Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**Национальный исследовательский университет «МЭИ»**

**Отчет**

**к курсовому проекту**

**по дисциплине «Защита данных»**

Тема: Программная реализация вычисления и проверки электронной подписи по алгоритму RSA

|  |  |  |
| --- | --- | --- |
| Студент:  Гайчуков Д.А. | А-13-21 |  |
| *фамилия, и., о.* | *группа* | *подпись* |

Преподаватель: доц., к.т.н.Хорев П.Б.

*должность, звание, фамилия, и., о.* *подпись*

Москва, 2024

# **Описание алгоритма**

Ссылка на описание алгоритма:

https://libeldoc.bsuir.by/bitstream/123456789/44205/1/Gerasimenok\_Kriptograficheskiy.pdf

## **1.1. Описание алгоритма RSA**

**I этап - генерация ключей:**

1. Выбираются два различных простых числа – заданного размера (например, 2048 бит)
2. Вычисляется их произведение , которое называется *модулем*
3. Вычисляется значение функции Эйлера
4. Выбирается целое число, взаимно простое со значением функции Эйлера значение лежит в диапазоне . Число называется *открытой экспонентой*
5. Вычисляется число d, оно, в свою очередь, должно удовлетворять сравнению

Число d называется *закрытой экспонентой*

1. Пара публикуется в качестве открытого ключа RSA

Пара играет роль закрытого ключа

**II этап - работа протокола:**

Выбираем целое число 𝑚 для шифрования, чтобы его зашифровать, воспользуемся открытым ключом:

Полученное 𝑐 и есть зашифрованное число. Для того чтобы из 𝑐 обратно получить исходное число 𝑚, необходимо использовать закрытый ключ:

## **1.2. Описание алгоритма RSA для электронной подписи**

**Алгоритмы цифровой подписи:**

Криптосистема RSA позволяет реализовать аутентификацию, создав цифровую подпись. Данная подпись подтверждает целостность отправляемой информации и может быть использована вместо стандартной подписи от руки.

**Алгоритм с открытым текстом:**

**Первый этап – генерация цифровой подписи:**

1. Возьмем открытый текст 𝑚.

2. Для создания подписи, обозначаемой 𝑠, необходимо воспользоваться сгенерированным ранее закрытым ключом:

3. Передаем пару {𝑠, 𝑚}, состоящую из подписи и открытого текста.

**Второй этап – проверка неизменности сообщения с помощью электронной подписи:**

1.Используя полученную подпись и открытый ключ, сгенерируем прообраз текста 𝑚, обычно обозначаемый 𝑚′:

2. Сравниваем прообраз 𝑚′ и сам текст 𝑚, тем самым выясняя подлинность подписи и целостность текста.

# **Программная реализация**

2.1. Открываем главное окно:

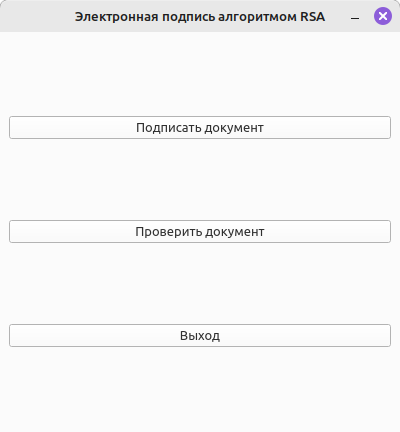


Рисунок 2.1 - Стартовое окно

2.2. При нажатии на кнопку “Подписать документ” открывается окно выбора:

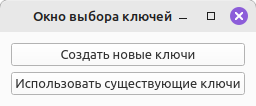


Рисунок 2.2 - Окно выбора ключей

2.3. При нажатии на кнопку “Создать новые ключи” необходимо ввести размер ключа:

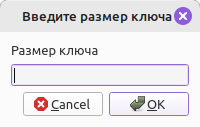


Рисунок 2.3 - Окно выбора размера ключа

2.4. Далее нужно выбрать файл для подписи и указать имя подписывающего:

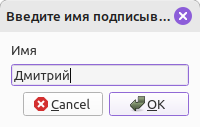


Рисунок 2.4 - Окно ввода имени подписывающего

2.5. Далее необходимо выбрать папку для подписи, ввести имя файла с подписью. Подпись будет храниться в выбранной папке в файле с выбранным именем.

2.6. После нужно выбрать папку хранения ключей. Ключи будут храниться в выбранной папке под именами public\_key и private\_key.

2.7. Далее нужно ввести кодовое слово:

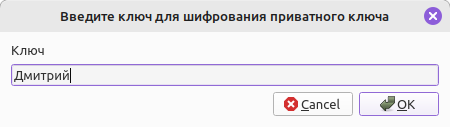


Рисунок 2.5 - Окно ввода кодового слова

2.8. Если выбрать “Использовать существующие ключи” в пункте 2.2., то необходимо выбрать папку с ключами и ввести кодовое слово (программа будет искать файл private\_key в выбранной папке):

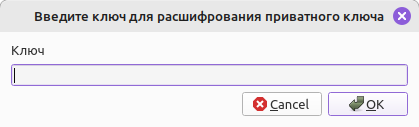


Рисунок 2.6 - Окно ввода кодового слова

2.9. Дальше нужно повторить пункты 2.4. и 2.5.

2.10. При выборе в главном окне кнопки “Проверить документ” (пункт 2.1.) нужно выбрать зашифрованный файл, файл с открытым ключом (public\_key) и выбрать файл с подписью. Далее мы увидим результат.