МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Информационной безопасности

ОТЧЕТ

по лабораторной работе №8
по дисциплине «Криптография и защита информации»
Тема: Изучение цифровой подписи

Студент гр.8382 _____ Нечепуренко Н.А.
Преподаватель _____ Племянников А.К.

Санкт-Петербург
2021

Цели работы.

Исследовать алгоритмы создания и проверки цифровой подписи, алгоритмы генерации ключевых пар для алгоритмов цифрой подписи RSA, DSA, ECDSA и получить практические навыки работы с ними, в том числе с использованием приложения Cryptool 1 и 2.

Генераторов ключевых пар.

Задание.

1. Перейти к утилите «Digital Signatures / PKI -> PKI / Generate…».
2. Сгенерировать ключевые пары по алгоритмам RSA-2048, DSA2048, EC-239. Зафиксируйте время генерации в таблице.
3. С помощью утилиты «Digital Signatures / PKI -> PKI / Display…» вывести сгенерированный открытый ключ и сохранить соответствующий скриншот.

Основные теоретические положения.

Генерация двух больших простых чисел p и q (p и q держатся в секрете).

1. Вычисление $n = pq$
2. Выбор произвольного $e, (e < n)$, взаимно простого с $\phi(n)$.
3. Вычисление $d : ed \equiv 1 \mod \phi(n)$.
4. Числа (e, n) – открытый ключ, d – закрытый ключ, p и q уничтожаются.

Генерация ключевых пар для алгоритма DSA

1. Выбирается число p: длина - [512,1024] битов и число битов в p должно быть кратно 64.
2. Выбирается число q, которое имеет тот же самый размер дайджеста 160 битов, такое, что: $p - 1 \equiv 0 \mod q$.
3. Выбирается $e_1 : e_1 q \equiv 1 \mod p$.

4. Выбирается целое число $d < q$ и вычисляется $e_2 = e_1 d \mod p$.
5. Числа (e_1, e_2, p, q)- открытый ключ, d – закрытый ключ.

Генерация ключевых пар для алгоритма ECDSA

- Выбирается эллиптическая кривая $E_p(a, b)$, p − простое число.
- Выбирается точка на кривой $e_1 = (x_1, y_1)$.
- Выбирается простое число q – порядок одной из циклических подгрупп группы точек эллиптической кривой: $q \times (x_1, y_1) = 0$.
- Выбирается закрытый ключ d.
- Вычисляется точка на кривой $e_2 = d \times d_1$.
- Открытый ключ - (a, b, q, p, e_1, e_2).

Генерация ключевых пар в Cryptool 1.

С помощью утилиты Generation of Asymmetric Key Pair сгенерируем 3 ключевые пары.

Генерация ключевой пары RSA-2048 заняла 0.795 секунд. Результат генерации представлен на рисунке 1.

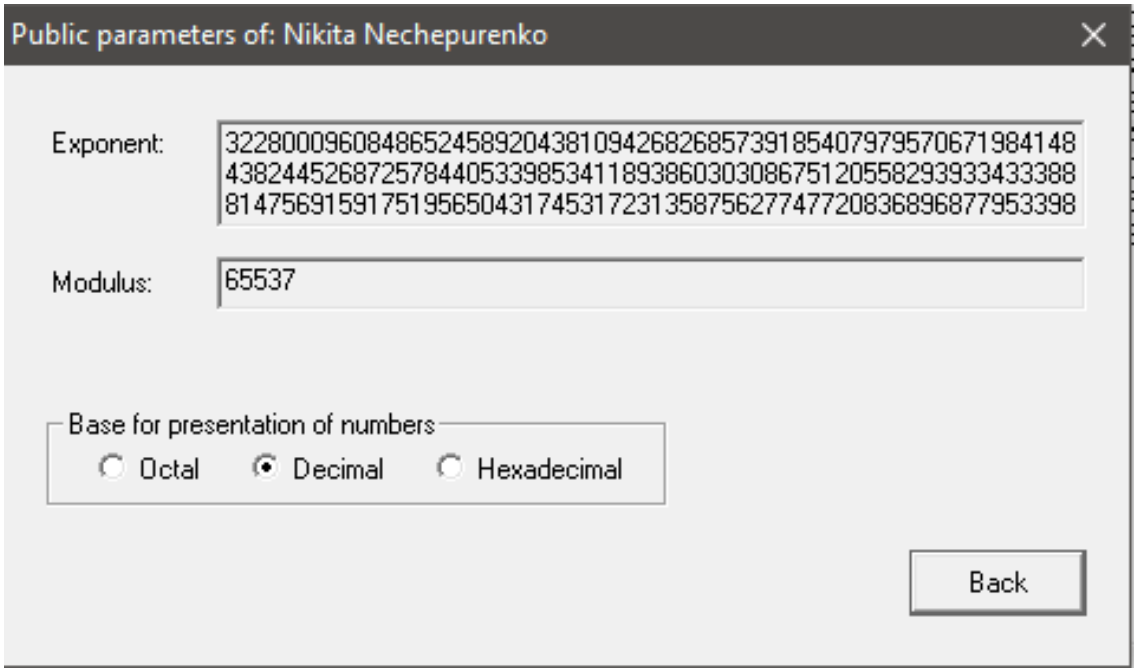


Рисунок 1 – Ключевая пара RSA-2048

Генерация ключевой пары DSA-2048 заняла 1.386 секунд. Результат генерации представлен на рисунке 2.

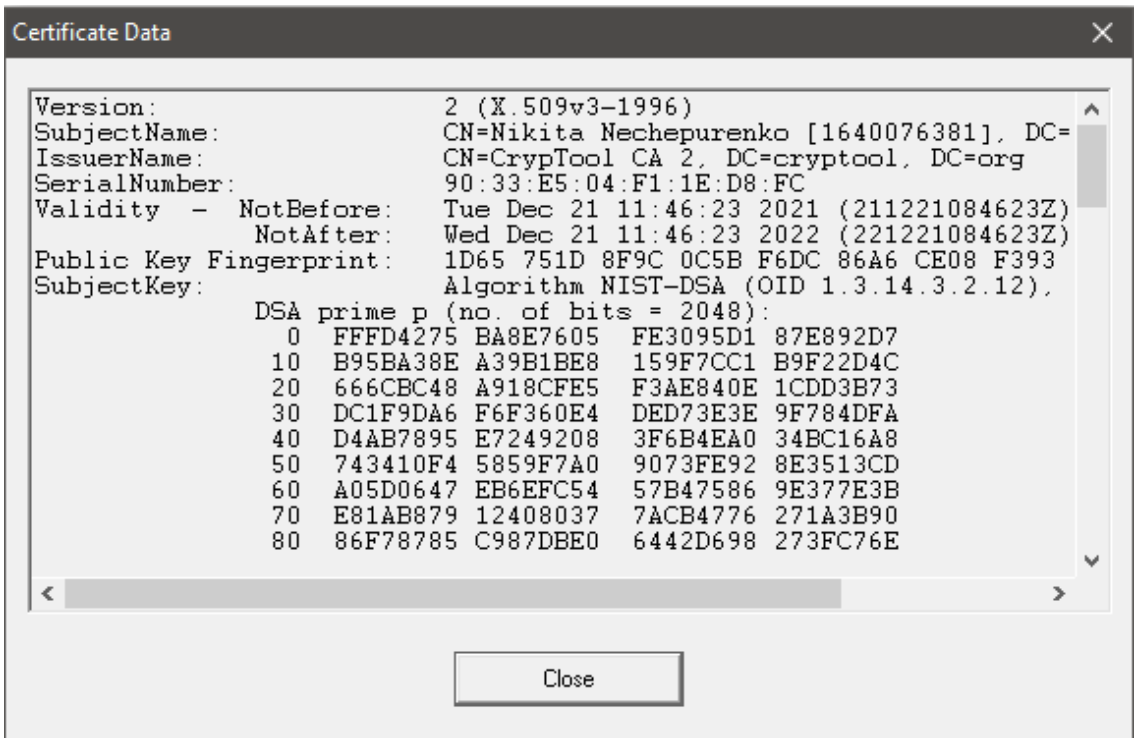


Рисунок 2 – Сертификат DSA-2048

Генерация ключевой пары EC239 заняла 0.02 секунд. Результат генерации представлен на рисунке 3.

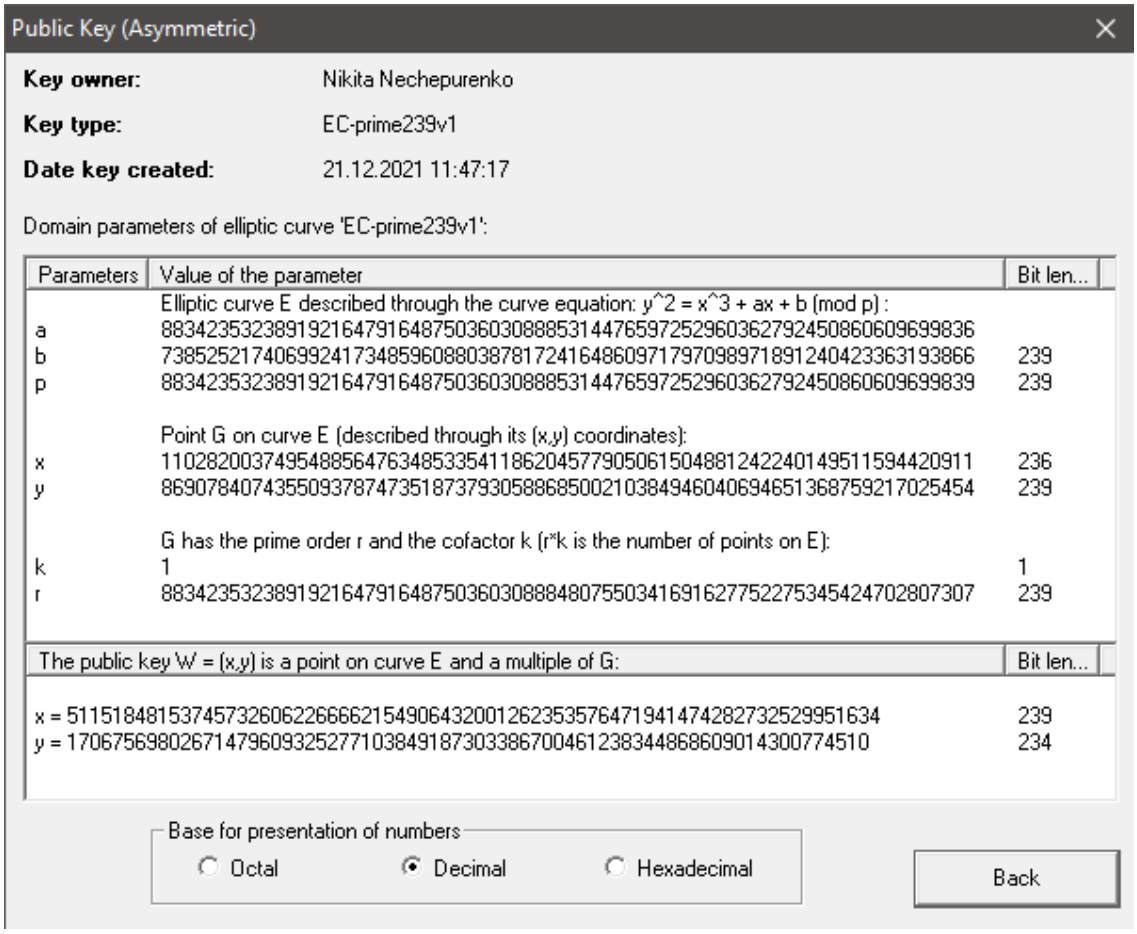


Рисунок 3 – Ключевая пара EC239

Выводы.

