**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИБ**

отчет

**по лабораторной работе №8**

**по дисциплине «Криптографические методы защиты информации»**

Тема: **Изучение цифровой подписи**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9363 |  | Труханова В.А. |
| Преподаватель |  | Племянников А.К. |

Санкт-Петербург

2023

**Цель 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.

**Генерация ключевых пар**

* + 1. **Задание**
* Перейти к утилите «Digital Signatures/PKI->PKI/Generate…».
* Сгенерировать ключевые пары по алгоритмам RSA-2048, DSA-2048, EC-239. Зафиксируйте время генерации в таблице.
* С помощью утилиты «*Digital Signatures/PKI-> PKI/Display…*» вывести сгенерированный открытый ключ и сохранить соответствующий скриншот.
  + 1. **Описание алгоритмов генерации ключевых пар.**

Для RSA:

* Выбираются два больших простых числа p и q;
* Вычисляется ;
* Выбирается произвольное число e (, взаимно простое с ;
* Вычисляется закрытый ключ d: ;
* Пара чисел (e, n) объявляются открытым ключом, d выбирается закрытым ключом;
* p и q нужно уничтожить.

Для DSA:

* Выбирается число p: длина – [512, 1024] бит, число бит в p должно быть кратно 64;
* Выбирается число q, которое имеет тот же размер в битах, что и размер дайджеста используемой хеш-функции (160 бит для SHA-1) и удовлетворяющее условию ;
* Выбирается ;
* Выбирается целое число и вычисляется ;
* Числа – открытый ключ, *d –* закрытый ключ.

Для ECDSA:

* Выбирается эллиптическая кривая , p – простое число;
* Выбирается точка на кривой ;
* Выбирается простое число q – порядок одной из циклических подгрупп группы точек эллиптической кривой: ;
* Выбирается закрытый ключ d;
* Вычисляется точка на кривой ;
* Открытый ключ – .
  + 1. **Таблица с фактическими временем генерации ключевых пар.**

|  |  |
| --- | --- |
| Алгоритм | Время (секунда) |
| RSA-2048 | 1.134 |
| DSA-2048 | 1.282 |
| EC-239 | 0.012 |

* + 1. **Скриншоты со значениями открытых ключей.**



Рисунок 1 – Сгенерированный ключ RSA-2048

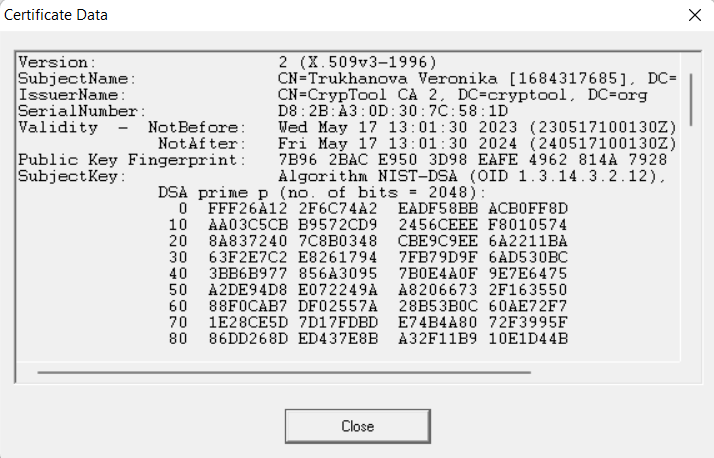


Рисунок 2 – Сгенерированный ключ DSA-2048

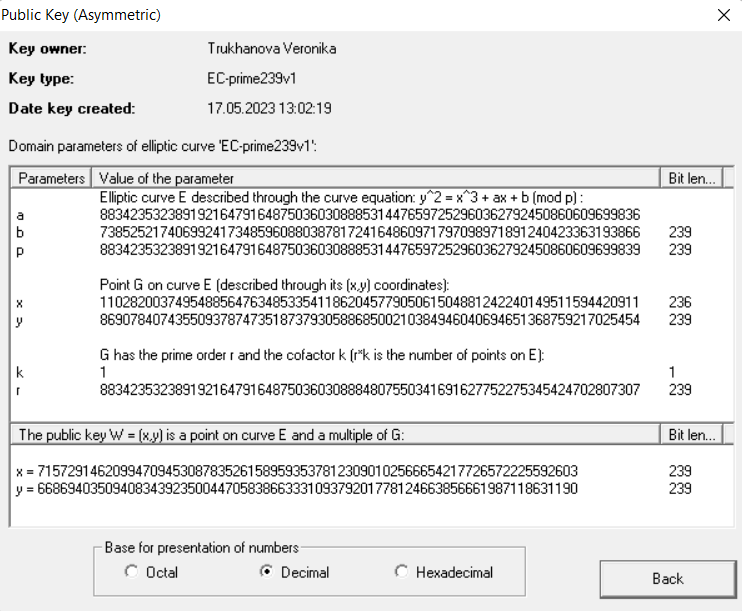


Рисунок 3 – Сгенерированный ключ EC-239

**Процессы создания и проверки цифровой подписи**

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



Рисунок 4 - Схема создания и проверки цифровой подписи

* + 1. **Таблица с фактическими временем генерации цифровой подписи.**

Выбранная хеш-функция – SHA-1

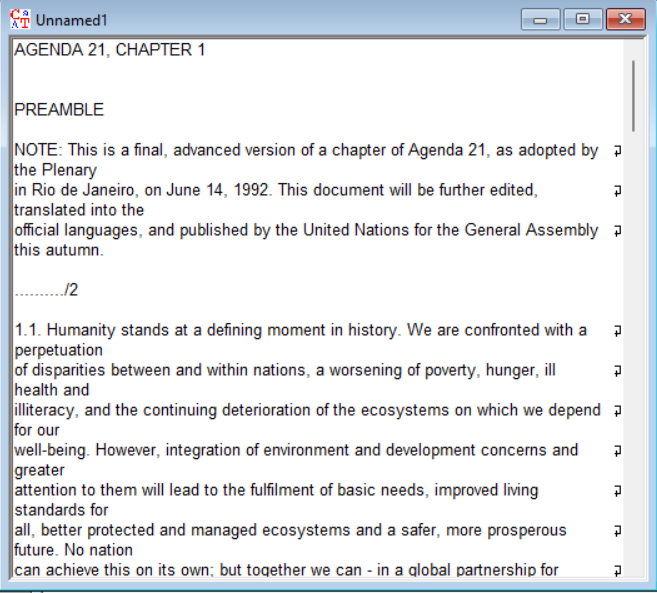
****

Рисунок 5 – Выбранный текcт

|  |  |
| --- | --- |
| **Ключ** | **Время (секунда)** |
| RSA-2048 | 0.006 |
| DSA-2048 | 0.004 |
| EC-239 | 0.005 |

