

Отчёт по лабораторной работе №8

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

Нгуен Дык Ань

Содержание

I.Цель работы	3
II. Задание	4
III. Выполнение задания	5
Код приложения	5
Анализ кода	6
Результат программы	8
IV. Вывод	9

I.Цель работы

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

II. Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочесть 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 прочесть оба текста, не зная ключа и не стремясь его определить.

III. Выполнение задания

Код приложения

```
#include <iostream>
#include <string>
using namespace std;

string xorOperator(const string &input, const string &key) {
    string output = input;
    for (size_t i = 0; i < input.size(); ++i) {
        output[i] = input[i] ^ key[i % key.size()];
    }
    return output;
}

int main() {
    string P1 = "ThisIsSecret";
    string P2 = "DontTellThat";
    string key = "123456789123";

    string ciphertext1 = xorOperator(P1, key);
    string ciphertext2 = xorOperator(P2, key);
```

```

cout << "Ciphertext 1: " << ciphertext1 << endl;
cout << "Ciphertext 2: " << ciphertext2 << endl;

string Text1 = xorOperator(xorOperator(ciphertext1,ciphertext2),P1);
string Text2 = xorOperator(xorOperator(ciphertext1,ciphertext2),P2);

cout << "Text 1: " << Text1 << endl;
cout << "Text 2: " << Text2 << endl;

return 0;
}

```

Анализ кода

- Мы используем метод шифрования: Выполнение операции сложения по модулю 2 (XOR).

```

string xorOperator(const string &input, const string &key) {
    string output = input;
    for (size_t i = 0; i < input.size(); ++i) {
        output[i] = input[i] ^ key[i % key.size()];
    }
    return output;
}

```

- Функция преобразует каждый элемент введенного текста в новый элемент, зашифрованный на основе ключа, с помощью операцией сложения по модулю 2 (XOR): $C_i = P_i + K_i \pmod{2}$.
- Где C_i — i -й символ получившегося зашифрованного послания, P_i — i -й символ открытого текста, K_i — i -й символ ключа, $i = 1, \dots, m$.

```

int main() {
    string P1 = "ThisIsSecret";
    string P2 = "DontTellThat";
    string key = "123456789123";

    string ciphertext1 = xorOperator(P1, key);
    string ciphertext2 = xorOperator(P2, key);

    cout << "Ciphertext 1: " << ciphertext1 << endl;
    cout << "Ciphertext 2: " << ciphertext2 << endl;

    string Text1 = xorOperator(xorOperator(ciphertext1,ciphertext2),P1);
    string Text2 = xorOperator(xorOperator(ciphertext1,ciphertext2),P2);

    cout << "Text 1: " << Text1 << endl;
    cout << "Text 2: " << Text2 << endl;

    return 0;
}

```

- В main мы определим 2 исходного текста с названиями P1 и P2 и ключ key.
- Использовать функцию “xorOperator” для генерации зашифрованного текста и вывода зашифрованного текста на экран.
- В ситуации, когда злоумышленник знал один из двух текста, он может прочитать остальные, не зная ключа и не стремясь его определить, на основе свойства операции XOR: $1 + 1 = 0$, $1 + 0 = 1$.
- Получаем $C1 + C2 = P1 + K + P2 + K = P1 + P2$, следует $C1 + C2 + P1 = P1 + P2 + P1 = P2$.

Результат программы

```
Ciphertext 1: eZZG|Ed]ZCWG  
Ciphertext 2: u]]@aS[TmYSG  
Text 1: DontTellThat  
Text 2: ThisIsSecret
```


IV. Вывод

После лабораторной работы я получил практические навыки по применению режима одноключевого кодирования на примере кодирования различных исходных текстов одним ключом.