**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8383 |  | Гречко В.Д. |
| Преподаватель |  | Племянников А.К. |

Санкт-Петербург

2021

**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.

**Генераторов ключевых пар.**

Описание алгоритмов генерации.

**RSA:** Выбираются два больших простых числа и . Вычисляется . Выбирается произвольное число , взаимно простое с . Вычисляется закрытый ключ (расширенный алгоритм Евклида):

Пара чисел объявляются открытым ключом, выбирается закрытым ключом, и нужно уничтожить.

**DSA:** Выбирается простое число , длиной между 512 и 1024 битами. Число битов в должно быть кратно 64. Выбирается другое простое число , которое имеет тот же самыйразмер, что и дайджест - 160 битов, такое , что

Выбирается , такое, что путем вычисления , где (теорема Ферма). Выбирается целое и вычисляется . Объявляется открытый ключ . Назначается закрытый ключ .

**ECDSA:** Выбирается эллиптическая кривая , 𝑝 – простое. Выбирается точка на кривой . Для дальнейших вычислений выбирается другое простое число - порядок циклической подгруппы группы точек эллиптической кривой : . Выбирается целое число , и назначается закрытым ключом. Вычисляется другая точка на кривой . Объявляется открытый ключ .

Задание.

1. Перейти к утилите *«Digital Signatures/PKI->PKI/Generate…».*
2. Сгенерировать ключевые пары по алгоритмам RSA-2048, DSA-2048, EC-239. Зафиксируйте время генерации в таблице.

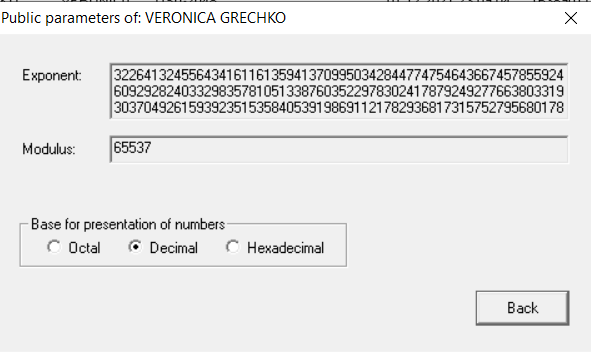
Выполним генерацию ключевых пар по указанным алгоритмам и зафиксируем время генерации в таблице 1.

*Таблица 1 – Время генерации ключевых пар*

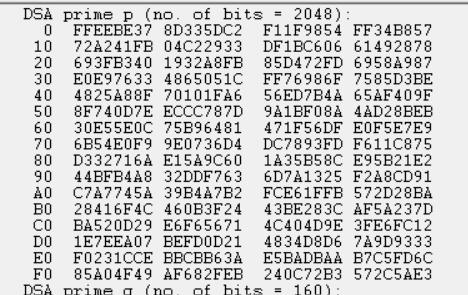
|  |  |
| --- | --- |
| Алгоритм | Время генерации |
| RSA-2048 | 1.334 секунд |
| DSA-2048 | 3.515 секунд |
| EC-239 | 0.012 секунд |

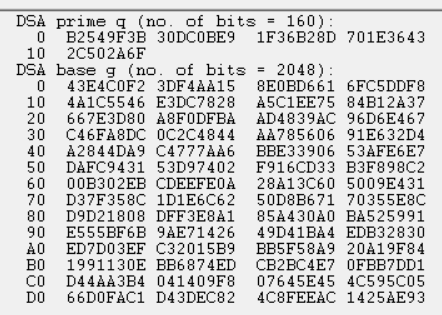
1. С помощью утилиты *«Digital Signatures/PKI-> PKI/Display…»* вывести сгенерированный открытый ключ и сохранить соответствующий скриншот.

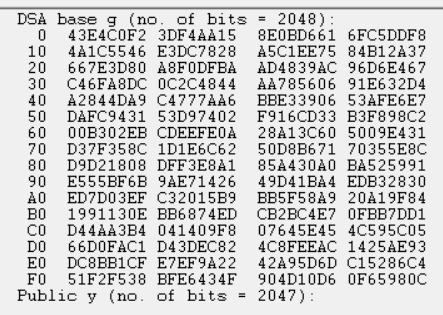
Полученные ключи представлены на рис. 1 – 3 соответственно.