* + 1. **Скриншоты со значениями цифровой подписи**

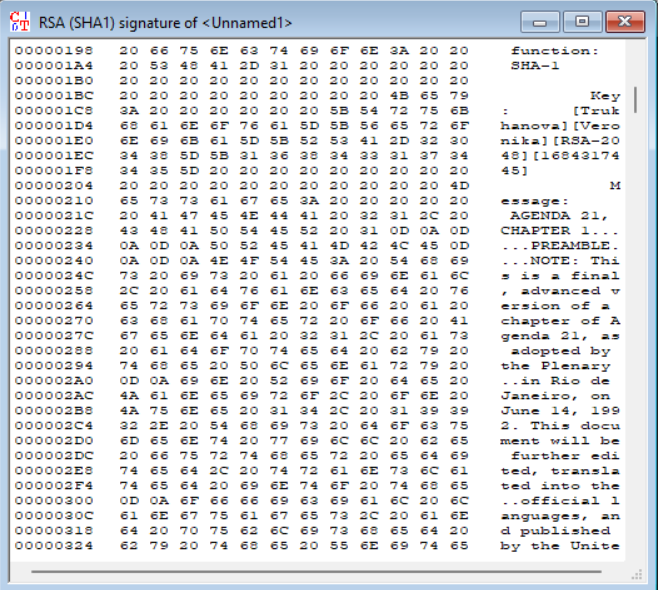


Рисунок 6 – Значение цифровой подписи для ключа RSA-2048

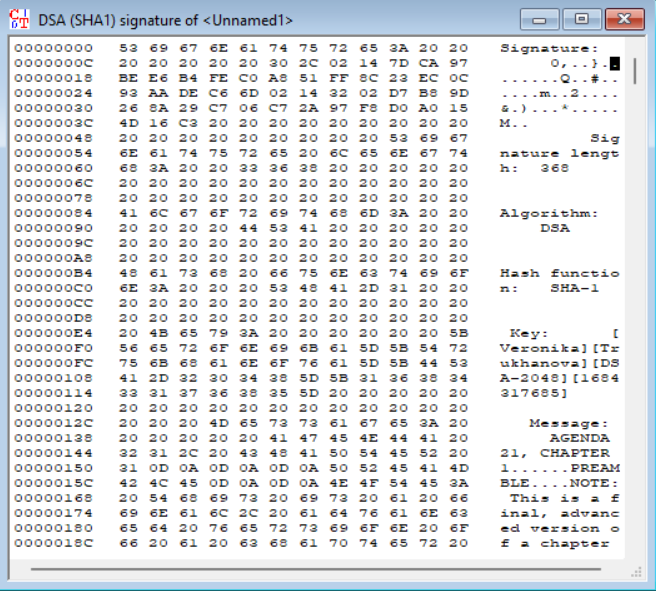


Рисунок 7 – Значение цифровой подписи для ключа DSA-2048

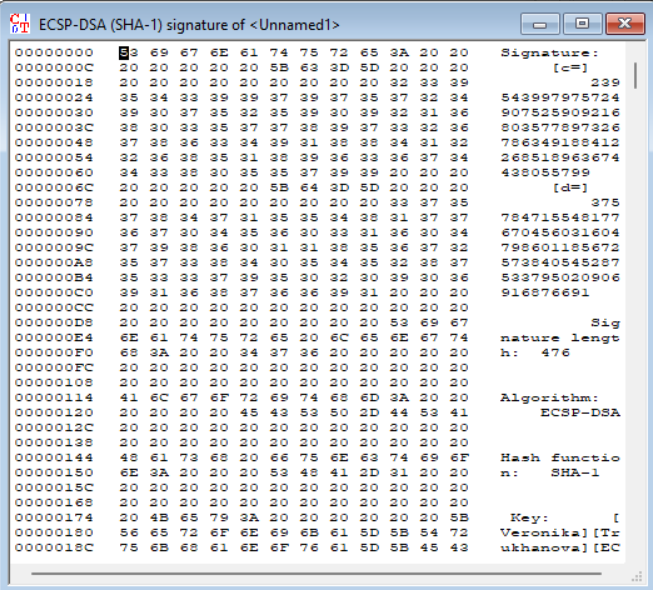
****

Рисунок 8 – Значение цифровой подписи для ключа EC-239

* + 1. **Скриншоты с результатами проверки цифрой подписи**

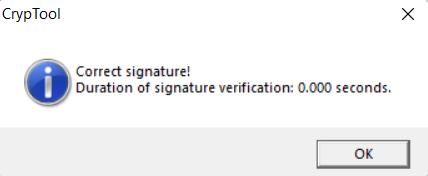


Рисунок 9 – Проверка для ключа RSA-2048 с исходным текстом

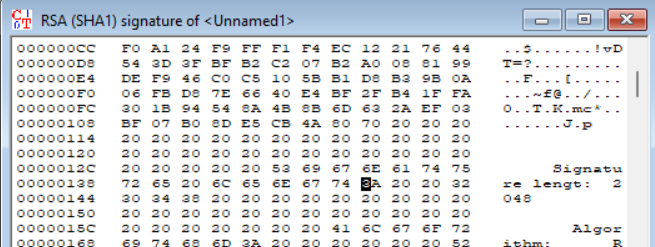


Рисунок 10 – Изменение цифровой подписи для ключа RSA-2048

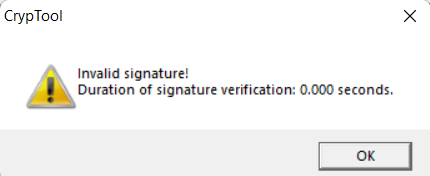


Рисунок 11 – Проверка для ключа RSA-2048 с изменённым текстом

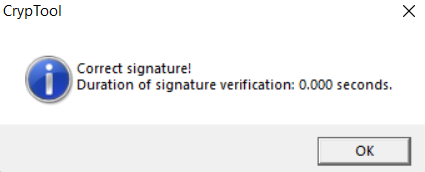


Рисунок 12 – Проверка для ключа DSA-2048 с исходным текстом

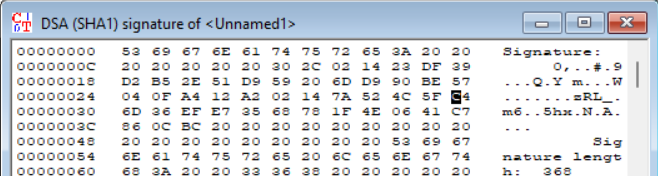


Рисунок 13 – Изменение цифровой подписи для ключа DSA-2048

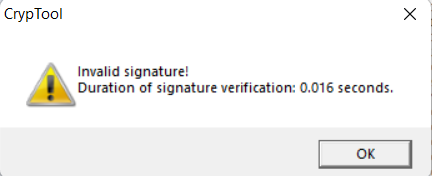


Рисунок 14 – Проверка для ключа DSA-2048 с изменённым текстом

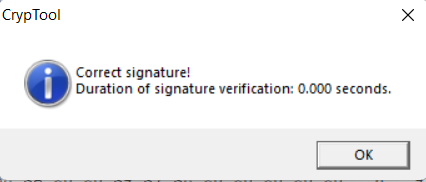


Рисунок 15 – Проверка для ключа EC-239 с исходным текстом

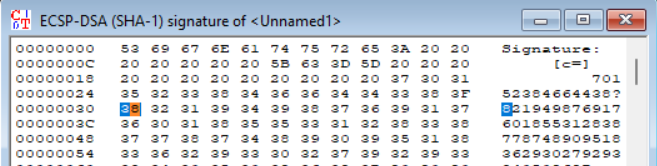


Рисунок 16 – Изменение цифровой подписи для ключа EC-239



Рисунок 17 – Проверка для ключа EC-239 с изменённым текстом

**Создание и проверка электронной подписи на основе эллиптических кривых**

* + 1. **Задание**
* Выполните процедуру создание подписи «*Digital Signatures/PKI-> Sign Document…*» алгоритмом ECSP-DSA в пошаговом режиме (*Display inter. results=ON*). Зафиксируйте скриншоты последовательности шагов.
* Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.
* 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*.
  + 1. **Описание алгоритма формирования и проверки подписи ECDSA**

Схема цифровой подписи ECDSA:

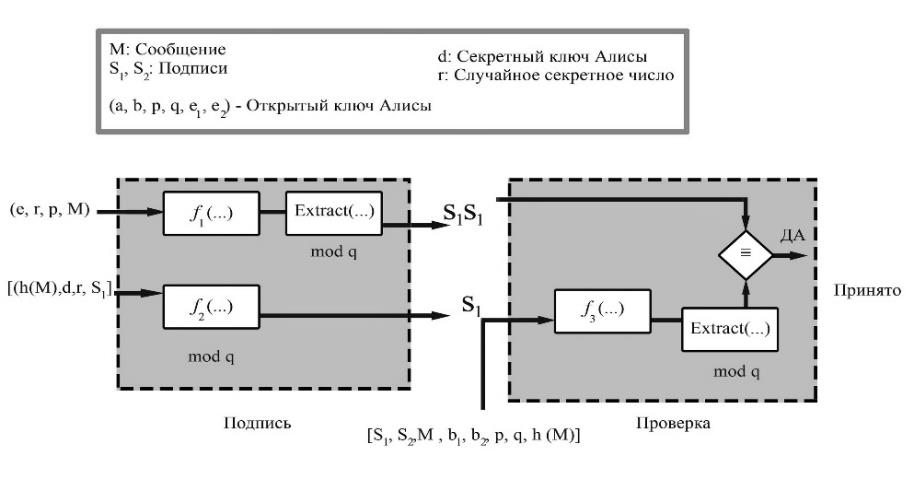


Рисунок 18 – Обобщенная схема цифровой подписи ECDSA

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

После того, как сгенерирована ключевая пара (закрытый ключ - d*,* и открытый ключ - , осуществляется подписание документа, затем на принимающей стороне осуществляется проверка.

