Отчет по лабораторной работе №8

Информационная безопасноть

Астафьева Анна Андреевна НПИбд-01-18

Содержание

List of Figures

1 Цель работы

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

2 Теоретическое описание

Исходные данные.

Две телеграммы Центра:

P_1 = НаВашисходящийот1204

P_2 = ВСеверныйфилиалБанка

Ключ Центра длиной 20 байт:

K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 OB B2 70 54

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$C_1 = P_1 \oplus K_1$$

$$C_2 = P_2 \oplus K_2$$

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. Таким образом, злоумышленник получает возможность определить те символы сообщения P_2, которые находятся

на позициях известного шаблона сообщения P_1.

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

В соответствии с логикой сообщения P_2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P_2. Затем используется подстановка вместо P_1 полученных на предыдущем шаге новых символов сообщения P_2. И так далее.
Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

3 Выполнение лабораторной работы

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста.
Исходные данные.
Две телеграммы Центра:

P_1 = НаВашисходящийот1204

P_2 = ВСеверныйфилиалБанка

1. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P_1 и P_2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C_1 и C_2 обоих текстов P_1 и P_2 при известном ключе ;

Используем функции из лабораторной работы №7:

1.1. Функция *encryption* с помощью однократного гаммирования из сообщения и ключа получает шифротекст (рис. 3.1).

```
In [1]: def encryption (message, key):
            cypher=[]
            cypher_1=[]
            if len(message)>len(key):
                for _ in range(len(message)-len(key)):
                    key.append('00')
            for i, j in zip(message, key):
                c = hex((int(i,16)^int(j,16)))[2:]
                c = (c, '0'+c)[len(c)==1]
                cypher.append(c)
                cypher_1.append(chra(int(i,16)^int(j,16)))
            return cypher, cypher_1
```

Figure 3.1: Код функции *encryption*

1.2. Функция *to_hex*, трансформирующая текст в шестнадцатиричное представление (рис. 3.2).

```
In [2]: def to_hex (text):
            hexa=[]
            for i in text:
                temp=hex(orda(i))[2:]
                temp = (temp, '0'+temp)[len(temp)==1]
                hexa.append(temp)
            return hexa
```

Figure 3.2: Код функции *to_hex*

Написаны еще несколько функций:

1.3. Функция *to_text*, трансформирующая текст в шестнадцатиричном представлении в символьное (рис. 3.3)

```
In [3]: def to_text (stroka):
            stroka=stroka.split()
            text=[]
            for i in stroka:
                text.append(chra(int(i,16)))
            return text
```

Figure 3.3: Код функции *to_text*

1.4. Функция *chra*, преобразовывающая число в символ (рис. 3.4)

```
In [4]: ascii_tabl=[]
        for i in range(32,128):
            ascii_tabl.append(chr(i))
        for i in range(1040,1104):
            ascii_tabl.append(chr(i))

        def chra(num):
            return ascii_tabl[num%len(ascii_tabl)]
```

Figure 3.4: Код функции *chra*

1.5. Функция *orda*, преобразовывающая символ в число (рис. 3.5)

```
In [5]: def orda(smb):
            return ascii_tabl.index(smb)
```

Figure 3.5: Код функции *orda*

Шифруем оба сообщения (рис. 3.6):

```
In [6]: P1='НаВашисходящийот1204'
        P2='ВСеверныйфилиалБанка'

        K='05 0c 17 7f 0e 4e 37 d2 94 10 09 2e 22 57 ff c8 0b b2 70 54'
        print('Исходные сообщения:')
        print('P1:\nшестн.: ',' '.join(to_hex(P1)),'\nсимв.:  ', '  '.join([i for i in P1]))
        print('P2:\nшестн.: ',' '.join(to_hex(P2)),'\nсимв.:  ', '  '.join([i for i in P2]))
        print('\nКлюч:\nшестн.: ',K,'\nсимв.:  ', '  '.join(to_text(K)))
        cypher_hex1, cypher1 = encryption(to_hex(P1), K.split())
        cypher_hex2, cypher2 = encryption(to_hex(P2), K.split())
        print('\nЗашифрованные сообщения:')
        print('P1:\nшестн.: ',' '.join(cypher_hex1),'\nсимв.:  ','  '.join(cypher1))
        print('P2:\nшестн.: ',' '.join(cypher_hex2),'\nсимв.:  ','  '.join(cypher2))

Исходные сообщения:
P1:
шестн.:  6d 80 62 80 98 88 91 95 8e 84 9f 99 88 89 8e 92 11 12 10 14
симв.:   Н  а  В  а  ш  и  с  х  о  д  я  щ  и  й  о  т  1  2  0  4
P2:
шестн.:  62 71 85 82 85 90 8d 9b 89 94 88 8b 88 80 8b 61 80 8d 8a 80
симв.:   В  С  е  в  е  р  н  ы  й  ф  и  л  и  а  л  Б  а  н  к  а

Ключ:
шестн.:  05 0c 17 7f 0e 4e 37 d2 94 10 09 2e 22 57 ff c8 0b b2 70 54
симв.:   %  ,  7  Я  .  n  W  R  ф  0  )  N  B  w     H  +  2  P  t

Зашифрованные сообщения:
P1:
шестн.:  68 8c 75 ff 96 c6 a6 47 1a 94 96 b7 aa de 71 5a 1a a0 60 40
симв.:   И  м  X     ц  F  &  g  :  ф  ц  7  *  ^  C  z  :     A  `
P2:
шестн.:  67 7d 92 fd 8b de ba 49 1d 84 81 a5 aa d7 74 a9 8b 3f fa d4
симв.:   З  Э  т  }  л  ^  :  i  =  д  б  %  *  W  Ф  )  л  _  z  T
```

