Разработка приложения для формирования цифровой подписи в электронном документообороте

Научный руководитель: Акуленок И. Н.
Исполнитель: Бочкарев В. Д.

Цели работы

- Проанализировать криптографические свойства алгоритмов;
- Реализовать приложение для работы с электронно-цифровой подписью.

Актуальность данной темы – переход крупных производств от бумажного документооборота к электронному, необходимость защищать подлинность таких документов.

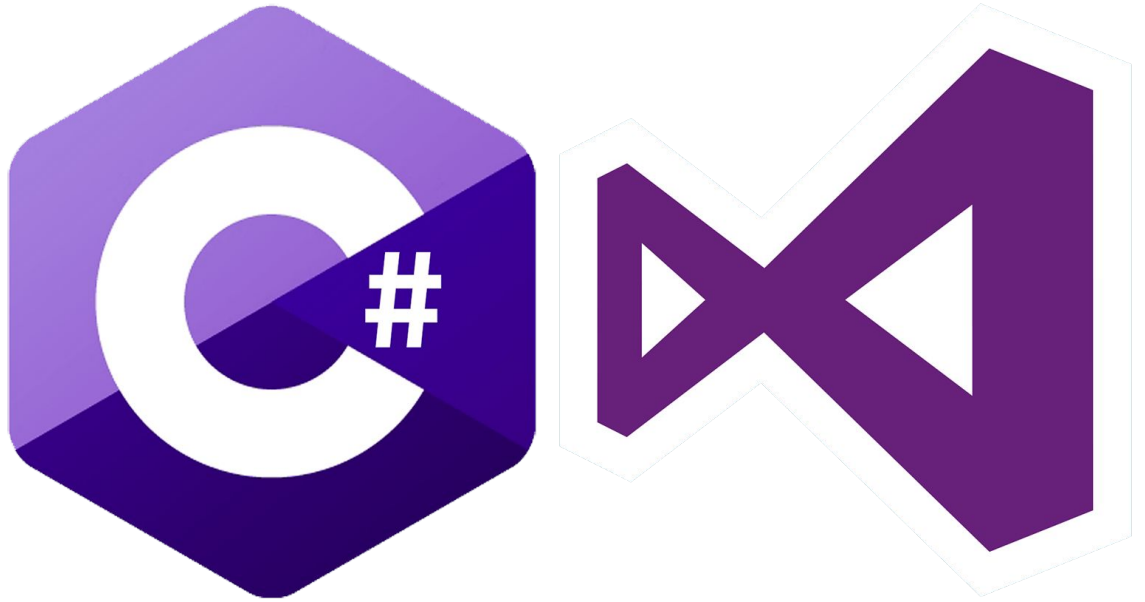
Требования к приложению

Приложение должно выполнять следующие функции:

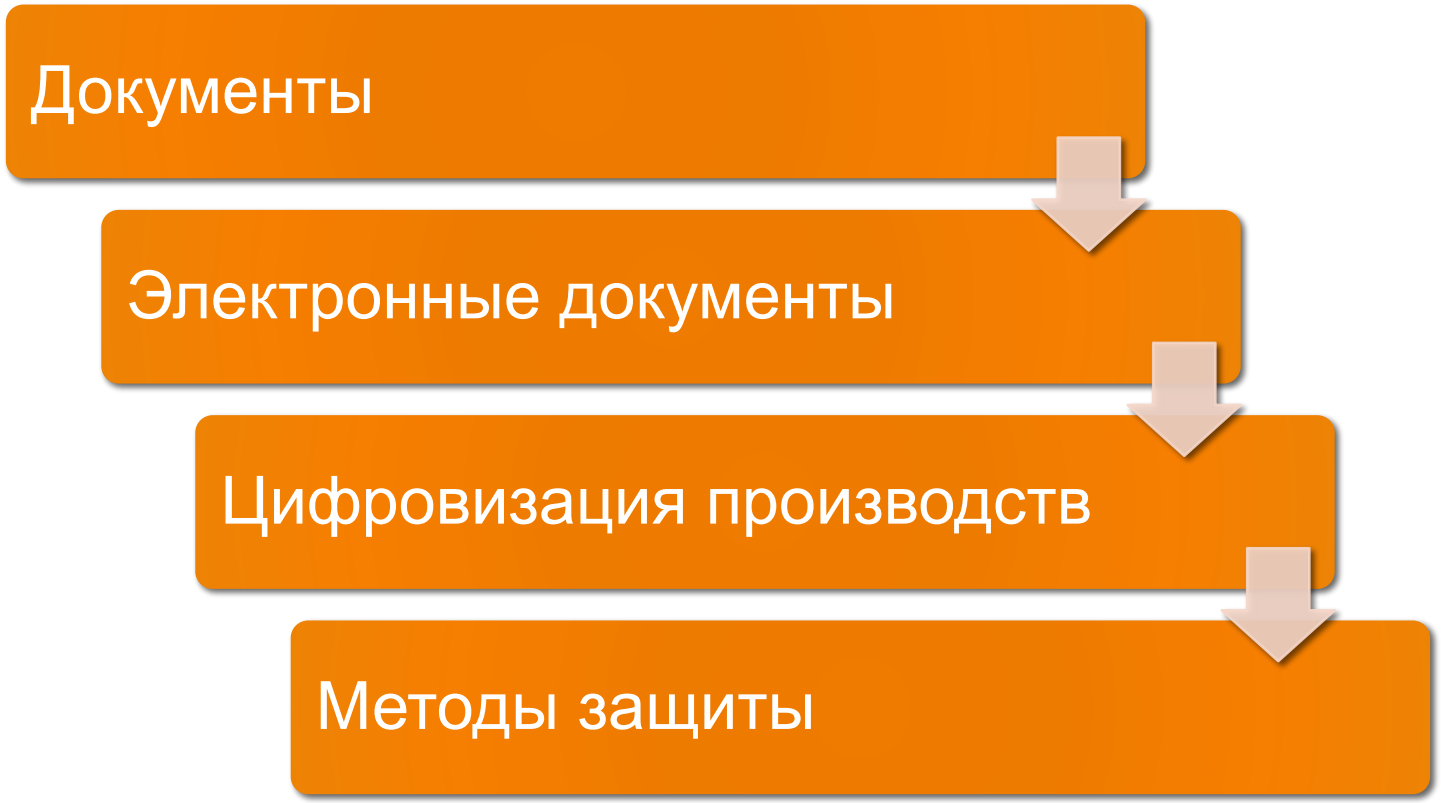
- Генерировать ключи;
- Формировать подпись и сертификат;
- Проверять корректность подписи и подлинность документа.