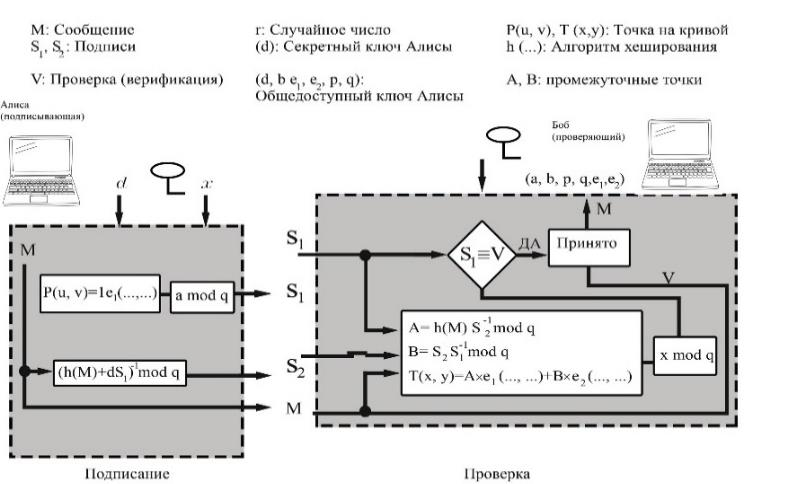


Рисунок 19 – Схема формирования и проверки подписи ECDSA

Алгоритм подписания ECDSA состоит из следующих операций:

* Выбирается секретное случайное число r: ;
* Выбирается третья точка на кривой: ;
* Вычисляется первая часть подписи по формуле: , где u- абсцисса.
* Вычисляется вторая часть подписи по формуле:

где ℎ(M)- дайджест сообщения, d- закрытый ключ.

Алгоритм проверки цифровой подписи ECDSA включает следующие операции:

* Вычисляем промежуточные результаты A и B:

,

.

* Восстанавливаем третью точку:

,

Верификатор сравнивается с первой частью цифровой подписи S1.

* + 1. **Результаты (скриншоты) пошагового выполнения ECDSA в CrypTool 1. Сравнение лекционной версии и реализации.**



Рисунок 20 – Установление подписи ECDSA

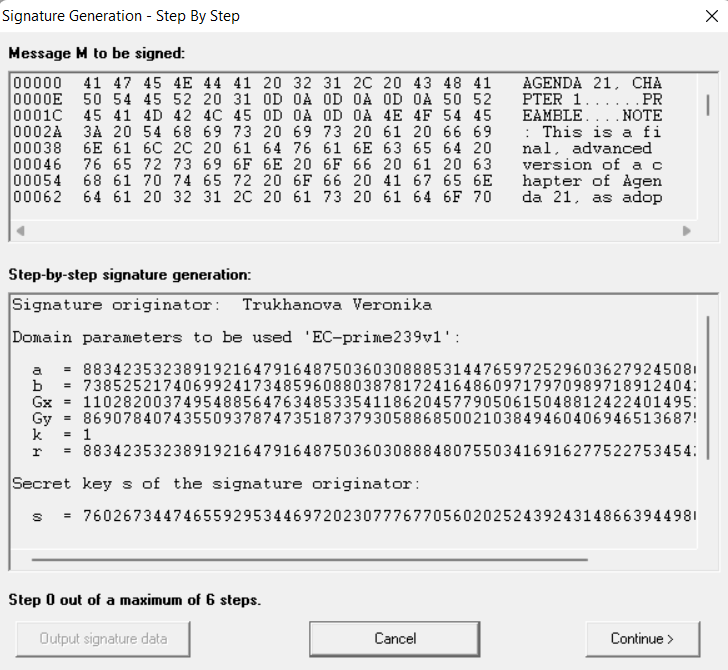


Рисунок 23 – Шаг 0

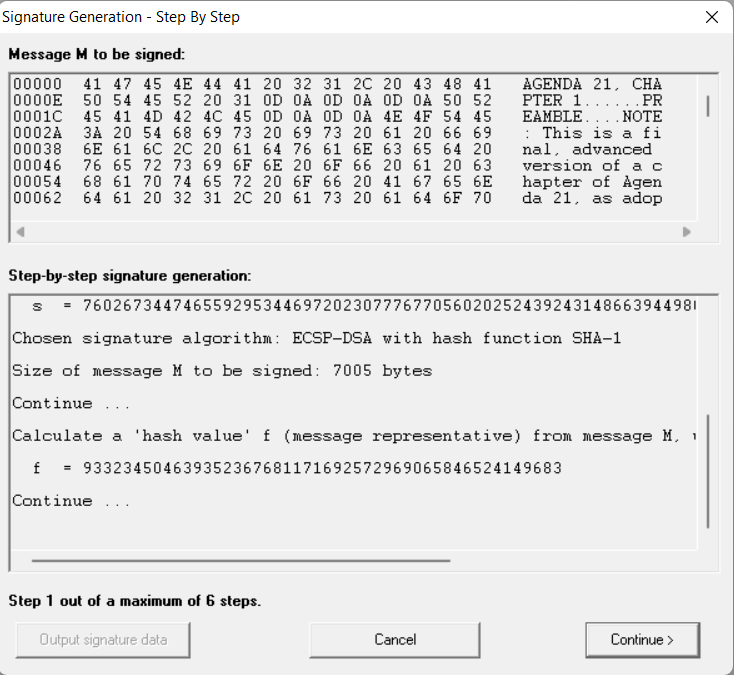


Рисунок 24 – Шаг 1

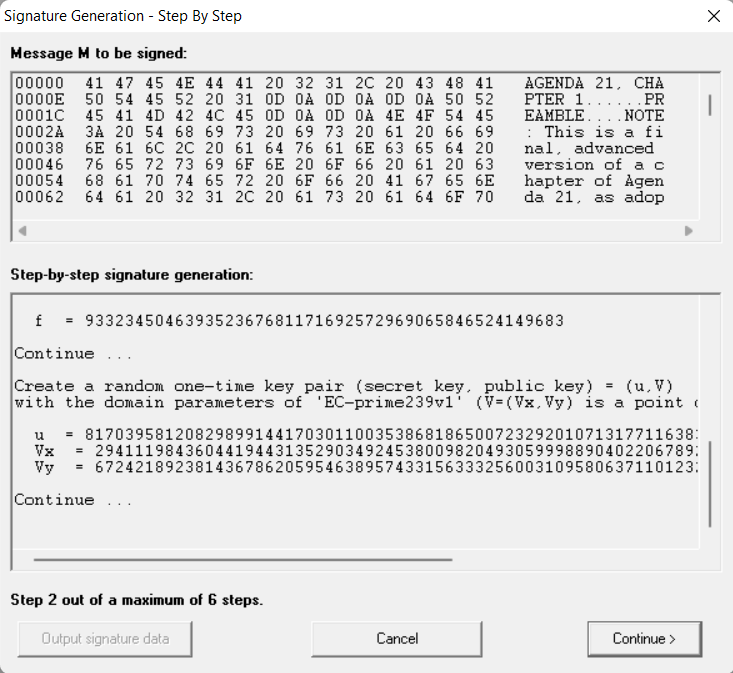


Рисунок 25 – Шаг 2

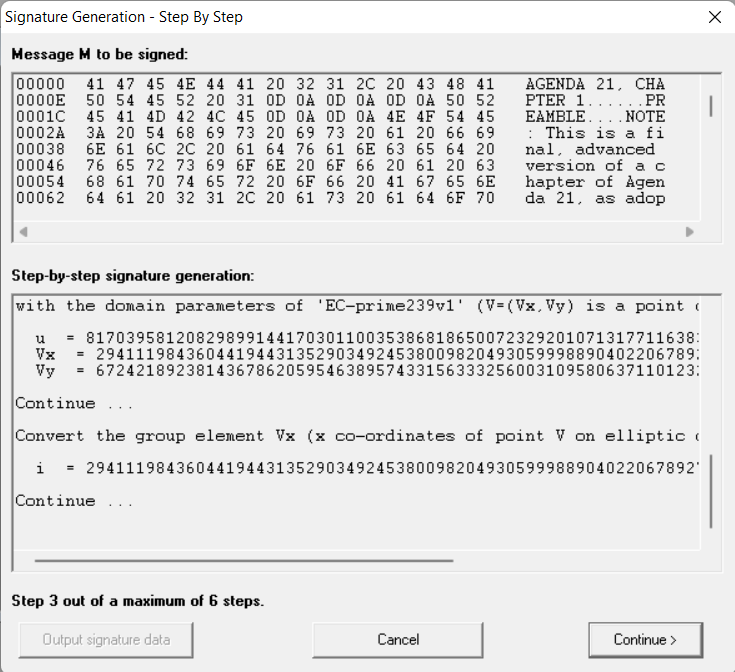


Рисунок 26 – Шаг 3



Рисунок 27 – Шаг 4

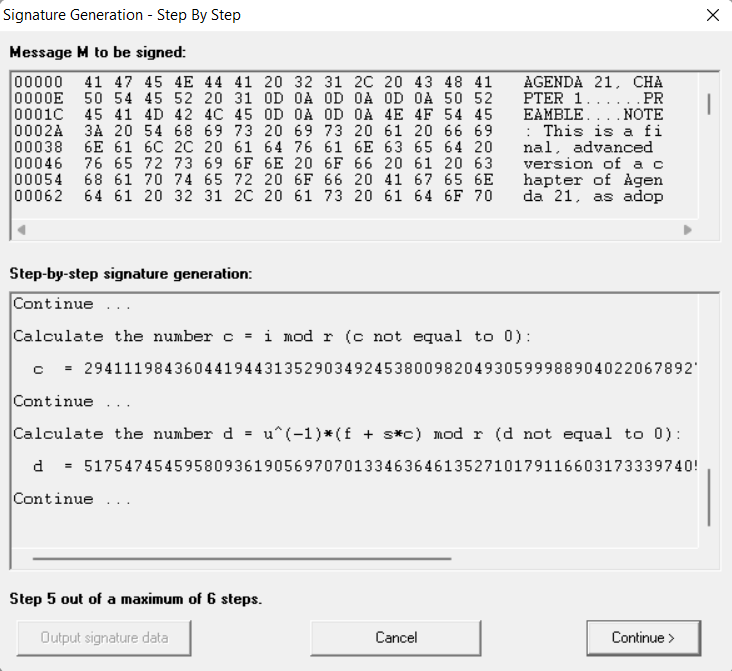


Рисунок 28 – Шаг 5



Рисунок 29 – Шаг 6

Алгоритм, представленный в лекции, и реализация в Cryptool являются одинаковыми. Различаются лишь названия переменных при вычислении.

