

# Лабораторная работа №8

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

---

Ким М. А.

23 октября 2023

Российский университет дружбы народов, Москва, Россия

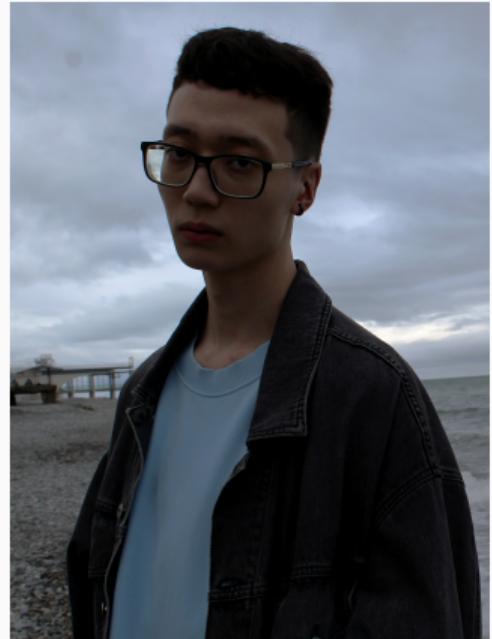
# Информация

---

# Докладчик

---

- Ким Михаил Алексеевич
- студент уч. группы НФИбд-01-20
- Российский университет дружбы народов
- 1032201664@pfur.ru
- <https://github.com/exmanka>



## **Вводная часть**

---

## Актуальность

---

- Необходимость навыков работы с различными ОС, git, Markdown, ЯП.

## Объект и предмет исследования

---

- Операционная система Rocky Linux
- Язык программирования C++
- Язык разметки Markdown
- Однократное гаммирование

## Цели и задачи

---

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

## **Процесс выполнения работы**

---

# Создание программы

```
1 //#include <cstdlib>
2 //#include <iostream>
3 //#include <vector>
4 //#include <string>
5
6
7 using std::cin;
8 using std::cout;
9 using std::endl;
10 using std::string;
11 using std::vector;
12
13
14 void printHex(const vector<unsigned char>& vec)
15 {
16     cout << std::hex << std::uppercase;
17
18     for (size_t i = 0; i < vec.size(); i++)
19     {
20         cout << static_cast<short>(vec[i]) << ' ';
21     }
22
23     cout << std::dec << std::nouppercase << endl;
24 }
25
26 void printHex(const vector<vector<unsigned char>>& vec)
27 {
28     cout << std::hex << std::uppercase;
29
30     for (size_t i = 0; i < vec.size(); i++)
31     {
32         for (size_t j = 0; j < vec[i].size(); j++)
33         {
34             cout << static_cast<short>(vec[i][j]) << ' ';
35         }
36         cout << endl;
37     }
38
39     cout << std::dec << std::nouppercase;
40 }
41
42 void printStrings(const vector<string>& str)
43 {
44     for (size_t i = 0; i < str.size(); i++)
45     {
46         cout << str[i] << endl;
47     }
48 }
```

```
49
50 function<vector<unsigned char>> cryptGaming(const vector<unsigned char> key, const vector<string> inputVector)
51 {
52     size_t elmsize = inputVector[0].size();
53     for (size_t i = 0; i < inputVector.size(); i++)
54     {
55         if (inputVector[i].size() != elmsize)
56         {
57             cout << "Input texts lengths are not the same!" << endl;
58             return vector<vector<unsigned char>>();
59         }
60
61         if (key.size() != elmsize)
62         {
63             cout << "Key length and Input texts length are not the same!" << endl;
64             return vector<vector<unsigned char>>();
65         }
66
67         vector<vector<unsigned char>> outTextVector(inputVector.size(), vector<unsigned char>(elmsize));
68         for (size_t k = 0; k < outTextVector.size(); k++)
69         {
70             for (size_t l = 0; l < elmsize; l++)
71             {
72                 outTextVector[k][l] = inputVector[k][l] ^ key[l];
73             }
74         }
75
76         return outTextVector;
77 }
78
79 function<string> cryptGaming(const vector<vector<unsigned char>> key, vector<vector<unsigned char>> inputVector)
80 {
81     size_t elmsize = inputVector[0].size();
82     for (size_t i = 0; i < inputVector.size(); i++)
83     {
84         if (inputVector[i].size() != elmsize)
85         {
86             cout << "Input texts lengths are not the same!" << endl;
87             return vector<string>();
88         }
89
90         if (key.size() != elmsize)
91         {
92             cout << "Key length and Input texts length are not the same!" << endl;
93             return vector<string>();
94         }
95
96         vector<string> outTextVector(inputVector.size(), string(elmsize, '0'));
97         for (size_t k = 0; k < outTextVector.size(); k++)
98         {
99             for (size_t l = 0; l < elmsize; l++)
100             {
101                 outTextVector[k][l] = inputVector[k][l] ^ key[l];
102             }
103         }
104
105         return outTextVector;
106 }
```

```
107
108 function<backTest> checkString(string templateP1, const vector<unsigned char> encText1, const vector<unsigned char> encText2)
109 {
110     if (templateP1.size() != encText1.size() || templateP1.size() != encText2.size())
111     {
112         cout << "Lengths of encrypted messages are not the same!" << endl;
113         return string("0");
114     }
115
116     string hashedText2(templateP1.size(), '0');
117
118     for (size_t i = 0; i < templateP1.size(); i++)
119     {
120         hashedText2[i] = (encText1[i] ^ encText2[i]) + templateP1[i];
121     }
122
123     return hashedText2;
124 }
125
126
127 int main()
128 {
129     std::setlocale(LC_ALL, "ru");
130
131     const string P1 = "Hello everyone!";
132     const string P2 = "Hello everyone!";
133     const string key = "P1_P2";
134
135     vector<string> messages = { P1, P2 };
136     cout << "Input texts:" << endl << messages;
137
138     const vector<unsigned char> key_c = { 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F };
139
140     cout << "Input key:" << endl;
141
142     vector<vector<unsigned char>> encText(cryptGaming(key, messages));
143     cout << "Encrypted texts:" << endl;
144     printVector(encText);
145
146     vector<string> decText(cryptGaming(key, encText));
147     cout << "Decrypted texts from encrypted texts:" << endl;
148     printVector(decText);
149
150     cout << "HashedText:" << endl;
151     for (size_t i = 0; i < templateP1.size() / 5; i++)
152     {
153         templateP1[i] = '1';
154     }
155
156     cout << "With template:\r\n" << templateP1 << endl;
157
158     string hashed2(backTest(templateP1, encText1, encText2));
159     cout << "Decrypted part of text P2:\r\n" << hashed2 << endl;
160
161     cout << "0" << endl;
162
163     return EXIT_SUCCESS;
164 }
```

# Результат работы программы

```
Input texts:  
Навашисходящийот1204  
ВСеверныйфилиалБанка  
  
Input key:  
5 C 17 7F E 4E 37 D2 94 10 9 2E 22 57 FF C8 B B2 70 54  
  
Encrypted texts:  
C8 EC D5 9F F6 A6 C6 27 7A F4 F6 D7 CA BE 11 3A 3A 80 40 60  
C7 DD F2 9D EB BE DA 29 7D E4 E1 C5 CA B7 14 9 EB SF 9A B4  
  
Decrypted texts from encrypted texts:  
Навашиходящийот1204  
ВСеверныйфилиалБанка  
  
P1 template:  
??????ходящийот1204  
  
Decrypted part of text P2:  
001=""'ныйфилиалБанка
```

## Результаты

---

# Результаты

---

- Выполнены все необходимые задания.

## Вывод

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