Выбор программных средств

- Язык программирования C#
- Платформа .NET Framework версии 7.2.0
- IDE Microsoft Visual Studio 2019
- СУБД SQLite



Предметная область



Основные методы защиты:

1. Загрузка в облачные сервисы
2. Шифрование данных документа
3. Электронно-цифровая подпись

Обзор существующих программ

КриптоАРМ

1. Платное ПО
2. Подпись для разного рода файлов
3. Поддержка стандартов Microsoft CryptoAPI

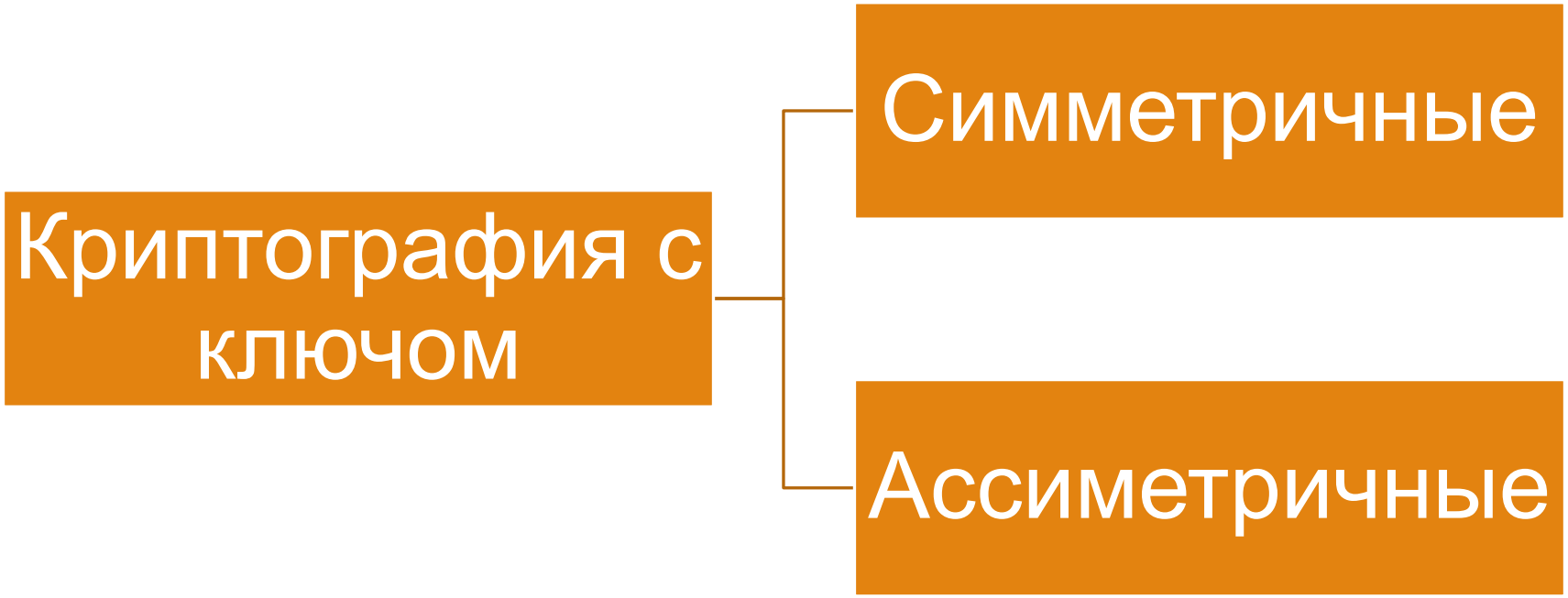
КРИПТО ПРО

1. Комплексный продукт
2. Поддержка стандартов Microsoft CryptoAPI

SignMachine

1. Бесплатное ПО
2. Необходим сертификат

Виды криптоалгоритмов



Электронно-цифровая подпись

ЭЦП – это цифровая информация, размер которой зависит от алгоритма, которая присоединяется к тексту, и передается уже вместе с ним.