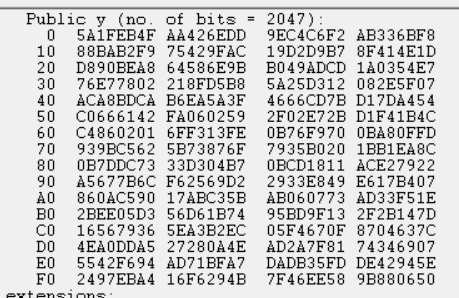


*Рисунок 1 – Открытый ключ для RSA-2048*

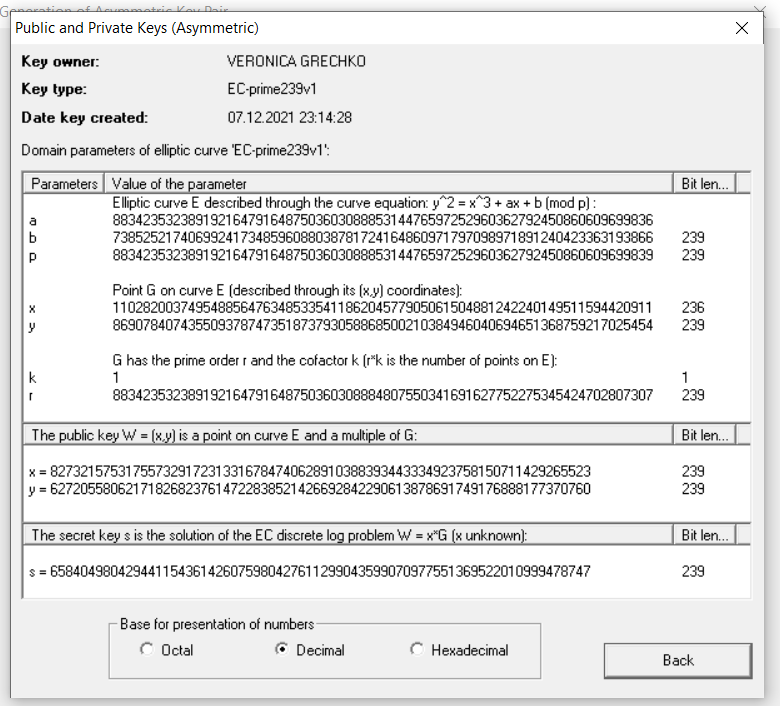








*Рисунок 2 – Открытый ключ для DSA-2048*

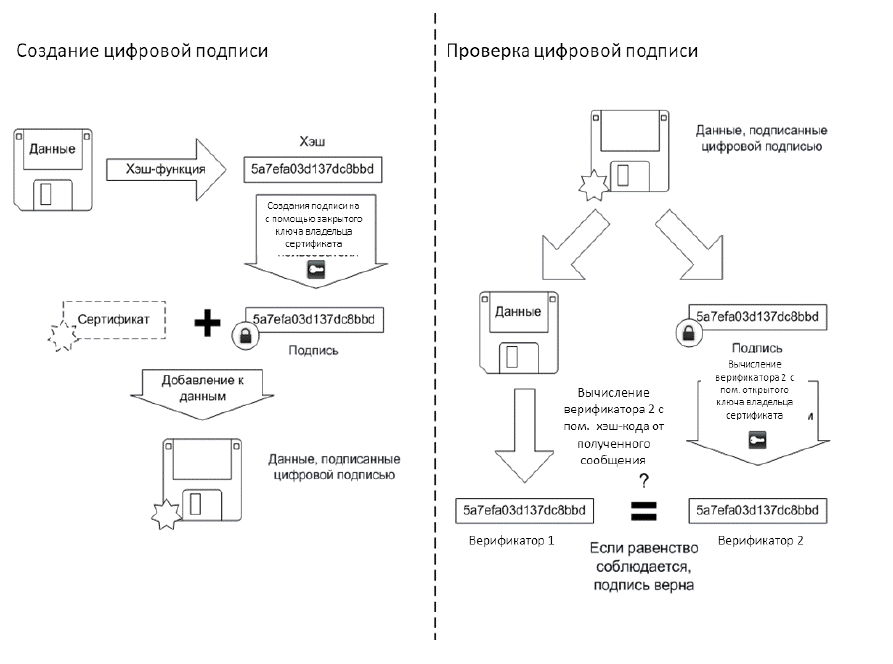


*Рисунок 3 – Открытый ключ для EC-239*

**Процессы создания и проверки цифровой подписи.**

Обобщенная схема создания и проверки цифровой подписи.

Обобщенная схема представлена на рис. 4.