Figure 3.6: Получение шифротекста сообщений

2. Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Далее преположим ситуацию, что злоумышленнику каким-то образом удалось заполучить оба сообщения в зашифрованном виде (рис. 3.7):

```
In [7]: print('Зашифрованные сообщения у злоумышленника:')
        print('P1:\nшестн.: ','  '.join(cypher_hex1))
        print('P2:\nшестн.: ','  '.join(cypher_hex2))

Зашифрованные сообщения у злоумышленника:
P1:
шестн.:  68  8c  75  ff  96  c6  a6  47  1a  94  96  b7  aa  de  71  5a  1a  a0  60  40
P2:
шестн.:  67  7d  92  fd  8b  de  ba  49  1d  84  81  a5  aa  d7  74  a9  8b  3f  fa  d4
```

Figure 3.7: Злоумышленник получил шифротексты

Складывая по модулю шифротексты можно получить гамму (рис. 3.8):

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

```
In [8]: gamma=[]
        for i in range(len(cypher1)):
            temp=hex(int(cypher_hex1[i],16)^int(cypher_hex2[i],16))[2:]
            temp = (temp, '0'+temp)[len(temp)==1]
            gamma.append(temp)
        print('Гамма:\nшестн.: ',' '.join(gamma),'\nсимв.:  ', '  '.join(to_text(' '.join(gamma))))

        Гамма:
        шестн.:  0f f1 e7 02 1d 18 1c 0e 07 10 17 12 00 09 05 f3 91 9f 9a 94
        симв.:   /  q  g  "  =  8  <  .  '  0  7  2     )  %  s  с  я  ъ  ф
```

Figure 3.8: Получение гаммы

Допустим, одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику известен формат телеграммы P_1. Ему известны в телеграмме P_1 выделенны жирным части:

P_1 = **НаВаш**исходящий**от**1204.

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

P_2 = **ВСеве**рныйфилиа**лБ**анка.

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

P_2 = **ВСеверный**филиа**лБ**анка.

Затем используется подстановка полученных на предыдущем шаге новых символов сообщения P_2. И так далее.

Работа описанного выше алгоритма реализована в программе (рис. 3.9), (рис. 3.10), (рис. 3.11).

```
In [9]: k='1'

        print('Расшифровка...\n')
        while k!='0':
            while 1:
                P=input('Введите известную часть сообщения, заменяя неизвестные символы вопросительным знаком (размер сообщения - %s):\n'
                if len(P)==len(gamma):
                    break
                else:
                    print('Неправильный размер. Попробуйте снова.\n')

            N_P=1
            while 1:
                N_P=input("Номер сообщения (1 или 2):\n")
                if N_P=='1' or N_P=='2':
                    break
                else:
                    print('Неправильный номер.\n')


            print('\nИзвестная часть сообщения P%s:'%N_P)
            print('шестн.: ',' '.join(to_hex(P)),'\nсимв.:  ', '  '.join([i for i in P]))

            cypher_hex, cypher = encryption(gamma, to_hex(P))
            print('\nРасшифровываем сообщение P%s:'%((1,2)[N_P=='1']))
            print('P%s'%((1,2)[N_P=='1']),':\nшестн.: ',' '.join(cypher_hex),'\nсимв.:  ','  '.join(cypher))
            k=input('\nПродолжить? (0 - нет, 1 - да)\n')
```