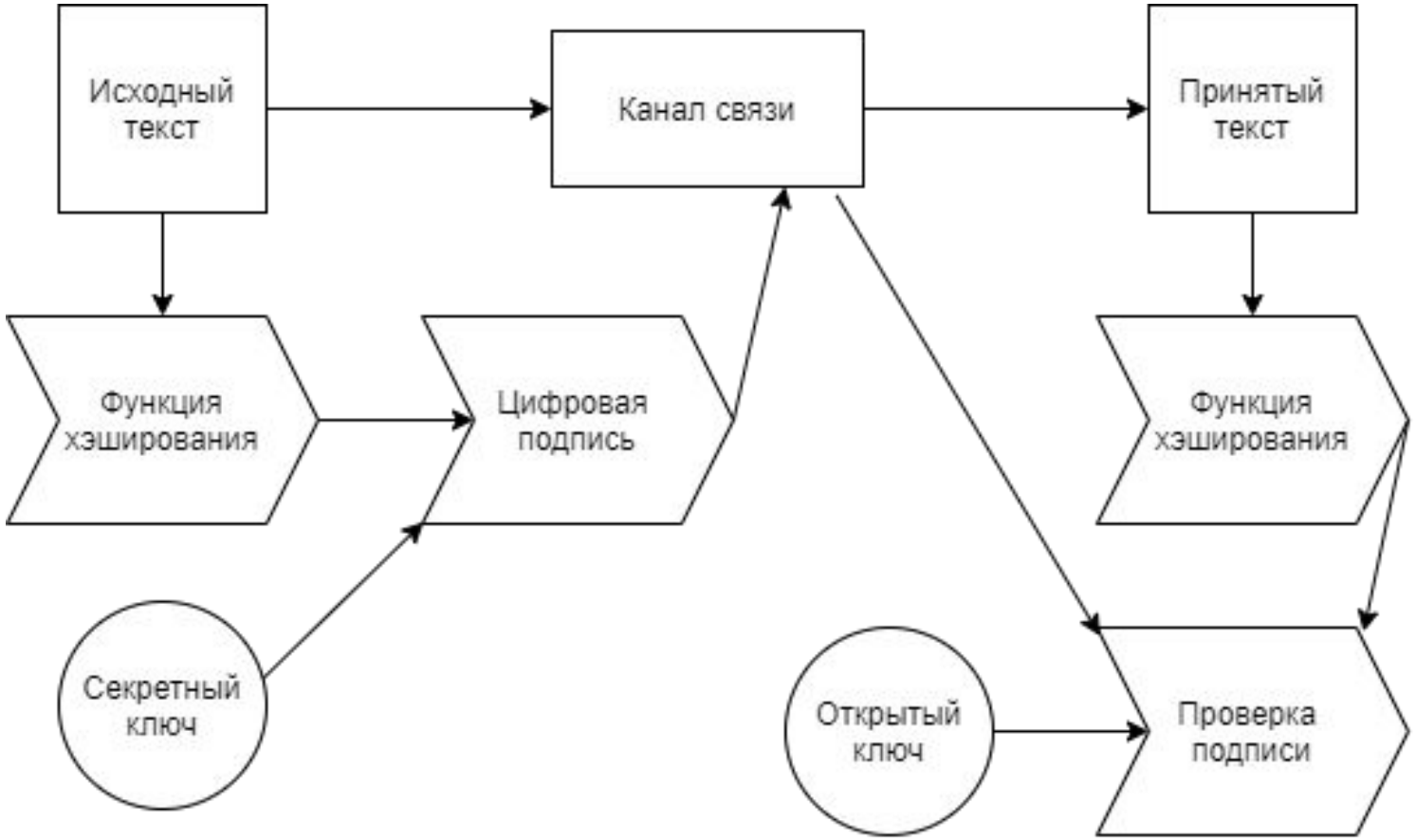
Функции ЭЦП		
Удостоверяет личность подписавшего документ	Гарантирует целостность документа	Обеспечивает неотказуемость от подписанного документа