* + 1. **Результаты проверки лекционного материала по ECDSA с использованием приложения *Indiv.Procedures->Number Theory…->Point Addition on EC*.**

Генерация:

* Выбрана эллиптическая кривая с параметрами a = 11, b = 20, p = 23;
* Выбрана точка P (в лекции e1) = (x1, y1) = (6, 7) на плоскости;
* Определено q перебором, пока не будет (q – порядок циклической подгруппы группы точек эллиптической кривой). Результат q =7;
* Выбрано число d = 5;
* Найдена точка
* Полученный открытый ключ: .

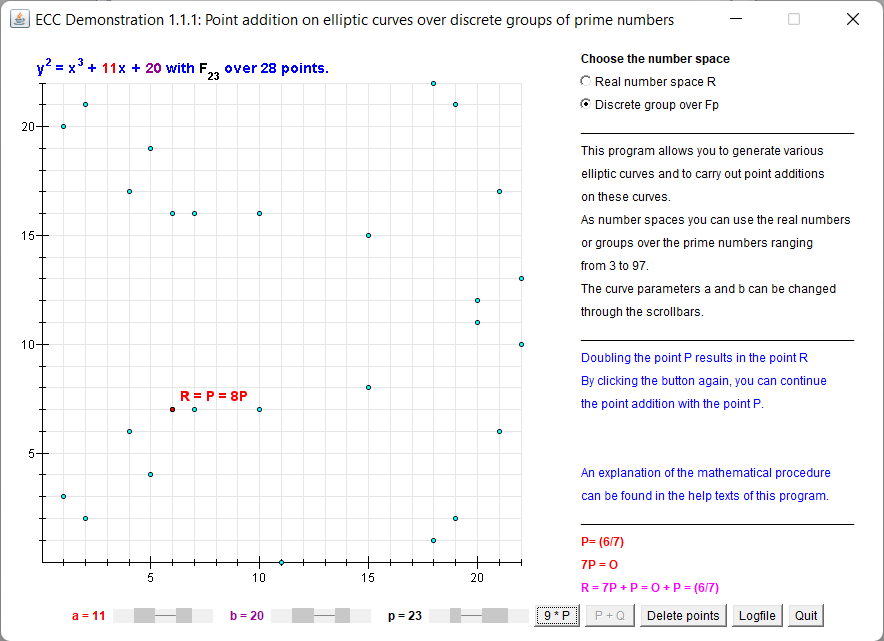


Рисунок 29 – Выбор P и вычисление q

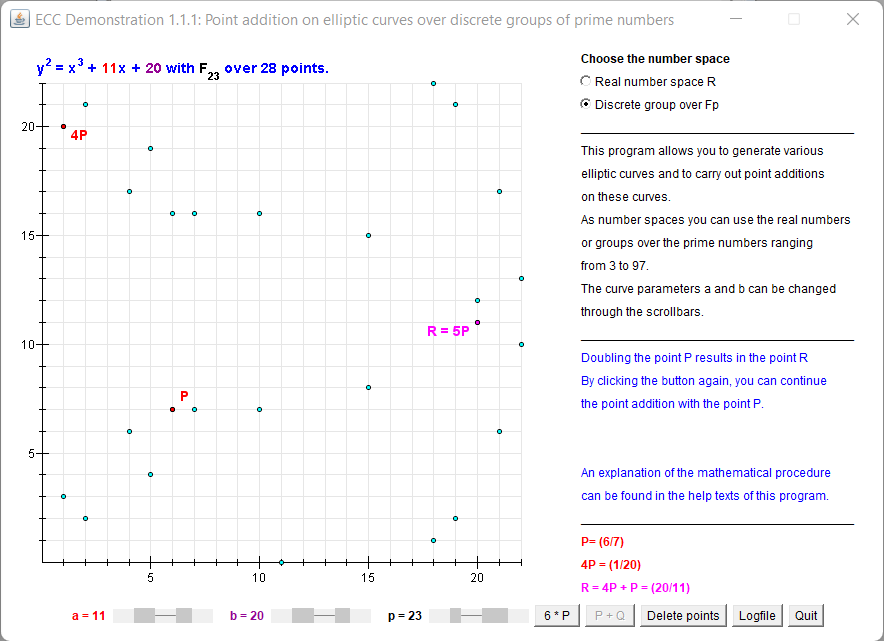


Рисунок 30 – Вычисление

Подписание: Выбрано число r: .

* + r = 5;
  + ;
  + Выбрано M: ;

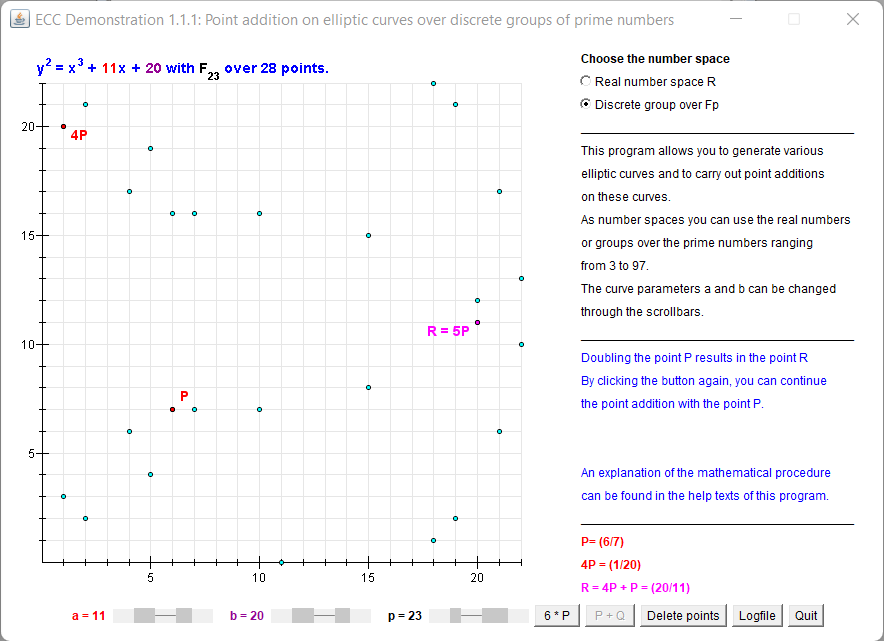


Рисунок 31 – Вычисление третьей точки на кривой при подписании

Проверка:

* + ;
  + ;
  + ;
  + Получили x = 20;
  + .

