Информационная безопасность. Отчет по лабораторной работе № 8

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

Горбунова Ярослава Михайловна

Содержание

List of Figures

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 OB B2 70 54

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

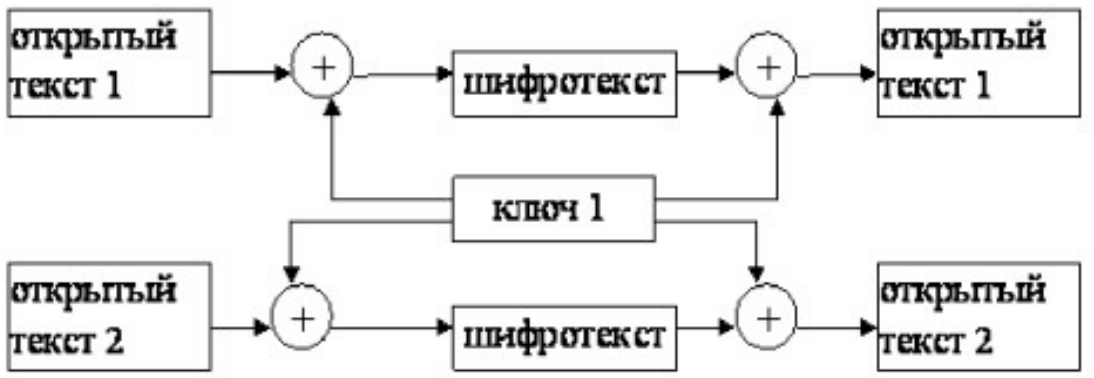


Figure 2.1: Общая схема шифрования двух различных текстов одним ключом

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

C1 = P1 ⊠ K, C2 = P2 ⊠ 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⊠1 = 0, 1⊠0 = 1 (8.2)

получаем:

C1 ⊠C2 = P1 ⊠K ⊠P2 ⊠K = P1 ⊠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 ⊠C2 (известен вид обеих шифровок). Тогда зная P1 и учитывая (8.2), имеем:

C1 ⊠C2 ⊠P1 = P1 ⊠P2 ⊠P1 = P2. (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, 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 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.

```
/*
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)

ciphertext_1 = text_1 ^ gamma
ciphertext_2 = text_2 ^ gamma

4) ciphertext_1 ^ ciphertext_2 ^ text_1 = text_2
   ciphertext_1 ^ ciphertext_2 ^ text_2 = text_1
*/


#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]) << " ";
}

```

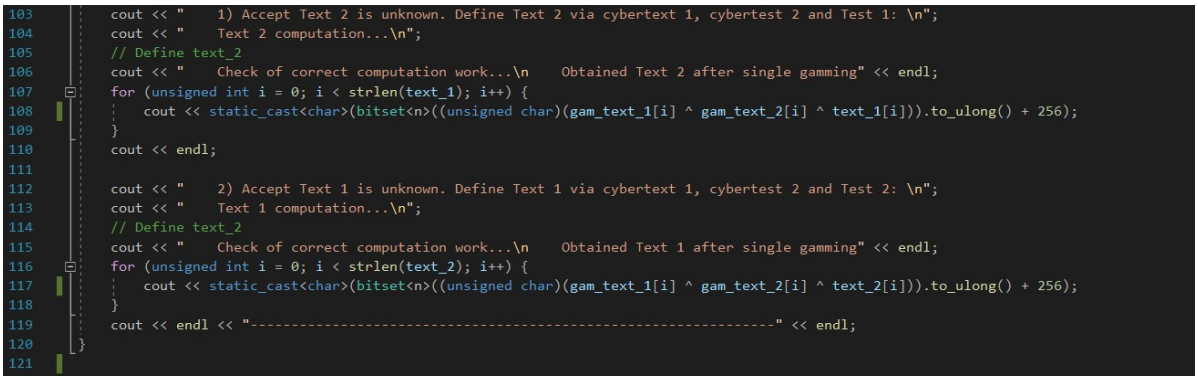
Figure 3.1: Программа (1)

```
// 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]) << " ";
}

int main()
{
    // Introduce variables
    //char text_1[m] = "НаВашисходящийот1204";
    char text_1[m] = "NaVasishodysiyot1204";
    //char text_2[m] = "ВСеверныйфилиалБанка";
    char text_2[m] = "VSevernyifilialBanka";
    int gam[m] = {
        0x05, 0x0C, 0x17, 0x7F, 0x0E, 0x4E, 0x37, 0xD2, 0x94, 0x10,
        0x09, 0x2E, 0x22, 0x57, 0xFF, 0xC8, 0x0B, 0xB2, 0x70, 0x54
    };  // given key (int)
    char gamma[m];  // key (char)
    char gam_text_1[1024], gam_text_2[1024];  // ciphertexts

    // Convert from integer to char
    for (unsigned int i = 0; i < size(gam); i++) {
        gamma[i] = (char)gam[i];
    }

    cout << "------------------------------------------------------------------" << endl << "Part 1" << endl;

    cout << "    Bitset of the key (gamma) :\n";
    print_bitset_text(gamma);
    cout << endl;

    // Text 1
    cout << "    Text 1 for single gamming: \n" << text_1 << endl;