Обобщенная схема работы ЭЦП

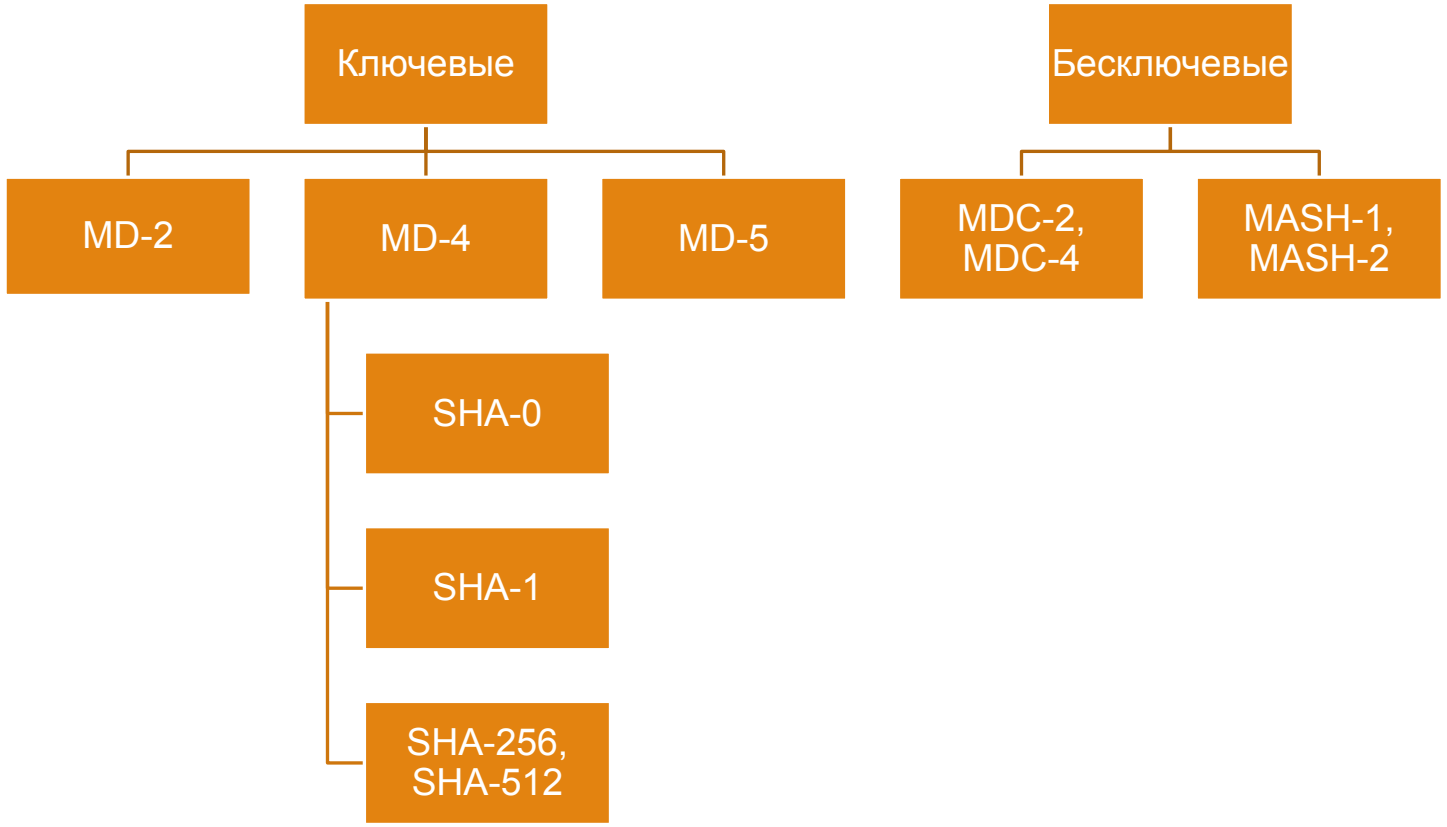
Генерация ключей

Подпись

Проверка



Функции хэширования



Требования к функциям хэширования:

- Стойкость к поиску первого прообраза
- Стойкость к поиску второго прообраза
- Стойкость к коллизиям

Диаграмма классов

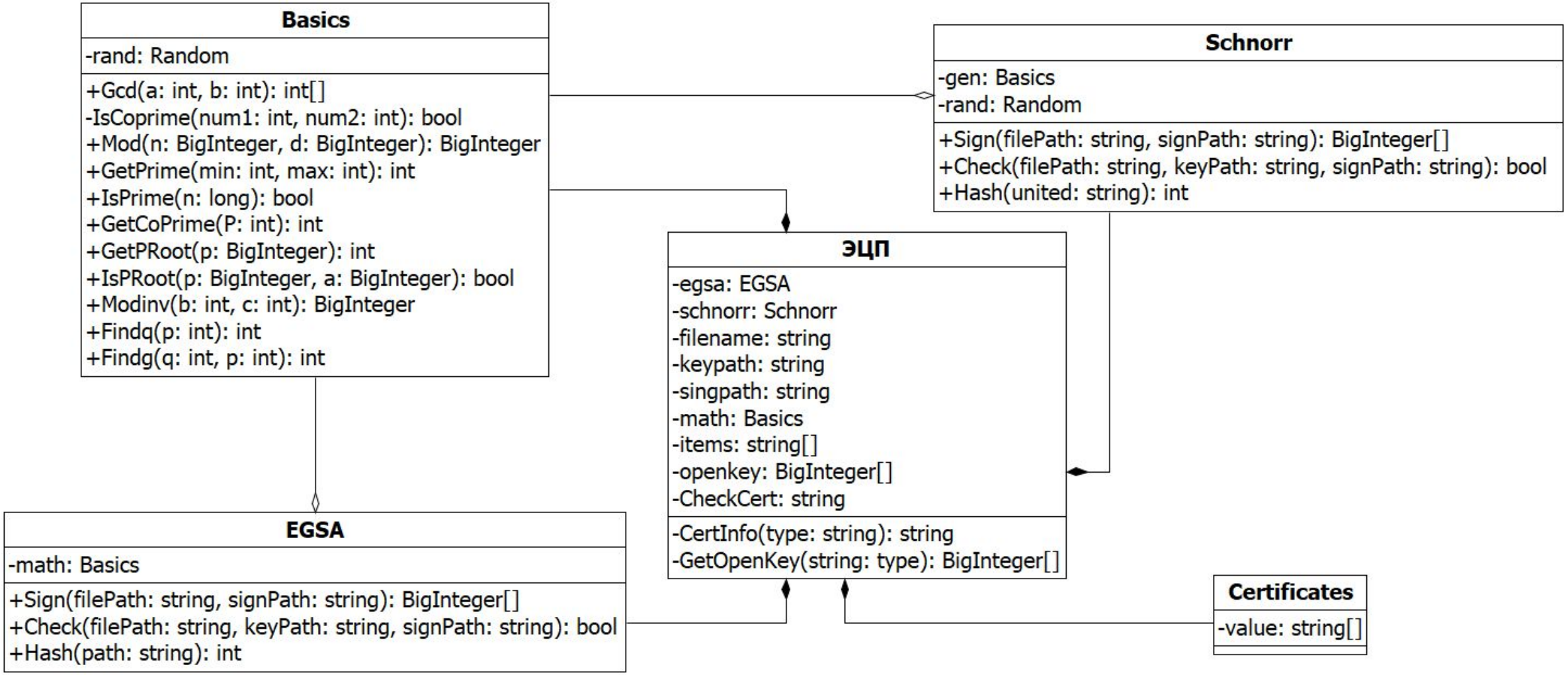
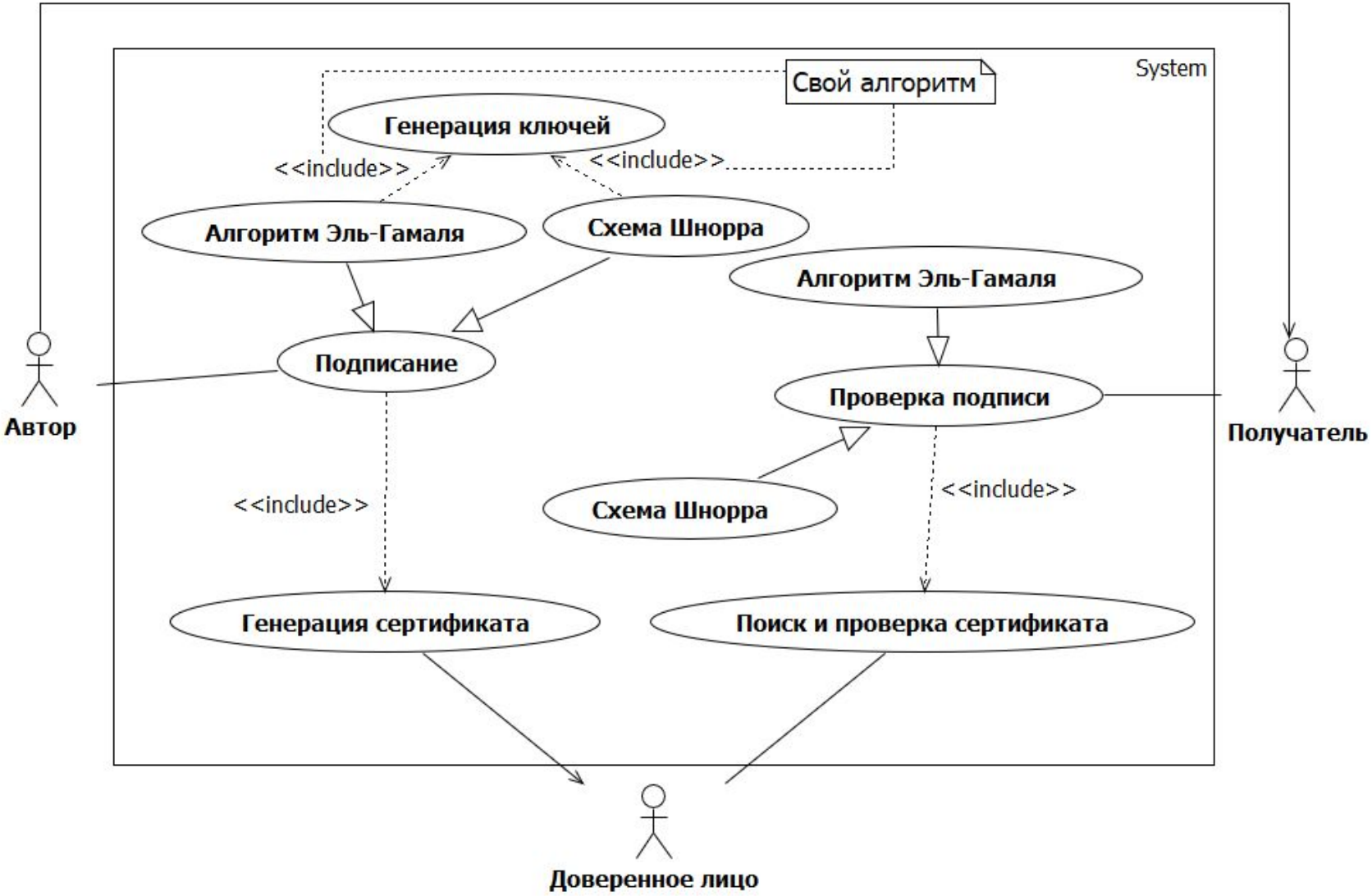


