

**Федеральное государственное автономное образовательное  
учреждение  
высшего образования  
"Национальный исследовательский университет  
"Высшая школа экономики"**

**Московский институт электроники и математики им.  
А.Н.Тихонова**

Направление подготовки/специальности  
10.05.01 «Компьютерная безопасность»  
Образовательная программа «Компьютерная безопасность»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**  
по дисциплине «Защита программ и данных»

## Описание задания

Реализовать программу на языке C/C++, которая должна включать следующую последовательность действий:

1. Запрос какой-либо информации от пользователя (логин, номер телефона, номер какого-либо документа, ключ любого вида и т.д.)
2. Произведение манипуляций по разработанному алгоритму с введённой пользователем информацией для генерации ответа на ключевой вопрос.
3. Запрос ключа от пользователя и сравнение его со сгенерированным ключом на этапе 2.
4. Выдача результата по принципу: сгенерированный и алгоритмом и введённый пользователем ключи совпадают – TRUE-> Поздравление или доступ к какой-либо информации; Не совпадают – ошибка.

При реализации программы должны быть соблюдены условия:

1. Внутри программы должен быть какой-либо алгоритм преобразования данных для генерации ключевой информации. Например, сложение по модулю кодов букв из таблицы ASCII, Подсчёт количества букв в логине пользователя и умножение его на текущий год, запросы системного времени или к каким-либо заранее определённым файлам.
2. Должны быть реализованы меры защиты от отладки: специальные функции языка, искусственное усложнение кода, директивы препроцессора, условия сборки компилятора, упаковщики и т. д.

Результатом выполнения задания должна быть разработанная программа в виде исходных кодов, make/cmake файла сборки или инструкции по сборке в случае применения дополнительных средств или специальных скриптов. Документация к программе с описанием алгоритма работы, описанием входных и выходных данных и применяемых в программе мер защиты.

## Описание работы программы

Программа запрашивает логин, являющийся любой комбинацией символов без пробелов. После этого производятся манипуляции с введенной информацией с помощью функции `generate_key`. Потом запрашивается пароль и сравнивается с результатом функции `generate_key`. В случае совпадения выводится сообщение с поздравлением, в противном – сообщение об ошибке.

## Описание алгоритма генерации ключа

Введенный логин разделяется на две равные части. Если в логине нечетное количество символов, то в конец добавляется символ «0». Первая половина логина записывается в прямой порядке, вторая – в обратном. Строки превращаются в свое численное представление на основе таблицы ASCII. Потом производится хог  $i$ -ого символа из первой строки и  $i$ -ого символа второй строки (во второй строке – вторая часть логина в обратном порядке), и к полученному результату код  $i$ -ого символа из первой строки. И берется  $\text{mod } 128$ . Если результат  $< 32$ , то к результату прибавляется 32. Таким образом, получаем первые  $(\text{длина логина})/2$  символов пароля.

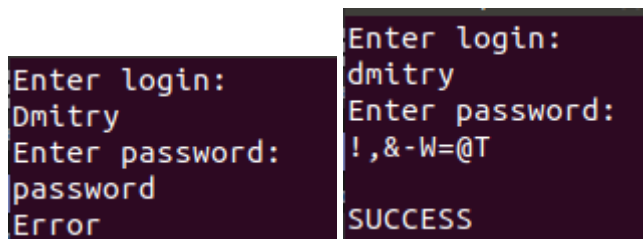
Потом вычисляется хэш с помощью простой хэш функции.

```
for (int i = 0; i < len_of_login; ++i){  
    hash = hash*seed + j;  
}
```

Где `seed` – сумма кодов символов логина, `j` – длина логина.

Полученный хэш делится на группы по 3 символа (числа), если длина хэша не делится на 3, то хэш дополняется символами «0». От каждого трехзначного числа берется  $\text{mod } 128$ . Если результат  $< 32$ , то к результату прибавляется 32. Число переводится в символ.

## Пример работы программы



```
Enter login:
Dmitry
Enter password:
password
Error
```

```
Enter login:
dmitry
Enter password:
!,&-W=@T
SUCCESS
```

## Примененные меры защиты

Большая часть текста вывода на экран в самом коде приобрела нечитаемый вид.

Были созданы некоторые макросы C++ для констант. Были добавлены неиспользуемые константы.

Был добавлен неиспользуемый буфер. Часть переменных были названы неосмысленными именами.

Были добавлены goto, чтоб снизить читаемость порядка выполнения алгоритма.