Были сгенерированы ключевые пары RSA-2048, DSA-2048, EC239. Время генерации алгоритма DSA примерно в 2 раза больше, чем у RSA-2048. Время генерации алгоритма EC239 наименьшее, примерно на 1-2 порядка меньше, чем RSA-2048.

Процессы создания и проверки цифровой подписи.

Задание.

1. Открыть текст не менее 5000 знаков. Перейти к приложению Digital Signatures / PKI -> Sign Document…
2. Задайте хэш-функцию, и другие параметры цифровой подписи.
3. Создайте подпись ключами, сгенерированными в предыдущем задании. Зафиксируйте время создания цифровой подписи для каждого ключа.
4. Сохраните скриншот цифровой подписи с помощью приложения Digital Signatures / PKI -> Extract Signature.
5. Выполните процедуру проверки подписи Digital Signatures / PKI -> Verify Signature для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

Основные теоретические положения.

Обобщенные схемы подписания и проверки цифровой подписи представлены на рисунке 4.

Рисунок 4 – Схемы работы технологии цифровой подписи

Процессы создания и проверки цифровой подписи в Cryptool 1.

Сгенерируем рыбный текст на 5001 символ.

Подписание полученного документа с хэш-функцией MD5 и алгоритмом подписания RSA с использованием PKI RSA-2048 заняло 0.008 секунд. Полученная сигнатура приведена на рисунке 5.

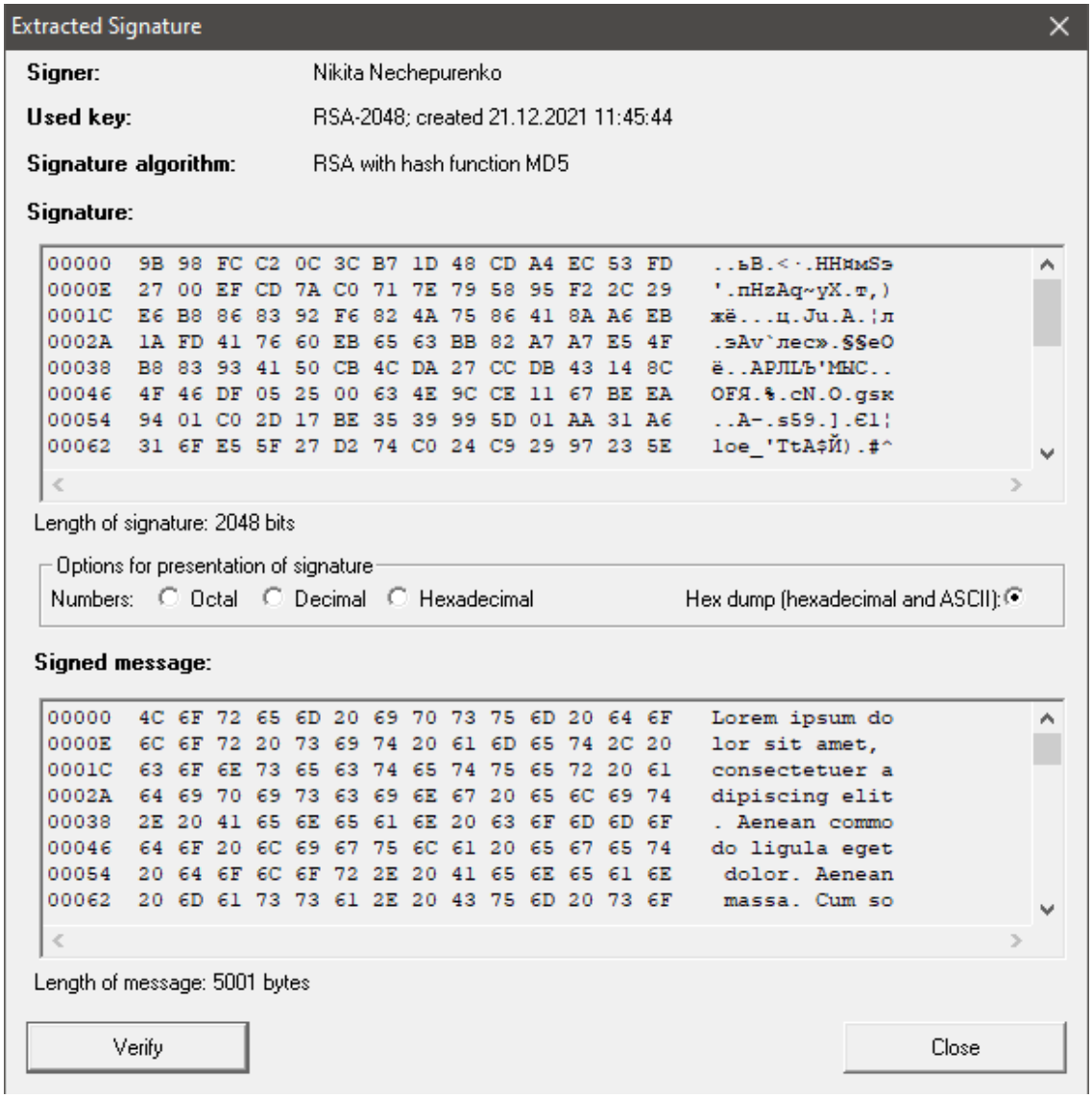


Рисунок 5 – Цифровая подпись с PKI RSA-2048

Корректность подписи была проверена, результаты приведены на рисунке 6.

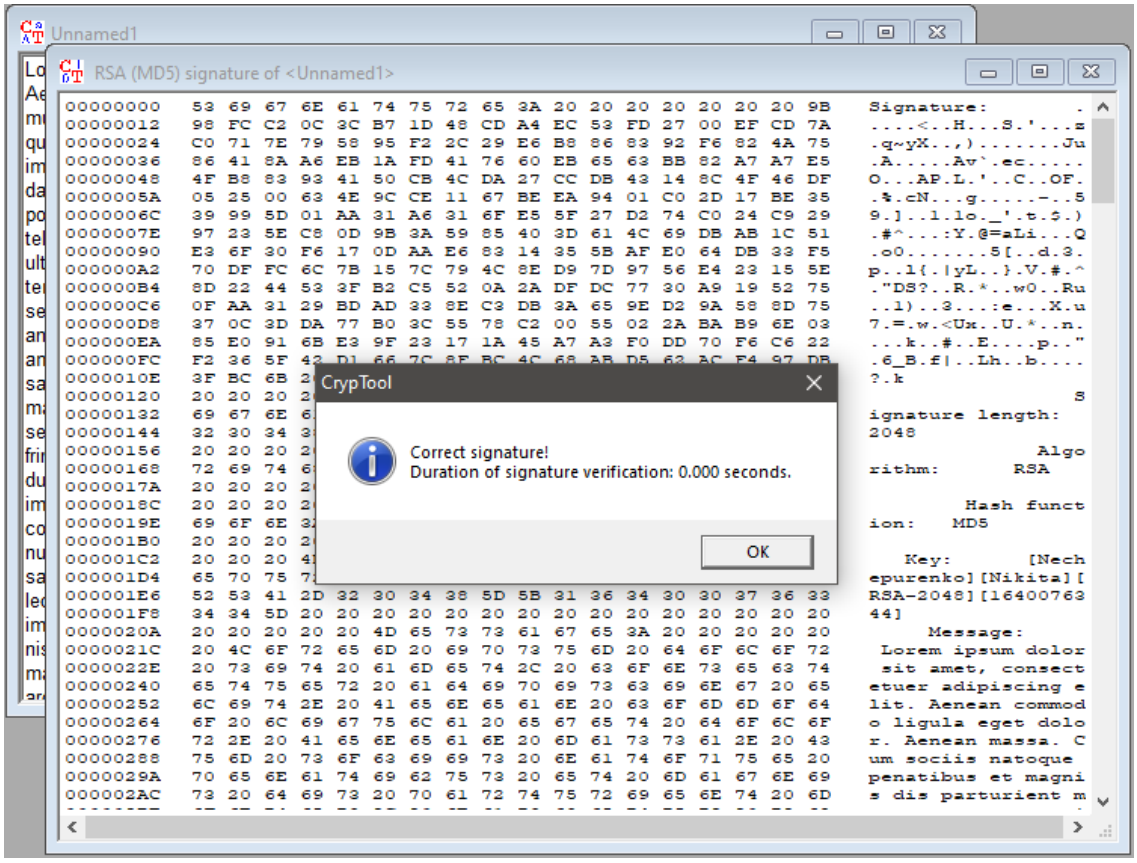


Рисунок 6 – Проверка подписи с PKI RSA-2048

Подписание полученного документа с хэш-функцией SHA-1 с использованием PKI DSA-2048 заняло 0.002 секунды. Полученная сигнатура приведена на рисунке 7.

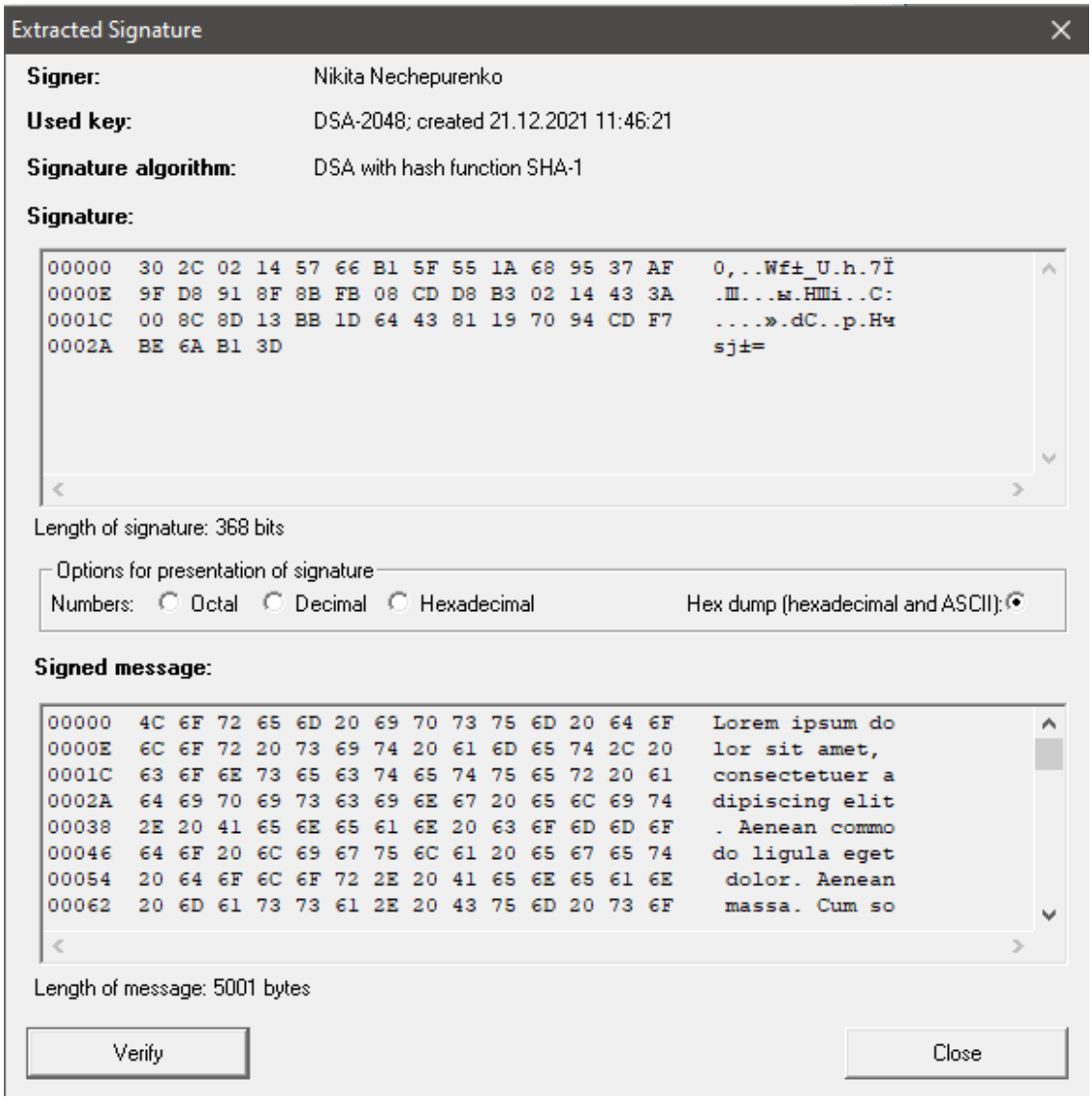


Рисунок 7 – Цифровая подпись с PKI DSA-2048

Корректность подписи была проверена, результаты приведены на рисунке 8.

