Отчёт по лабораторной работе №8

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

Жижченко Валерия Викторовна

Содержание

List of Figures

1 Цель работы

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

2 Выполнение работы

Разработали приложение, позволяющее шифровать и дешифровать тексты P_1 и P_2 в режиме однократного гаммирования. Приложение определяет вид шифротекстов C_1 и C_2 обоих текстов P_1 и P_2 при известном ключе:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <map>

using namespace std;

using binType = vector<unsigned char>;

binType generateKey(size_t len);
pair<binType, binType> encrypt(binType p1, binType p2, binType k);
binType decrypt(binType c1, binType c2, binType p1);

int main() {
  char str1[] = {"Как ваши дела?"};
  char str2[] = {"Как вас зовут?"};

  binType p1(str1, str1 + sizeof(str1));
```

```
binType p2(str2, str2 + sizeof(str2));

auto key = generateKey(p1.size());
auto encr = encrypt(p1, p2, key);
auto decrP2 = decrypt(encr.first, encr.second, p1);

cout << "P1: ";

for (auto i: p1) {
  cout << i;
}

cout << endl << "P2: ";

for (auto i: p2) {
  cout << i;
}

cout << endl << "C1: ";

for (auto i: encr.first) {
  cout << i;
}

cout << endl << "C2: ";

for (auto i: encr.second) {
  cout << i;
}
```

```
int count = 0;

cout << endl << endl << "C1 hex:" << endl;

for (auto i: encr.first) {
  printf("%#x\t", i);

  if (count++ >= 4) {
    count = 0;
    cout << endl;
  }
}

count = 0;

cout << endl << "C2 hex:" << endl;

for (auto i: encr.second) {
  printf("%#x\t", i);

  if (count++ >= 4) {
    count = 0;
    cout << endl;
  }
}

count = 0;
```

```
  cout << endl << endl << "Key: " << endl;

  for (auto i: key) {
    printf("%#x\t", i);

    if (count++ >= 4) {
      count = 0;
      cout << endl;
    }
  }

  cout << endl << endl << "Decrypted P2: ";

  for (auto i: decrP2) {
    cout << i;
  }

  cout << endl;

  return 0;
}

binType generateKey(size_t len) {
  binType out;

  srand(time(nullptr));

  for (int i = 0; i < len; i++) {
    out.push_back(rand() % (1 << 8 * sizeof(unsigned char)));
```

```
  }

  return out;
}

pair<binType, binType> encrypt(binType p1, binType p2, binType k) {
  binType c1;
  binType c2;

  for (int i = 0; i < k.size(); i++) {
    c1.push_back(p1[i] ^ k[i]);
    c2.push_back(p2[i] ^ k[i]);
  }

  return make_pair(c1, c2);
}

binType decrypt(binType c1, binType c2, binType p1) {
  binType out;

  for (int i = 0; i < p1.size(); i++) {
    out.push_back(c1[i] ^ c2[i] ^ p1[i]);
  }

  return out;
}
```

1. Вывод работы программы:

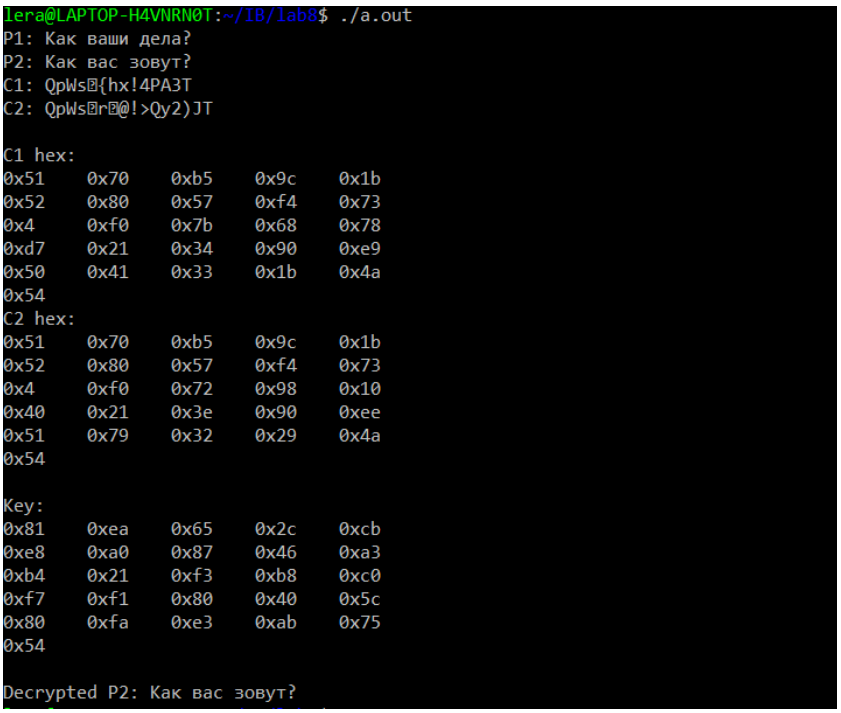

Figure 2.1: Вывод программы

3 Ответы на контрольные вопросы

1. Как, зная один из текстов (P_1 или P_2), определить другой, не зная при этом ключа?

Необходимо воспользоваться формулой:
$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2$,
где C_1 и C_2 – шифротексты.

2. Что будет при повторном использовании ключа при шифровании текста?

Тогда мы получим исходное сообщение.

3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?

Режим шифрования однократного гаммирования одним ключом двух открытых текстов реализуется по следующей формуле:
$C_1 = P_1 \oplus K$
$C_2 = P_2 \oplus K$,
где C_i – шифротексты, P_i – открытые тексты, K – ключ шифрования.

4. Перечислите недостатки шифрования одним ключом двух открытых текстов.

- Во-первых, имея на руках одно из сообщений в открытом виде и оба шифротекста, злоумышленник способен расшифровать каждое сообщение, не зная ключа.

- Во-вторых, зная шаблон сообщений, злоумышленник получает возможность определить те символы сообщения P_2, которые находятся на позициях известного шаблона сообщения P_1.

- В соответствии с логикой сообщения P_2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P_2. Таким образом, применяя формулу из п. 1, с подстановкой вместо P_1 полученных на предыдущем шаге новых символов сообщения P_2 злоумышленник если не прочитает оба сообщения, то значительно уменьшит пространство их поиска. Наконец, зная ключ, злоумышленник смоет расшифровать все сообщения, которые были закодированы при его помощи.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Такой подход помогает упростить процесс шифрования и дешифровки. Также, при отправке сообщений между 2-я компьютерами, удобнее пользоваться одним общим ключом для передаваемых данных.

4 Вывод

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