# Информационная безопасность. Лабораторная работа № 7 на тему “Элементы криптографии. Однократное гаммирование”

Горбунова Ярослава Михайловна

RUDN University, Moscow, Russian Federation

# Содержание

# Содержание

# Цели и задачи

# Цели и задачи

Освоить на практике применение режима однократного гаммирования

# Выполнение

# Выполнение

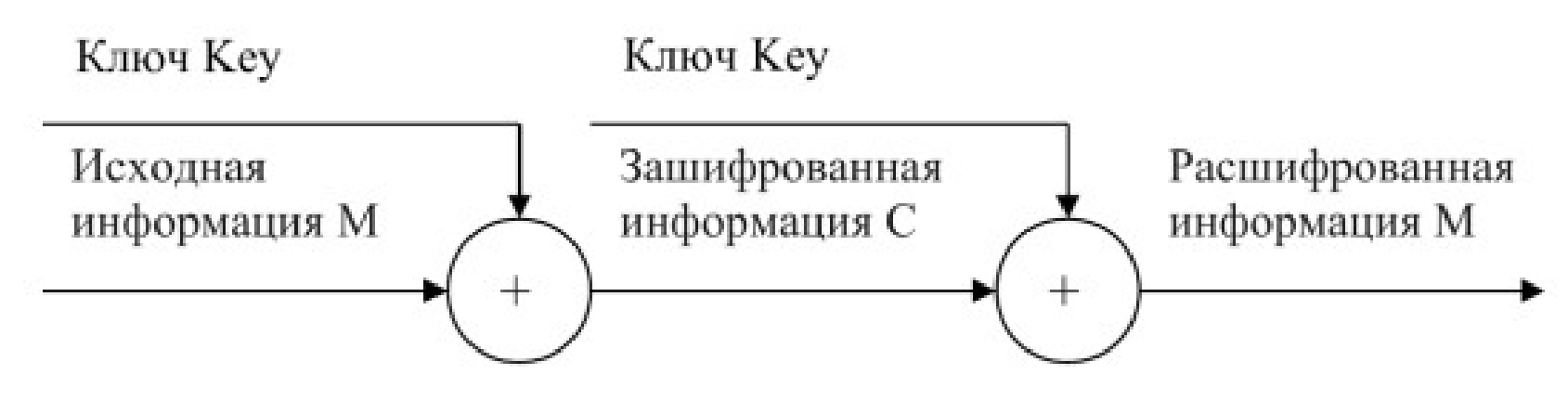


**Figure 1:** Схема однократного использования Вернама

# Выполнение

$$C_i = P_i \oplus K_i,$$

**Figure 2:** Формула 7.1

$$C_i \oplus P_i = P_i \oplus K_i \oplus P_i = K_i,$$
$$K_i = C_i \oplus P_i.$$

**Figure 3:** Формула 7.2

где C_i — i-й символ получившегося зашифрованного послания, P_i — i-й символ открытого текста, K_i — i-й символ ключа, i = 1, m. Размерности открытого текста и ключа должны совпадать, и полученный шифротекст будет такой же длины

# Выполнение

```
/*
Gorbunova Y.M., NFI-01-19, 2022

Single gamming
Principle:
1) text ^ gamma = ciphertext;
2) ciphertext ^ gamma = text;
3) gamma = ciphertext ^ text;
where "^" is additions modulo 2 (XOR)
*/


#include <iostream>
#include <string>
#include <cstring>
#include <bitset>

using namespace std;

const int m = 1024;  // array dimension
const int n = 8;  // bitset dimension

// Print for unmodified text
void print_bitset_text(char arr[]) {
    for (unsigned int i = 0; i < strlen(arr); i++)
        cout << bitset<n>((unsigned char)arr[i]) << " ";
}

// Print for modified text
void print_bitset_gamma(char arr[], char text[]) {
    for (unsigned int i = 0; i < strlen(text); i++)
        cout << bitset<n>((unsigned char)arr[i]) << " ";
}
```