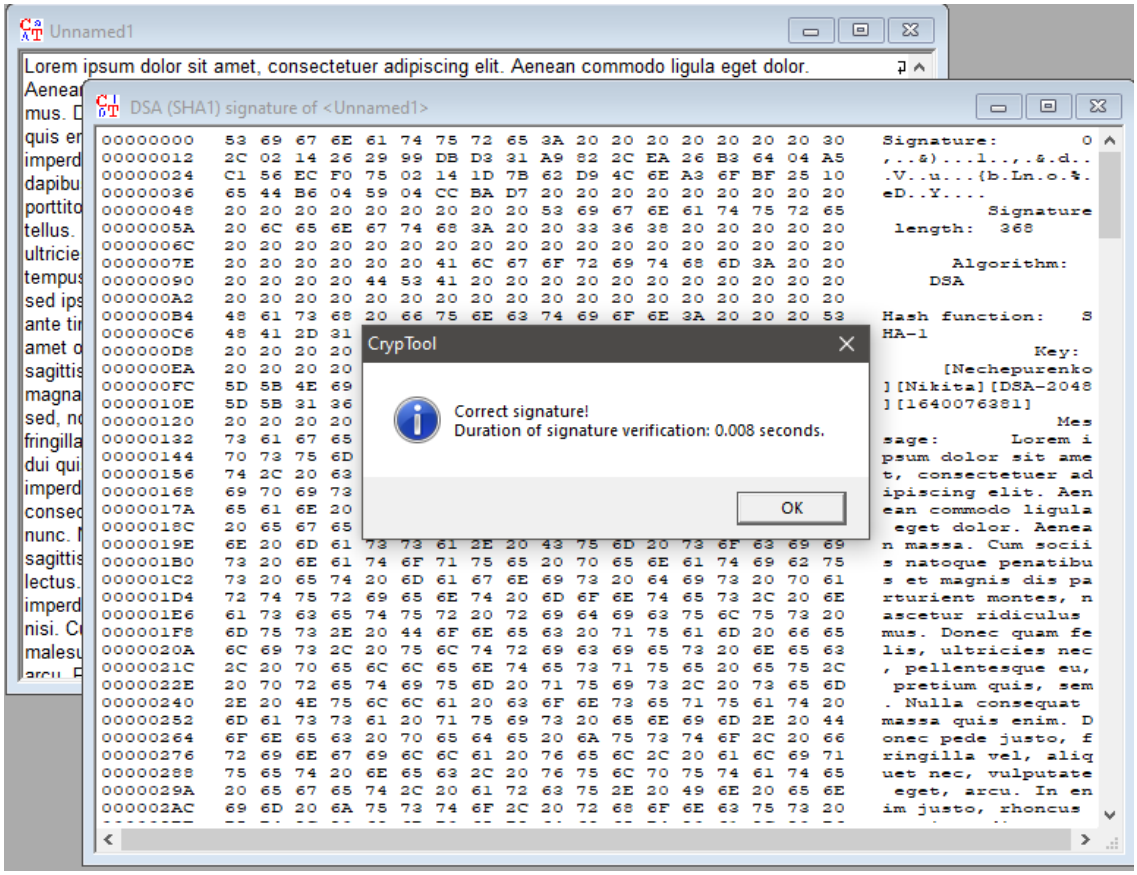


Рисунок 8 – Проверка подписи с PKI DSA-2048

Подписание полученного документа с хэш-функцией SHA-1 и алгоритмом подписания ECSP-DSA с использованием PKI EC239 выполнилось быстрее, чем разрядность числа Cryptool 1 (0.000 seconds). Полученная сигнатура приведена на рисунке 9.

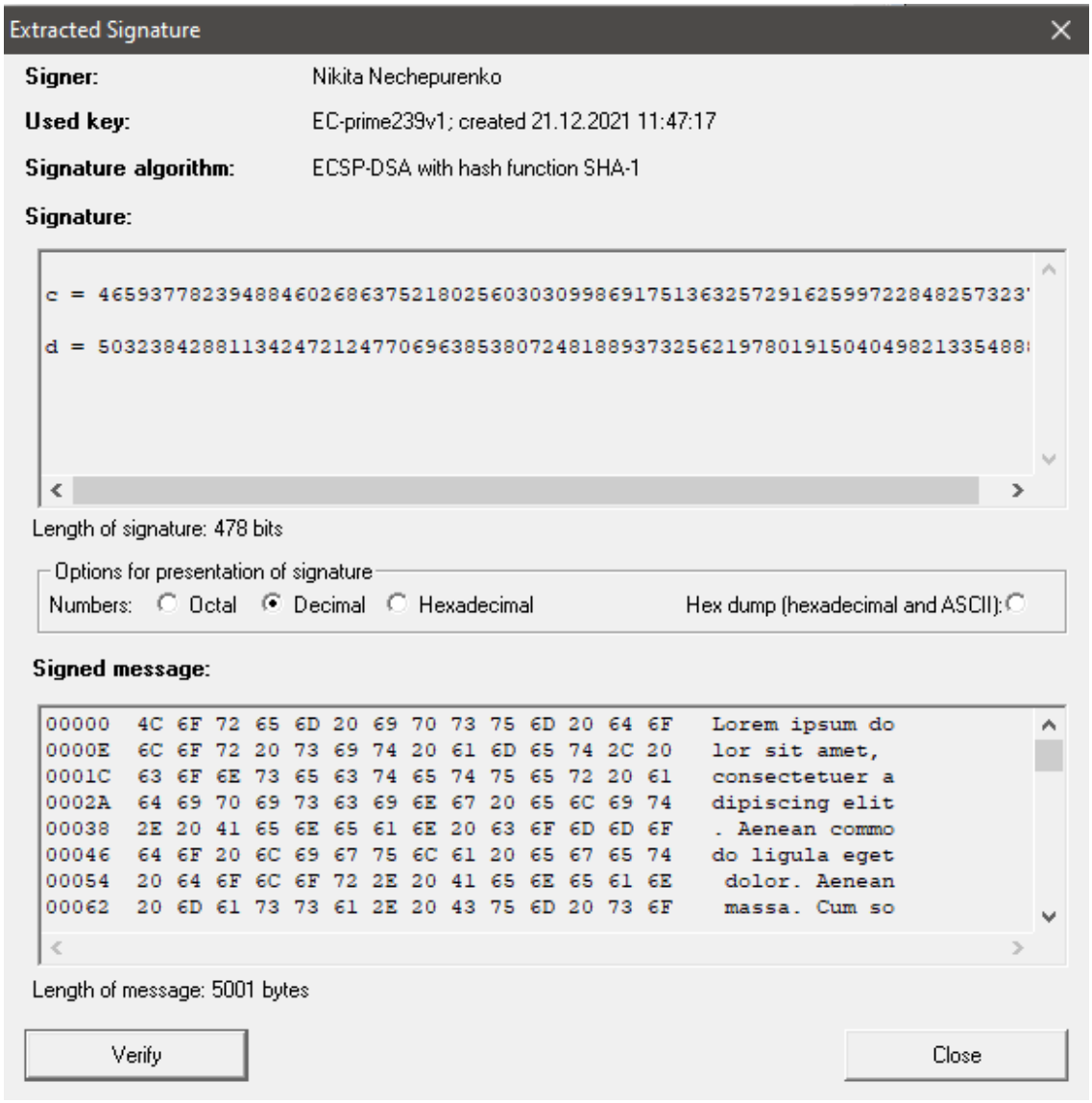


Рисунок 9 – Цифровая подпись с PKI EC239

Корректность подписи была проверена, результаты приведены на рисунке 10.

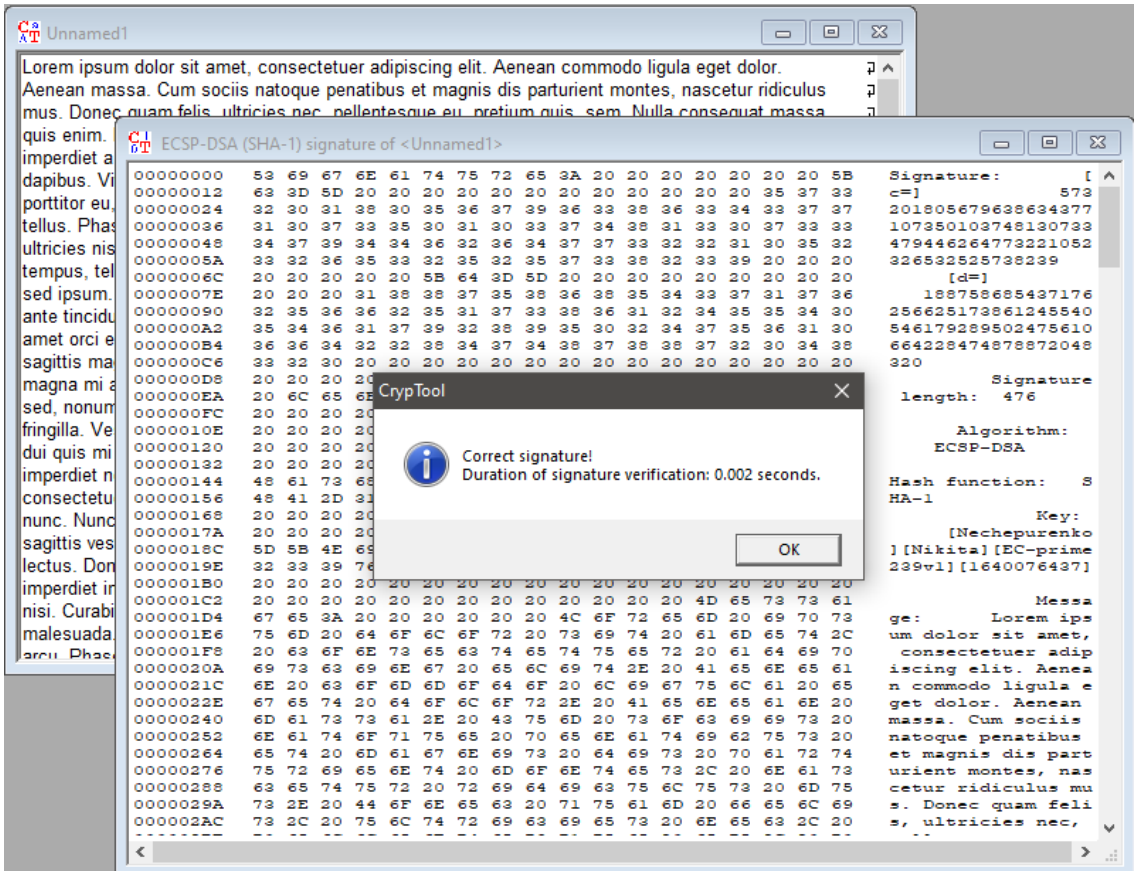


Рисунок 10 – Проверка подписи с PKI EC239

Выводы.