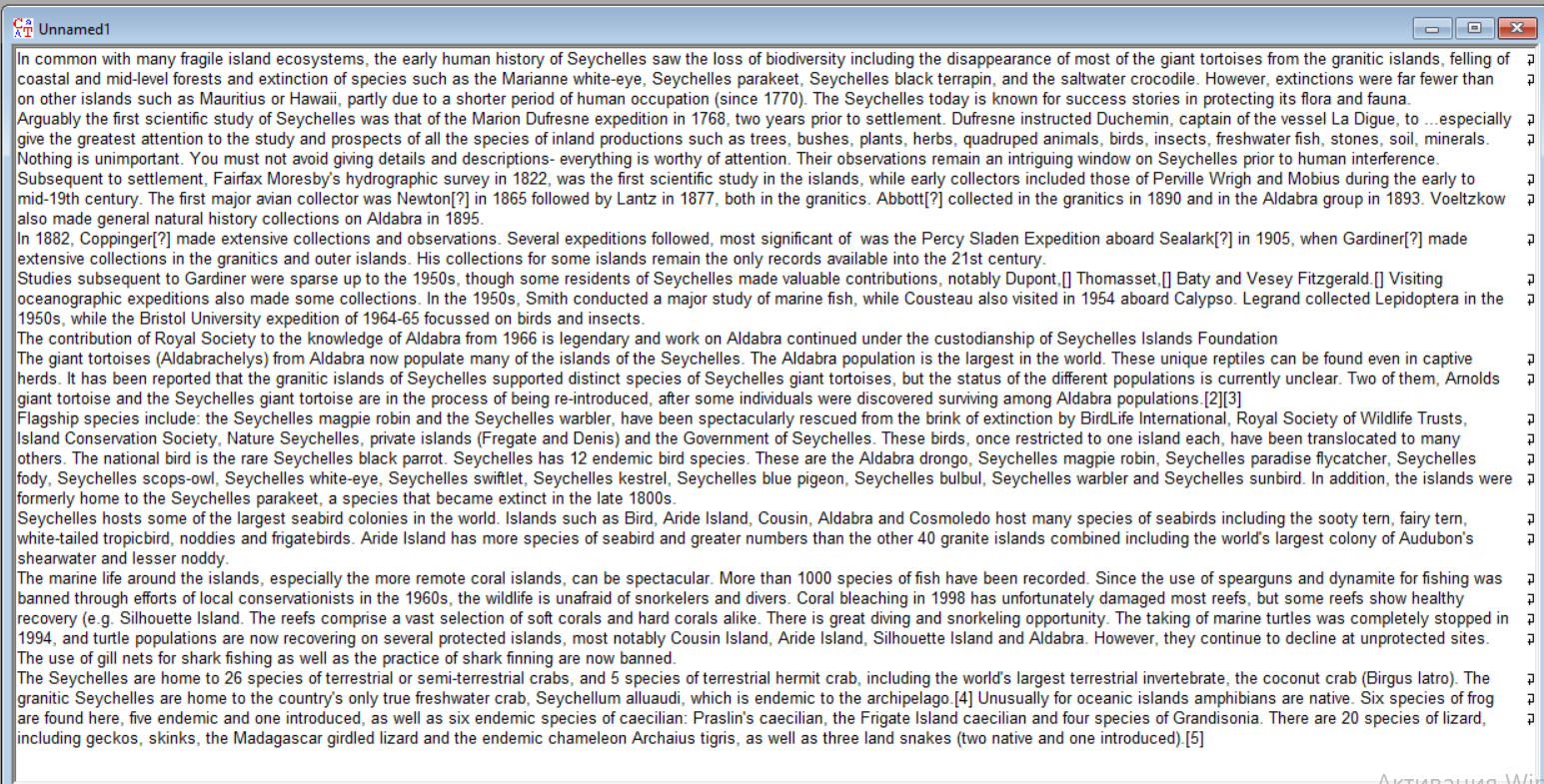


*Рисунок 4 – Обобщенная схема*

Задание.

1. Открыть текст не менее 5000 знаков. Перейти к приложению *Digital Signatures/PKI-> Sign Document…*

Исходный текст представлен на рис. 5.



*Рисунок 5 – Исходный текст*

1. Задайте хэш-функцию, и другие параметры цифровой подписи.
2. Создайте подпись ключами, сгенерированными в предыдущем задании. Зафиксируйте время создания цифровой подписи для каждого ключа.

Выполним указанные действия, и сведем полученное время в таблицу 2.

*Таблица 2 – Время создания цифровой подписи в зависимости от ключей хэш-функций*

|  |  |  |
| --- | --- | --- |
| Алгоритм | Хэш-функция | Время создания подписи |
| RSA-2048 | MD5 | 0.010 секунд |
| SHA-1 | 0.008 секунд |
| DSA-2048 | MD5 | - |
| SHA-1 | 0.002 секунд |
| EC-239 | MD5 | - |
| SHA-1 | 0 секунд |

1. Сохраните скриншот цифровой подписи с помощью приложения *Digital Signatures/PKI-> Extract Signature.*

Скриншоты полученных цифровых подписей представлены на рис. 6 – 9 соответственно.

|  |  |
| --- | --- |
| *Рисунок 6 – Цифровая подпись RSA с MD5* | *Рисунок 7 – Цифровая подпись RSA с SHA-1* |
| *Рисунок 8 – Цифровая подпись DSA с SHA-1* | |
| *Рисунок 9 – Цифровая подпись ECDSA с SHA-1* | |

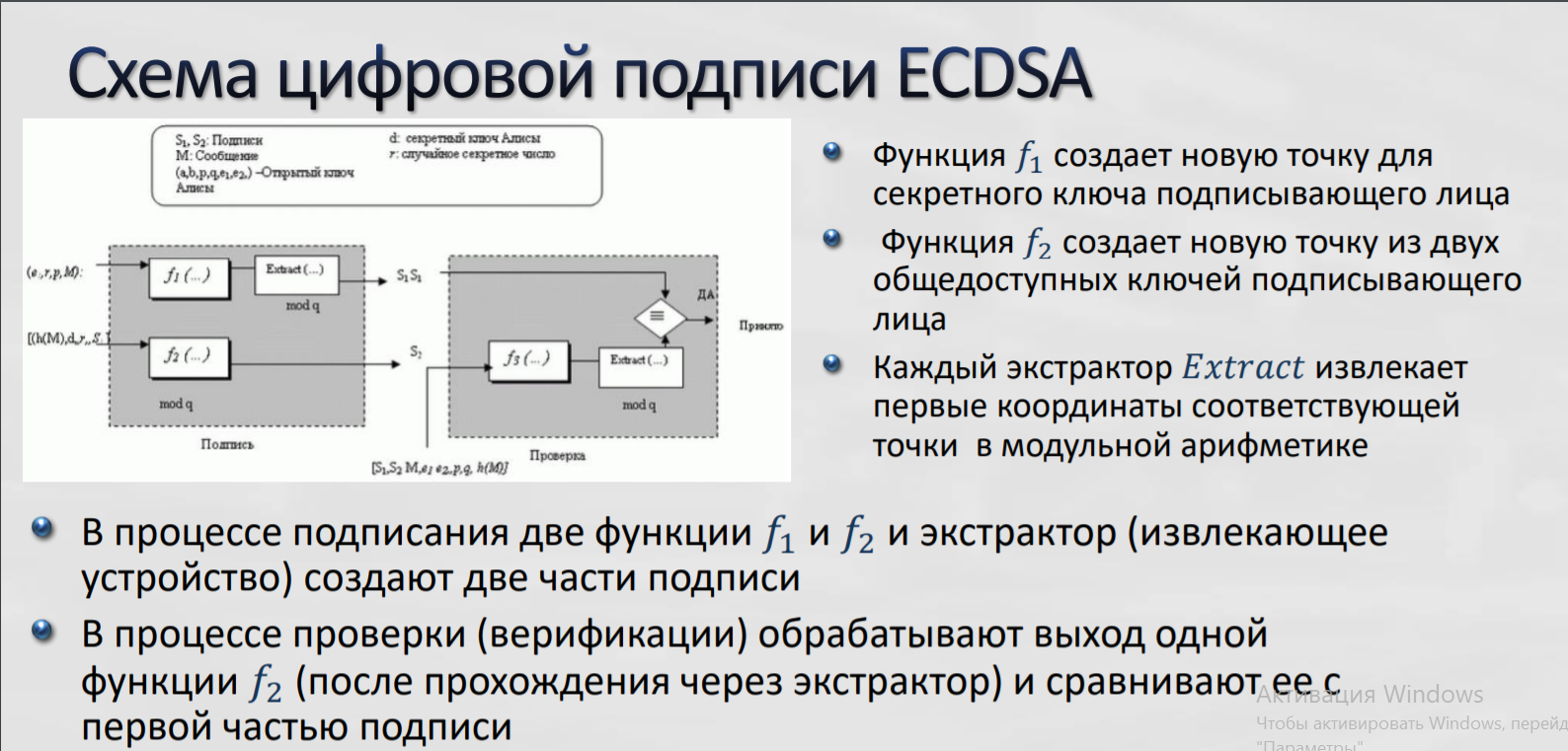
1. Выполните процедуру проверки подписи *Digital Signatures/PKI-> Verify Signature* для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