**Демонстрация процесса подписи в среде PKI**

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

Version: 2 (X.509v3-1996)

SubjectName: CN=Veronika Trukhanova [1684317445], DC=cryptool, DC=org

IssuerName: CN=CrypTool CA 2, DC=cryptool, DC=org

SerialNumber: C2:1B:E9:F0:16:05:08:F0

Validity - NotBefore: Wed May 17 12:57:36 2023 (230517095736Z)

NotAfter: Fri May 17 12:57:36 2024 (240517095736Z)

Public Key Fingerprint: 8472 660A A06A 2D5F 3201 E517 ED2C 7758

SubjectKey: Algorithm rsa (OID 2.5.8.1.1), Keysize = 2048

Public modulus (no. of bits = 2048):

0 FF2E3967 CA994B36 2D5AFF09 0B1DA222

10 AAF7D4E1 A54AFA50 4F0087E2 25025DBE

20 B0CA2DFC 406B775D 90729926 0BDB423F

30 2C100455 A2C841C9 D7807C84 FBF8F1C9

40 35D37ECB 195D2AF7 726B6CBB DCA401F9

50 1656C9A2 3DE4C1FB 9D9D5530 F5FEE12C

60 FF84C6E5 9E36D3E2 0282A7A9 60ECAC50

70 91E1AC00 95016E4E 690014AB EBF70E0C

80 38C020EA C4BB1B1B BA9609D7 2D00A642

90 3289D36A 220820BD A62D3E72 30317AE5

A0 8D7283C9 FA1B9B25 A3869F75 86C7E3A3

B0 8CCAD148 E6468818 3D10AAAC 3BFF4E95

C0 FBC87B27 3B25E542 692421C2 7CA60C6A

D0 F5A78D27 34AD2DED 31AB500E CC63DB6C

E0 E0C4709A FC137D2C 2E2B3997 EE5A928B

F0 50C1AC48 7A26CAF4 61D2541A AB743065

Public exponent (no. of bits = 17):

0 010001

Certificate extensions:

Private extensions:

OID 2.206.5.4.3.2:

PrintableString:

|[Trukhanova][Veronika][RSA-2048]|

|[1684317445] |

SHA1 digest of DER code of ToBeSigned:

0 26E4853E 3D0E9263 71EA99DF B2855136

10 B1762DB2

Signature: Algorithm sha1WithRSASignature (OID 1.3.14.3.2.29), NULL

0 A849DB30 19B52644 5D216335 5981E41C

10 F98A11DF 6A20031F 6C6D84AE EA981367

20 000D978D 78D31EEB FBCAE32F 628828BB

30 C76B0B8B 3CC712CE 13B74F1E AD883B8E

40 99639B3F 78F6136F E853B926 14769538

50 EBF26E28 B454E31D 1DABBC50 65FB9F8D

60 34A76097 FA24CACC A952B0CF FA593DEC

70 0F85833B B51429EE 1DD1EF1D 6D11D239

80 9F98D06B 459BE78D E855FADD DA8C86D6

90 BFD9A3E2 738A0AFC 851EC661 9DB10488

A0 E462E137 7EDE3330 095DF2E4 97B71757

B0 BC582A02 57B18603 E22BF429 DD09702F

C0 47535965 2410575D F7674E42 5C4FA595

D0 BD7524B3 0EFC3342 AAD91A69 B58E71F9

E0 BD62B0E9 D1659955 095DC529 EC9CECBD

F0 9382CB82 2B3D6D9B 4B5039E9 11267B57

Certificate Fingerprint (MD5): E0:EC:C3:9F:BE:1C:DF:9F:58:9C:C5:77:F4:D0:41:B7

Certificate Fingerprint (SHA-1): FD85 06B7 1725 98BD 8B32 F192 79

Между сертификатами из лекции и сгенерированным в программе Cryptool 1 есть несколько отличий в структуре:

* В сгенерированном сертификате нет полей: кому выдан, кем выдан, назначение, срок действия закрытого ключа идентификатор ключ центра сертификатов.
* В сертификате из лекции не показан отпечаток сертификата.
  + 1. **Схема процедуры подписания из CrypTool**