**Figure 4:** Программа (1)

# Выполнение

```
34
35  int main()
36  {
37      // Introduce variables
38      //char text[m] = "Штирлиц - Вы Герой!!";
39      char text[m] = "Stirlis - Vy Geroy!!";
40      //char target_text[m] = "С Новым Годом друзья";
41      char target_text[m] = "S Novym Godom drysya";
42      int gam[m] = {
43          0x05, 0x0C, 0x17, 0x7F, 0x0E, 0x4E, 0x37, 0xD2, 0x94, 0x10,
44          0x09, 0x2E, 0x22, 0x57, 0xFF, 0xC8, 0x0B, 0xB2, 0x70, 0x54
45      };  // given key
46      char gamma_1[m];  // given key
47      char gamma_2[m];  // target key to obtain target_text from text
48      char gam_text[1024];
49
50      // Convert from integer to char
51      for (unsigned int i = 0; i < size(gam); i++) {
52          gamma_1[i] = (char)gam[i];
53      }
54
55      cout << "----------------------------------------------------------------" << endl << "Part 1" << endl;
56
57      cout << "    Text for single gamming: \n" << text << endl;
58
59      cout << "    Bitset of the Text :\n";
60      print_bitset_text(text);
61      cout << endl;
62
63      cout << "    Bitset of the given key (gamma_1) :\n";
64      print_bitset_text(gamma_1);
65      cout << endl;
66
67      cout << "    Ciphertext computation...\n";
```

**Figure 5:** Программа (2)

# Выполнение

```
68
69      // Convert text to ciphertext
70      for (unsigned int i = 0; i < strlen(text); i++) {
71          gam_text[i] = text[i] ^ gamma_1[i];
72      }
73      cout << "    Bitset of the Ciphertext (gam_text) :\n";
74      print_bitset_gamma(gam_text, text);
75      cout << endl;
76      cout << "-----------------------------------------------------------------" << endl;
77      cout << endl << "-----------------------------------------------------------------" << endl << "Part 2" << endl;
78      cout << "    Target text for single gamming: \n" << target_text << endl;
79      cout << "    Bitset of the Target text :\n";
80      print_bitset_text(target_text);
81      cout << endl;
82      cout << "    Key computation...\n";
83
84      // Find key to obtain target_text from text
85      for (unsigned int i = 0; i < strlen(target_text); i++) {
86          gamma_2[i] = gam_text[i] ^ target_text[i];
87      }
88
89      cout << "    Bitset of the obtained key (gamma_2) :\n";
90      print_bitset_gamma(gamma_2, target_text);
91      cout << endl;
92
93      cout << "    Check of correct computation work...\n    Obtained target text after single gamming" << endl;
94      for (unsigned int i = 0; i < strlen(text); i++) {
95          cout << static_cast<char>(bitset<n>((unsigned char)(gam_text[i] ^ gamma_2[i])).to_ulong() + 256);
96      }
97
98      cout << endl << "-----------------------------------------------------------------" << endl;
99  }
100
```

**Figure 6:** Программа (3)

# Выполнение

```
---------------------------------------------------------------
Part 1
    Text for single gamming:
Stirlis - Vy Geroy!!
    Bitset of the Text :
01010011 01110100 01101001 01110010 01101100 01101001 01110011 00100000 00101101 00100000 01010110 01111001 00100000 01000111 01100101 01110010 01101111 01111001 00100001 00100001
    Bitset of the given key (gamma_1) :
00000101 00001100 00010111 01111111 00001110 01001110 00110111 11010010 10010100 00010000 00001001 00101110 00100010 01010111 11111111 11001000 00001011 10110010 01110000 01010100
    Ciphertext computation...
    Bitset of the Ciphertext (gam_text) :
01010110 01111000 01111110 00001101 01100010 00100111 01000100 11110010 10111001 00110000 01011111 01010111 00000010 00010000 10011010 10111010 01100100 11001011 01010001 01110101
---------------------------------------------------------------

---------------------------------------------------------------
Part 2
    Target text for single gamming:
S Novym Godom drysya
    Bitset of the Target text :
01010011 00100000 01001110 01101111 01110110 01111001 01101101 00100000 01000111 01101111 01100100 01101111 01101101 00100000 01100100 01110010 01111001 01110011 01111001 01100001
    Key computation...
    Bitset of the obtained key (gamma_2) :
00000101 01011000 00110000 01100010 00010100 01011110 00101001 11010010 11111110 01011111 00111011 00111000 01101111 00110000 11111110 11001000 00011101 10111000 00101000 00010100
    Check of correct computation work...
    Obtained target text after single gamming
S Novym Godom drysya
---------------------------------------------------------------
```

**Figure 7:** Вывод работы программы

# Результаты

# Результаты

Освоено на практике применение режима однократного гаммирования

# Список литературы

# Список литературы

1. Методические материалы курса