Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Вариант №1

по дисциплине

Средства и методы защиты информации в интеллектуальных системах

Выполнил: А. С. Терлеев

Студент группы

121703

Проверил: В. В. Захаров

Минск 2023

Тема: Простейшие криптографические преобразования

Задача: Разработать программу на языке C++, реализующую следующие функции:

- реализовать в виде программы шифр (зашифрование и расшифрование) Цезаря;
- реализовать атаку полным перебором ключа, используя для оценки правильности выбора ключа визуальный метод или исходный текст лоя автоматического сравнения результата дешифрования;
- оценить криптографическую стойкость реализованного ключа

Листинг программы:

```cpp
#include "string"

class Caesar{
private:
    std::string alphabet;
    int key;

    Caesar(int key);
    Caesar(int key, std::string alphabet);

    std::string encode(std::string& password);
    std::string decode(std::string& encodedPassword);

    int getIndexOfAlphabet(char ch);

public:
    static std::string encode(int key, std::string password);
    static std::string decode(int key, std::string encodedPassword);
};
```

```cpp
#include "Caesar.h"
#include "iostream"


std::string Caesar::encode(int key, std::string password) {
    return Caesar(key).encode(password);
}

std::string Caesar::decode(int key, std::string encodedPassword) {
    return Caesar(key).decode(encodedPassword);
}

Caesar::Caesar(int key): alphabet("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_"), key(key) { };

Caesar::Caesar(int key, std::string alphabet) : key(key), alphabet(alphabet) { }


std::string Caesar::encode(std::string& password) {
    std::string encodedPassword = "";
    for (char ch : password) {
        int index = getIndexOfAlphabet(ch);
        encodedPassword += alphabet.at((index + key) % alphabet.size());
    }
    return encodedPassword;
}

std::string Caesar::decode(std::string& encodedPassword) {
    std::string password = "";
    for (char ch : encodedPassword) {
        int index = getIndexOfAlphabet(ch);
        password += alphabet.at((index - (key % alphabet.size()) + alphabet.size()) % alphabet.size());
    }
    return password;
}


int Caesar::getIndexOfAlphabet(char ch) {
    for (int i = 0; i < alphabet.size(); i++) {
        if (alphabet.at(i) == ch)
            return i;
    }
    throw std::invalid_argument("Element not found");
};
```

```cpp
#include "iostream"
#include "src/Caesar.h"

int findKey(std::string encodedPassword, std::string password) {
    std::string attempPassword;
    for (int i = 1; encodedPassword != attempPassword; i++){
        attempPassword = Caesar::decode(i, encodedPassword);
        if (password == attempPassword)
            return i;
    }
    throw std::invalid_argument("Key can't be found");
}


int main(){
    std::string password, encodedPassword;
    int key;

    std::cout << "Enter password: "; std::cin >> password;
    std::cout << "Enter key: "; std::cin >> key;

    encodedPassword = Caesar::encode(key, password);

    std::cout << "Encoded password: " << encodedPassword << std::endl;
    std::cout << "Key: " << findKey(encodedPassword, password) << std::endl;
}
```

Выполнение реализованной программы:

```
alexey_t@VivoBook:~/git/labs5sem/SiMZIIS/lab2$ ./lab2
Enter password: test_password
Enter key: 12
Encoded password: 5q45L1m44803p
Key: 12
```

Программа выполняется моментально

Оценка криптографической стойкости шифра Цезаря:

Шифр Цезаря не обладает высокой криптографической стойкостью и считается очень простым шифром. Это связано с тем, что шифр имеет ограниченное количество ключей (количество ключей равно количеству символов алфавита). Это означает, что потенциальный злоумышленник может легко перебрать все возможные ключи и расшифровать сообщение.

Варианты усложнения шифра Цезаря:

- Множественный шаг: Вместо одного фиксированного сдвига можно использовать несколько шагов сдвига для каждого символа. Например, первая буква сдвигается на 3 позиции, вторая на 5 позиций, третья на 2 позиции и так далее. Это усложнит обратное расшифрование без знания правила сдвига для каждого символа.

- Ключевое слово: Добавьте ключевое слово перед шифруемым сообщением. Каждая буква ключевого слова определяет сдвиг для соответствующей буквы в сообщении. Например, если ключевое слово "SECRET"и сообщение "HELLO то первая буква "H"будет сдвигаться на 18 позиций (S), вторая буква "E на 4 позиции (E), и так далее. Это усложнит расшифровку без знания ключа.

Вывод:

В данной лабораторной работе были выполнены следующие задачи:

- Была реализована программа, осуществляющая шифрование и расшифрование текста с использованием шифра Цезаря. Шифр Цезаря представляет собой простой метод шифрования, основанный на сдвиге символов в алфавите на фиксированное количество позиций.

- Для оценки правильности выбора ключа была реализована программа, осуществляющая атаку полным перебором ключа. Атака состояла в переборе всех возможных ключей и сравнении расшифрованного текста с оригинальным текстом. Визуальный метод или автоматическое сравнение результатов дешифрования могли быть использованы для оценки правильности выбора ключа.

- Была проведена оценка криптографической стойкости реализованного шифра Цезаря. Шифр Цезаря не обладает высокой стойкостью, так как имеет ограниченное количество ключей, не меняет распределение частотности букв и не предоставляет дополнительных

сложностей для расшифровки. Это делает его уязвимым для различных атак, включая атаку полным перебором ключа и частотный анализ.

- Были предложены варианты усложнения шифра Цезаря. Некоторые из вариантов включают использование множественного шага сдвига для каждого символа, введение ключевого слова для определения сдвига, использование нескольких алфавитов или комбинирование шифра Цезаря с другими методами шифрования. Варианты усложнения могут повысить уровень безопасности шифра Цезаря.