Государственное учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Кафедра: Интеллектуальных информационных технологий

Дисциплина: Средства и методы защиты информации в интеллектуальных системах

**Отчет по лабораторной работе №7**

**“Установка, использование и анализ специализированных средств криптографического пакета OpenSSL”**

Выполнил:

Студент гр. 221701 Телица И. Д.

Проверил: Захаров В. В.

Минск 2024

#### **Цель**:

Ознакомится с криптографической библиотекой OpenSSL.

#### **Задания:**

1. Выполнить тестирование скорости выполнения различных алгоритмов шифрования:

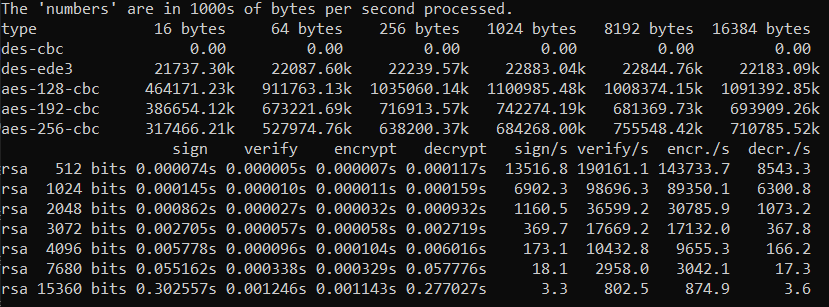


Рис. 1. Скорость выполнения алгоритмов DES, AES, RSA

DES vs AES:

В тестах AES почти всегда показывает более высокую производительность по сравнению с DES, AES был разработан с учетом современных вычислительных архитектур и поддерживает аппаратное ускорение на многих устройствах.

AES vs RSA:

В сравнении с AES, RSA работает значительно медленнее при обработке данных. RSA обычно используется для шифрования небольших объемов данных (например, для обмена ключами), а не для больших файлов. Это объясняется тем, что асимметричное шифрование RSA требует сложных математических вычислений, таких как операции с большими числами и возведение в степень, которые занимают больше времени, чем побитовые операции, выполняемые AES.

1. Создать криптографические ключи. Выбрать несколько произвольных файлов и выполнить:
   1. шифрование (зашифрование и расшифрование) посредством различных симметричных алгоритмов;

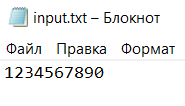


Рис. 2. Исходный файл для симметричного шифрования

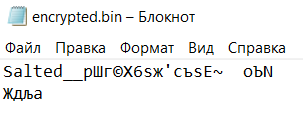


Рис. 3. Зашифрованный файл алгоритмом AES

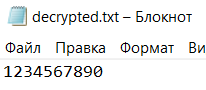


Рис. 4. Зашифрованный файл алгоритмом AES

* 1. шифрование (зашифрование и расшифрование) посредством различных асимметричных алгоритмов;

