

Информационная безопасность. Отчет по лабораторной работе № 8

**Элементы криптографии. Шифрование (кодирование) различных
исходных текстов одним ключом**

Горбунова Ярослава Михайловна

Содержание

1	Цель работы	5
2	Указание к работе	6
3	Выполнение лабораторной работы	8
3.1	Контрольные вопросы	11
4	Выводы	12
5	Список литературы	13

List of Figures

2.1	Общая схема шифрования двух различных текстов одним ключом	6
3.1	Программа (1)	9
3.2	Программа (2)	9
3.3	Программа (3)	10
3.4	Программа (4)	10
3.5	Вывод работы программы	11

List of Tables

1 Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом [1].

2 Указание к работе

Исходные данные. Две телеграммы Центра:

P1 = НаВашисходящийот1204 P2 = ВСеверныйфилиалБанка

Ключ Центра длиной 20 байт:

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

Режим шифрования однократного гаммирования одним ключом двух видов открытого текста реализуется в соответствии со схемой, приведённой на fig. 2.1.

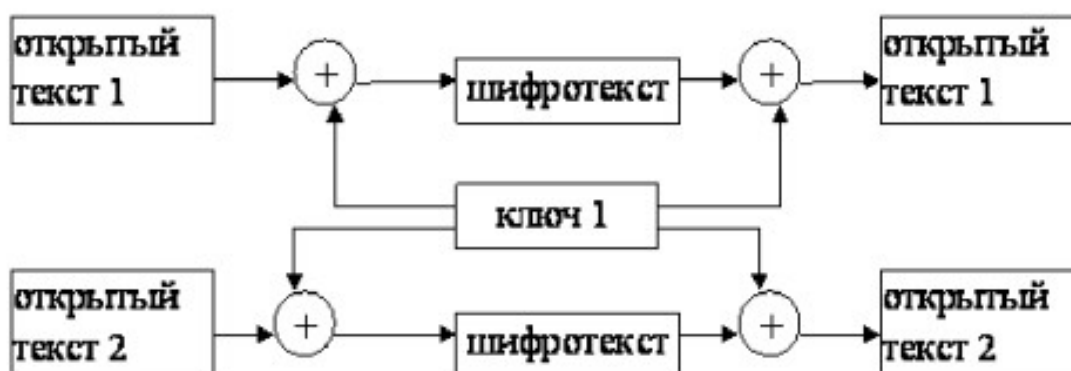


Figure 2.1: Общая схема шифрования двух различных текстов одним ключом

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$C1 = P1 \boxtimes K, C2 = P2 \boxtimes K. (8.1)$$

Открытый текст можно найти в соответствии с (8.1), зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства (8.1) складываются по модулю 2. Тогда с учётом свойства операции XOR

$$1 \otimes 1 = 0, 1 \otimes 0 = 1 \quad (8.2)$$

получаем:

$$C1 \otimes C2 = P1 \otimes K \otimes P2 \otimes K = P1 \otimes P2.$$

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 $C1 \otimes C2$ (известен вид обеих шифровок). Тогда зная $P1$ и учитывая (8.2), имеем:

