**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Петербургский государственный университет путей сообщения**

**Императора Александра I»**

**(ФГБОУ ВО ПГУПС)**

Факультет «Автоматизация и интеллектуальные технологии»

Кафедра «Информатика и информационная безопасность»

Лабораторная работа № 4

по дисциплине

«Защита электронного технологического документооборота»

«Реализация электронной подписи на основе открытого коллективного ключа»

|  |  |  |
| --- | --- | --- |
| **Выполнил обучающийся**  Курс 5  Группа КИБ-012 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Я. И. Геласимов |
|  |  |  |
| **Проверил** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | М.Ф. Соломатова |

Санкт-Петербург

2024

**Цель выполнения лабораторной работы:**

Изучение возможности применения механизмов электронной подписи для технологического электронного документооборота.

**Порядок выполнения лабораторной работы:**

Изучить стандартные алгоритмы создания электронной подписи на основе открытого коллективного ключа (ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012, RSA, схема Шнорра).

Разработать алгоритм формирования и проверки электронной подписи на основе открытого коллективного ключа (ЭП ОКК), который в общем виде состоит из трёх этапов: формирования ЭП ОКК на основе генерации индивидуальных параметров подписи и применения функции, отображающих их в коллективный параметр подписи; собственно, создание ЭП ОКК и отправка её вместе с электронным документом соответствующим пользователям АСУ предприятия; проверка ЭП ОКК с использованием открытого коллективного ключа.

На основе одного из указанных стандартов разработать схему формирования и проверки ЭП ОКК, соответствующую обобщенному методу ЭП ОКК.

Реализовать разработанную схему на языке программирования С++ (С#) и протестировать полученный код.

**Выполнение лабораторной работы:**

В качестве криптографического протокола был выбран RSA

Ассиметричный алгоритм криптографии RSA, датой возникновения концепции которого считается 1976 год сейчас очень активно используется для обмена данными, верификацией источника программного обеспечения и в других сферах, где необходимо обмениваться данными или верифицировать отправителя. Кроме того, он является базовой частью HTTPS протокола, использование которого в России достигло 98% по данным Яндекс.Радара.

RSA это алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших полупростых чисел.

RSA-ключи генерируются следующим образом:

1) выбираются два различных случайных простых числа p и q заданного размера (например, 1024 бита каждое);

2) вычисляется их произведение

n = pq, которое называется модулем;

3) вычисляется значение функции Эйлера от числа

n=(p-1)(q-1)

4) выбирается целое число e – взаимно простое с n и 1<e< n . Оно называется открытой экспонентой

5) вычисляется число d, мультипликативно обратное к числу e по модулю n, т.е.

d∙e = 1 (mod φn)

Число d называется секретной экспонентой; обычно оно вычисляется при помощи расширенного алгоритма Евклида

6) пара (e, n) публикуется в качестве открытого ключа RSA

7) пара (d, n) играет роль закрытого ключа RSA

Шифрование происходит по следующему алгоритму: берется число m и возводится в степень e по модулю n. На выходе будет получено значение c – зашифрованная информация.

Для расшифровки необходимо значение с возвести в степень d по модулю n. Полученное выражение будет совпадать с исходным.

Схема коллективной подписи методом RSA изображена на рисунке ниже.

