Шифр гаммирования

Рябцева Маргарита Михайловна НБИ-01-19 18 октября, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

Цель лабораторной работы

Изучение алгоритма шифрования гаммированием

Выполнение лабораторной

работы

Гаммирование

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.

Гаммирование

Наложение (или снятие) гаммы на блок сообщения в рассматриваемом нами стандарте реализуется с помощью операции побитного сложения по модулю 2 (XOR). То есть при шифровании сообщений каждый блок открытого сообщения ксорится с блоком криптографической гаммы, длина которого должна соответствовать длине блоков открытого сообщения. При этом, если размер блока исходного текста меньше, чем размер блока гаммы, блок гаммы обрезается до размера блока исходного текста (выполняется процедура усечения гаммы).

Алгоритм



Figure 1: Шифрование

Алгоритм



Figure 2: Дешифровка

В аддитивных шифрах символы исходного сообщения заменяются числами, которые складываются по модулю с числами гаммы. Ключом шифра является гамма, символы которой последовательно повторяются. Перед шифрованием символы сообщения и гаммы заменяются их номерами в алфавите и само кодирование выполняется по формуле

$$Ci = (Ti + Gi) mod N$$

Пример работы алгоритма

<i>T</i>	K	A	Ф	E	Д	P	A		С	И	С	Т	E	М		И	н	Φ	o	P	М	A	T	И	K	И
G	С	И	М	В	o	Л	С	И	М	В	0	Л	C	И	М	В	0	Л	С	И	M	В	o	Л	C	И
<i>T</i>	12	1	22	6	5	18	1	34	19	10	19	20	6	14	34	10	15	22	16	18	14	1	20	10	12	10
G	19	10	14	3	16	13	19	10	14	3	16	13	19	10	14	3	16	13	19	10	14	3	16	13	19	10
T+G	31	11	36	9	21	31	20	44	33	13	35	33	25	24	48	13	31	35	35	28	28	4	36	23	31	20
mod N	31	11	36	9	21	31	20	0	33	13	35	33	25	24	4	13	31	35	35	28	28	4	36	23	31	20
<i>0</i> → <i>N</i>	31	11	36	9	21	31	20	44	33	13	35	33	25	24	4	13	31	35	35	28	28	4	36	23	31	20
C	Э	Й	1	3	У	Э	T	9	Я	Л	0	Я	ч	Ц	Γ	Л	Э	0	0	ъ	ъ	Γ	1	x	Э	T

Figure 3: Работа алгоритма гаммирования

Пример работы программы

```
autput#2.txt 🗵 📙 test.txt 🗵
                                                         ■ output.txt ☑
                 Hello world
                                                                POXINUMINARY PERGRAPATION NAK
          length: 11 lines: 1
                               In:1 Col:1 Pos:1
                                                                  Windows (CR LF) UTF-8
                                                                                                 INS
                   for line in file:
                      res str = ""
                      # Посимвольное гаммирование
                      for i in range(len(line)):
                          if main offset!=0 and main offset % len(string) == 0:
                              str offset += 1
                          res_str += chr(ord(line[i])^ord(string[(main_offset+str_offset)%len(string)]))
                          main offset += 1
                      # Запись в файл
                      filewrite.write(res str)
       def main():
           Main function
           string = input("Введите строку:\n")
           # strina = "test strina"
           xor cipher("test.txt","output.txt", string)
           xor cipher("output.txt", "output#2.txt", string)
           print("DONE!")
           input("Нажмите Enter для завершения программы")
In [2]: if name == " main ":
           main()
        Введите строку:
        KevString
       DONE!
       Нажмите Enter для завершения программы
```

Выводы



Изучили алгоритм шифрования с помощью гаммирования