

Шифр гаммирования

Агоннудэ Месседэ Мишель НКНбд-01-19

19 октября, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

Цель лабораторной работы

Изучение алгоритма шифрования гаммированием

Выполнение лабораторной работы

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.

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 XORится 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).



Figure 1: Шифрование

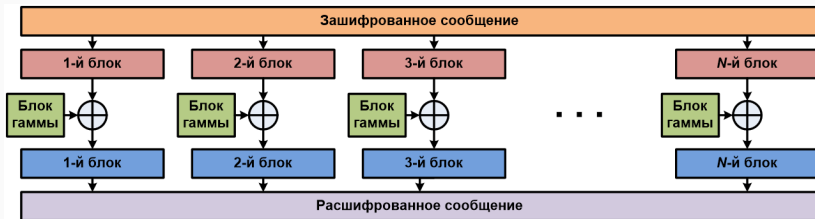


Figure 2: Дешифровка

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

$$C_i = (T_i + G_i) \bmod N$$

Пример работы алгоритма

<i>T</i>	К	А	Ф	Е	Д	Р	А		С	И	С	Т	Е	М		И	Н	Ф	О	Р	М	А	Т	И	К	И
<i>G</i>	С	И	М	В	О	Л	С	И	М	В	О	Л	С	И	М	В	О	Л	С	И	М	В	О	Л	С	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
<i>G</i>	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
<i>T+G</i>	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
<i>mod N</i>	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 → 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
<i>C</i>	Э	Й	1	З	У	Э	Т	9	Я	Л	0	Я	Ч	Ц	Г	Л	Э	0	0	Ъ	Ъ	Г	1	Х	Э	Т

Figure 3: Работа алгоритма гаммирования

Пример работы программы

```
In [4]: main()

Введите текст гаммыпроверкакода
Введите текст для кодированияпроверкакода
Числа текста [17, 18, 16, 3, 6, 18, 12, 1, 12, 16, 5, 1]
Числа гаммы [17, 18, 16, 3, 6, 18, 12, 1, 12, 16, 5, 1]
Числа шифровки: [1]
шифровка: а
расшифровка: п
Числа шифровки: [1, 3]
шифровка: ав
расшифровка: пр
Числа шифровки: [1, 3, 32]
шифровка: авя
расшифровка: про
Числа шифровки: [1, 3, 32, 6]
шифровка: авяе
расшифровка: пров
Числа шифровки: [1, 3, 32, 6, 12]
шифровка: авяек
расшифровка: прове
Числа шифровки: [1, 3, 32, 6, 12, 3]
шифровка: авяекв
расшифровка: провер
Числа шифровки: [1, 3, 32, 6, 12, 3, 24]
шифровка: авяеквц
расшифровка: проверк
Числа шифровки: [1, 3, 32, 6, 12, 3, 24, 2]
шифровка: авяеквцб
расшифровка: проверка
Числа шифровки: [1, 3, 32, 6, 12, 3, 24, 2, 24]
шифровка: авяеквцбц
расшифровка: проверкак
Числа шифровки: [1, 3, 32, 6, 12, 3, 24, 2, 24, 32]
шифровка: авяеквцбця
расшифровка: проверкако
Числа шифровки: [1, 3, 32, 6, 12, 3, 24, 2, 24, 32, 10]
шифровка: авяеквцбцяи
расшифровка: проверкакод
Числа шифровки: [1, 3, 32, 6, 12, 3, 24, 2, 24, 32, 10, 2]
```

Выводы

Изучили алгоритм шифрования с помощью гаммирования