Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчёт**

**по лабораторной работе №12**

**Дисциплина: КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ**

**Тема: «ЦИФРОВАЯ ПОДПИСЬ ДОКУМЕНТОВ. АЛГОРИТМЫ СОСТАВЛЕНИЕ И ПРОВЕРКИ ЦИФРОВОЙ ПОДПИСИ»**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. А. Корнилов

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение

компьютерных технологий

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А. А. Крамаренко

**Тема:** Цифровая подпись документов. Алгоритмы составления и проверки цифровой подписи.

**Цель:** Изучение алгоритмов составление и проверки цифровой подписи документов.

**Ход работы:**

1. Провести подпись и проверку подписи для алгоритма ГОСТ34.10-2018 или реализовать данный алгоритм.

Для реализации поставленной задачи был написан класс GOST, который реализует алгоритм цифровой подписи на основе эллиптических кривых, соответствующий российскому стандарту **ГОСТ Р 34.10-2012**. Алгоритм работает с эллиптическими кривыми и использует хеш-функцию **Streebog** (ГОСТ Р 34.11-2012) для хеширования сообщений.

Внутри класса в качестве параметров задаются характеристики эллиптической кривой: простое число p, коэффициенты a и b, порядок группы точек m и порядок подгруппы q​. На основании них в классе используется объект EllipseCurve, который предоставляет методы для выполнения операций с точками на кривой, таких как сложение точек и умножение точки на скаляр.

При инициализации экземпляра класса проверяется корректность заданных параметров.

Для генерации открытого ключа используется метод generate\_decryption\_key. Этот метод принимает базовую точку P на эллиптической кривой и закрытый ключ d. Открытый ключ Qвычисляется как Q=d⋅P с использованием метода prod\_point, который выполняет умножение точки на скаляр.

Для создания цифровой подписи используется метод subscribe. Этот метод принимает сообщение, базовую точку P и закрытый ключ d. Сначала сообщение хешируется с использованием алгоритма **Streebog**, и результат хеширования преобразуется в целое число z. Затем вычисляется значение e=zmod  q. Если e=0, оно заменяется на 1. Далее генерируется случайное число kиз интервала [1,q−1], и вычисляется точка C=k⋅P. Затем значение r определяется как r=xCmod  q, где xC​ — координата x точки C. Если r=0, процесс генерации k повторяется. Затем вычисляется значение s=(r⋅d+k⋅e)mod  q. Если s=0, процесс также повторяется. В результате подпись формируется как объединение значений r и s в виде бинарной строки.

Для проверки подписи используется метод check\_correct. Этот метод принимает подпись, сообщение, открытый ключ Q и базовую точку P. Сначала подпись разделяется на значения r и s, которые проверяются на корректность. Затем сообщение хешируется с использованием алгоритма **Streebog**, и результат хеширования преобразуется в целое число z. Вычисляется значение e=zmod  q, и если e=0, оно заменяется на 1. Далее вычисляется значение v=e^−1mod  q, которое используется для вычисления z1=(s⋅v) mod  qи z2=(−r⋅v) mod  q. Затем вычисляется точка C=z1⋅P+z2⋅Q, и проверяется, что r=xCmod  q, где xC​ — координата x точки C. Если условие выполняется, подпись считается верной.

В классе также реализованы вспомогательные методы, такие как euler\_totient, который вычисляет значение функции Эйлера для заданного числа, и методы для работы с точками на эллиптической кривой.

1. Реализовать программный продукт, позволяющий подписывать и проверять подпись вводимого сообщения согласно схеме RSA. Возможно пользоваться встроенными библиотеками языков для хэш функции.

Для реализации поставленной задачи был написан класс RSA, который реализует алгоритм шифрования и цифровой подписи на основе асимметричного криптографического алгоритма RSA. Алгоритм RSA использует пару ключей: открытый ключ для шифрования и проверки подписи, и закрытый ключ для расшифрования и создания подписи. Для хеширования сообщений используется алгоритм SHA-256, который обеспечивает целостность данных и используется при создании и проверке цифровой подписи.

Внутри класса реализованы методы для генерации ключей, создания и проверки цифровой подписи, а также для шифрования и расшифрования данных. Основные математические операции, такие как расширенный алгоритм Евклида и модульная арифметика, используются для вычисления ключей и выполнения криптографических операций.

При инициализации экземпляра класса ключи не создаются автоматически, а генерируются с помощью метода create\_keys. Этот метод принимает размеры простых чисел p и q в битах и возвращает пару ключей: открытый и закрытый. Для генерации простых чисел используется функция randprime из библиотеки sympy, которая гарантирует, что числа p и q являются простыми и имеют заданный размер.

Для создания цифровой подписи используется метод subcribe. Этот метод принимает сообщение и закрытый ключ. Сначала сообщение хешируется с использованием алгоритма SHA-256, и результат хеширования преобразуется в целое число. Затем подпись вычисляется как результат возведения хеша в степень закрытого ключа по модулю n. Подпись возвращается вместе с исходным сообщением.

Для проверки подписи используется метод check\_sub. Этот метод принимает сообщение, подпись и открытый ключ. Сообщение хешируется, и подпись расшифровывается с использованием открытого ключа. Если расшифрованное значение совпадает с хешем сообщения, подпись считается верной.

Для шифрования данных используется метод encrypt. Этот метод принимает сообщение, подпись и открытый ключ. Сообщение и подпись объединяются в одну строку, которая затем преобразуется в целое число и шифруется с использованием открытого ключа. Результатом является зашифрованное число.

Для расшифрования данных используется метод decrypt. Этот метод принимает зашифрованные данные и закрытый ключ. Данные расшифровываются с использованием закрытого ключа, после чего преобразуются обратно в строку. Строка разделяется на исходное сообщение и подпись, которые возвращаются как результат.

В классе также реализованы вспомогательные методы, такие как \_\_extended\_gcd, который используется для вычисления секретной экспоненты d в процессе генерации ключей. Этот метод реализует расширенный алгоритм Евклида, который находит НОД двух чисел и коэффициенты для решения линейного уравнения.

**Вывод:** Были изучены алгоритмы для генерации и проверки цифровой подписи документов.