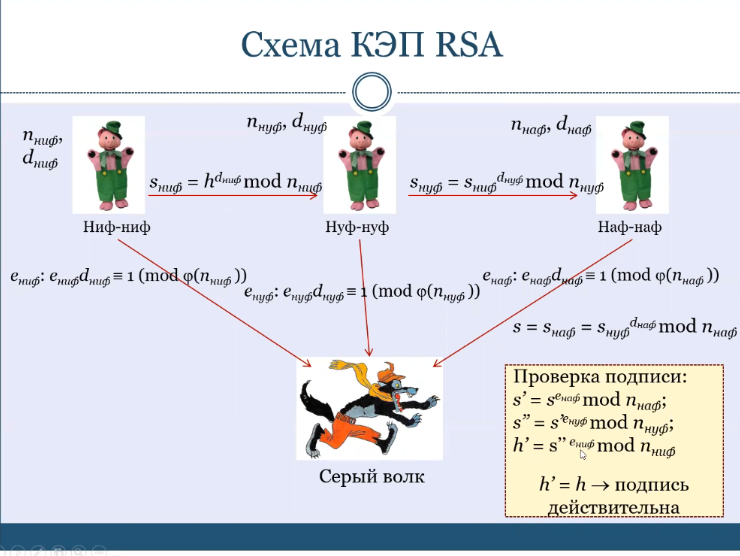


Рисунок 1 -  Схема коллективной электронной подписи RSA

Описание процесса можно сформулировать следующим образом: каждому участнику коллективной подписи присваивается номер в соответствии с модулем n (чем больше n, тем выше номер). Эти номера используются для обозначения порядка подписи и проверки. На вход подается числовое значение, часто представляющее собой хэш, которое первый участник подписывает, возводя в степень d по модулю n (где значения d и n у каждого участника свои). Полученное значение передается следующему участнику, который также подписывает его и передает следующему участнику. Процесс заканчивается, когда последний участник подписывает полученное им значение. Этот цикл формирует коллективную подпись.

Проверка подписи выполняется аналогичным образом, начиная с подписи последнего участника и двигаясь к первому. После проверки подписей всех участников восстанавливается изначальное значение хэша.

**Примеры работы**

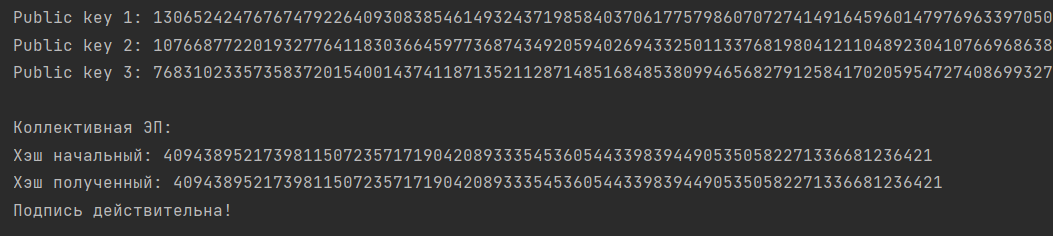


Рисунок 2 – Пример №1

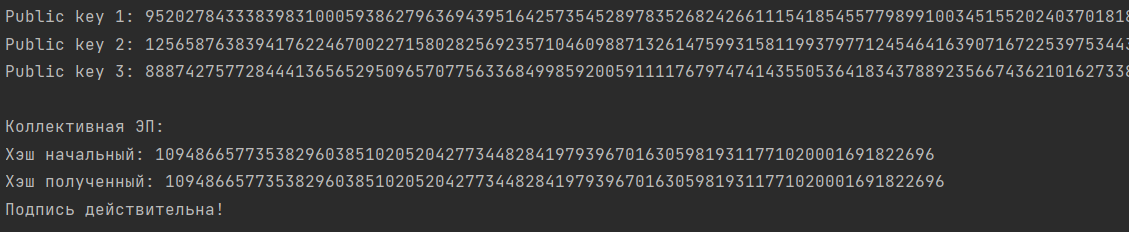


Рисунок 3 – Пример №2

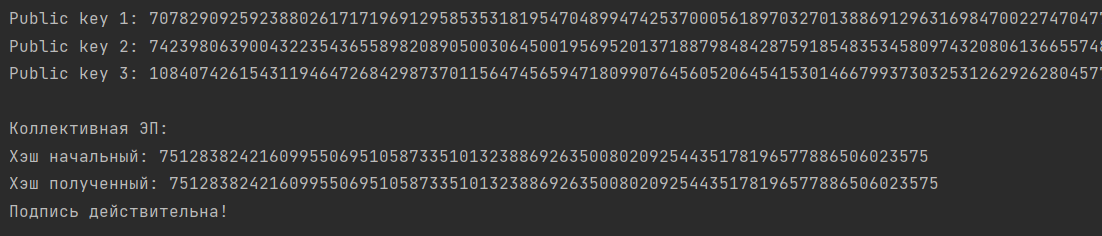


Рисунок 4 – Пример №3