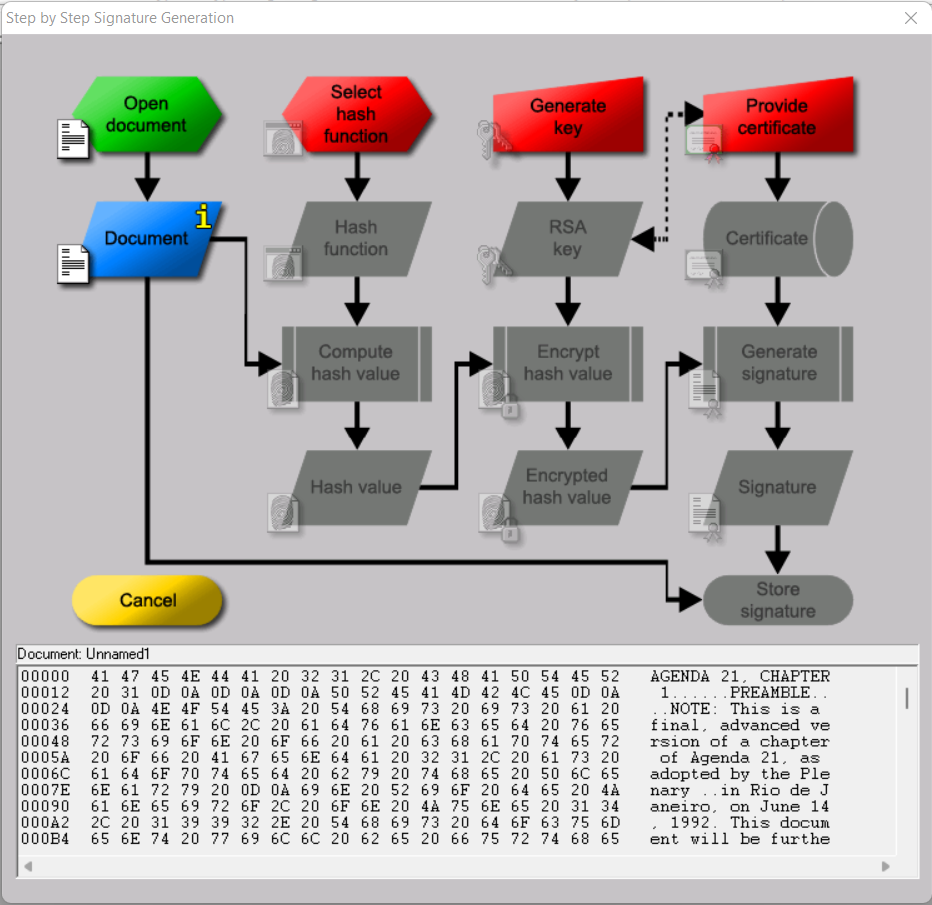


Рисунок 33 – Схема процедуры подписания из CrypTool

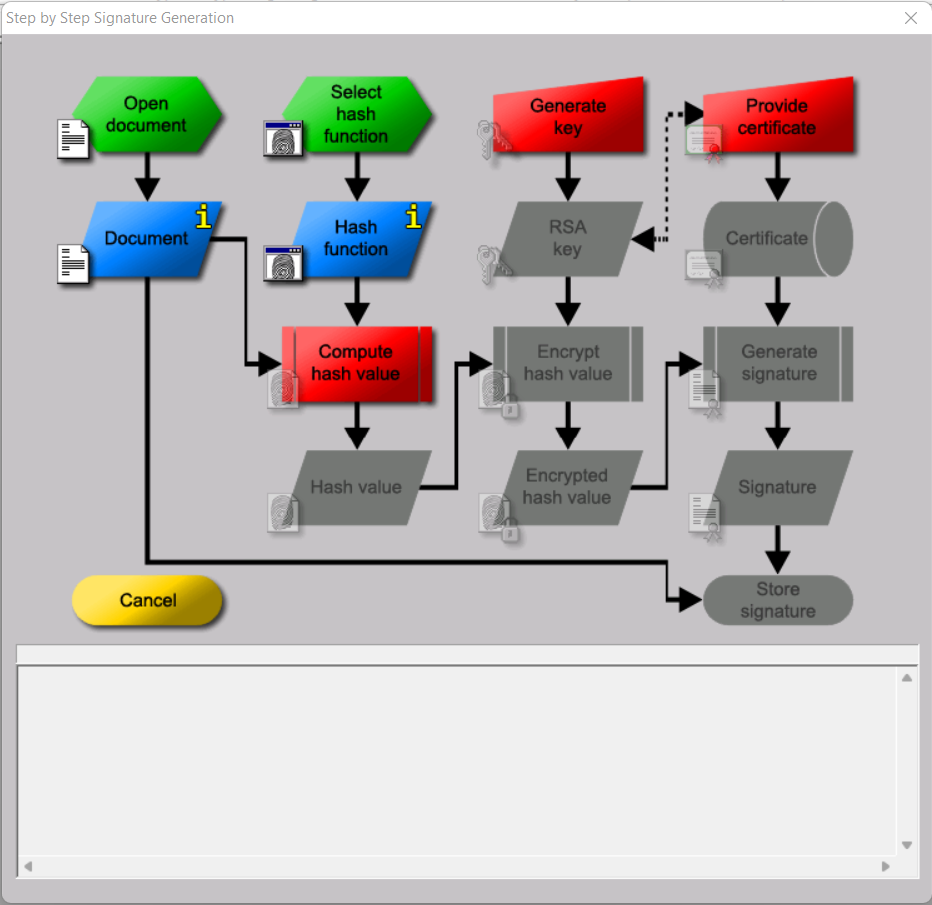


Рисунок 34 – Процесс выбора хэш-функции

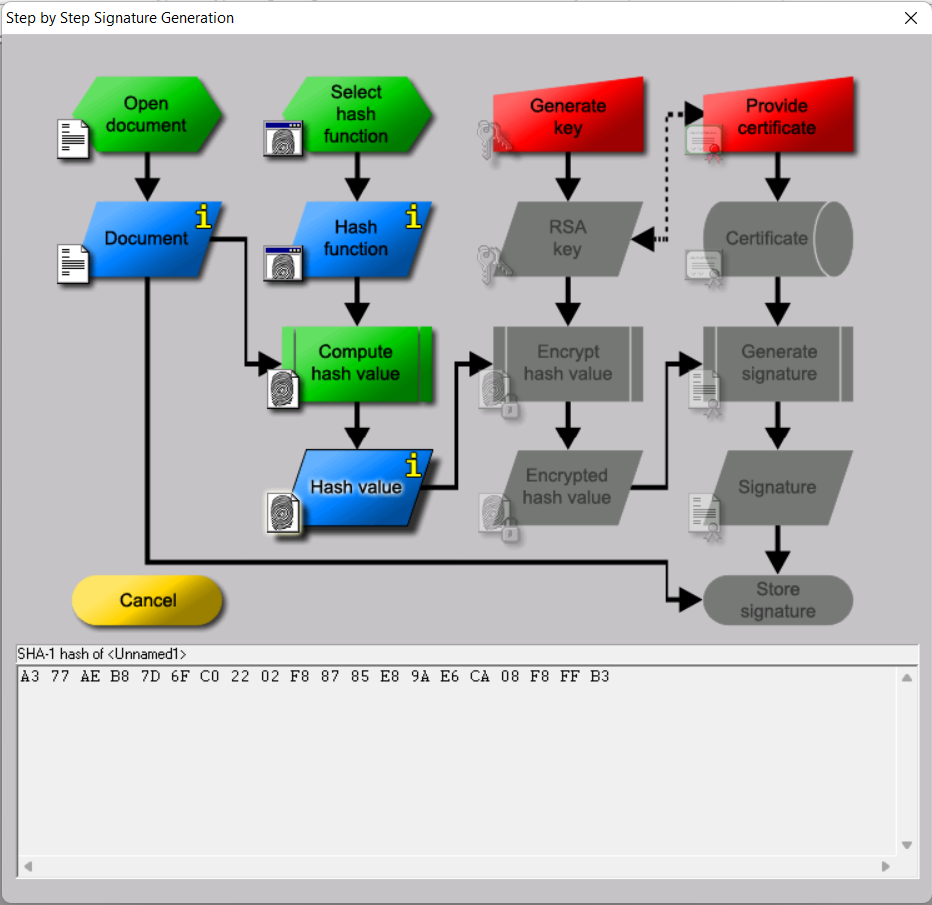


Рисунок 35 – Вычисление значения хэш-функции документа

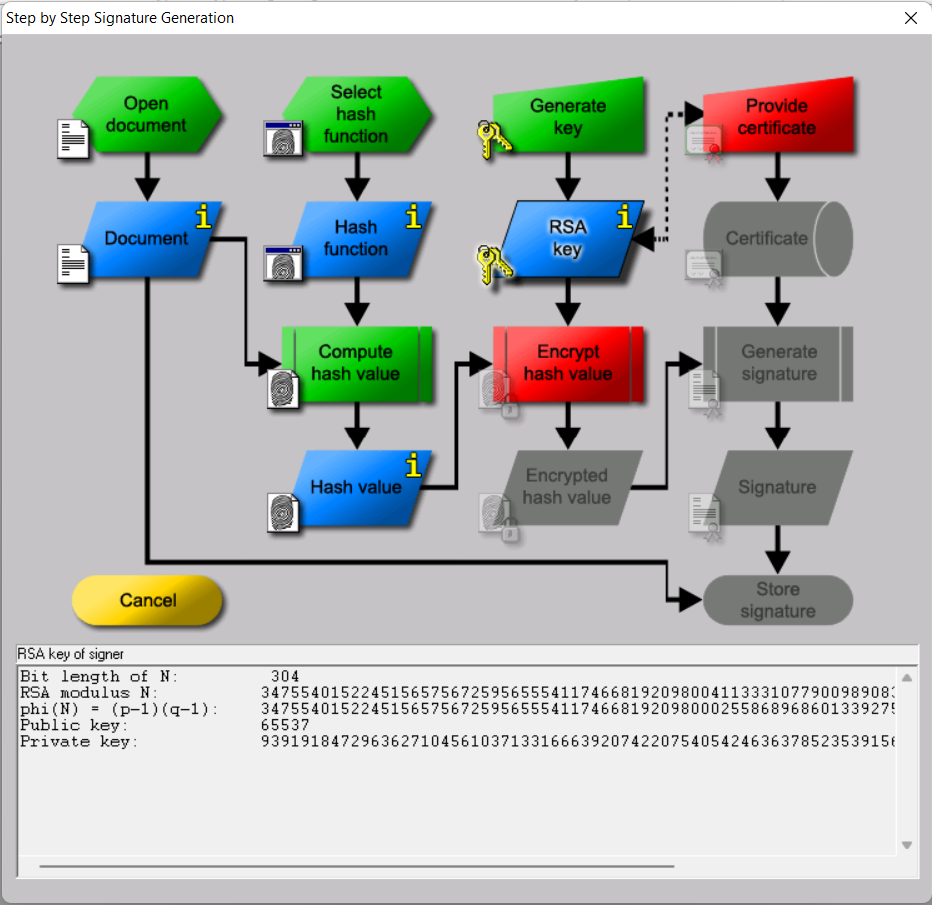


Рисунок 36 – Генерация ключевых пар для RSA

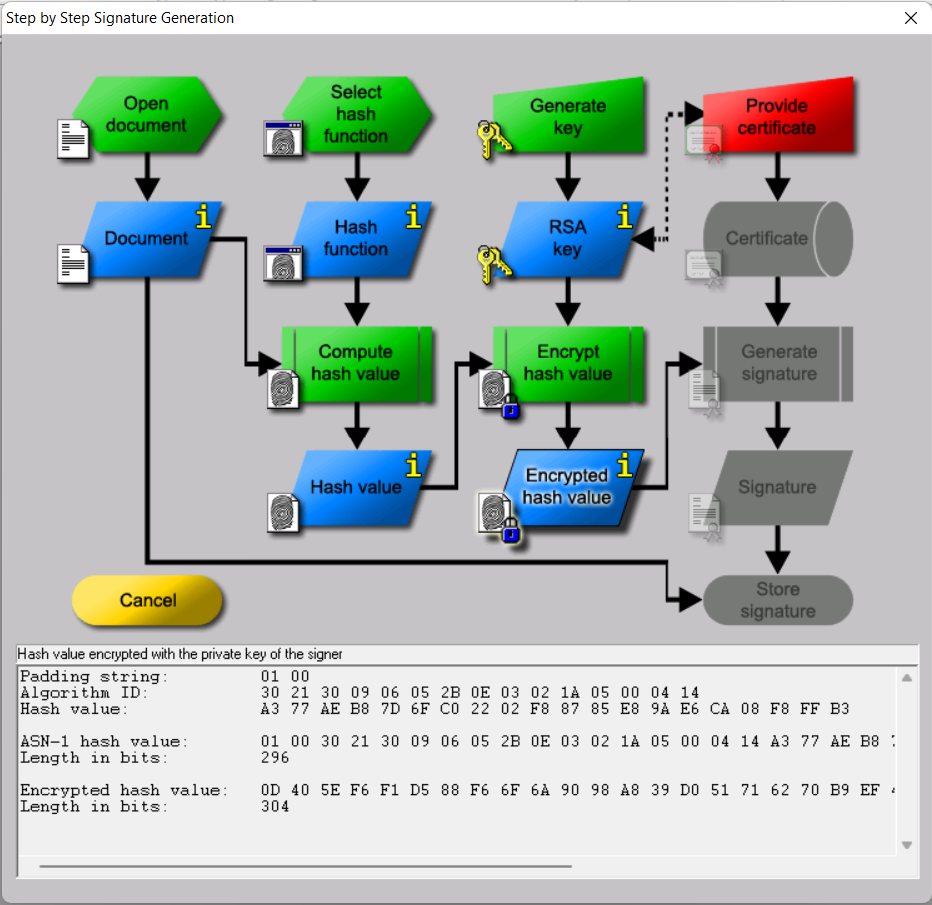


Рисунок 37 – Шифровка значения хэш-функции

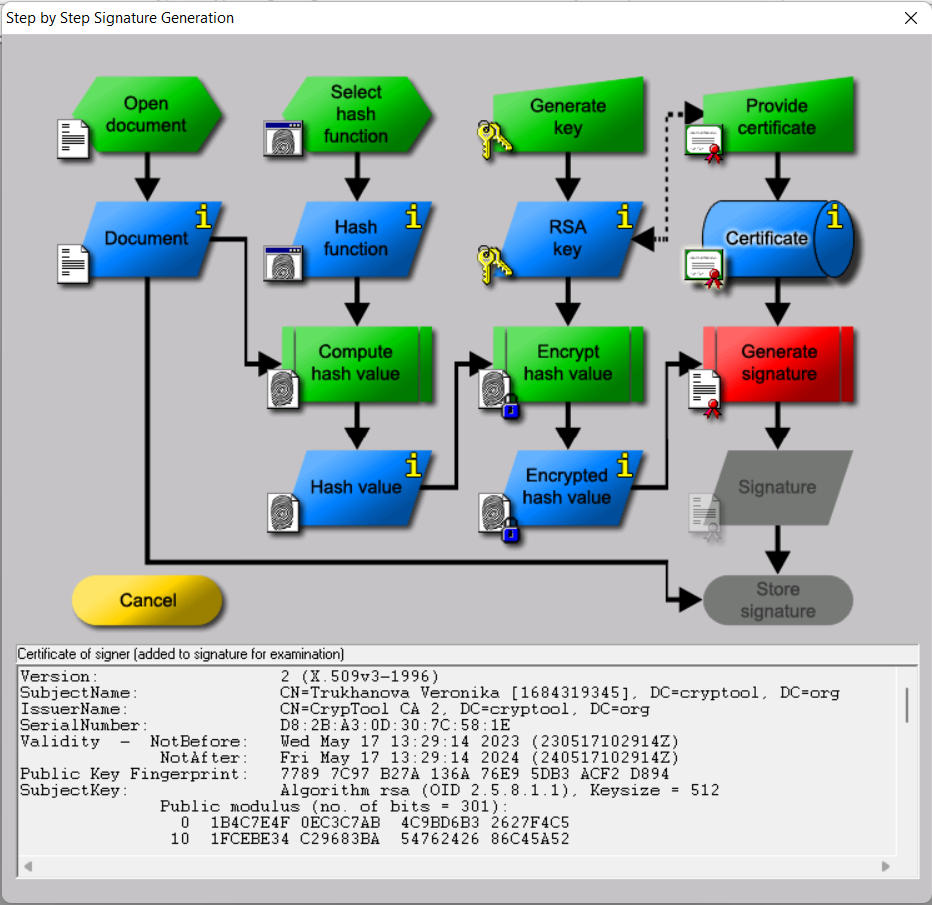


Рисунок 38 – Обеспечение сертификата

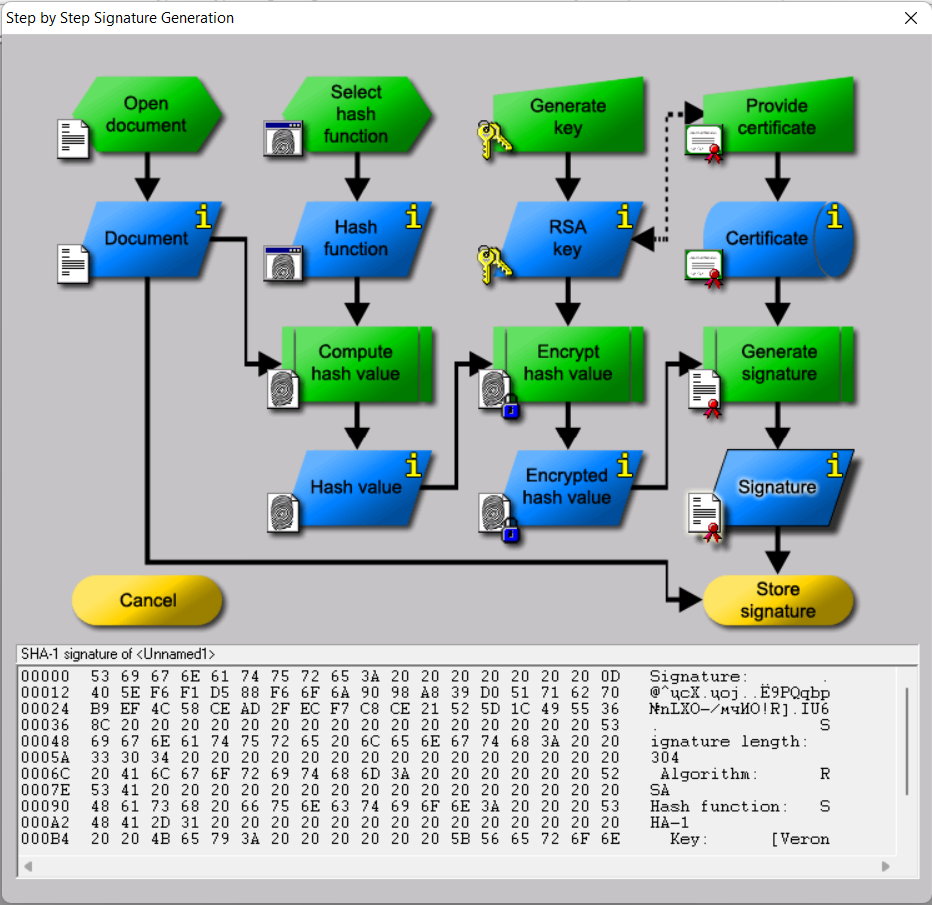


Рисунок 39 – Создание подписи

**Подписание своего отчета**

* + 1. **Задание**
* Сконвертируйте отчет в формат pdf.