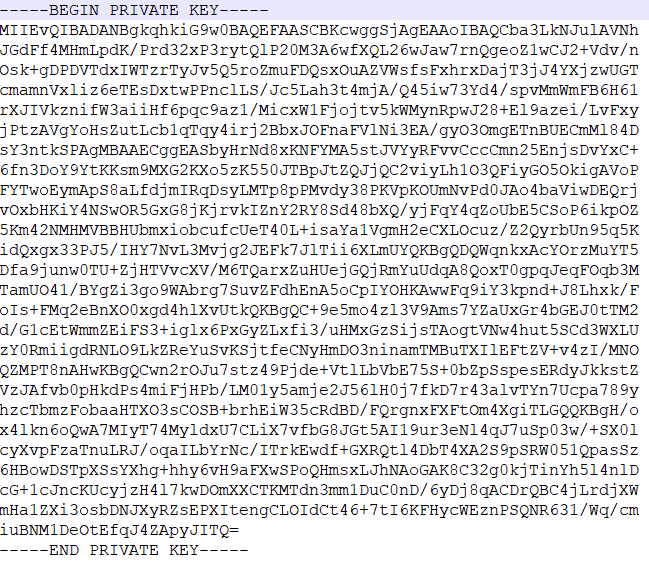


Рис. 5. Приватный ключ алгоритма RSA

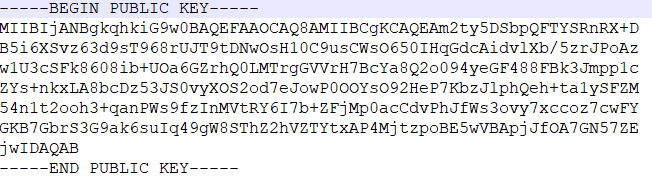


Рис. 6. Публичный ключ алгоритма RSA

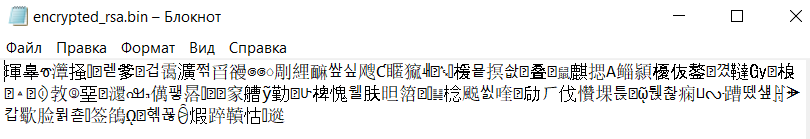


Рис. 7. Зашифрованный файл алгоритмом RSA

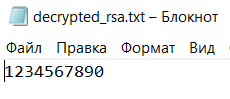


Рис. 8. Расшифрованный файл алгоритмом RSA

* 1. хэширование различных файлов различными алгоритмами (обязательно md5 и sha1).



Рис. 9. Хэш алгоритмом MD5



Рис. 10. Хэш алгоритмом SHA1

1. Создать самоподписанный сертификат X509. Изучить состав сертификата и назначение его компонентов.

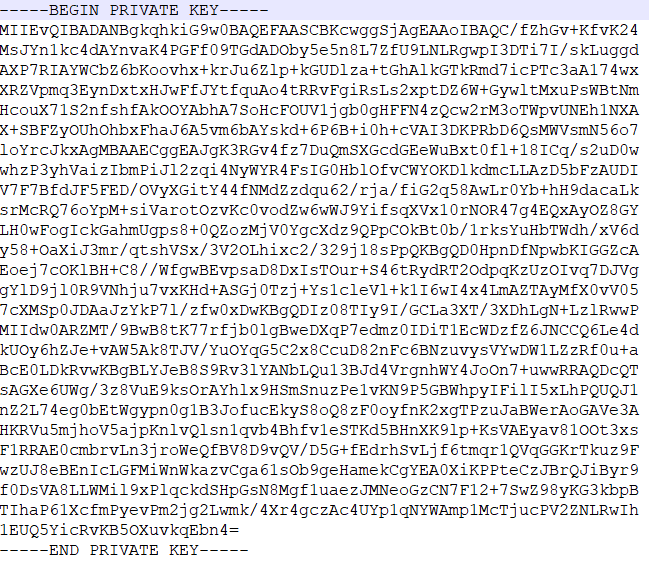
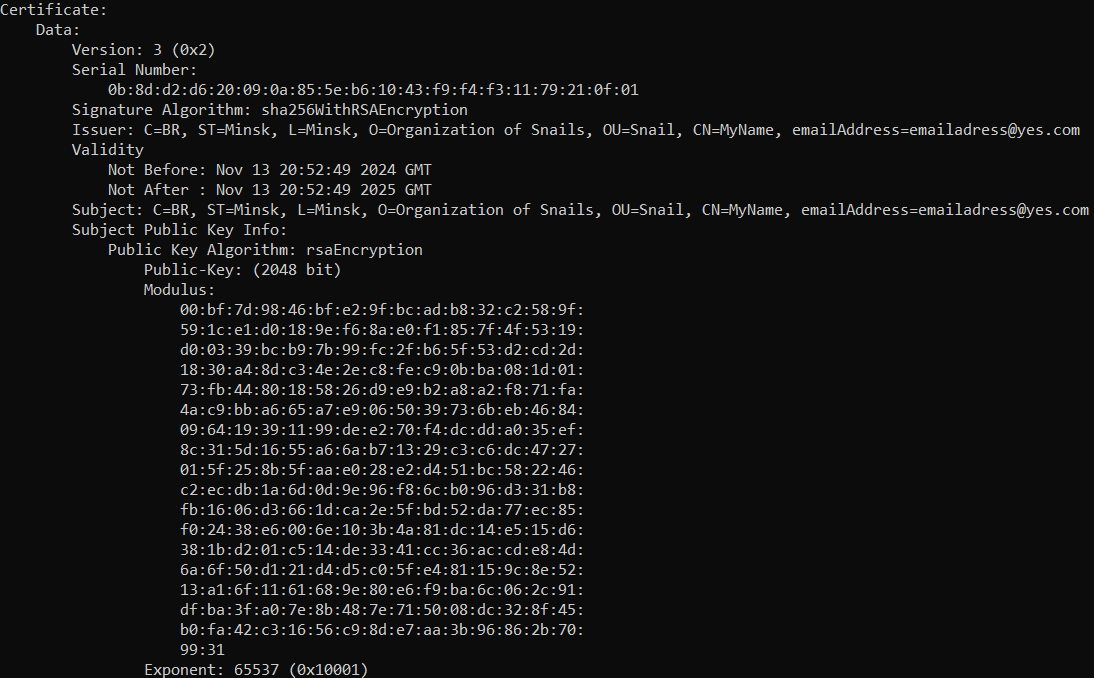


