздық тудыруы мүмкін.

5) Кодтау және шифрлау екі түрлі ұғым, бірақ екеуі де деректерді түрлендіруді қамтиды.

Кодтау – үйлесімділікті қамтамасыз ету немесе деректерді тасымалдауды жеңілдету үшін ақпаратты бір пішімнен екіншісіне түрлендіру. Кодтаудың негізгі мақсаты деректерді сақтауға және тасымалдауға ыңғайлы форматта сақтау болып табылады. Кодтаудың мысалы мәтінді тек мәтіндік деректерді қолдайтын жүйелермен үйлесімді пішімде (мысалы, URL мекенжайында немесе электрондық поштада) тасымалдауға болатындай етіп мәтінді Base64 түріне түрлендіру болып табылады. Кодтау әдетте қорғаныс шарасы болып табылмайды, себебі кодтау ережелері немесе схемалары берілген кодталған деректер оңай декодталады.

Мысалдар: ASCII (мәтіндер үшін), Base64 (мәтіндік протоколдар арқылы екілік деректерді беру үшін), URL кодтауы.

Нәтиже: Шифрланған мәліметтерді құпия кілтсіз оңай оқуға және декодтауға болады, себебі мақсат ақпаратты қорғау емес, оның дұрыс берілуін немесе үйлесімділігін қамтамасыз ету.

Шифрлау ақпаратты рұқсатсыз кіруден қорғауға бағытталған. Шифрлау кезінде хабарлама арнайы шифрды шешу кілті болған жағдайда ғана оқуға болатындай түрлендіріледі.

Мысалдар: Шифрлаудың танымал әдістеріне AES, RSA, DES алгоритмдері жатады.

Нәтиже: Егер біреу шифрланған хабарламаны тартып алса немесе сол ақпаратқа қол жеткізсе кілтсіз оның шифрын шеше алмайды.

6) Вернам шифры – шифрлау әдістерінің ішіндегі ең қауіпсізі деп саналады. Бұл әдісте бастапқы мәтін кездейсоқ кілтпен биттік деңгейде XOR операциясы арқылы шифрланады.

Вернам шифрының негізгі ерекшеліктері мен шарттары:

\* Кілттің ұзындығы бастапқы мәтіннің ұзындығына тең болуы қажет. Кілт символдарының саны шифрланатын мәтіннің символдарына дәл сәйкес келуі тиіс.

\* Кілт толығымен кездейсоқ болуы керек. Бұл шифрдың қауіпсіздігі кілттің кездейсоқтығына байланысты, себебі тек шынайы кездейсоқ кілт шифрдың сындырылуына қарсы тұрады.

\* Кілт тек бір рет қана қолданылуы керек. Егер кілт бір реттен артық қолданылса, шифр қауіпсіздігінен айырылады және криптоанализге әлсіздік танытады.

Вернам шифрының басты артықшылығы – ол теориялық тұрғыдан бұзылмайды, егер жоғарыдағы шарттар толық орындалса. Бұл шифрдың кемшілігі – кілтті сақтау және беру мәселелерінің күрделілігінде.

7) Блоктық және ағындық шифрлау – симметриялық шифрлау әдістерінің екі негізгі түрі, және олардың айырмашылықтары шифрлау үдерісінің құрылымында жатыр.

Блоктық шифрлау мәліметтерді белгілі бір өлшемдегі (мысалы, 64 бит, 128 бит) блоктарға бөліп, әрбір блокты жеке шифрлайды.

Мысалы, AES және DES – блоктық шифрлардың танымал үлгілері.

Әрбір блок жеке шифрланғандықтан, бұл әдіс үлкен көлемдегі мәліметтермен жұмыс істеуге ыңғайлы.

Артықшылықтары:

\* Бірнеше блоктар арасында тәуелділіктің болмауы арқылы параллельді шифрлау жасауға болады.

\* Блоктық шифрлау кезінде әрбір блок толық өңделгендіктен, мәлімет бүтіндігін қамтамасыз етуге жақсы мүмкіндік береді.

Кемшіліктері:

\* Блоктың толық болмауы жағдайында оны толтыру (padding) қажет болуы мүмкін, бұл қосымша жадты талап етеді.

\* Кейбір жағдайларда кешігу тудырып, уақыт тиімділігін төмендетуі мүмкін.

Ағындық шифрлау мәліметті бит немесе байт деңгейінде шифрлайды, яғни бүкіл мәтінді үздіксіз ағын ретінде өңдейді. Ағындық шифрларда кілт ағыны (key stream) мәлімет ағынымен XOR операциясы арқылы біріктіріледі. RC4 – ағындық шифрлаудың танымал мысалы.

Артықшылықтары:

\* Өте жылдам және шифрлауды нақты уақыт режимінде жүргізуге болады.

\* Ағындық шифрлау блокты толтыруды талап етпейді, сондықтан деректер көлеміне қарай ыңғайлы.

Кемшіліктері:

\* Қолданылатын кілт ағынының кездейсоқтығына тәуелділік жоғары, егер ол дұрыс емес жасалса, шифрлау әлсіз болады.

\* Тек бір бағытта ғана шифрлайтындықтан, толық мәтінмен жұмыс істегенде қауіпсіздік тұрғысынан блоктық шифрлауға қарағанда әлсіз болуы мүмкін.

Қорытындылай келе, блоктық шифрлау – мәліметтерді блоктарға бөледі және үлкен көлемдегі деректер үшін тиімді, әсіресе мәліметтер бүтіндігін қажет ететін жағдайларда. Ал ағындық шифрлау – мәліметтерді үздіксіз ағын ретінде шифрлайды, және нақты уақыттағы, кіші көлемдегі деректер үшін қолайлы.