С помощью ключей RSA-2048, DSA-2048 и EC239 были сгенерированы цифровые подписи. Самое большое время генерации было у ключа RSA-2048. Время генерации подписи для ключа EC239 меньше точности числа в Cryptool 1.

Подписанный документ был проверен для каждой подписи, проверки были пройдены успешно.

Схемы цифровой подписи на эллиптических кривых.

Задание.

1. Выполните процедуру создание подписи «Digital Signatures / PKI -> Sign Document…» алгоритмом ECSP-DSA в пошаговом режиме (Display inter.

results=ON). Зафиксируйте скриншоты последовательности шагов.

2. Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.
3. Проверить лекционный материал по ECDSA, выполнив создание и проверку подписи сообщения M (принять $M = h(M)$) приложением Indiv.Procedures -> Number Theory… -> Point Addition on EC.

Основные теоретические положения.

Схема цифровой подписи ECDSA (рисунок 11)

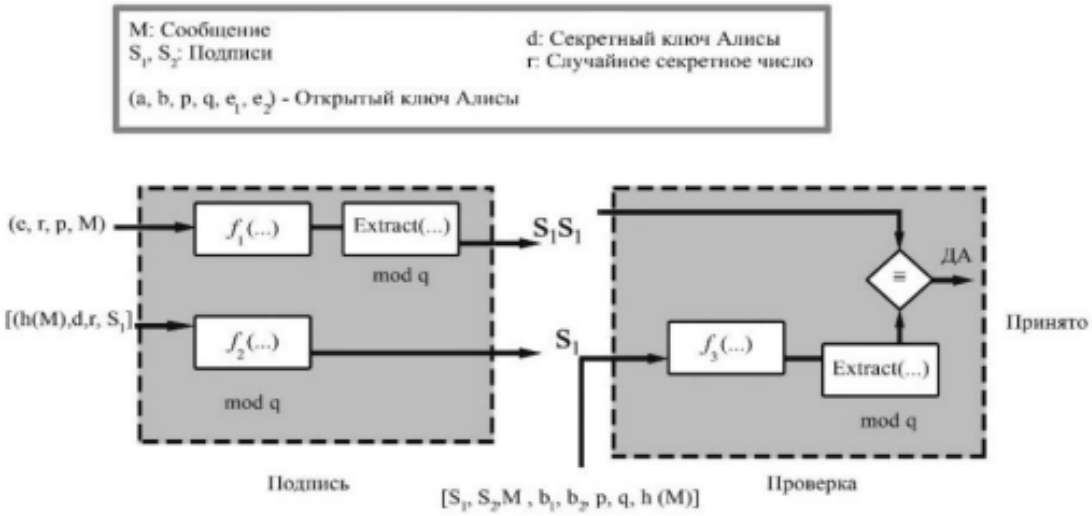


Рисунок 11 – Схема цифровой подписи ECDSA

В процессе подписания две функции f_1 и f_2 и экстрактор Extract создают две части подписи. В процессе проверки (верификации) обрабатывают выход одной функции f_2 (после прохождения через экстрактор) и сравнивают ее с первой частью подписи.

После того, как сгенерирована ключевая пара (закрытый ключ - d, и открытый ключ - (a, b, q, p, e_1, e_2) (см. раздел 1), осуществляется подписание документа, затем на принимающей стороне осуществляется проверка (рисунок

12).

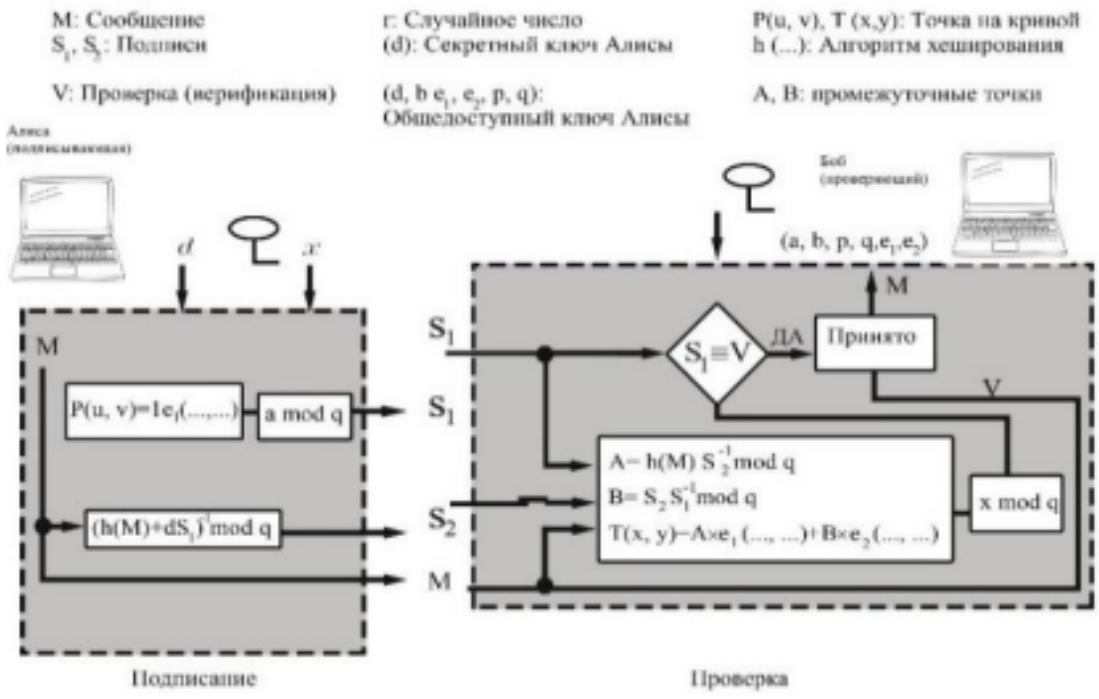


Рисунок 12 – Проверка подписи ECDSA

Алгоритм подписания ECDSA состоит из следующих операций:

- Выбирается секретное случайное число $r : r \in (1, q - 1)$.
- Выбирается третья точка на кривой: $P(u, v) = r \times e_1$.
- Вычисляется первая часть подписи по формуле: $S_1 = u \mod q$, где u-абсцисса.
- Вычисляется вторая часть подписи по формуле: $S_2 = (h(M) + d \times S_1) \times r^{-1} \mod q$, где $h(M)$- дайджест сообщения, d - закрытый ключ.

Алгоритм проверки цифровой подписи ECDSA включает следующие операции:

- Вычисляем промежуточные результаты A и B: $A = h(M) \times S_2^{-1} \mod q$ и $B = S_2^{-1} \times S_2 \mod q$.
- Восстанавливаем третью точку: $T(x, y) = A \times e_1 + B \times e_2$.

- Верификатор $V = x \mod q$ сравнивается с первой частью цифровой подписи S_1.

Цифровая подпись на эллиптических кривых в Cryptool 1.

Приведем скриншоты пошагового создания подписи.