Рис. 11. Приватный ключ алгоритма цифровой подписи



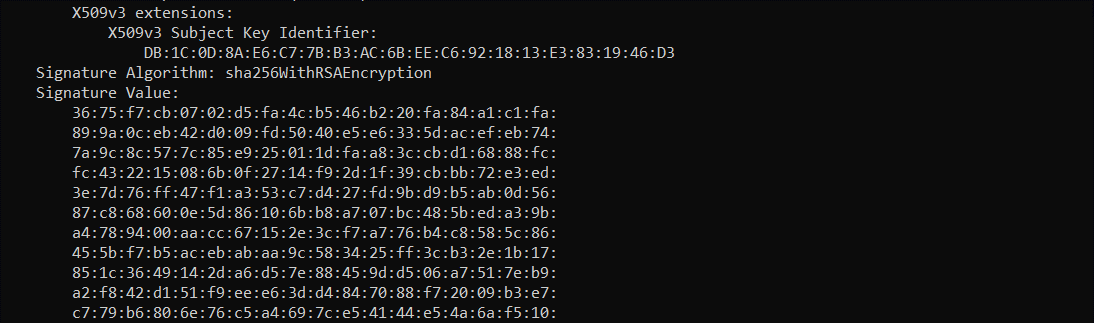


Рис. 12. Содержимое сертификата

Описание компонентов сертификата

* **Субъект (Subject)**: указывает информацию о владельце сертификата (например, имя компании, домен).
* **Издатель (Issuer)**: информация о центре сертификации (для самоподписанного сертификата субъект и издатель совпадают).
* **Публичный ключ**: используется для проверки подлинности владельца сертификата.
* **Срок действия (Validity)**: период, в течение которого сертификат действителен.
* **Алгоритм подписи**: алгоритм, используемый для подписи сертификата.

## **Вывод:**

Познакомился с криптографической библиотекой OpenSSL, ознакомился с её функционалом, таким как:

* Использование различных методов шифрования
* Проверка скорости методов шифрования
* Составление цифровой подписи
* Хэширование данных

На основе полученных данных выяснили, что различия в скоростях между AES и RSA в несколько десятков тысяч раз. Такие большие отличия связаны с тем, что AES был разработан для шифрования больших объёмов данных и не имеет таких сложных вычислений, как в RSA