# Лабораторная работа 8

Попов Дмитрий Павлович, НФИбд-01-19

# Содержание

**1 Цель работы** **5**

**2 Выполнение лабораторной работы** **6**

**3 Выводы** **10**

**4 Список литературы** **11**

# List of Figures

2.1 encrypt_fuction . . . . . 6
2.2 output_prog . . . . . 7
2.3 finding_mess . . . . . 7
2.4 Main . . . . . 8
2.5 console_output . . . . . 9

# List of Tables

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

дисциплина: Информационная безопасность

Преподователь: Кулябов Дмитрий Сергеевич

Студент: Попов Дмитрий Павлович

Группа: НФИбд-01-19

МОСКВА

2022 г.

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

** Постановка задачи ** Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать 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 этого у меня есть функция позволяющая зашифровывать, расшифровывать данные с помощью сообщения и ключа. А также позволяющая получить ключ (fig. 2.1).

```
7   vector<uint8_t> encrypt(vector<uint8_t> message, vector<uint8_t> key)
8   {
9       if (message.size() != key.size())
10      {
11          return {};
12      }
13      vector<uint8_t> encrypted;
14      for (int i = 0; i < message.size(); i++)
15      {
16          encrypted.push_back(message[i] ^ key[i]);
17      }
18      return encrypted;
19  }
```

Figure 2.1: encrypt_fuction

Функция для вывода результатов (fig. 2.2)

```
49  void print_bytes(vector<uint8_t> message)
50  {
51      for (const auto& e : message)
52      {
53          cout << hex << unsigned(e) << " ";
54      }
55      cout << endl;
56  }
57
58  void print_text(vector<uint8_t> message)
59  {
60      string str(message.begin(), message.end());
61      cout << str << endl;
62  }
```

Figure 2.2: output_prog

Фунцкия определения текста, зная два шифротекста и оригинальный текст одного из них (fig. 2.3)

```
64  vector<uint8_t> get_message_with_three_pieces(vector<uint8_t> cr1, vector<uint8_t> cr2, vector<uint8_t> msg1)
65  {
66      if (cr1.size() != cr2.size() and cr1.size() != msg1.size())
67      {
68          return {};
69      }
70      vector<uint8_t> msg2;
71      for (int i = 0; i < cr1.size(); i++)
72      {
73          msg2.push_back(cr1[i] ^ cr2[i] ^ msg1[i]);
74      }
75      return msg2;
76  }
```

Figure 2.3: finding_mess

Главная функция (fig. 2.4)

```
78  int main()
79  {
80      string message1 = "message1 in the chat";
81      string message2 = "chat have message2 !";
82      vector<uint8_t> first(message1.begin(), message1.end());
83      vector<uint8_t> second(message2.begin(), message2.end());
84
85      string keystr = "ourkeyisthebestworld";
86      vector<uint8_t> key(keystr.begin(), keystr.end());
87
88      vector<uint8_t> crypt1 = encrypt(first, key);
89      vector<uint8_t> crypt2 = encrypt(second, key);
90
91
92      cout << "Original Message number 1: " << endl;
93      print_text(first);
94      cout << endl << "Original Message number 2: " << endl;
95      print_text(second);
96      cout << endl << "Crypted message number 1: " << endl;
97      print_bytes(crypt1);
98      cout << endl << "Crypted message number 2: " << endl;
99      print_bytes(crypt2);
100
101
102     cout << endl <<"Finding message 2:" << endl;
103     vector<uint8_t> msg_found = get_message_with_three_pieces(crypt1, crypt2, first);
104     print_text(msg_found);
105     return 0;
106 }
```

Figure 2.4: Main

Затем я запускаю программу, получаю два шифротекста для каждого текста при известном ключе. Далее не зная ключа и не стремясь его определить, получаю текст (fig. 2.5)

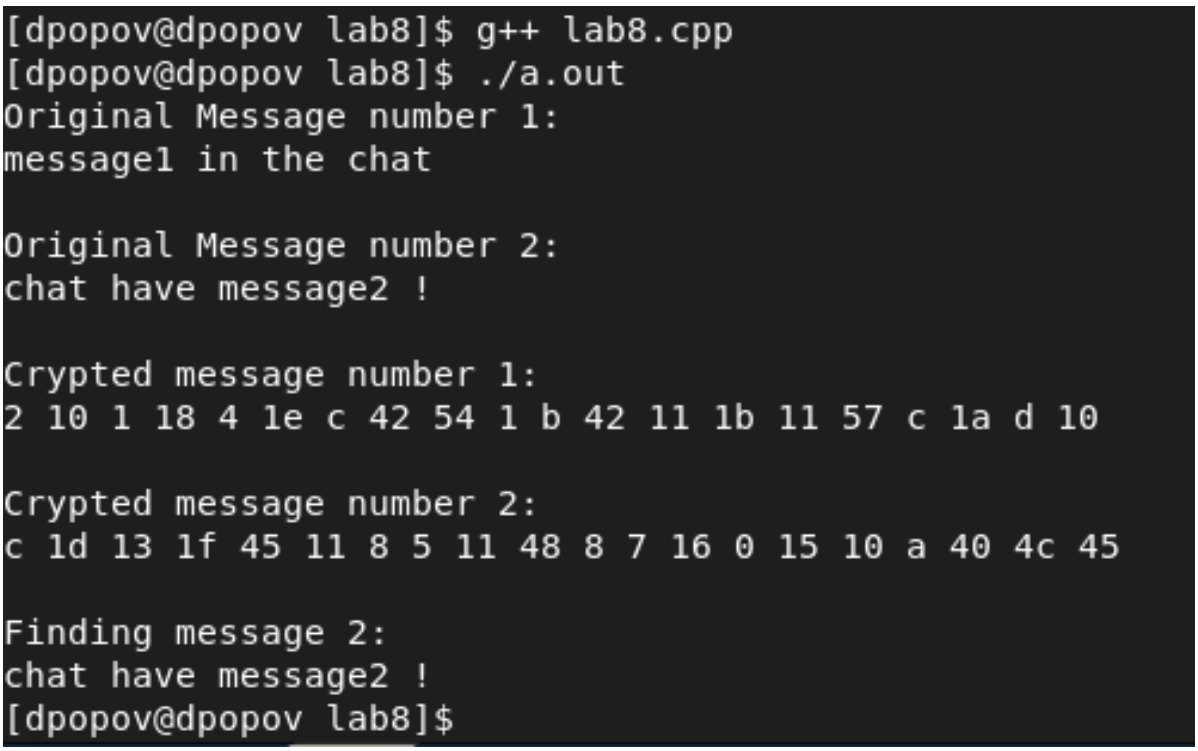


Figure 2.5: console_output

Способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить: злоумышленник может получить два зашифрованных текста, например, во время передачи информации через сеть. Также если он сможет получить часть оригинального сообщения одного из двух зашифрованных текстов, он сможет прочитать оба текста и без ключа.

# 3 Выводы

В результате выполнения работы я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

# 4 Список литературы

1. Методические материалы курса