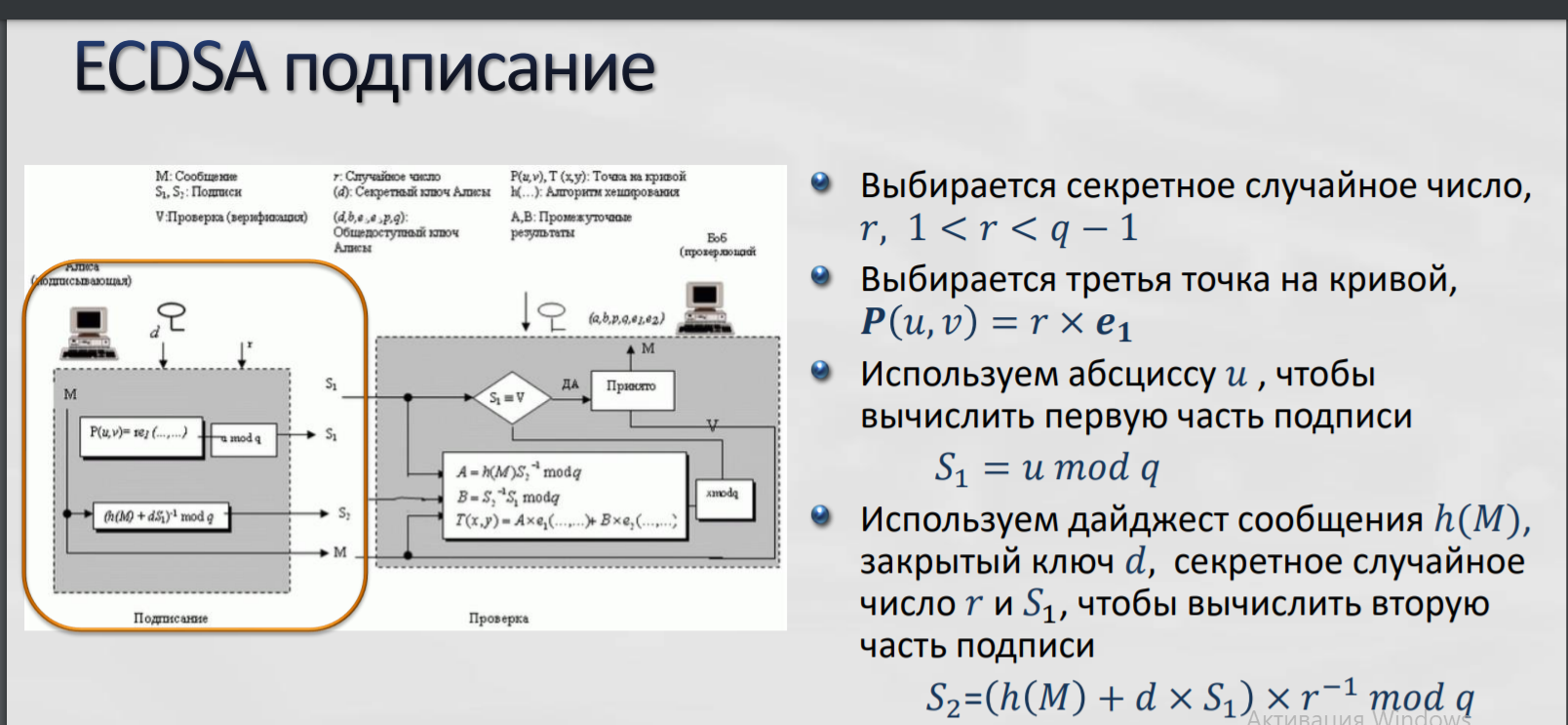
Выполним проверку и сведем полученные значения в таблицу 3.