* Экспортируйте ранее созданный сертификат ключевой пары RSA Digital Signatures/PKI->PKI/Generate…->Export PSE(#PKCS12).
* Откройте pdf-версию отчета и попытайтесь подписать с использованием этого сертификата.
* Создайте собственный самоподписанный сертификат в среде Adobe Reader и используйте его для подписи отчета.
* Сохраните скриншоты свойств подписи и сертификата.
* Внесите изменения (маркеры, комментарии) в отчет и проверьте подпись.
  + 1. **Скриншот титульного листа с цифровой подписью.**

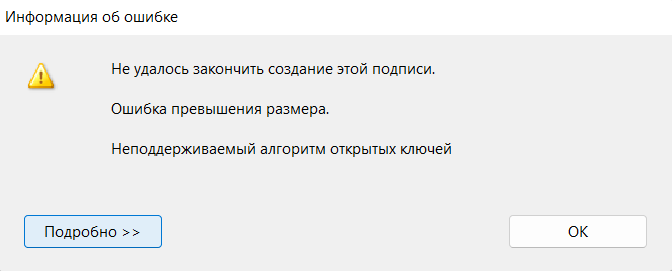
****

Рисунок 40 – Попытка подписания сертификатом из CrypTool

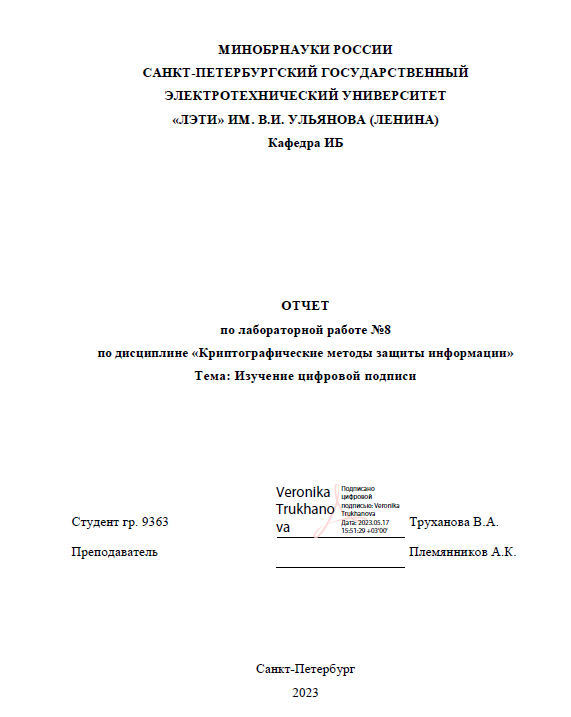


Рисунок 41 – Титульный лист с электронной подписью с использованием подписанного сертификата в среде Adobe Acrobat Reader