$$C1 \otimes C2 \otimes P1 = P1 \otimes P2 \otimes P1 = P2. \quad (8.3)$$

Таким образом, злоумышленник получает возможность определить те символы сообщения $P2$, которые находятся на позициях известного шаблона сообщения $P1$. В соответствии с логикой сообщения $P2$, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения $P2$. Затем вновь используется (8.3) с подстановкой вместо $P1$ полученных на предыдущем шаге новых символов сообщения $P2$. 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, прочитав оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Для выполнения работы была написана программа (fig. 3.1 - fig. 3.5) с помощью языка программирования C++, которая получает на вход два открытых текста "NaVasishodysiyot1204", "VSevernyifilialBanka" и ключ "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54", затем шифрует открытые тексты методом однократного гаммирования и получает два шифротекста. После этого предполагаем два случая. В первом неизвестен открытый текст 2, во втором - открытый текст 1. Методом сложения по модулю 2, однократного гаммирования, определяется открытый текст 2 и текст 1 для случаев соответственно. Определение открытых текстов происходит без ключа, не осуществляются попытки его определения.


```

1  /*
2  Gorbunova Y.M., NFI-01-19, 2022
3  :
4  Single gamming
5  Principle:
6  1) text ^ gamma = ciphertext;
7  2) ciphertext ^ gamma = text;
8  3) gamma = ciphertext ^ text;
9  where "^" is additions modulo 2 (XOR)
10 :
11 ciphertext_1 = text_1 ^ gamma
12 ciphertext_2 = text_2 ^ gamma
13 :
14 4) ciphertext_1 ^ ciphertext_2 ^ text_1 = text_2
15 :
16   ciphertext_1 ^ ciphertext_2 ^ text_2 = text_1
17 */
18
19 #include <iostream>
20 #include <string>
21 #include <cstring>
22 #include <bitset>
23
24 using namespace std;
25
26 const int m = 1024; // array dimension
27 const int n = 8; // bitset dimension
28
29 // Print for unmodified text
30 void print_bitset_text(char arr[]) {
31     for (unsigned int i = 0; i < strlen(arr); i++)
32         cout << bitset<n>((unsigned char)arr[i]) << " ";
33 }
34

```

Figure 3.1: Программа (1)

```

35 // Print for modified text
36 void print_bitset_gamma(char arr[], char text[]) {
37     for (unsigned int i = 0; i < strlen(text); i++)
38         cout << bitset<n>((unsigned char)arr[i]) << " ";
39 }
40
41 int main()
42 {
43     // Introduce variables
44     //char text_1[m] = "НаВашисходящийот1204";
45     char text_1[m] = "NaVasishodysiyot1204";
46     //char text_2[m] = "ВСеверныйфилиалБанка";
47     char text_2[m] = "VSevernyifilialBanka";
48     int gam[m] = {
49         0x05, 0x0C, 0x17, 0x7F, 0x0E, 0x4E, 0x37, 0xD2, 0x94, 0x10,
50         0x09, 0x2E, 0x22, 0x57, 0xFF, 0xC8, 0x0B, 0xB2, 0x70, 0x54
51     }; // given key (int)
52     char gamma[m]; // key (char)
53     char gam_text_1[1024], gam_text_2[1024]; // ciphertexts
54
55     // Convert from integer to char
56     for (unsigned int i = 0; i < size(gam); i++) {
57         gamma[i] = (char)gam[i];
58     }
59
60     cout << "-----" << endl << "Part 1" << endl;
61
62     cout << "    Bitset of the key (gamma) :\n";
63     print_bitset_text(gamma);
64     cout << endl;
65
66     // Text 1
67     cout << "    Text 1 for single gamming: \n" << text_1 << endl;
68

```

Figure 3.2: Программа (2)

```

69     cout << "    Bitset of the Text 1 :\n";
70     print_bitset_text(text_1);
71     cout << endl;
72
73     cout << "    Ciphertext computation...\n";
74
75     // Convert text to ciphertext
76     for (unsigned int i = 0; i < strlen(text_1); i++) {
77         gam_text_1[i] = text_1[i] ^ gamma[i];
78     }
79     cout << "    Bitset of the Ciphertext (gam_text_1) :\n";
80     print_bitset_gamma(gam_text_1, text_1);
81     cout << endl;
82
83     // Text 2
84     cout << "    Text 2 for single gamming: \n" << text_2 << endl;
85
86     cout << "    Bitset of the Text 2 :\n";
87     print_bitset_text(text_2);
88     cout << endl;
89
90     cout << "    Ciphertext computation...\n";
91
92     // Convert text to ciphertext
93     for (unsigned int i = 0; i < strlen(text_2); i++) {
94         gam_text_2[i] = text_2[i] ^ gamma[i];
95     }
96     cout << "    Bitset of the Ciphertext (gam_text_2) :\n";
97     print_bitset_gamma(gam_text_2, text_2);
98
99     cout << endl;
100    cout << "-----" << endl;
101    cout << endl << "-----" << endl << "Part 2" << endl;
102