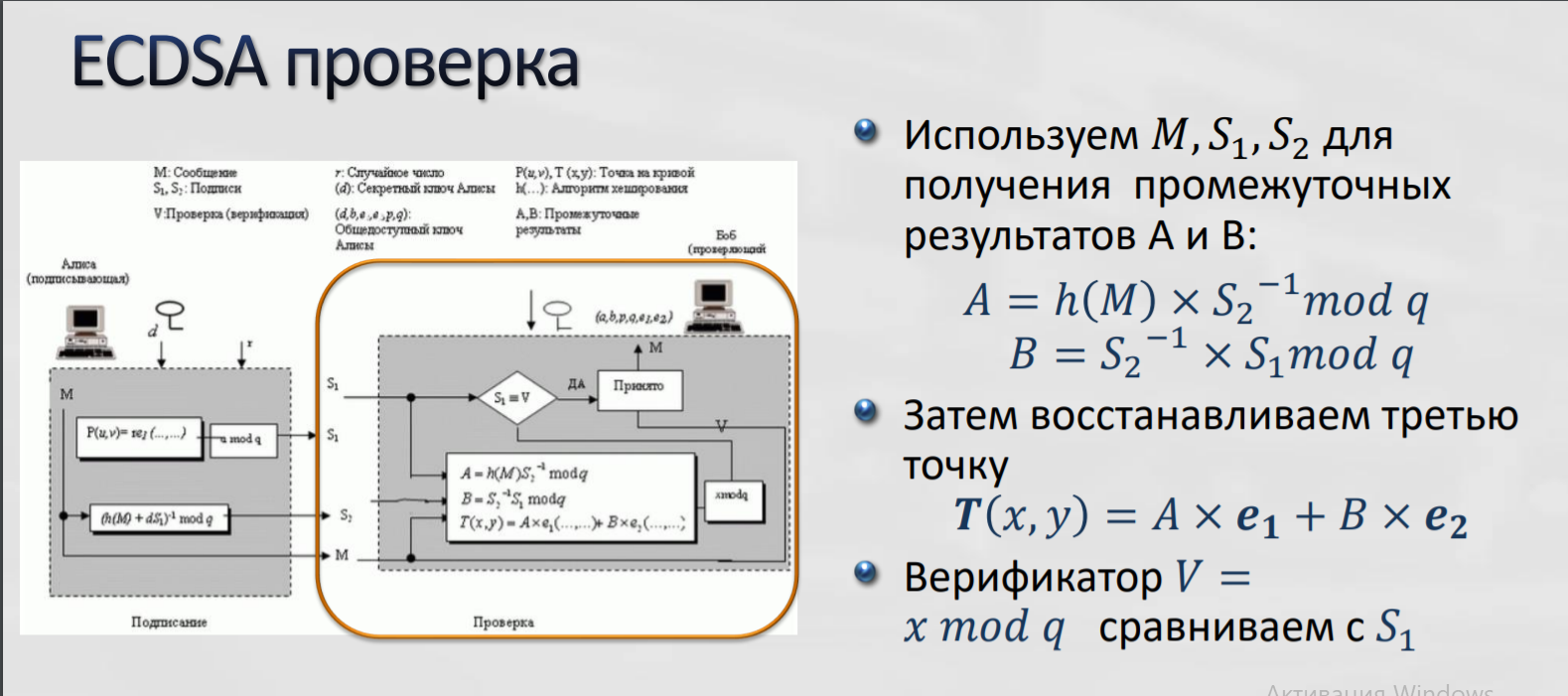
|  |  |  |
| --- | --- | --- |
| Алгоритм | Сохранение целостности | Нарушение целостности |
| RSA-2048 |  |  |
|  |  |
| DSA-2048 |  |  |
| EC-239 |  |  |

**Схемы цифровой подписи на эллиптических кривых.**

Описание алгоритма формирования и проверки подписи ECDSA.



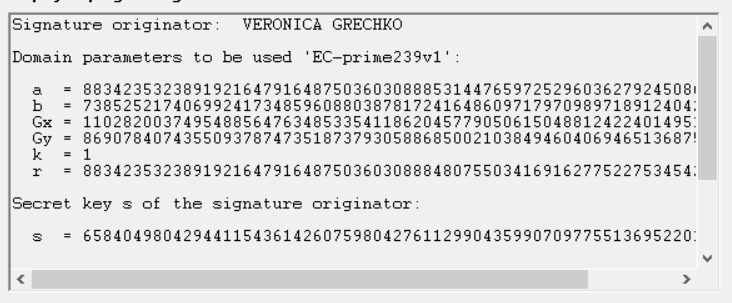


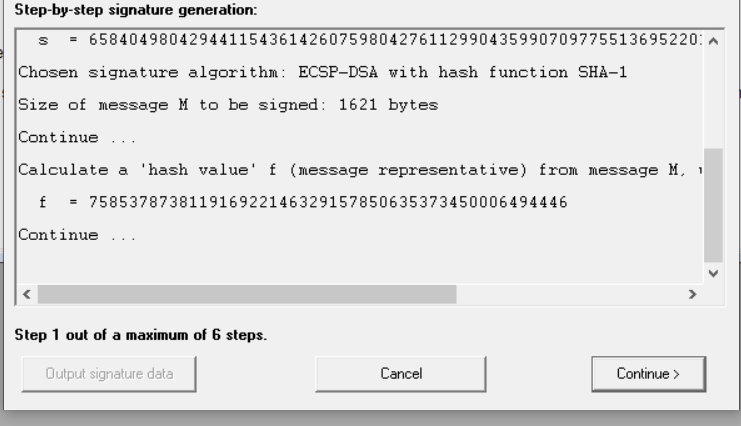


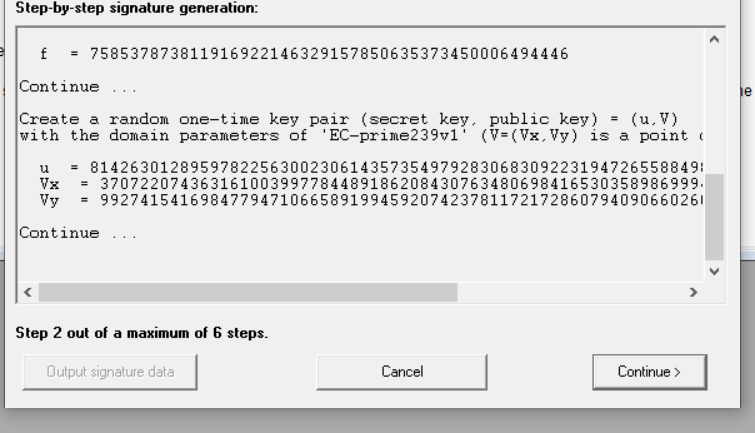
Задание.