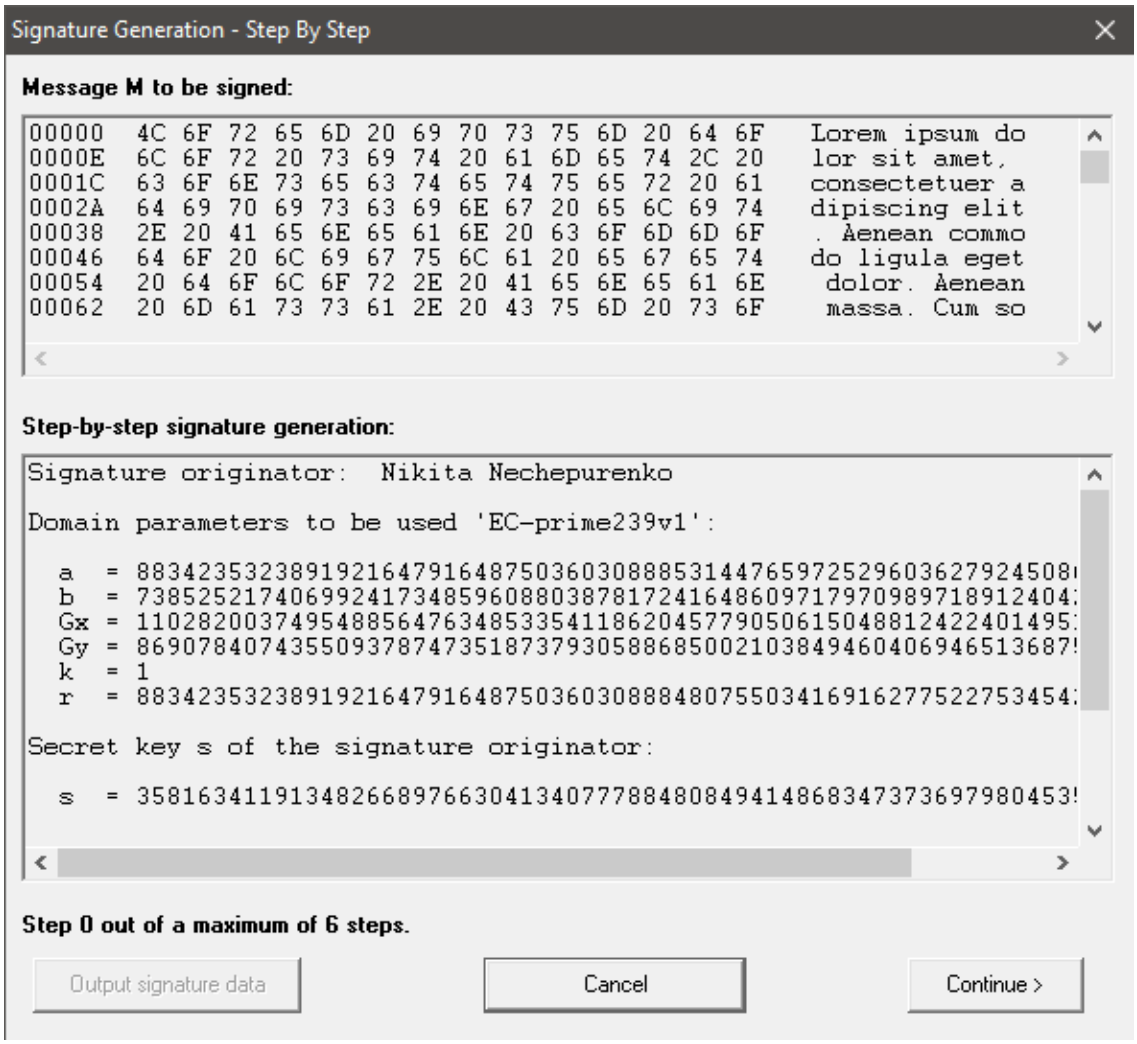


Рисунок 13 – Создание подписи ECSP-DSA, шаг 0

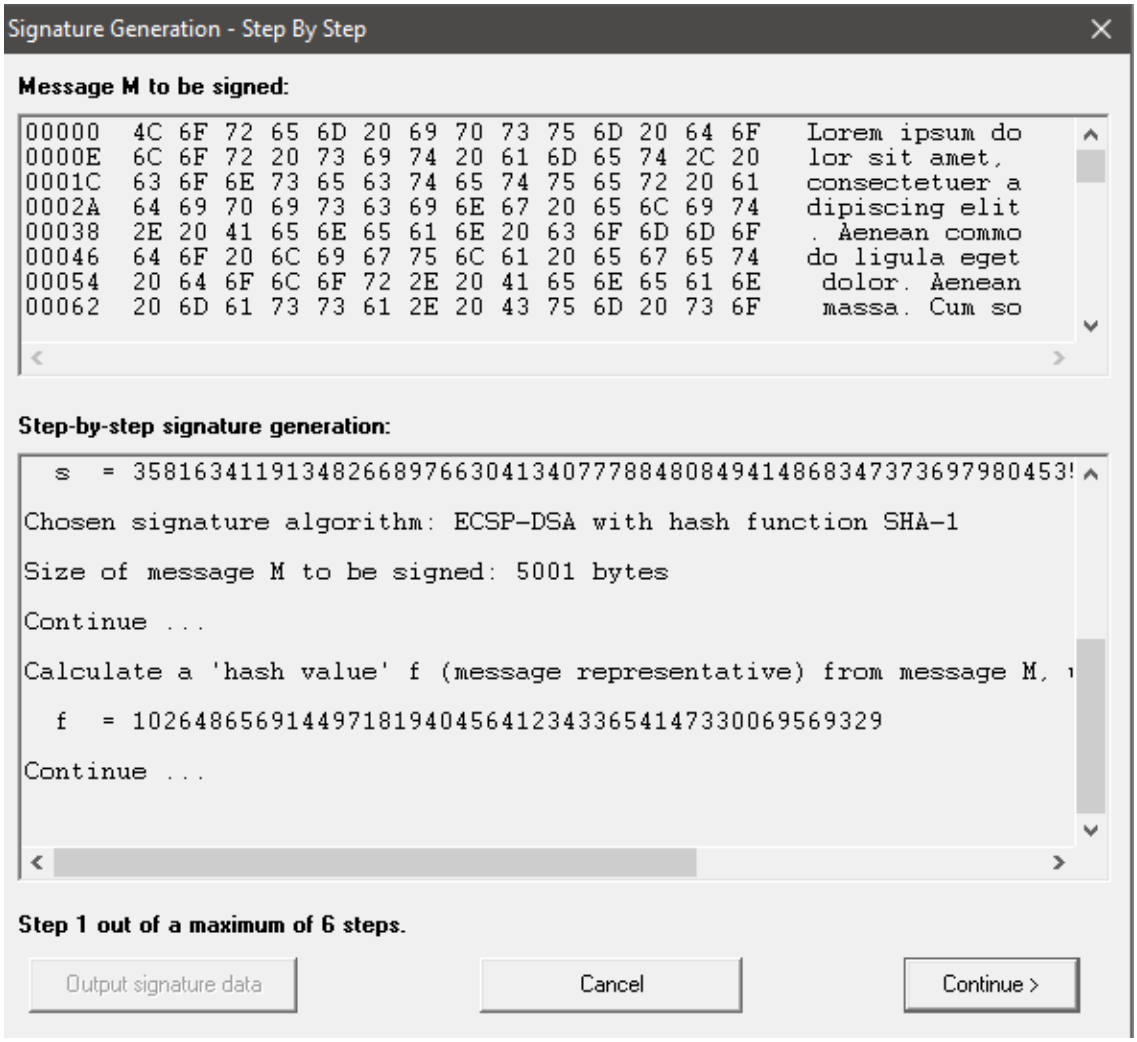


Рисунок 14 – Создание подписи ECSP-DSA, шаг 1

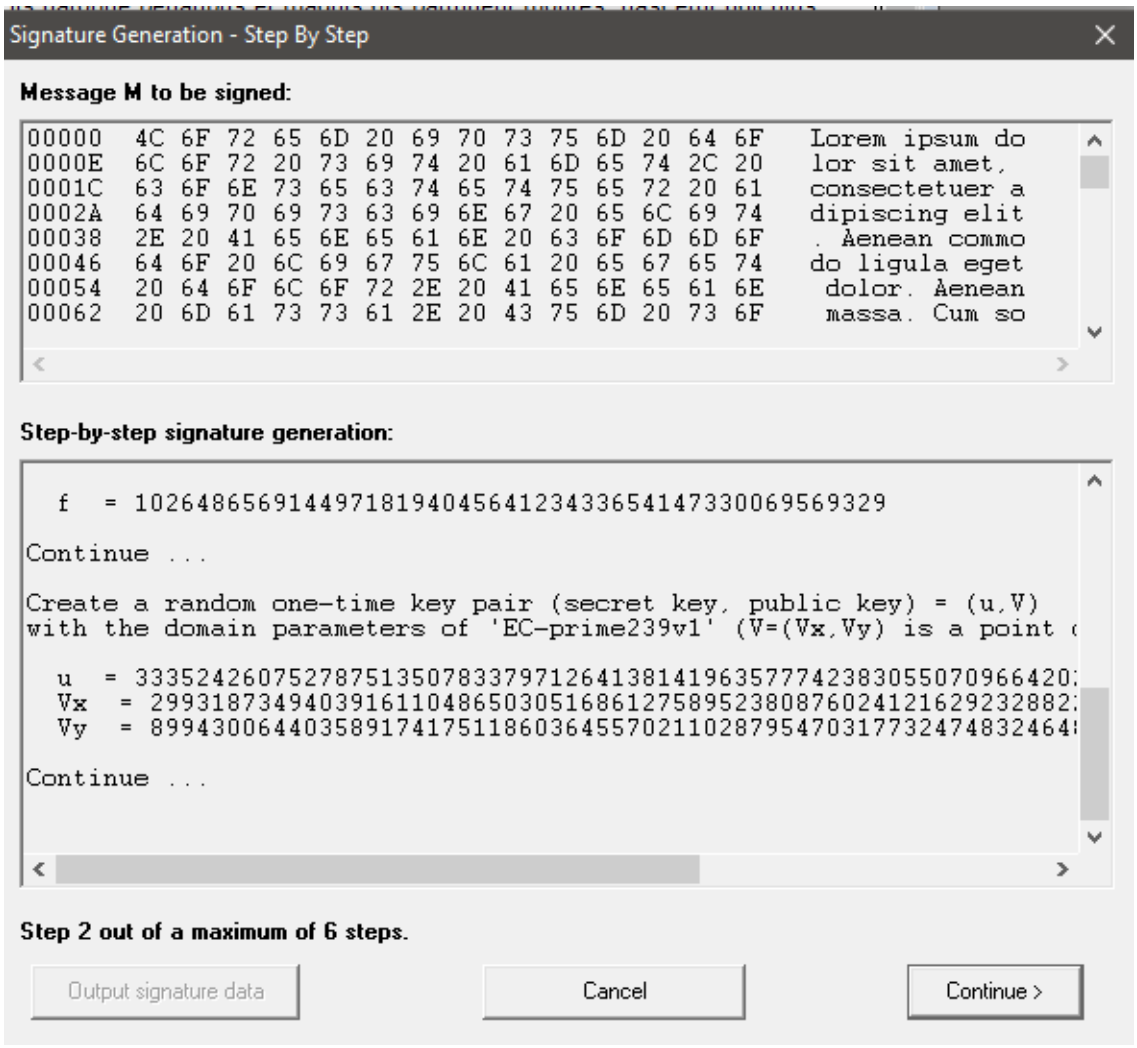


Рисунок 15 – Создание подписи ECSP-DSA, шаг 2

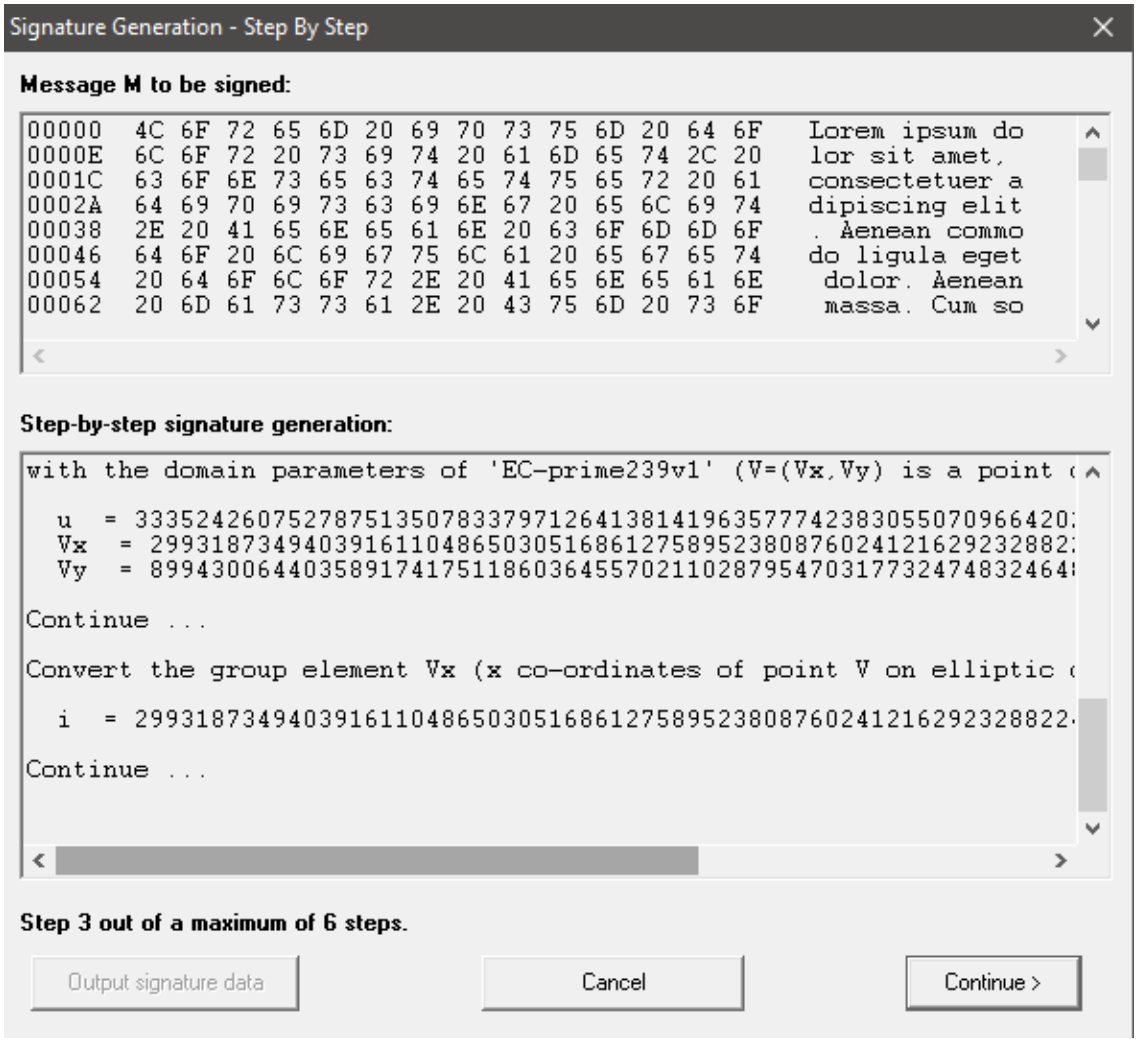


Рисунок 16 – Создание подписи ECSP-DSA, шаг 3