Диаграмма прецедентов



Алгоритм Эль-Гамаля

Основные методы разработанного приложения:

- **Sign** реализует основные алгоритмы Эль-Гамаля: генерации ключей и формирования подписи;
- **Check** реализует основные алгоритмы проверки подписи;
- **Gcd** реализует расширенный алгоритм Евклида ;
- **Mod** реализует корректное вычисление деления по модулю отрицательных чисел;
- **GetCoPrime** реализует проверку чисел на взаимную простоту;
- **Hash** реализует хэш-функцию SHA-256;
- **IsPRoot** проверяет, является ли число первообразным корнем другого числа.

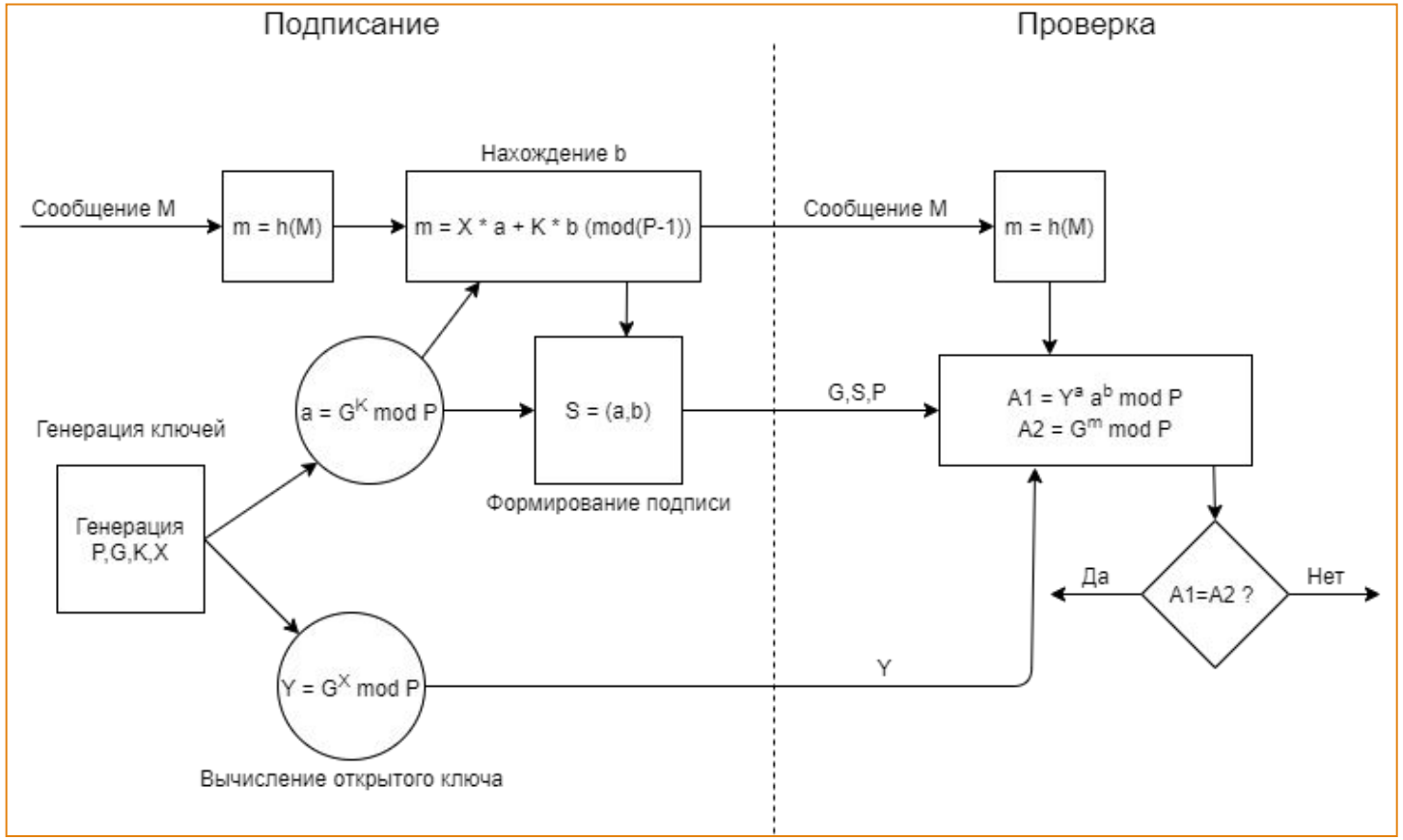
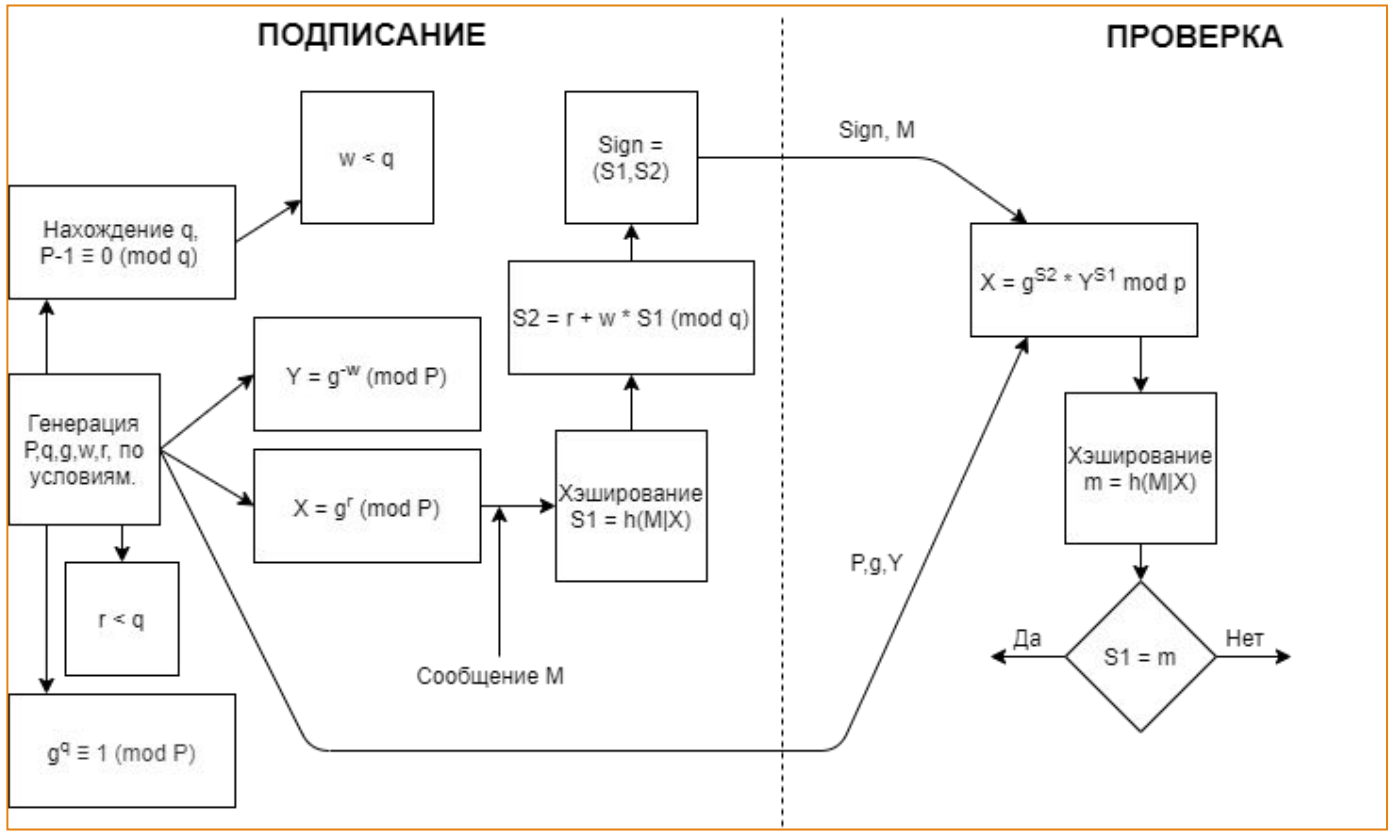


