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 №1

по курсу “Средства и методы защиты информации в интеллектуальных системах”

|  |  |
| --- | --- |
| Выполнил:  студент гр. 221702 | Юргилевич Е. В. |
| Проверил: | Крищенович В.А. |

Минск 2024

Лабораторная работа №1

Генерация паролей

**Задание:**

1. Разработать программу на C++, реализующую следующие функции:
2. Генерация строки с заданной пользователем длиной, состоящей из символов алфавита в соответствии с вариантом задания(использовать функции rand(), srand() и инициализацию от таймера);
3. Проверка равномерности распределения символов путём визуализации частотного распределения;
4. Вычисление среднего времени подбора пароля, выбираемого из сгенерированной строки.
5. Построить график зависимости среднего времени подбора пароля от его длины.
6. Дать практические рекомендации по выбору пароля исходя из предположений об алфавите пароля; производительности вычислительного средства атакующего и времени атаки

**Вариант 5**

При генерации пароля использовать арабские цифры.

**Код программы**#include <iostream>

#include <ctime>

#include <map>

#include <cmath>

using namespace std;

string generateRandomString(int length, string alphabet);

void createHistogram(string str);

void calculateAverageTime(int str\_len, int literal\_size);

void main() {

setlocale(LC\_ALL, "ru");

string alphabet = "0123456789";

int length;

cout << "Ввебите длину пароля (100макс.): ";

cin >> length;

cout << endl;

if (length > 100) {

cout << "неверный ввод";

return;

}

string random\_str = generateRandomString(length, alphabet);

cout << random\_str;

createHistogram(random\_str);

calculateAverageTime(length, alphabet.size());

return;

}

string generateRandomString(int length, string alphabet) {

string result;

srand(time(0));

for (int i = 0; i < length; i++) {

result += alphabet[rand() % alphabet.length()];

}

return result;

}

void createHistogram(string str) {

map<char, int> histogram;

for (char c: str)

{

histogram[c]++;

}

cout << "\nЧастота вхождения символов в строку:\n";

for (auto h: histogram)

{

cout << h.first << ": ";

for (int i = 0; i < h.second; i++)

{

cout << "[]";

}

cout << " (" << h.second << ")" << endl;

}

}

void calculateAverageTime(int str\_len, int alphabet\_size) {

for (int i = 1; i <= str\_len; i++)

{

long double combinationCounter = 0;

for (int j = 0; j <= i; j++) {

combinationCounter += pow(alphabet\_size, j);

}

long double averageTime = combinationCounter / (2 \* 1000000);

cout << "Среднее время подбора пароля методом перебора длины " << i << " составило: " << averageTime << 'с' << std::endl;

}

}

**Описание основных функций программы  
1)** **string generateRandomString(int length, string alphabet)**

Генерирует случайную строку длиной length. Символы выбираются из диапазона 0 до 9, а затем возвращает сгенерированную строку.

**2) void createHistogram(string str)**

Вычисляет и выводит частотное распределение символов в переданной строке str. Для каждого символа, встречающегося в строке, строится графическое представление частоты в виде строки символов [].

**3)** **void calculateAverageTime(int str\_len, int alphabet\_size)**

Оценивает время, необходимое для brute-force атаки на строку длиной str\_len с алфавитом размера literal\_size.

Подсчитывает общее количество возможных комбинаций символов с учётом длины строки и размера алфавита. Возвращает оценочное время, исходя из скорости 1 миллиона попыток в секунду.

**График зависимости среднего времени подбора пароля от его длины:**



**Вывод:** рекомендации по выбору пароля зависят от различных факторов:  
**Состав пароля**: Рекомендуется включать в пароль цифры, заглавные и строчные буквы, а также специальные символы. Чем разнообразнее используемый набор символов, тем сложнее подобрать пароль методом перебора.

**Длина пароля**: Чем больше символов содержит пароль, тем труднее его взломать. Для обеспечения надёжной защиты рекомендуется использовать пароли длиной от 12 символов.