Министерство образования Республики Беларусь

Учреждение образования “Белорусский государственный университет информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Дисциплина: Средства и методы защиты информации в интеллектуальных системах

Отчёт к лабораторной работе №5

Выполнил: Потоцкий Д.А

Группа: 221702

Проверила: Крищенович В.А.

Минск 2024

Лабораторная работа №5

АСИММЕТРИЧНОЕ ШИФРОВАНИЕ И ЭЛЕКТРОННАЯ

ЦИФРОВАЯ ПОДПИСЬ

**Задание:** разработать программное обеспечение, реализующее функции генерации секретного и открытого ключей, шифрования и цифровой подписи для алгоритма RSA. Обмен входными и выходными данными должен осуществляться через файлы:

– открытого ключа;

– секретного ключа;

– исходного сообщения;

– зашифрованного сообщения.

Для повышения скорости шифрования использовать метод

последовательного возведения в квадрат и умножения.

Выполнить тестирование разработанного программного обеспечения на 10 наборах тестовых данных.

Длина чисел p и q должна быть не менее 1024 бит.

**Описание основных функций программы:   
1. mod\_exp(base, exp, mod)**

Выполняет возведение числа base в степень exp по модулю mod с использованием быстрого алгоритма возведения в степень по модулю. Это ключевая операция для шифрования и дешифрования в RSA.

**2. encrypt(message, key)**

Шифрует строку message с использованием открытого ключа RSA, содержащегося в объекте RSAKey (с ключами e и n).

**3. decrypt(ciphertext, key)**

Дешифрует зашифрованное сообщение (ciphertext) с использованием закрытого ключа RSA (с ключами d и n).

**4. generate\_rsa(bits, randfunc=None, e=65537)**

Генерирует пару ключей RSA с заданной длиной в битах (bits). Используется алгоритм генерации простых чисел для создания двух простых чисел p и q, после чего рассчитываются модули и экспоненты. Параметр e задает публичную экспоненту, а randfunc — функцию для случайных чисел.

**5. save\_keys(rsa\_key, pub\_file, priv\_file)**

Сохраняет открытый и закрытый ключи из объекта RSAKey в файлы. Открытый ключ сохраняется в файл pub\_file, а закрытый — в файл priv\_file.

**6. load\_keys(pub\_file, priv\_file)**

Загружает ключи RSA из файлов pub\_file и priv\_file.

**7. save\_message(message, file\_path)**

Сохраняет строку message в файл по пути file\_path.

**8. load\_message(file\_path)**

Загружает сообщение из файла по пути file\_path.

**9. save\_encrypted\_message(encrypted\_message, file\_path)**

Сохраняет зашифрованное сообщение (число Integer) в файл по пути file\_path.

**10. load\_encrypted\_message(file\_path)**

Загружает зашифрованное сообщение (число) из файла по пути file\_path.