Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**Криптографические методы защиты информации**

**Лабораторная работа №12**

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ГЕНЕРАЦИИ И ВЕРИФИКАЦИИ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ

Вариант №6

Выполнил:

Студент 3 курса 4 группы ФИТ

Левша Марк Сергеевич

Минск 2023

**Цель**: изучение алгоритмов генерации и верификации электронной цифровой подписи и приобретение практических навыков их реализации.

Задачи: 1.

Закрепить теоретические знания по алгебраическому описанию и алгоритмам реализации операций генерации и верификации электронной цифровой подписи (ЭЦП).

2. Получить навыки практической реализации методов генерации и верификации ЭЦП на основе хеширования подписываемых сообщений и алгоритмов RSA, Эль-Гамаля и Шнорра, а также DSA.

3. Разработать приложение для реализации заданных алгоритмов генерации и верификации ЭЦП.

4. Оценить скорость генерации и верификации ЭЦП.

5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Практическое задание**

В ходе лабораторной работы было разработано приложение, которое выполняет генерацию и верификацию ЭЦП на основе алгоритмов RSA, Эль-Гамаля и Шнорра. Так же был выполнен замер времени выполнения всех операций каждого алгоритма.

**ЭЦП на основе RSA**:

*S* ≡ (*H*(*Mo*))*dо* mod *n*o

*S* – подпись, *H*(*Mo) –* хеш начального сообщения, *d*0 и *n*0 – элементы тайного ключа отправителя.

*H*(*Mo*) ≡ (*S*)*ео*mod *no*

*e*0 и *n*0 – элементы открытого ключа.

**ЭЦп на основе Эль-Гамаля:**

*S* = {*a*, *b*}

*a* ≡ *gk* mod *p*;

С помощью расширенного алгоритма Евклида решается уравнение:

*Н*(*Mо*) ≡ (*xa* + *kb*) mod (*p* – 1)

*k* – взаимно простое с *p* – 1, *g* – первообразный корень *p*.

Для верификации:

*ya*×*ab* ≡ *gh* mod *p*

**ЭЦП на основе Шнорра:**

Выбираются числа: *p* – простое число в диапазоне от 512 до 1024 битов; *q* –160-битное простое число, делитель (*p* – 1); любое число *g* (*g* ≠ 1) такое, что

*gq* ≡ 1 mod *p*

Числа *p*, *g*, *q* являются открытыми.

Выбирается число *х* < *q* (*х* является тайным ключом) и вычисляется последний элемент открытого ключа:

*y* ≡ *g* – *х* mod *p*

Секретный ключ имеет длину не менее 160 битов. Для подписи сообщения Мо выбирается случайное число *k* (1 < *k* < *q*) и вычисляет параметр *а*:

*а* ≡ *gk*mod *p*.

Далее создаётся хеш-образ:

*h* = *H*(*Mo*||*a*)

*b* ≡ (*k* + *xh*) mod *q*.

*М*' = *Мо|*|*S*; *S* = {*h*, *b*}

*М*' отправляется. Для проверки получатель вычисляет:

*Х* ≡ *gb*×*yh* (mod *p*)

Затем он проверяет выполнение равенства: *h* = *Н*(*Mп*||*Х*)

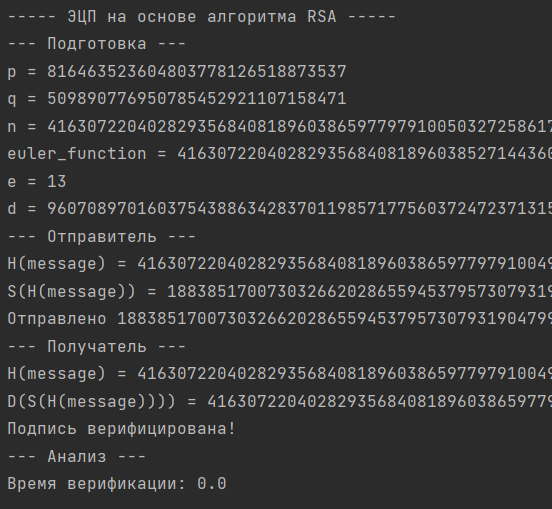
Вывод консоли представлен на рисунках 1-3.

Рис. 1 – вывод консоли генерации и верификации ЭЦП на основе алгоритма RSA

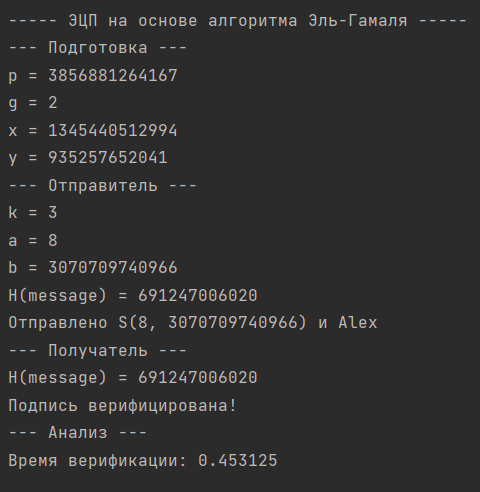
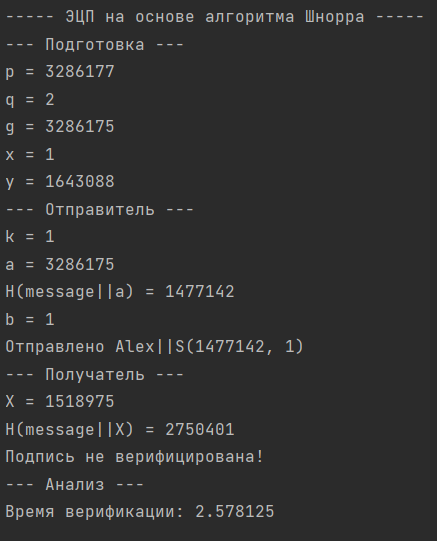
Рис. 2 – вывод консоли генерации и верификации ЭЦП на основе алгоритма Эль-Гамаля

Рис. 3 – вывод консоли генерации и верификации ЭЦП на основе алгоритма Шнорра

**Вывод**: в ходе лабораторной работы были изучены алгоритмы создания электронной цифровой подписи и была разработано консольное приложение в соответствии с целью работы.