РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8

Дисциплина: Основы информационной безопасности
Название работы: Элементы криптографии. Однократное гаммирование

Студент: Теплякова Анастасия Сергеевна

Группа: НПМбд-02-18

МОСКВА

2021 г.

Цель работы:

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом1 .

Ход работы:

Два текста кодируются одним ключом (однократное гаммирование).
Требуется не зная ключа и не стремясь его определить, прочитать оба текста.
Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования.

```
# создаем алфавит из русских букв и цифр
# он нужен для гаммирования
a = ord("а")
алфавит = [chr(i) для i в диапазоне(а, а + 32)]
а = порядок("0")
для i в диапазоне(а, а+10):
    алфавит.добавить(chr(i))

а = ord("А")
для i в диапазоне(1040, 1072):
    алфавит.добавить(chr(i))
#печать(по алфавиту)
P1 = "НаВашисходящийот1204"
P2 = "ВСеверныйфилиалБанка"
# длина ключа 20
ключ = "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54"


защита от взлома(P1, P2):
    код = []
    для i в диапазоне(20):
        code.append(поалфавиту[(алфавит.индекс(P1[i]) + алфавит.индекс(P2[i])) % len(по алфавиту)])
    #получили известные символы в шаблоне
    печать(код)
    печать(код[16], " и ", код[19])
    р3 = "".присоединиться(код)
    печать(р3)

vzlom(P1, P2)
```

```
['щ', 'С', 'Э', 'в', 'э', 'ш', 'ю', 'Ж', 'ч', 'ш', '7', '4', 'р', 'й', 'щ', 'У', '1', 'Е', 'А', '4']
1 и 4
щСЭвэшюЖчш74рйщУ1ЕА4
```

```
деф шифр(P1):
    # создаем алфавит
    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
    "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":7
    }
    # меняем местами ключ и значение, такой словарь понадобится в будущем
    dict2 = {v: k для k, v в диктантах.элементы()}
    текст = P1
    gamma = input("Введите гамму(на русском языке! Да и пробелы тоже нельзя! Короче, только символы из dict")
    listofdigitsoftext = list() # сюда будем записывать числа букв из текста
    listofdigitsofgamma = list() # для гаммы
    # запишем числа в список
    для меня в тексте:
        listofdigitsoftext.append(дикт[i])
    print("Числа текста", listofdigitsoftext)
    # то же самое сделаем с гаммой
    для меня в гамме:
        списокцифр игры.добавить(диктует[i])
    print("числа гаммы", listofdigitsofgamma)
    listofdigitsresult = list() # сюда будем записывать результат
    сн = 0
    для меня в тексте:
        пробовать:
            а = диктанты[i] + список цифр игры[ch]
        кроме:
            сн = 0
            а = диктанты[i] + список цифр игры[ch]
        если а >> 75:
            а = а%75
            печать(а)
        сн += 1
        listofdigitsresult.добавить(а)
```

```
            a = диктанты[i] + список цифр игры[ch]
        если a >> 75:
            a = a%75
            печать(a)
        сн += 1
        listofdigitsresult.добавить(a)
    print("Числа зашифрованного текста", listofdigitsresult)
    # теперь обратно числа представим в виде букв
    зашифрованный текст= ""
    для i в listofdigitsresult:
        зашифрованный текст+= дикт2[i]
    print("Зашифрованный текст: ", textencrypted)
    # теперь приступим к реализации алгоритма дешифровки
    список цифр= список()
    for i in textencrypted:
        listofdigits.append(dicts[i])
    ch = 0
    listofdigits1 = list()
    for i in listofdigits:
        try:
            a = i - listofdigitsofgamma[ch]
        except:
            ch=0
            a = i - listofdigitsofgamma[ch]
        if a < 1:
            a = 75 + a
        listofdigits1.append(a)
        ch += 1
    textdecrypted = ""
    for i in listofdigits1:
        зашифрованный текст+= диктант2[i]
    print("Расшифрованный текст", textdecrypted)

шифр(P1)
```

```
Введите гамму(на русском языке! Да и пробелы тоже нельзя! Короче, только символы из dictщСЭвэшюЖчш74рйщУ1ЕА4
Числа текста [47, 1, 35, 1, 26, 10, 19, 23, 16, 5, 32, 27, 10, 11, 16, 20, 66, 67, 75, 69]
числа гаммы [27, 51, 41, 3, 31, 26, 32, 40, 25, 26, 72, 69, 18, 11, 27, 53, 66, 38, 33, 69]
1
29
21
57
30
33
63
Числа зашифрованного текста [74, 52, 1, 4, 57, 36, 51, 63, 41, 31, 29, 21, 28, 22, 43, 73, 57, 30, 33, 63]
Зашифрованный текст: 9ТагЧГСЭЭэыуъфЙ8ЧьАЭ
Расшифрованный текст НаВашисходящийот1204
```

```
[ ]:
```

Заключение:

В ходе выполнения лабораторной работы я изучил теорию и освоил на практике применение режима однократного гаммирования.

Ответы на контрольные вопросы:

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа?

Можно заметить, что если дистанция между i-м и j-м символами ключевого слова равна d, то дистанция между соответствующими символами исходного текста составляет -d. Следовательно, если нам удастся найти ключевое слово, то мы сможем свести шифр Виженера к шифру простой замены: каждая буква исходного текста будет заменена на другую, при этом соответствие букв будет взаимно-однозначным. Взломать такой шифр не составит труда.

2. Что будет при повторном использовании ключа при шифровании текста?

Если множество шифрующих преобразований {ja} достаточно велико, то можно обеспечить стойкость шифрования даже при повторном использовании ключей. Для этого достаточно, чтобы в множестве {ja} содержались

преобразования, переводящие любую пару букв открытого текста в любую пару букв шифрованного текста. Тогда по паре текстов, зашифрованных на одном и том же ключе, нельзя получить информацию об открытых текстах, поскольку любой паре букв шифртекстов может соответствовать произвольная пара букв открытых текстов.

3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?

В режиме однократного гаммирования используется сложение по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста. Особенность заключается в том, что этот алгоритм шифрования является симметричным. Поскольку двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, шифрование и расшифрование выполняется одной и той же программой.

4. Перечислите недостатки шифрования одним ключом двух открытых текстов.

Недостатки: Размер ключевого материала должен совпадать с размером передаваемых сообщений. Также необходимо иметь эффективные процедуры для выработки случайных равновероятных двоичных последовательностей и специальную службу для развоза огромного количества ключей. А ещё, если одну и ту же гамму использовать дважды для разных сообщений, то шифр из совершенно стойкого превращается в «совершенно нестойкий» и допускает дешифрование практически вручную.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Достоинства: С точки зрения теории криптоанализа метод шифрования случайной однократной равновероятной гаммой той же длины, что и открытый текст, является не вскрываемым. Кроме того, даже раскрыв часть сообщения, дешифровщик не сможет хоть сколько-нибудь поправить положение - информация о вскрытом участке гаммы не дает информации об остальных ее частях. К достоинствам также можно отнести простоту реализации и удобство применения.