```

Figure 3.3: Программа (3)

```

103    cout << "    1) Accept Text 2 is unknown. Define Text 2 via cybertext 1, cybertest 2 and Test 1: \n";
104    cout << "    Text 2 computation...\n";
105    // Define text_2
106    cout << "    Check of correct computation work...\n    Obtained Text 2 after single gamming" << endl;
107    for (unsigned int i = 0; i < strlen(text_1); i++) {
108        cout << static_cast<char>(bitset<n>((unsigned char)(gam_text_1[i] ^ gam_text_2[i] ^ text_1[i])).to_ulong() + 256);
109    }
110    cout << endl;
111
112    cout << "    2) Accept Text 1 is unknown. Define Text 1 via cybertext 1, cybertest 2 and Test 2: \n";
113    cout << "    Text 1 computation...\n";
114    // Define text_2
115    cout << "    Check of correct computation work...\n    Obtained Text 1 after single gamming" << endl;
116    for (unsigned int i = 0; i < strlen(text_2); i++) {
117        cout << static_cast<char>(bitset<n>((unsigned char)(gam_text_1[i] ^ gam_text_2[i] ^ text_2[i])).to_ulong() + 256);
118    }
119    cout << endl << "-----" << endl;
120
121

```

Figure 3.4: Программа (4)

```

-----
Part 1
  Bitset of the key (gamma) :
00000101 00001100 00010111 01111111 00001110 01001110 00110111 11010010 10010100 00010000 00001001 00101110 00100010 01010111 11111111 11001000 00001011 10110010 01110000 01010100
  Text 1 for single gamming:
NaVasishodysiyot1204
  Bitset of the Text 1 :
01001110 01100001 01010110 01100001 01110011 01101001 01110011 11010000 01101111 01100100 01111001 01110011 01101001 01111001 01101111 01110100 00110001 00110010 00110000 00110100
  Ciphertext computation...
  Bitset of the Ciphertext (gam_text_1) :
01001011 01011011 01000001 00011110 01111101 01000100 10111010 11111011 01110100 01110000 01011110 01001011 00101110 10010000 10111100 00111010 10000000 01000000 01100000
  Text 2 for single gamming:
VSevernyifilialBanka
  Bitset of the Text 2 :
01010110 01010011 01100101 01110110 01100101 01110010 01101110 01111001 01101001 01100110 01101001 01101110 01101001 01100001 01101100 01000010 01100001 01101110 01101011 01100001
  Ciphertext computation...
  Bitset of the Ciphertext (gam_text_2) :
01010011 01011111 01110010 00001001 01101011 00111100 01011001 10101011 11111101 01110110 01110000 01000010 01001011 00110110 10010011 10001010 01101010 11011100 00011011 00110101
-----
Part 2
  1) Accept Text 2 is unknown. Define Text 2 via cybertext 1, cybertest 2 and Test 1:
  Text 2 computation...
  Check of correct computation work...
  Obtained Text 2 after single gamming
VSevernyifilialBanka
  2) Accept Text 1 is unknown. Define Text 1 via cybertext 1, cybertest 2 and Test 2:
  Text 1 computation...
  Check of correct computation work...
  Obtained Text 1 after single gamming
NaVasishodysiyot1204
-----

```

Figure 3.5: Вывод работы программы

3.1 Контрольные вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа? – По формулам: $C1 \boxtimes C2 \boxtimes P1 = P2$, $C1 \boxtimes C2 \boxtimes P2 = P1$.
2. Что будет при повторном использовании ключа при шифровании текста? – Расшифровка текста.
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов? – Ключ применяется к каждому из текстов в отдельности, получаются два различных шифротекста.
4. Перечислите недостатки шифрования одним ключом двух открытых текстов. – При наличии минимум двух шифротекстов и хотябы одного открытого текста можно получить другой открытый текст даже не имея ключа.
5. Перечислите преимущества шифрования одним ключом двух открытых текстов. – Нет необходимости в хранении двух последовательностей символов ключа.

4 Выводы

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

5 Список литературы

1. Методические материалы курса