Отчёта по лабораторной работе №8

дисциплина: Информационная безопасность

Кашкин Иван Евгеньевич

Содержание

Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Теоретические сведения

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

Метод гаммирования с обратной связью заключается в том, что для получения сегмента гаммы используется контрольная сумма определенного участка шифруемых данных.
Например, если рассматривать гамму шифра как объединение непересекающихся множеств H(j)

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$C_1 = P_1 \oplus K\ C_2$$
$$= P_2 \oplus K$$

Открытый текст можно найти, зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства складываются по модулю 2. Тогда с учётом свойства операции XOR получаем:

$$C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$

Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей.

Допустим, что злоумышленнику этот формат известен.

Тогда он получает достаточно много пар $C_1 \oplus C_2$ (известен вид обеих шифровок). Тогда зная P_1 имеем:

$$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2$$

Таким образом, злоумышленник получает возможность определить те символы сообщения P_2, которые находятся на позициях известного шаблона сообщения P_1.

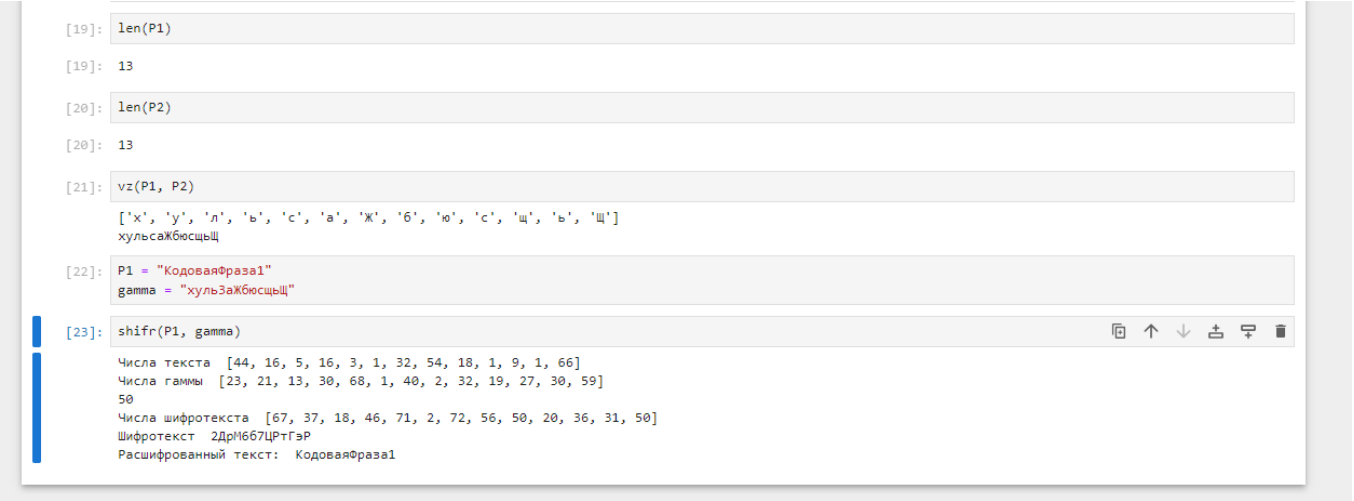
В соответствии с логикой сообщения P_2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P_2.

Затем вновь используется равенство с подстановкой вместо P_1 полученных на предыдущем шаге новых символов сообщения P_2.

Выполнение работы

[17]:
```
a = ord("а")
liters = [chr(i) for i in range(a, a + 32)]
a = ord("0")
for i in range(a, a+10):
    liters.append(chr(i))

a = ord("А")
for i in range(1040, 1072):
    liters.append(chr(i))

P1 = "КодоваяФраза1"
P2 = "Безопасность2"

def vz(P1, P2):
    code = []
    for i in range(len(P1)):
        code.append(liters[(liters.index(P1[i]) + liters.index(P2[i])) % len(liters)])
    print(code)
    pr = "".join(code)
    print(pr)
```

```
def shifr(P1, gamma):
    dicts = {"а": 1, "б": 2, "в": 3, "г": 4, "д": 5, "е": 6, "ё": 7, "ж": 8, "з": 9, "и": 10, "й": 11, "к": 12, "л": 13,
             "м": 14, "н": 15, "о": 16, "п": 17, "р": 18, "с": 19, "т": 20, "у": 21, "ф": 22, "х": 23, "ц": 24, "ч": 25,
             "ш": 26, "щ": 27, "ъ": 28, "ы": 29, "ь": 30, "э": 31, "ю": 32, "я": 32, "А":33 , "Б": 34, "В": 35 , "Г":36,
             "Д":37 , "Е":38 , "Ё":39 , "Ж":40 , "З":41, "И":42,"Й":43 , "К":44 , "Л":45 , "М":46 , "Н":47 , "О":48 ,
             "П":49 , "Р":50 , "С":51 , "Т":52 , "У":53 , "Ф":54 , "Х":55 , "Ц":56 , "Ч":57 , "Ш":58,"Щ":59 , "Ъ":60 ,
             "Ы":61 , "Ь":62 , "Э":63 , "Ю":64 , "Я":65 , "1":66 , "2":67 , "3":68 , "4":69 , "5":70 , "6":71 , "7": 72,
             "8":73 , "9":74 , "0":75
    }

    dicts2 = {v: k for k, v in dicts.items()}
    text = P1
    digits_text = []
    digits_gamma = []

    for i in text:
        digits_text.append(dicts[i])
    print("Числа текста ", digits_text)

    for i in gamma:
        digits_gamma.append(dicts[i])
    print("Числа гаммы ", digits_gamma)

    digits_result = []
    ch = 0
    for i in text:
        try:
            a = dicts[i] + digits_gamma[ch]
        except:
            ch = 0
            a = dicts[i] + digits_gamma[ch]
        if a > 75:
            a = a%75
            print(a)
        ch += 1
        digits_result.append(a)
    print("Числа шифротекста ", digits_result)

    text_cr = ""
    for i in digits_result:
        text_cr += dicts2[i]
    print("Шифротекст ", text_cr)

    digits = []
    for i in text_cr:
        digits.append(dicts[i])
    ch = 0
    digits1 = []
    for i in digits:
        try:
            a = i - digits_gamma[ch]
        except:
            ch = 0
            a = i - digits_gamma[ch]
        if a < 1:
            a = 75 + a
        digits1.append(a)
        ch += 1

    text_decr = ""
    for i in digits1:
        text_decr += dicts2[i]
    print("Расшифрованный текст: ", text_decr)
```



Выводы

В ходе выполнения лабораторной работы было разработано приложение, позволяющее шифровать тексты в режиме однократного гаммирования.

Список литературы{.unnumbered}

1. Шифрование методом гаммирования
2. Режим гаммирования в блочном алгоритме шифрования