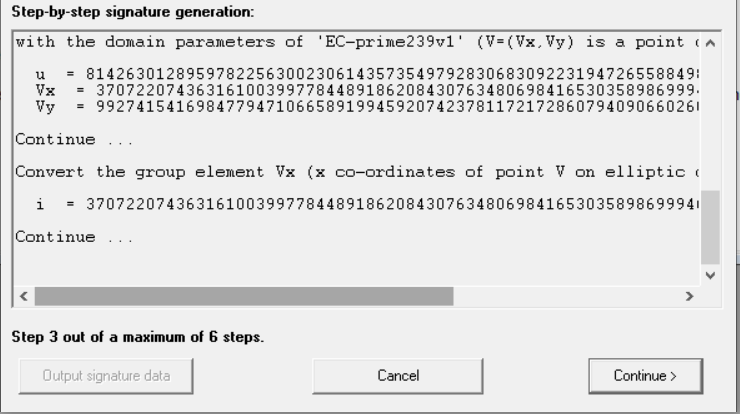
1. Выполните процедуру создание подписи *«Digital Signatures/PKI-> Sign Document…»* алгоритмом ECSP-DSA в пошаговом режиме (*Display inter. results=ON*). Зафиксируйте скриншоты последовательности шагов.

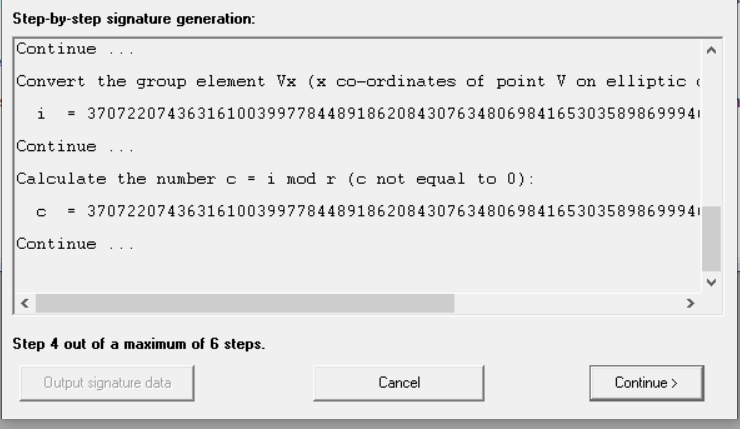
Шаги представлены на рис. 10

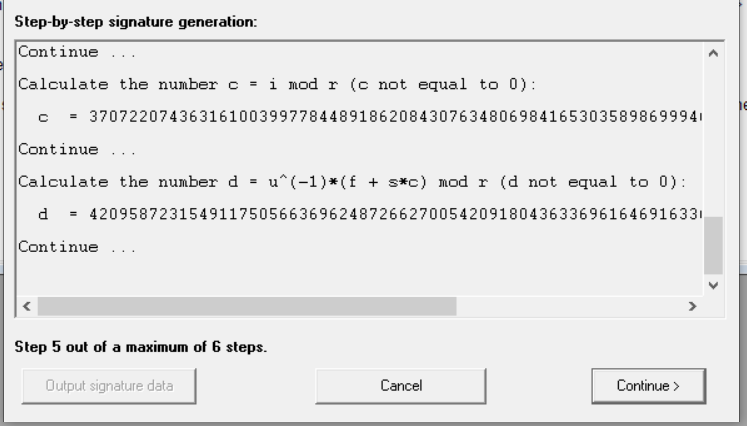


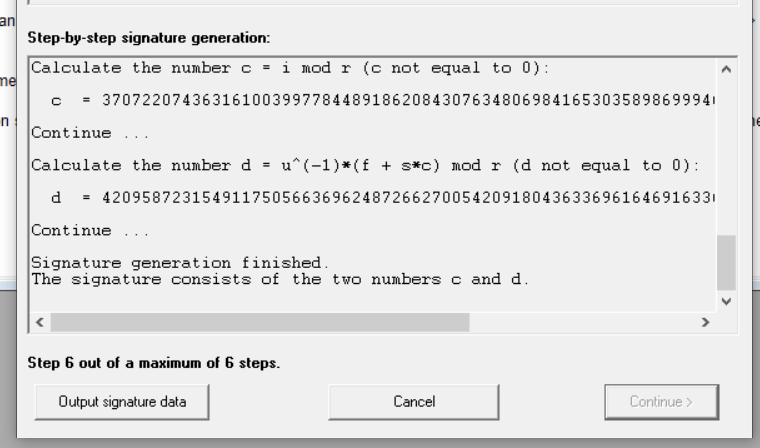












*Рисунок 10 – Этапы создания подписи*

Выполним сравнение реализаций алгоритма в CrypTool и в лекционных материалах.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cryptool** |  |  |  |  |  |  |  |  |  |  |  |
| **Лекция** |  |  |  |  |  |  |  |  |  |  |  |

