

Национальный исследовательский университет «МЭИ»

Институт информационных и вычислительных технологий

**Кафедра вычислительной техники**

**Лабораторная работа №2**

**по курсу** «**Защита информации**»

**Тема: «Изучение алгоритмов симметричного и асимметричного шифрования с использованием библиотек с использованием криптобиблиотек»**

|  |  |
| --- | --- |
|  | **Выполнил** |
| **Студент** | Швец Григорий Владиславович |
| **Группа** | А-06-19 |
| **Дата** | 09.04.2023 |

|  |  |
| --- | --- |
|  | **Принял** |
| **Преподаватель** | Андреева Ирина Николаевна |
| **Оценка** |  |
| **Дата** |  |
| **Подпись** |  |

Москва 2023

Содержание

[**Цель работы** 3](#_Toc131928803)

[**Задание** 3](#_Toc131928804)

[**Выполнение работы** 3](#_Toc131928805)

[**Исходный код программ** 4](#_Toc131928806)

# **Цель работы**

Целью данной лабораторной работы является изучение алгоритмов симметричного и асимметричного шифрования с использованием криптобиблиотек.

# **Задание**

1. Подготовить текстовый файл, содержимое которого будет шифроваться и расшифровываться в соответствии с пунктами задания.
2. Разработанное приложение должно:
   1. Зашифровывать содержимое файла, **имя которого вводится с клавиатуры,** на основе **ДВУХ** крипто алгоритмов: симметричного и асимметричного.
   2. Расшифровывать данные файла, **имя которого вводится с клавиатуры,** на основе **ДВУХ** крипто алгоритмов: симметричного и асимметричного.
3. Расшифрованное содержимое файла **должно** выводиться на экран.

|  |  |  |
| --- | --- | --- |
| **№** | **Библиотека, алгоритм симметричного шифрования** | **Библиотека, алгоритм асимметричного шифрования** |
| 17 | OpenSSL, BlowFish | wolfCrypt, RSA |

# **Выполнение работы**

Для реализации поставленной задачи было принято решение использовать 2 объектно-ориентированных языка программирования, а именно: для библиотеки OpenSSL и асимметричного алгоритма шифрования BlowFish – C++, а для библиотеки wolfCrypt и асимметричного алгоритма шифрования RSA – Python. Так же стоит сказать, что оба разработанных приложения были разработаны под операционную систему Linux на дистрибутиве Ubuntu, поскольку импорт данных библиотек на ОС Windows и их установка может вызвать некоторые трудности, т.к. возникают ошибки при сборке библиотек на данной ОС. Именно исходя из этих соображений выбор пал именно на Linux, потому что работа на данной ОС с указанными библиотеками не составляет подобных проблем.

# **Исходный код программ**

1. **OpenSSL, BlowFish**

|  |
| --- |
| #include <openssl/blowfish.h>  #include <cstdio>  #include <iostream>  #include <string>  #include <string.h>  #define BUFSIZE 1024  using namespace std;  /\* Функция криптопреобразования информации. Параметры функции: дескрипторы входного и выходного  файлов и режим работы - BF\_ENCRYPT для шифрования и BF\_DECRYPT для дешифрования \*/  int do\_crypt(FILE \*in, FILE \*out, int mode)  {  int num = 0;  unsigned char inbuf[BUFSIZE], outbuf[BUFSIZE];  BF\_KEY bfkey;  const unsigned char key[16] = "111111111111111";  unsigned char iv[8] = "1111111";  /\* Помещаем ключ в структуру bfkey \*/  BF\_set\_key(&bfkey, sizeof(key), key);  /\* Шифруем блоки входного файла \*/  cout << "Результат выполнения: ";  for(;;) {  int inlen = fread(inbuf, 1, BUFSIZE, in);  if(inlen <= 0) break;  BF\_cfb64\_encrypt(inbuf, outbuf, (long)inlen, &bfkey, iv, &num, mode);  fwrite(outbuf, 1, inlen, out);  cout << outbuf;  }  cout << endl;  return 1;  }  int main()  {  string infile, ofile;  char\* cinfile, \*cofile;  string menu;  int mode;  FILE\* in\_ptr, \*out\_ptr;  do {  cout << "1.Зашифровать файл\n2.Расшифровать файл\n3.Выход\nВыберите действие: ";  getline(cin, menu);  if (menu == "1")  {  mode = BF\_ENCRYPT;  cout << "Введите путь к файлу с текстом: ";  getline(cin, infile);  cout << "Введите путь к файлу с шифротекстом: ";  getline(cin, ofile);  }  else if (menu == "2")  {  mode = BF\_DECRYPT;  cout << "Введите путь к файлу с шифротекстом: ";  getline(cin, infile);  cout << "Введите путь к файлу с текстом: ";  getline(cin, ofile);  }  in\_ptr = fopen(infile.c\_str(), "r");  out\_ptr = fopen(ofile.c\_str(), "wb+");  int rev = do\_crypt(in\_ptr, out\_ptr, mode);  fclose(in\_ptr);  fclose(out\_ptr);  } while ((menu == "1") ||(menu == "2"));  } |

1. **wolfCrypt, RSA**

|  |
| --- |
| from wolfcrypt.ciphers import RsaPrivate, RsaPublic  from wolfcrypt.utils import h2b  menu = input("1.Зашифровать файл\n2.Расшифровать файл\n3.Выход\nВыберите действие: ")  while menu in ['1', '2']:  if menu == '1':  pubfile = input("Введите путь до файла с открытым ключом: ")  with open(pubfile) as f:  public = f.read()  textfile = input("Введите путь до файла с текстом: ")  with open(textfile) as f:  text = f.read()  text = text.encode('utf-8')  pub = RsaPublic(h2b(public[:-1]))  ciphertext = pub.encrypt(text)  ciphertextfile = input("Введите путь до файла с шифротекстом: ")  with open(ciphertextfile, 'wb+') as f:  text = f.write(ciphertext)  elif menu == '2':  privfile = input("Введите путь до файла с закрытым ключом: ")  with open(privfile) as f:  private = f.read()  ciphertextfile = input("Введите путь до файла с шифротекстом: ")  with open(ciphertextfile, 'rb') as f:  ciphertext = f.read()  prv = RsaPrivate(h2b(private[:-1]))  text = prv.decrypt(ciphertext)  print(f"Расшифровано содержимое: {text}")  textfile = input("Введите путь до файла с текстом: ")  with open(textfile, 'w+') as f:  text = f.write(text.decode("utf-8"))  menu = input("1.Зашифровать файл\n2.Расшифровать файл\n3.Выход\nВыберите действие: ") |