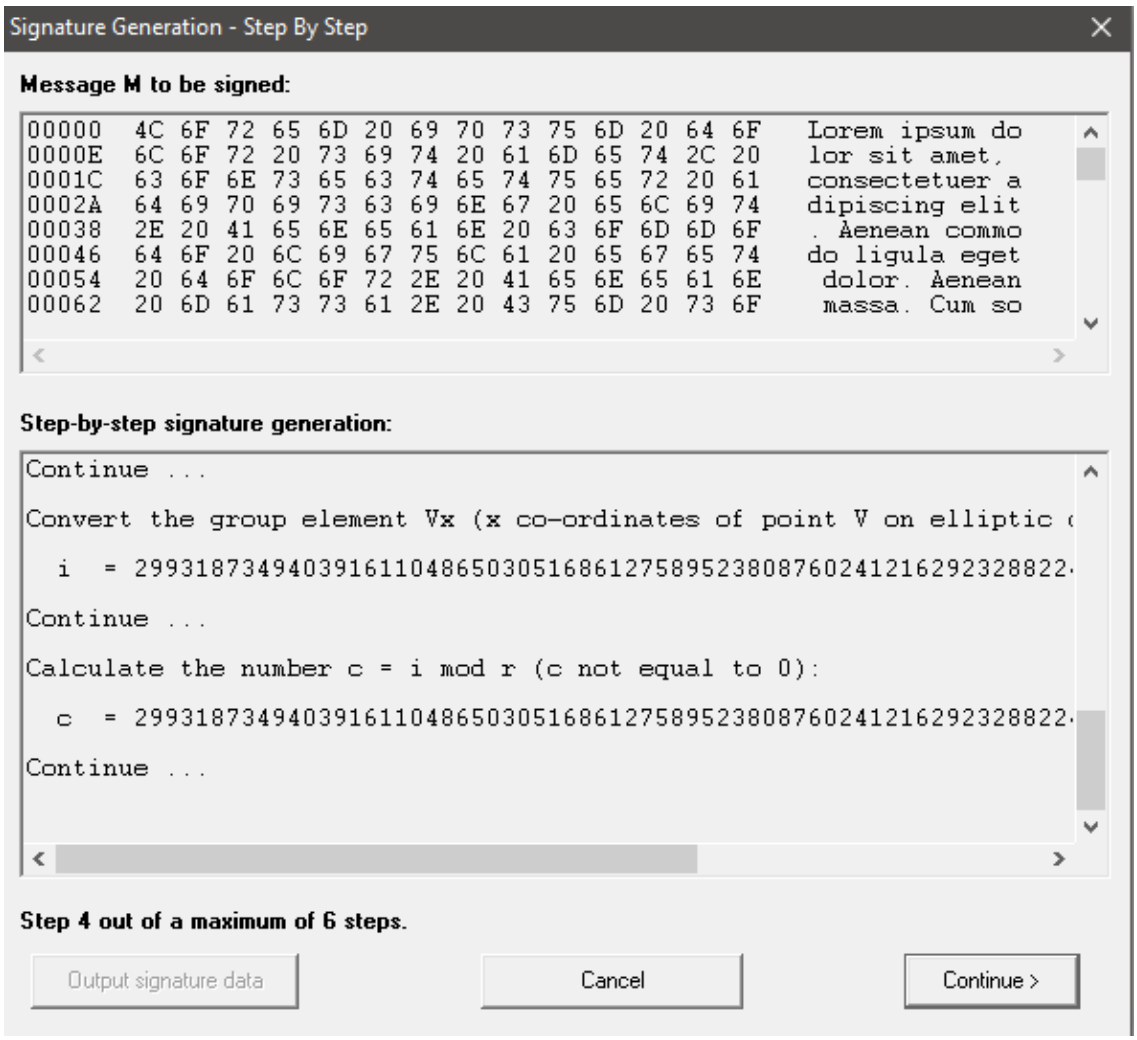


Рисунок 17 – Создание подписи ECSP-DSA, шаг 4

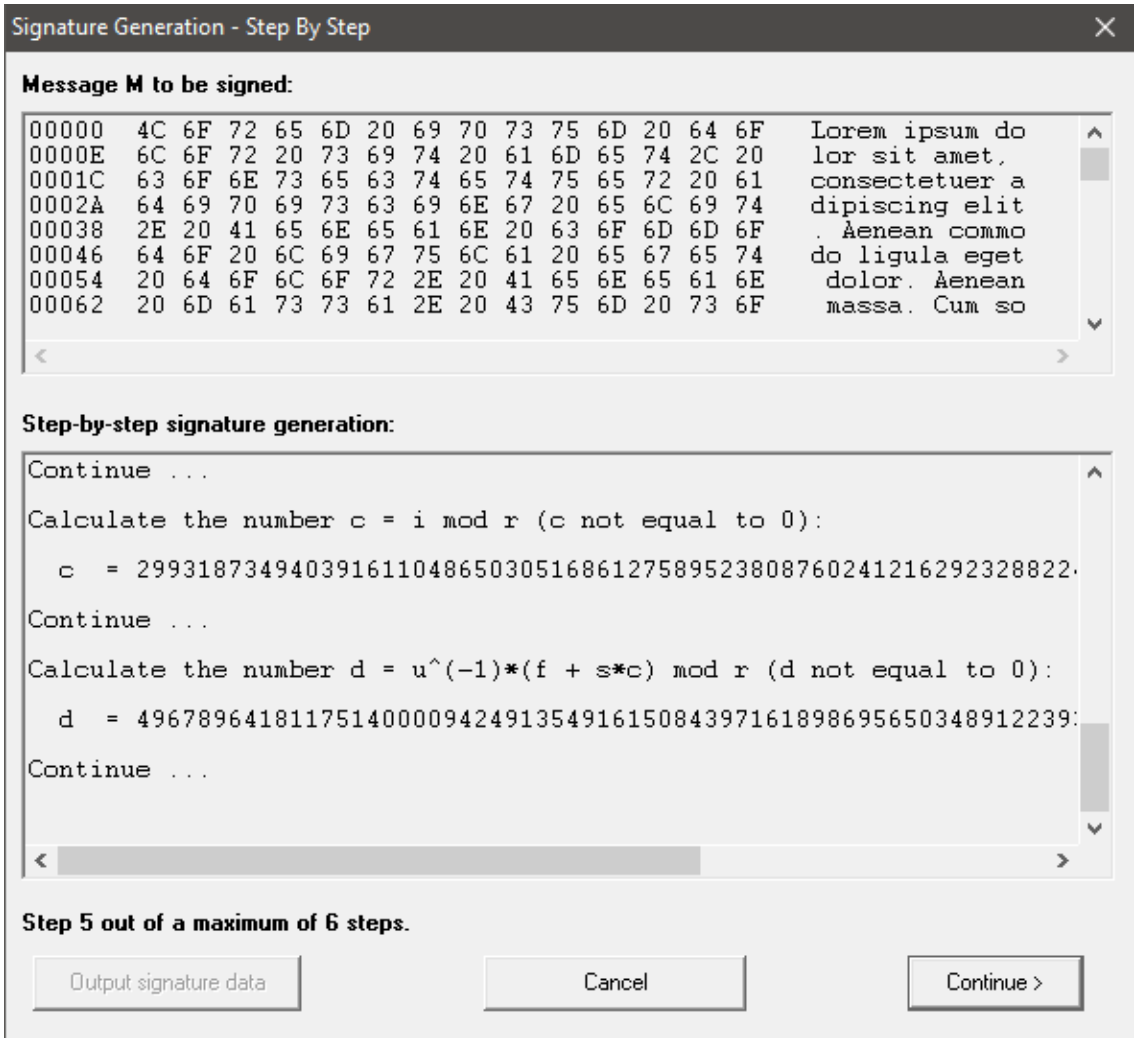


Рисунок 18 – Создание подписи ECSP-DSA, шаг 5

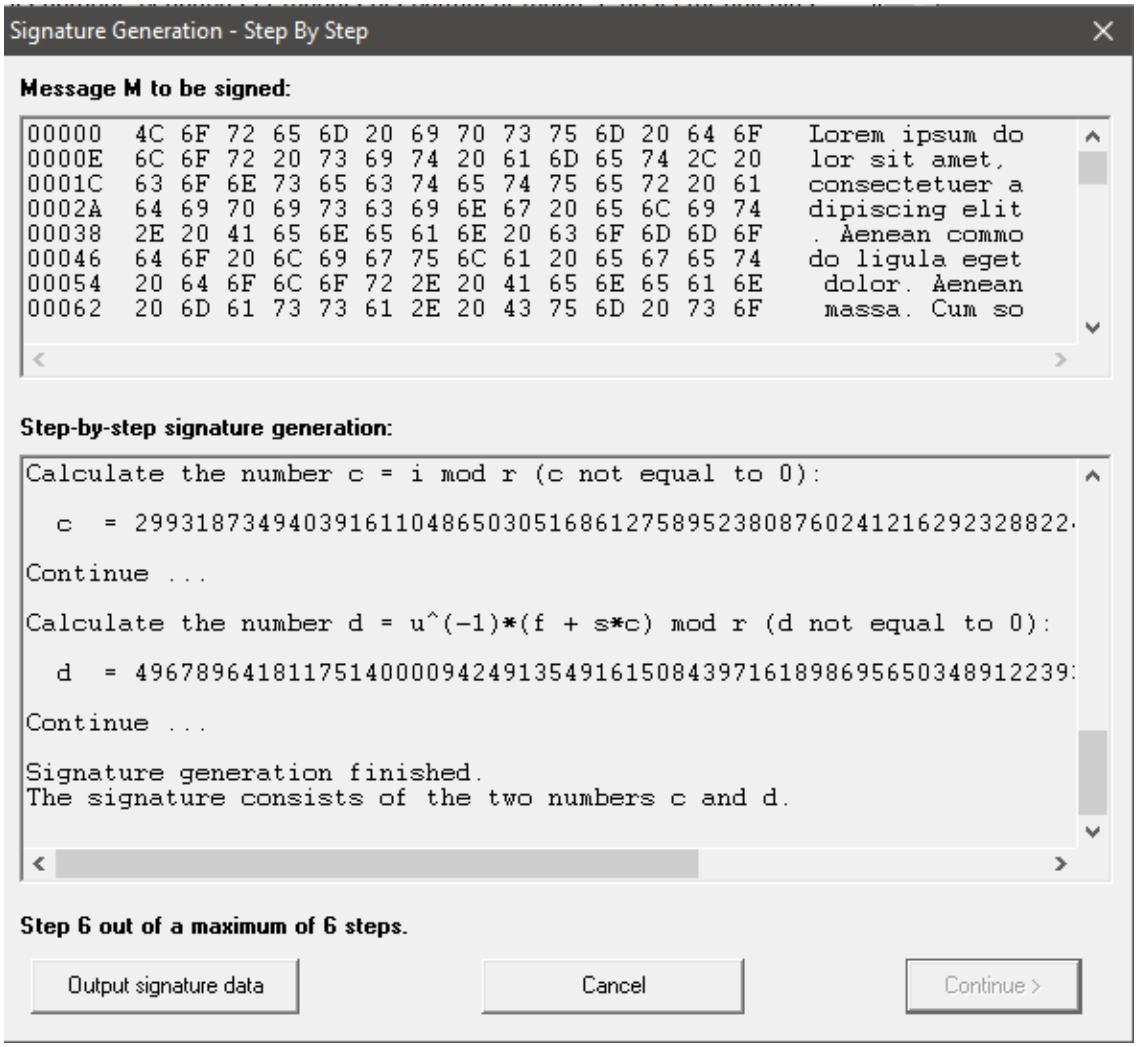


Рисунок 19 – Создание подписи ECSP-DSA, шаг 6

Проверим подпись при соблюдении целостности исходного текста (см. рис. 20).