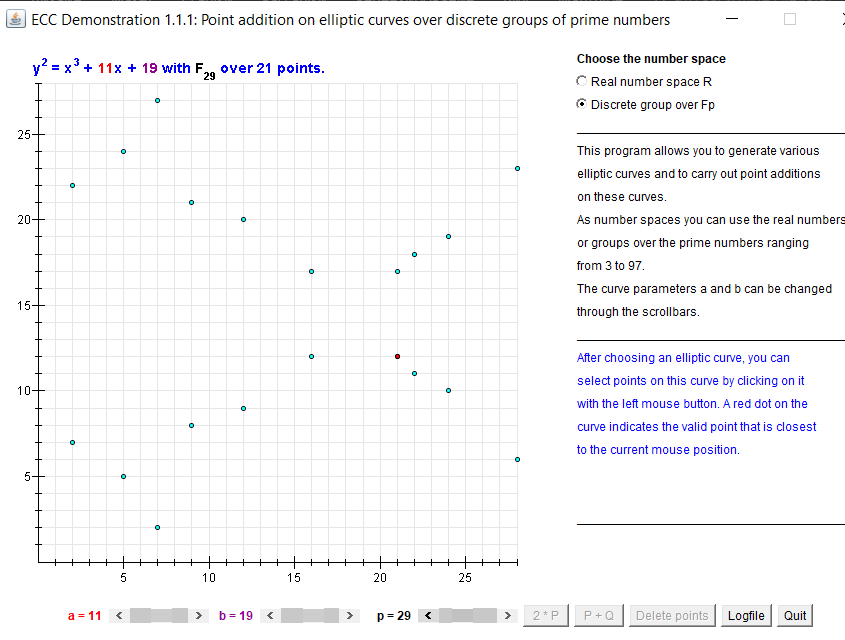
1. Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

Данная проверка уже выполнялась в пункте 8.1.

1. 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.*

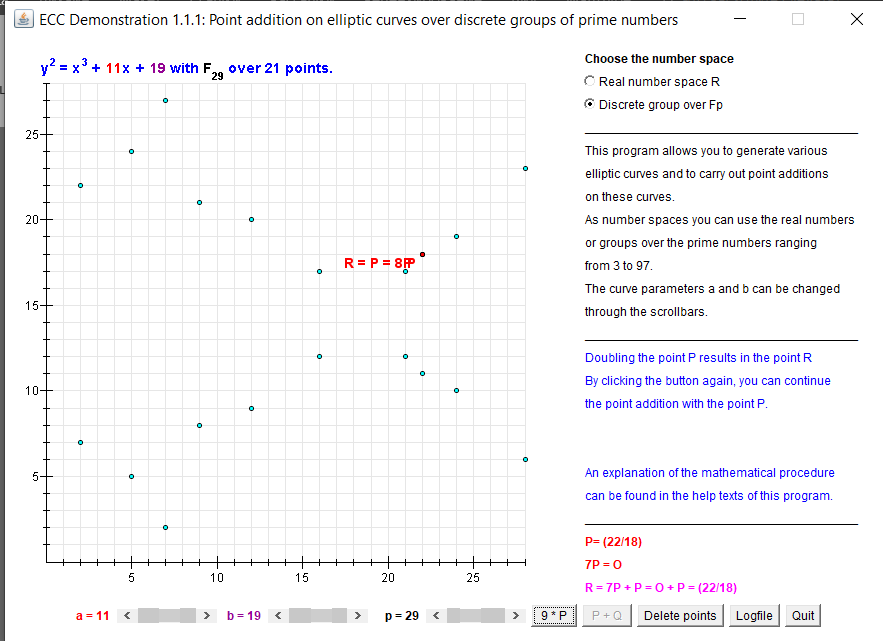
Подписание:

Выбрана эллиптическая кривая с параметрами .



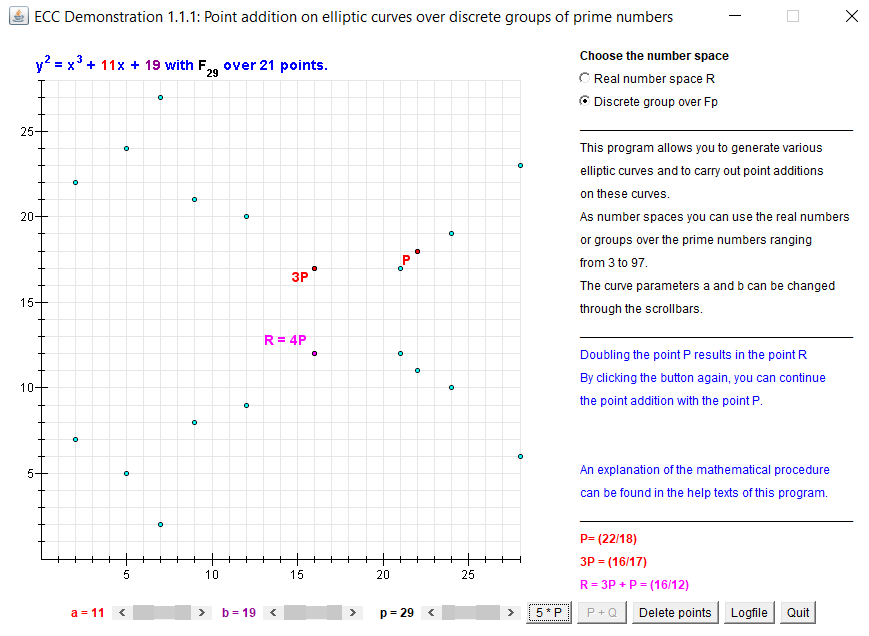
Выбрана точка P (в лекции ) на плоскости.

Определено перебором, пока не будет ( – порядок циклической подгруппы группы точек эллиптической кривой). .



Выбрано число .