```

Figure 3.2: Программа (2)

```
    cout << "    Bitset of the Text 1 :\n";
    print_bitset_text(text_1);
    cout << endl;

    cout << "    Ciphertext computation...\n";

    // Convert text to ciphertext
    for (unsigned int i = 0; i < strlen(text_1); i++) {
        gam_text_1[i] = text_1[i] ^ gamma[i];
    }
    cout << "    Bitset of the Ciphertext (gam_text_1) :\n";
    print_bitset_gamma(gam_text_1, text_1);
    cout << endl;

    // Text 2
    cout << "    Text 2 for single gamming: \n" << text_2 << endl;

    cout << "    Bitset of the Text 2 :\n";
    print_bitset_text(text_2);
    cout << endl;

    cout << "    Ciphertext computation...\n";

    // Convert text to ciphertext
    for (unsigned int i = 0; i < strlen(text_2); i++) {
        gam_text_2[i] = text_2[i] ^ gamma[i];
    }
    cout << "    Bitset of the Ciphertext (gam_text_2) :\n";
    print_bitset_gamma(gam_text_2, text_2);

    cout << endl;
    cout << "-----------------------------------------------------------------" << endl;
    cout << endl << "-----------------------------------------------------------------" << endl << "Part 2" << endl;

```

Figure 3.3: Программа (3)



```
    cout << "    1) Accept Text 2 is unknown. Define Text 2 via cybertext 1, cybertest 2 and Test 1: \n";
    cout << "    Text 2 computation...\n";
    // Define text_2
    cout << "    Check of correct computation work...\n    Obtained Text 2 after single gamming" << endl;
    for (unsigned int i = 0; i < strlen(text_1); i++) {
        cout << static_cast<char>(bitset<n>((unsigned char)(gam_text_1[i] ^ gam_text_2[i] ^ text_1[i])).to_ulong() + 256);
    }
    cout << endl;

    cout << "    2) Accept Text 1 is unknown. Define Text 1 via cybertext 1, cybertest 2 and Test 2: \n";
    cout << "    Text 1 computation...\n";
    // Define text_2
    cout << "    Check of correct computation work...\n    Obtained Text 1 after single gamming" << endl;
    for (unsigned int i = 0; i < strlen(text_2); i++) {
        cout << static_cast<char>(bitset<n>((unsigned char)(gam_text_1[i] ^ gam_text_2[i] ^ text_2[i])).to_ulong() + 256);
    }
    cout << endl << "-----------------------------------------------------------------" << endl;
}

```

Figure 3.4: Программа (4)

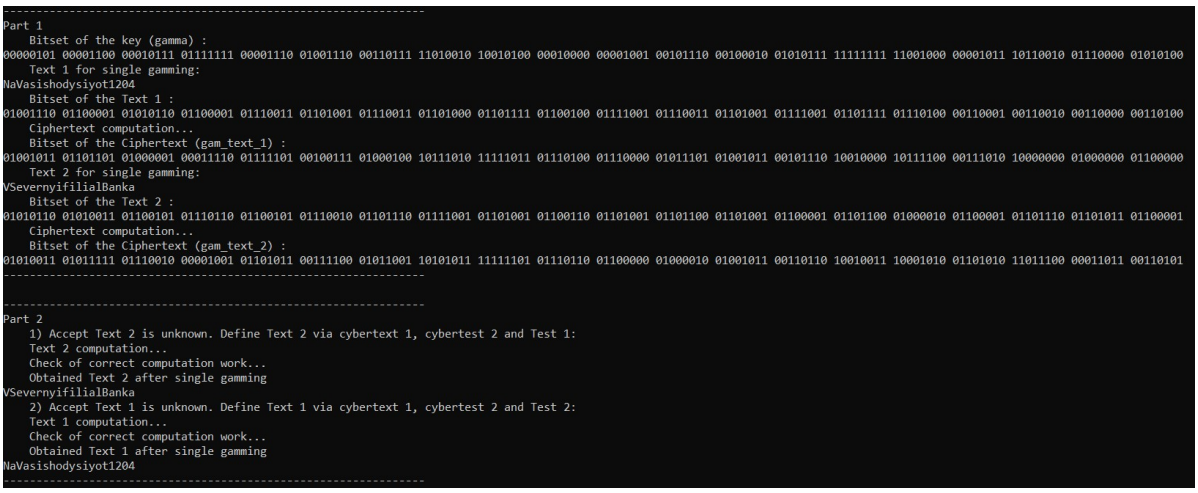


Figure 3.5: Вывод работы программы

3.1 Контрольные вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа? – По формулам: C1 ⊠ C2 ⊠ P1 = P2, C1 ⊠C2 ⊠ 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