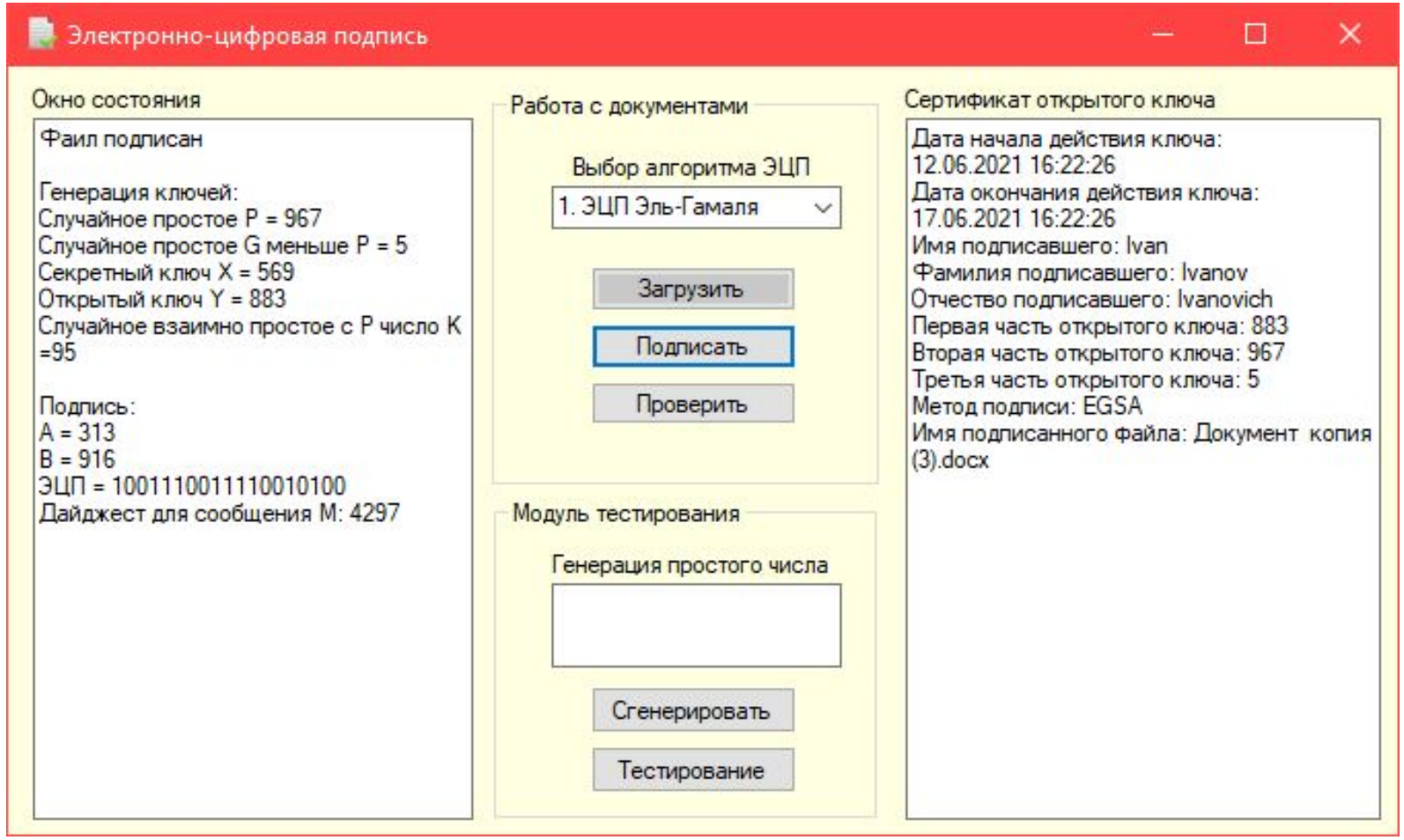
Схема Шнорра

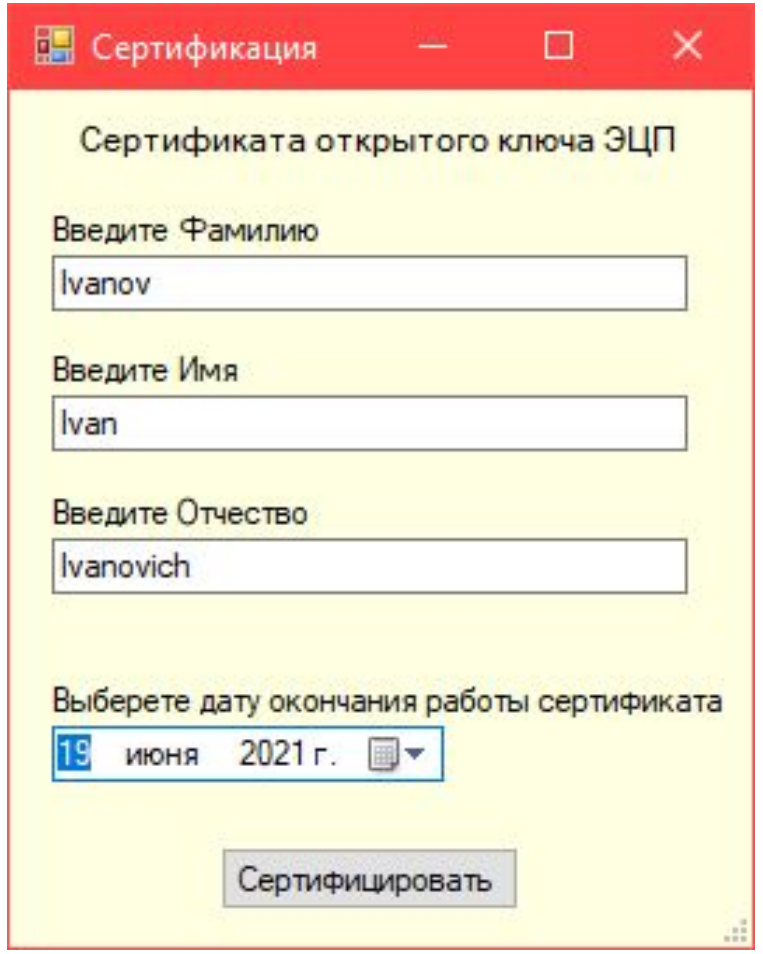


Основные методы разработанного приложения:

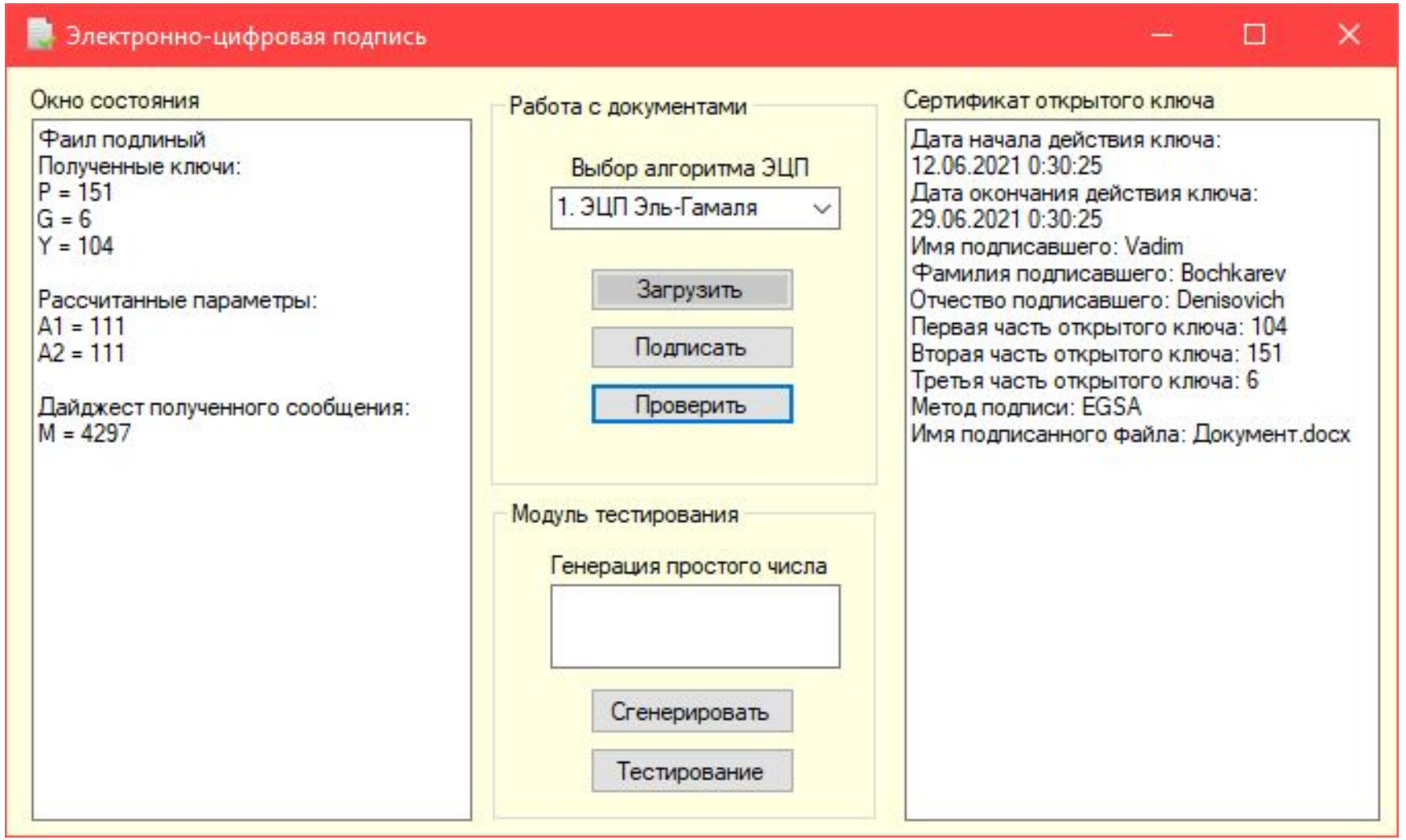
- **Sign** реализует основные алгоритмы Шнорра: генерации ключей и формирования подписи;
- **Check** реализует основные алгоритмы проверки подписи;
- **Hash** реализует хэш-функцию MD5;
- **Modinv** реализует нахождение мультипликативного обратного.

Руководство пользователя - подпись





Руководство пользователя - проверка



Тестирование и отладка

В данной работе проверялись:

- главная форма приложения и элементы управления;
- операция возведения в степень на переполнение (типа данных BigInteger);
- стандартный оператор модульного деления C# на работу с отрицательными числами;
- корректность выполнения алгоритмов формирования ЭЦП;
- корректность проверки сертификата на принадлежность к документу.

Заключение

- Проанализированы два алгоритма формирования ЭЦП.
- Реализовано приложение для работы с электронно-цифровой подписью.
- Реализованное приложение успешно протестировано.

Спасибо за внимание!