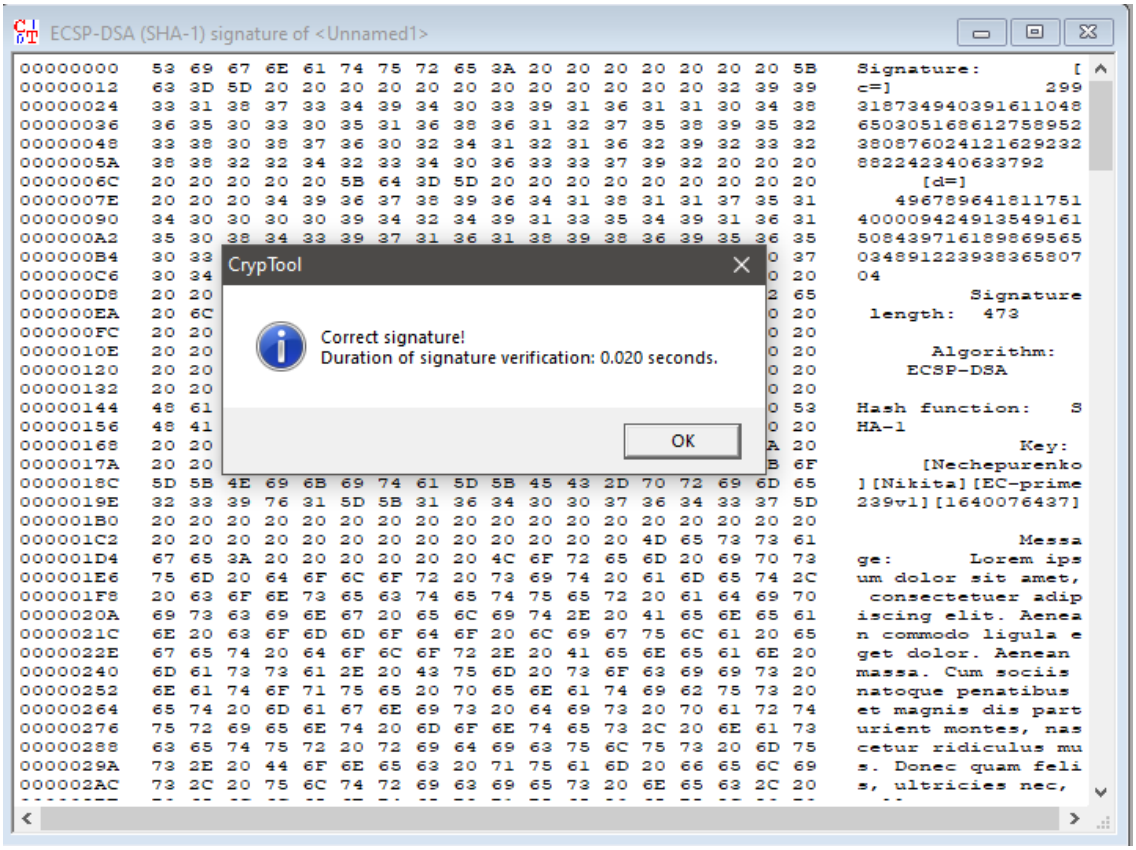


Рисунок 20 – Проверка подписи, корректный текст

Удалим последнее слово открытого текста и проверим целостность сообщения с помощью подписи.

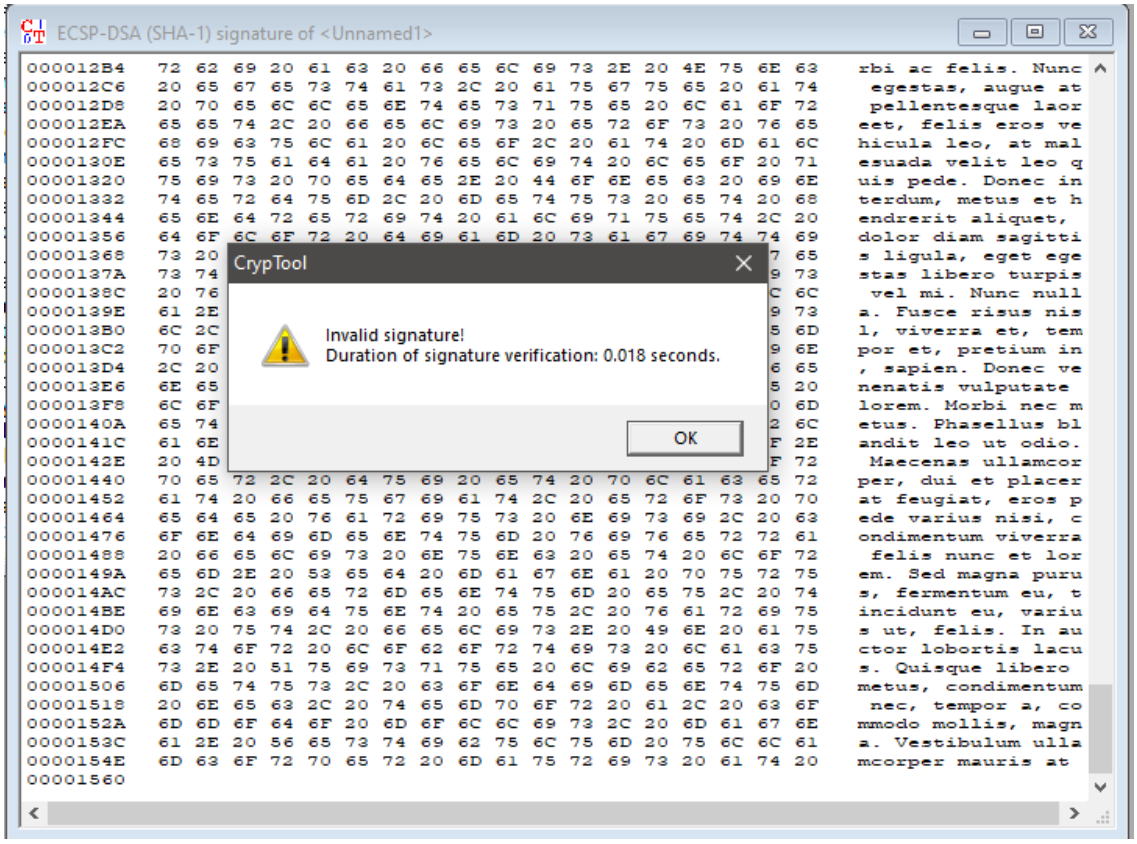


Рисунок 21 – Проверка подписи, измененный текст

Проверим лекционный материал по эллиптическим кривым. Положим $M = h(M) = 25$.

Выберем $E_{13}(-10, 15) : p = 13, a = -10, b = 15$. Также выберем $q = 11$, закрытый ключ $d = 5$, точку на кривой $e_1 = (5, 8)$.

Вторая точка будет $e_2 = d \times e_1 = 5 \times (5, 8) = (5, 5)$. Проверим это с помощью демонстрационного приложения.

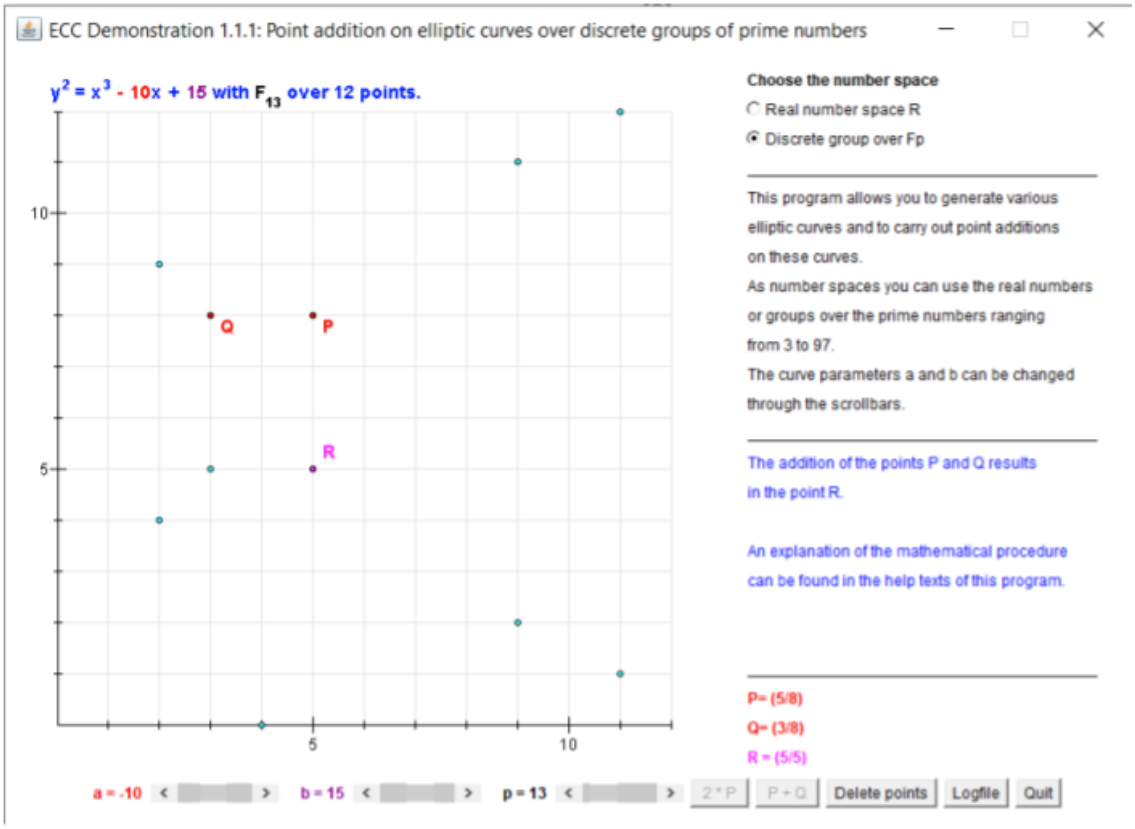


Рисунок 22 – Вычисление точки e_2

Таким образом, получается открытый ключ

$$(a, b, q, p, e_1, e_2) = (-10, 15, 11, 13, (5, 8), (5, 5))$$

Процесс подписания: выберем секретное $r = 7$, тогда третья точка на кривой $P(u, v) = 7 \times (5, 8) = (5, 8)$.

Первая часть подписи $S_1 = u \mod q = 5 \mod 11 = 5$.

Вторая часть подписи $S_2 = (h(M)+d\times S_1)\times r^{-1} \mod q = (25+5\times 5)\times 8 \mod 11 = 4$.

Процесс проверки: вычисляем промежуточные результаты

$$A = h(M) \times S_2^{-1} \mod q = 25 \times 3 \mod 11 = 9$$

$$B = S_2^{-1} \times S_1 \mod q = 3 \times 5 \mod 11 = 4$$

тогда третья точка

$$T(x,y) = A \times e_1 + B \times e_2 = 9 \times (5,8) + 4 \times (5,5) =$$
$$= (4,0) + (3,5) = (5,5)$$

Верификатор $V = x \mod q = 5 \mod 11 = 5$, целостность не нарушена.

Выводы.

Был рассмотрен пошаговый процесс генерации цифровой подписи алгоритмом ECSP-DSA.

Подпись была проверена для исходного и модифицированного сообщения, полученные результаты соответствуют ожиданиям.

В ручном режиме было произведено подписание и верификация открытого текста с помощью ECDSA.

Демонстрация процесса подписи в среде PKI.

Задание.

1. Запустить демонстрационную утилиту «Digital Signatures / PKI -> Signature Demonstration…».
2. Получите сертификат на ранее сгенерированную ключевую пару RSA-2048.
3. Выполните и сохраните скриншоты всех этапов создания цифровой подписи документа.
4. Сохраните скриншот сертификата для проверки этой цифровой подписи.

Основные теоретические положения.

Инфраструктура открытых ключей (ИОК, PKI – Public Key Infrastructure) — набор средств (технических, материальных, организационных и т. д.), распределённых служб и компонентов, в совокупности используемых для поддержки решения основных задач криптографии, а именно:

1. Обеспечение конфиденциальности информации;
2. Обеспечение целостности информации;
3. Обеспечение аутентификации пользователей и ресурсов, к которым обращаются пользователи;
4. Обеспечение возможности подтверждения совершенных пользователями действий

Решение перечисленных задач основано на использовании сертификатов открытых ключей. Сертификат открытого ключа — это электронный документ, который содержит:

1. Открытый ключ пользователя
2. Информацию о пользователе, которому принадлежит сертификат
3. Информацию о сроке действия сертификата
4. Информацию об издателе сертификата
5. Другие атрибуты
6. Цифровую подпись этих данных, созданную удостоверяющим центром, издавшим и выдавшим этот сертификат.