Figure 3.9: Взлом сообщений

```
Расшифровка...

Введите известную часть сообщения, заменяя неизвестные символы вопросительным знаком (размер сообщения - 20):
НаВаш????????от????
Номер сообщения (1 или 2):
1

Известная часть сообщения P1:
шестн.:  6d 80 62 80 98 1f 1f 1f 1f 1f 1f 1f 1f 8e 92 1f 1f 1f 1f
симв.:   Н  а  В  а  ш  ?  ?  ?  ?  ?  ?  ?  ?  о  т  ?  ?  ?  ?

Расшифровываем сообщение P2:
P2 :
шестн.:  62 71 85 82 85 07 03 11 18 0f 08 0d 1f 16 8b 61 8e 80 85 8b
симв.:   В  С  е  в  е  '  #  1  8  /  (  -  ?  б  л  Б  о  а  е  л

Продолжить? (0 - нет, 1 - да)
1
Введите известную часть сообщения, заменяя неизвестные символы вопросительным знаком (размер сообщения - 20):
ВСеверный?????лБ????
Номер сообщения (1 или 2):
2

Известная часть сообщения P2:
шестн.:  62 71 85 82 85 90 8d 9b 89 1f 1f 1f 1f 1f 8b 61 1f 1f 1f 1f
симв.:   В  С  е  в  е  р  н  ы  й  ?  ?  ?  ?  ?  л  Б  ?  ?  ?  ?

Расшифровываем сообщение P1:
P1 :
шестн.:  6d 80 62 80 98 88 91 95 8e 0f 08 0d 1f 16 8e 92 8e 80 85 8b
симв.:   Н  а  В  а  ш  и  с  х  о  /  (  -  ?  б  о  т  о  а  е  л

Продолжить? (0 - нет, 1 - да)
1
```

Figure 3.10: Взлом сообщений

```
Введите известную часть сообщения, заменяя неизвестные символы вопросительным знаком (размер сообщения - 20):
НаВашисходящийот????
Номер сообщения (1 или 2):
1

Известная часть сообщения P1:
шестн.:  6d 80 62 80 98 88 91 95 8e 84 9f 99 88 89 8e 92 1f 1f 1f 1f
симв.:   Н  а  В  а  ш  и  с  х  о  д  я  щ  и  й  о  т  ?  ?  ?  ?

Расшифровываем сообщение P2:
P2 :
шестн.:  62 71 85 82 85 90 8d 9b 89 94 88 8b 88 80 8b 61 8e 80 85 8b
симв.:   В  С  е  в  е  р  н  ы  й  ф  и  л  и  а  л  Б  о  а  е  л

Продолжить? (0 - нет, 1 - да)
1
Введите известную часть сообщения, заменяя неизвестные символы вопросительным знаком (размер сообщения - 20):
ВСеверныйфилиалБанка
Номер сообщения (1 или 2):
2

Известная часть сообщения P2:
шестн.:  62 71 85 82 85 90 8d 9b 89 94 88 8b 88 80 8b 61 80 8d 8a 80
симв.:   В  С  е  в  е  р  н  ы  й  ф  и  л  и  а  л  Б  а  н  к  а

Расшифровываем сообщение P1:
P1 :
шестн.:  6d 80 62 80 98 88 91 95 8e 84 9f 99 88 89 8e 92 11 12 10 14
симв.:   Н  а  В  а  ш  и  с  х  о  д  я  щ  и  й  о  т  1  2  0  4

Продолжить? (0 - нет, 1 - да)
0
```

Figure 3.11: Взлом сообщений

4 Контрольные вопросы

1. Как, зная один из текстов (P_1 или P_2), определить другой, не зная при этом ключа?
   По формуле $C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2$

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

3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?
   по формулам режима однократного гаммирования:

$$C_1 = P_1 \oplus K_1$$

$$C_2 = P_2 \oplus K_2$$

4. Перечислите недостатки шифрования одним ключом двух открытых текстов.

- ключ, попав не в те руки, даст возможность злоумышленнику расшифровать оба текста;

- можно расшифровать с помощью открытого текста другие известные шифротексты;

- можно узнать часть текста, используя заранее известный шаблон и формат другого текста.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

- скорость шифрования выше;

- простой алгоритм шифрования;

- шифротекст сильно меняется, если изменяется ключ или открытый текст.

5 Выводы

На основе проделанной работы освоила на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.