* + 1. **Скриншоты свойств подписи и сертификата.**

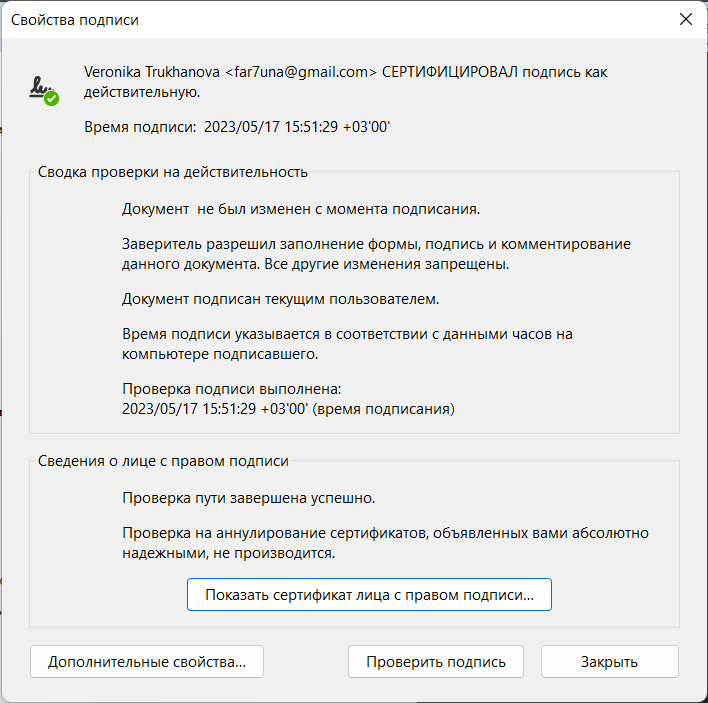


Рисунок 42 – Свойства подписи в Adobe Acrobat Reader

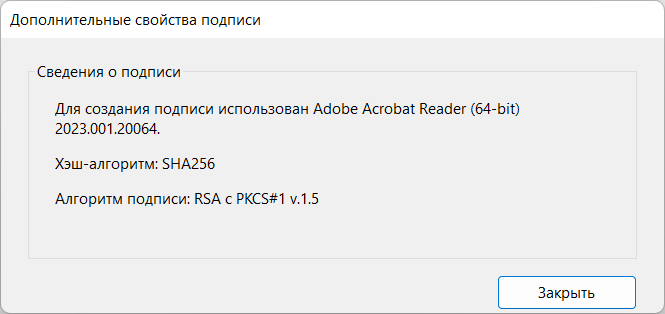


Рисунок 43 – Дополнительные свойства подписи в Adobe Acrobat Reader

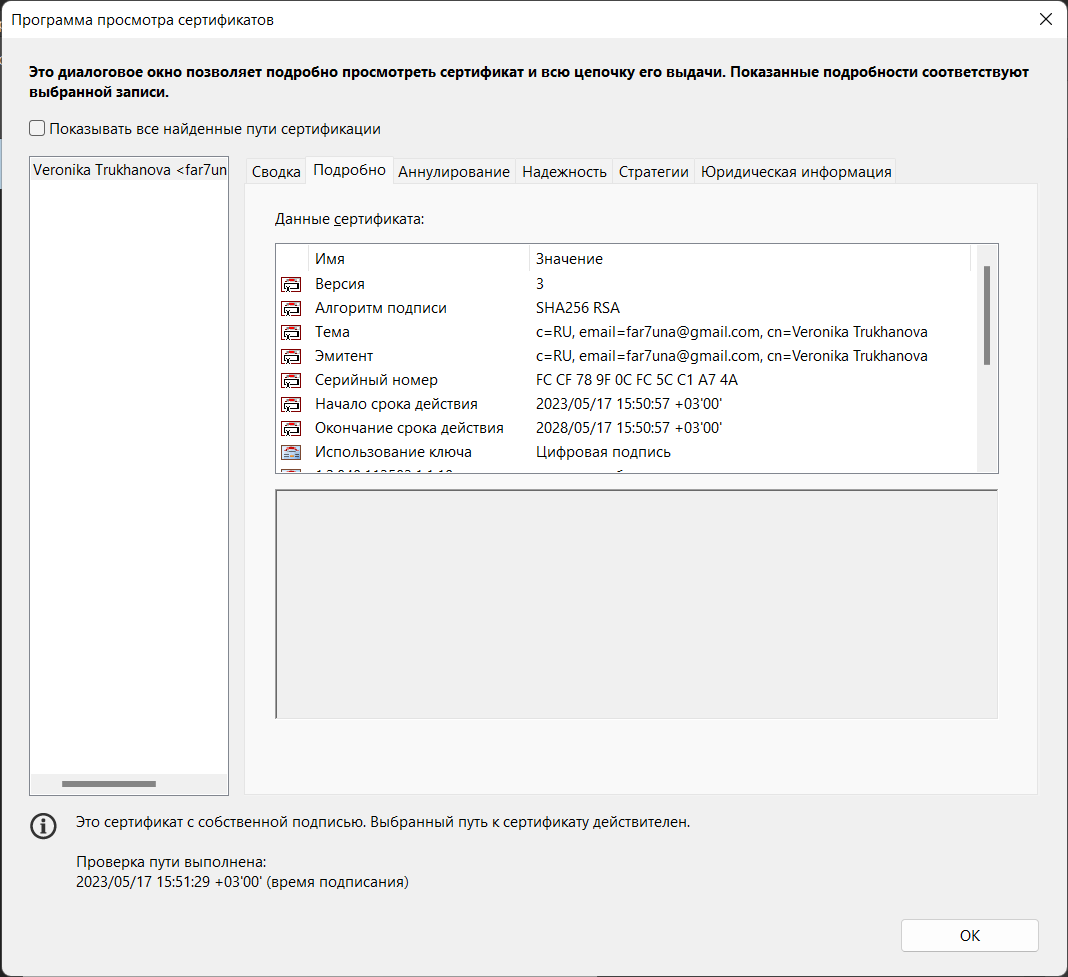


Рисунок 44 – Сертификат ключа проверки электронной подписи в Adobe Acrobat Reader

* + 1. **Скриншот результата проверки после внесения изменений в отчет**

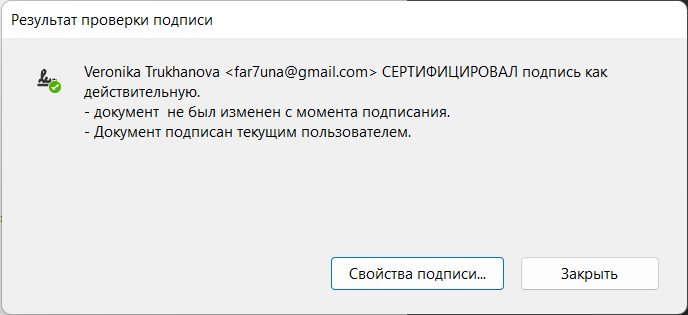
****

Рисунок 45 – Результат проверки подписи

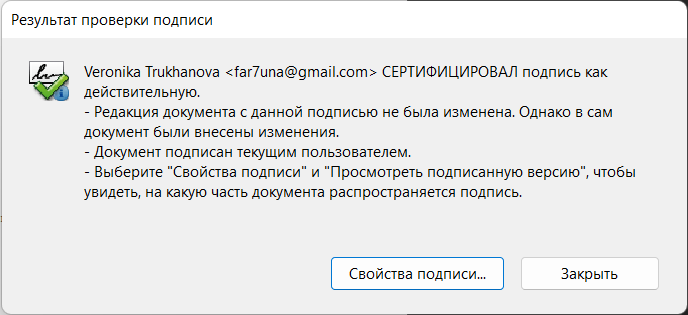
****

Рисунок 46 – Результат проверки подписи после изменения в документе

**Заключение**

В ходе выполнения данной лабораторной работы были исследованы алгоритмы создания и проверки цифровой подписи, алгоритмы генерации ключевых пар для алгоритмов цифрой подписи RSA, DSA, ECDSA.

1. С помощью программы CrypTool 1 были созданы ключевые пары RSA-2048, DSA-2048, EC-239. Среди них алгоритм ECDSA имел самую быструю генерацию ключевых пар – 0.012 сек.. Это связано с вычислением на эллиптических кривых, требующее меньше ресурсов для того же уровня криптостойкости.
2. Были созданы ЦП с использованием ранее сгенерированных ключевых пар RSA-2048, DSA-2048, EC-239, после чего был сделан вывод что вычисление подписи происходило быстрее всего при помощи алгоритма DSA – 0.004 сек.
3. Была рассмотрена схема цифровой подписи ECDSA. Алгоритм генерирует открытый ключ и закрытый ключ d, сгенерированная ЦП состоит из двух частей – и . Проверка заключается в сравнении с прошедшей через экстрактор .
4. Был изучен процесс подписи в среде PKI и сгенерирован сертификат, в котором хранится информация об открытом ключе. Закрытый ключ известен только владельцу сертификата.
5. С помощью Adobe Acrobat Reader был подписан отчет, а также проведена проверка подписи после модификации документа, которая говорила о том что подпись является действительной, но подписанный ею документ был изменен.