Существует несколько вариантов использования сертификатов открытых ключей:

1. Для зашифрования и расшифрования электронных документов;
2. Для подписания электронного документа и проверки подписи;
3. Для аутентификации отправителя документа.

Создание сертификата в Cryptool 1.

С помощью демонстрационной утилиты сгенерируем сертификат на пару RSA-2048 (см. рис. 23).

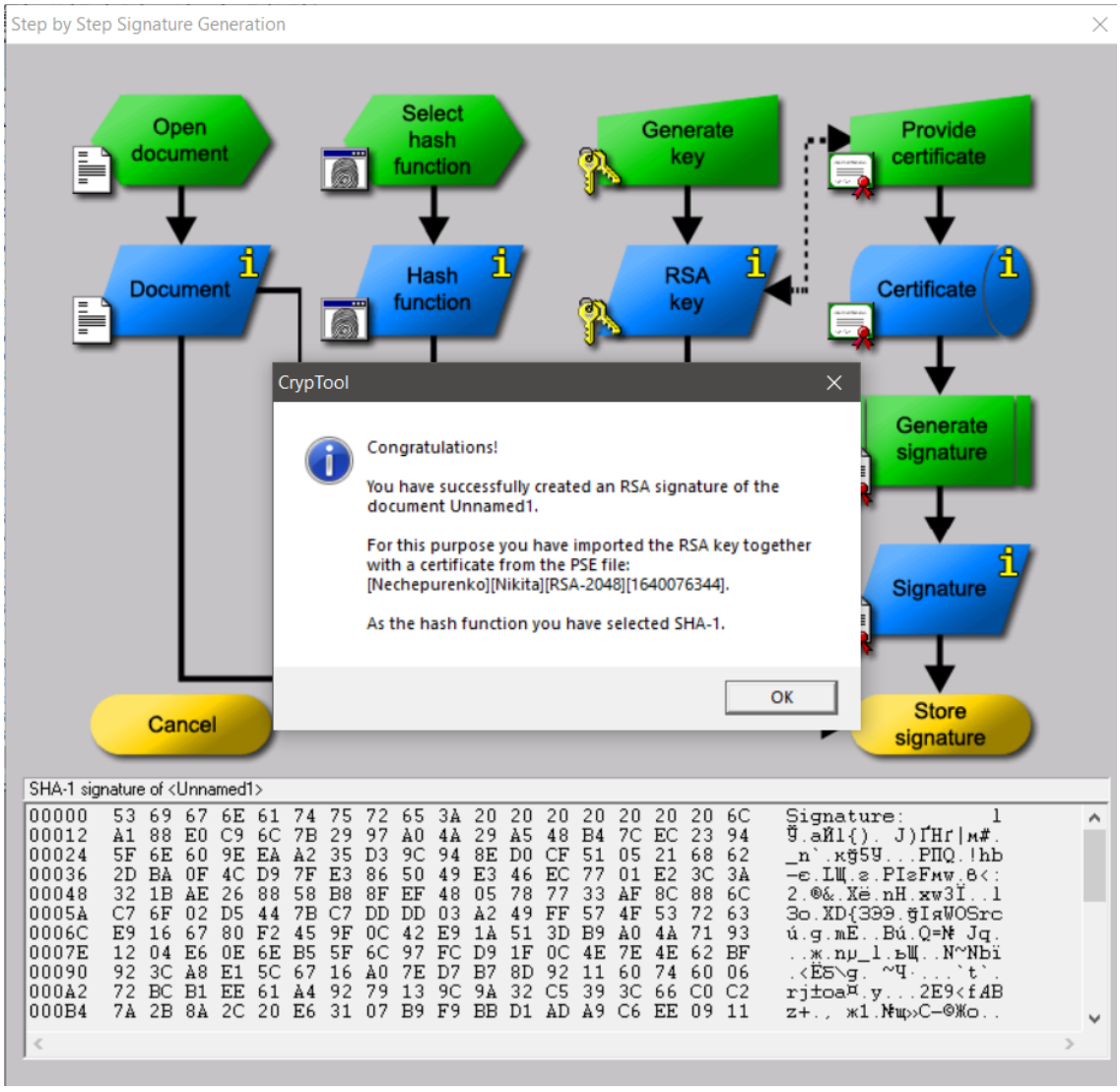


Рисунок 23 – Генерация сертификата

Проверка:

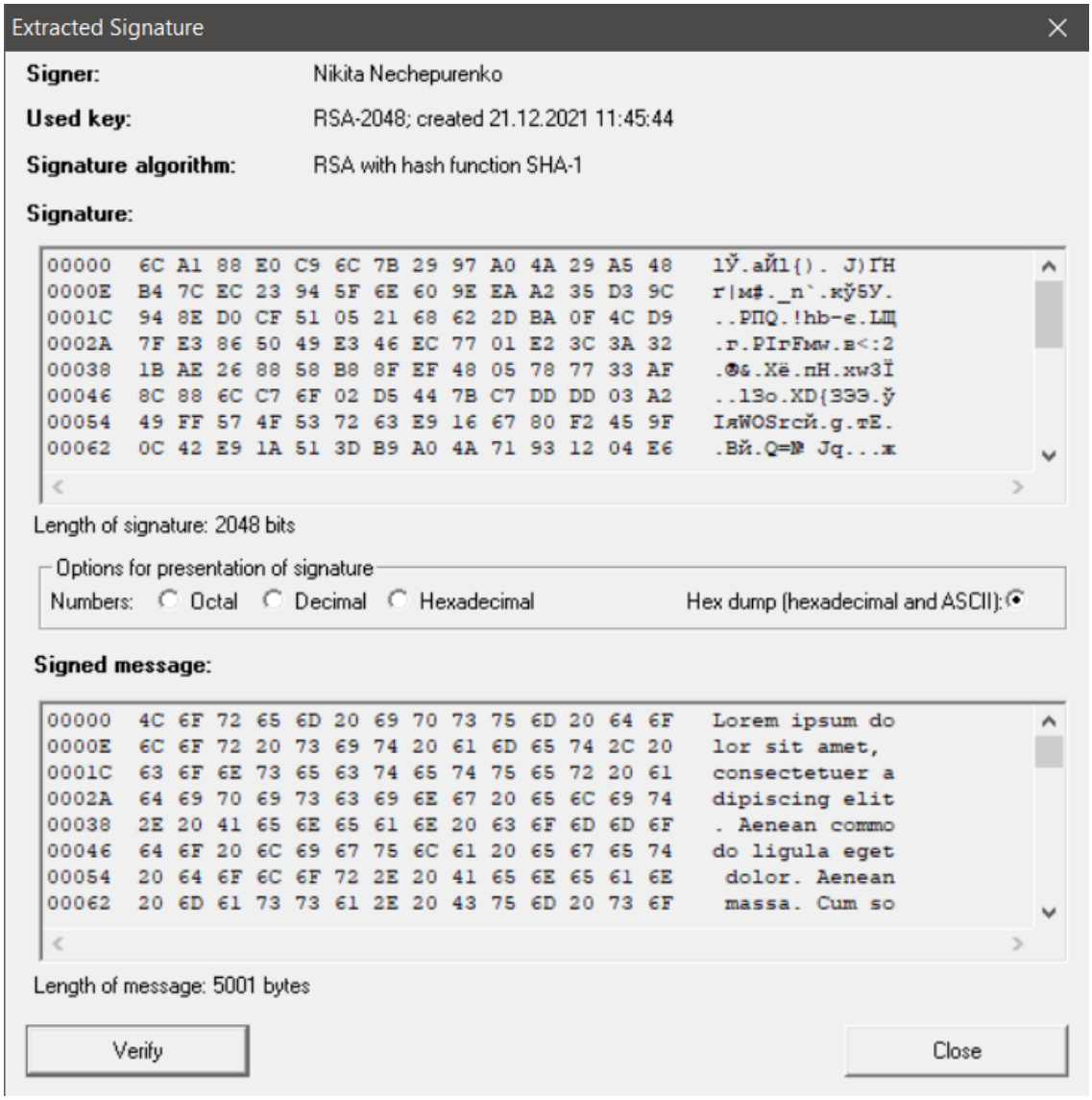


Рисунок 24 – Проверка подписи

Подписание своего отчета.

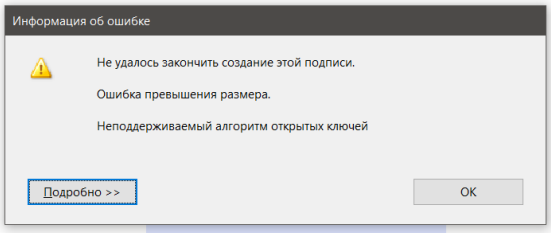
Задание.

1. Сконвертируйте отчет в формат pdf.
2. Экспортируйте ранее созданный сертификат ключевой пары RSA Digital Signatures / PKI -> PKI / Generate… -> Export PSE(#PKCS12).
3. Откройте pdf-версию отчета и попытайтесь подписать с использованием этого сертификата.

4. Создайте собственный самоподписанный сертификат в среде Adobe Reader и используйте его для подписи отчета.
5. Сохраните скриншоты свойств подписи и сертификата.
6. Внесите изменения (маркеры, комментарии) в отчет и проверьте подпись.

Подписание своего отчета в Adobe Acrobat.

Пара ключей, сгенерированная в Cryptool 1 не подошла для подписания отчета в Adobe Acrobat (см. рис. ниже).



Студент гр.8382 _____ Нечепуренко Н.А.

Преподаватель _____ Племянников А.К.

Рисунок 25 – Ошибка подписания

В Adobe Acrobat была сгенерирована новая пара, отчет был подписан ей.

ОТЧЕТ

по лабораторной работе №8

по дисциплине «Криптография и защита информации»

Тема: Изучение цифровой подписи

Студент гр.8382 Nikita Nechepurenko Подписано цифровой подписью: Nikita Nechepurenko Дата: 2021.12.25 13:24:51 +03'00' Нечепуренко Н.А.

Преподаватель _____ Племянников А.К.

Санкт-Петербург

2021

Рисунок 26 – Цифровая подпись на титульном листе

При внесении изменений цифровая подпись перестает быть действительной.

Заключение.

В ходе выполнения лабораторной работы были исследованы алгоритмы создания и проверки цифровой подписи, алгоритмы генерации ключевых пар RSA, DSA, ECDSA, с помощью программного продукта CrypTool 1.

1. Были сгенерированы ключевые пары по алгоритмам RSA-2048, DSA-2048 и EC-239 и зафиксировано время их генерации. Время генерации алгоритма DSA примерно в 2 раза больше, чем у RSA-2048. Время генерации алгоритма EC239 наименьшее, примерно на 1-2 порядка меньше, чем RSA-2048.
2. Затем были созданы подписи ключами, сгенерированными ранее, и зафиксировано время. Все подписи были созданы примерно за одинаковое время (разница в несколько тысячных секунды). Также была выполнена процедура проверки этих подписей для случаев сохранения и нарушения целостности исходного текста. При нарушении целостности текста проверка была провалена.
3. Так же был изучен способ формирования и проверки подписи алгоритмом ECDSA, основанный на эллиптической кривой над конечными полями.
4. Далее был изучен механизм подписи в среде PKI Public Key Infrastructure и была рассмотрена одна из областей применения (а также создания) сертификатов – подпись документов.
5. Промежуточная версия отчета была подписана с использованием Adobe Acrobat. После внесения изменений в подписанный pdf-файл программа установила, что файл был изменен.