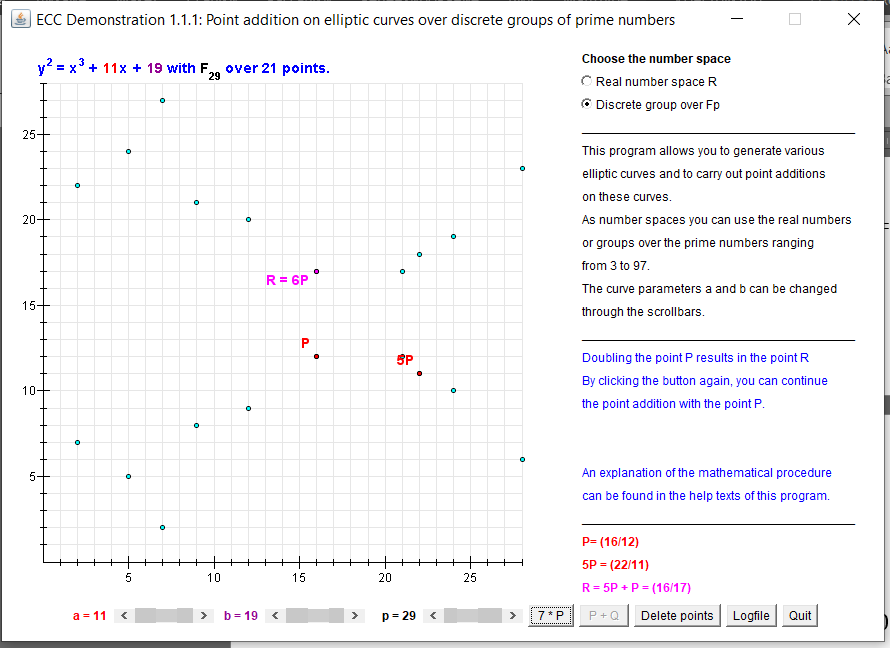
Найдена точка



Полученный открытый ключ: .

Пусть , тогда:

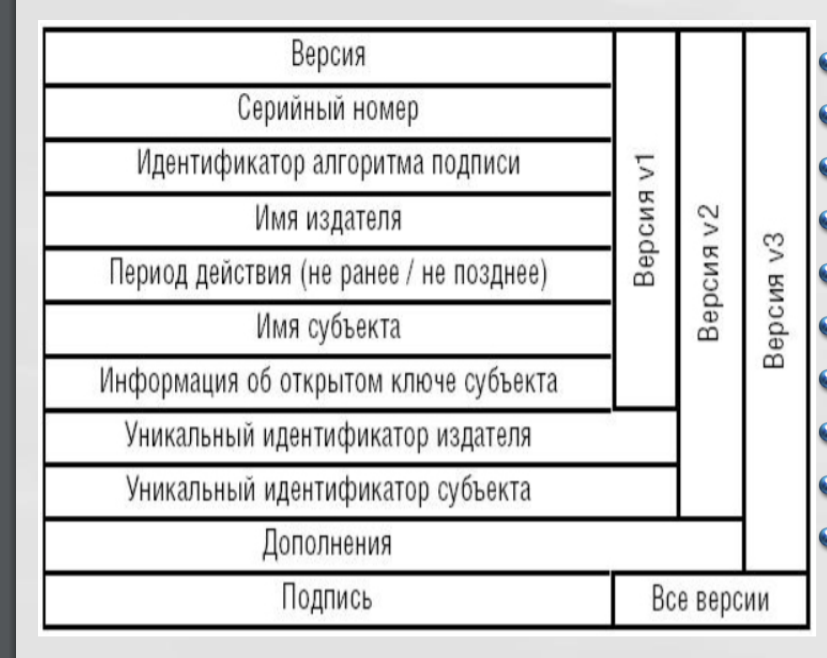
Проверка:



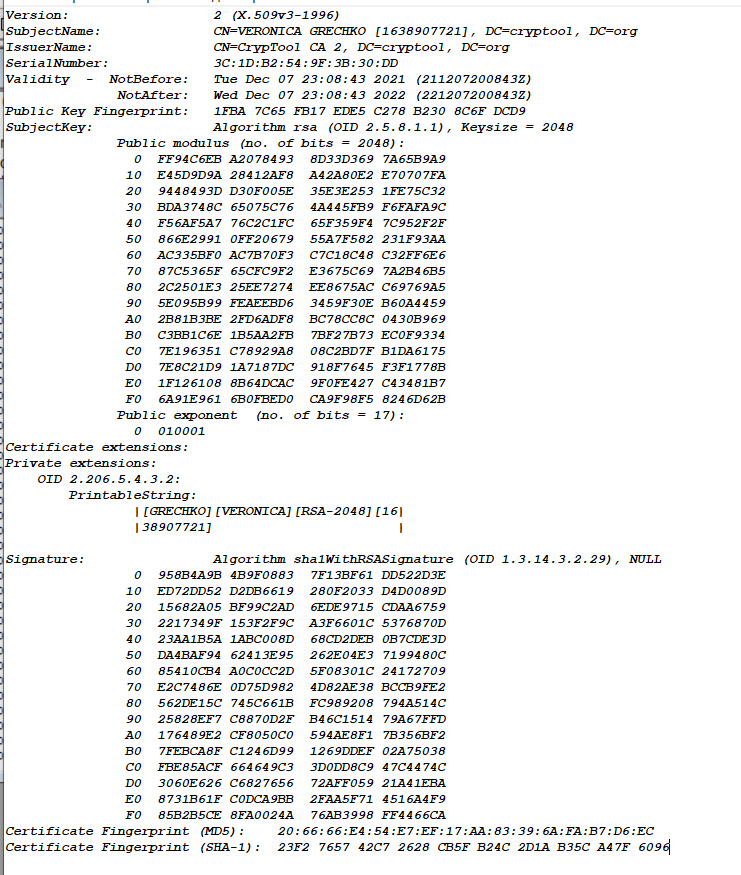
**Схемы цифровой подписи на эллиптических кривых.**

Сравнение структуры сертификата (как в лекции) и сертификата из CrypTool 1.0.

Структура сертификата, представленная в лекции:



Сертификат, полученный в CrypTool:



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

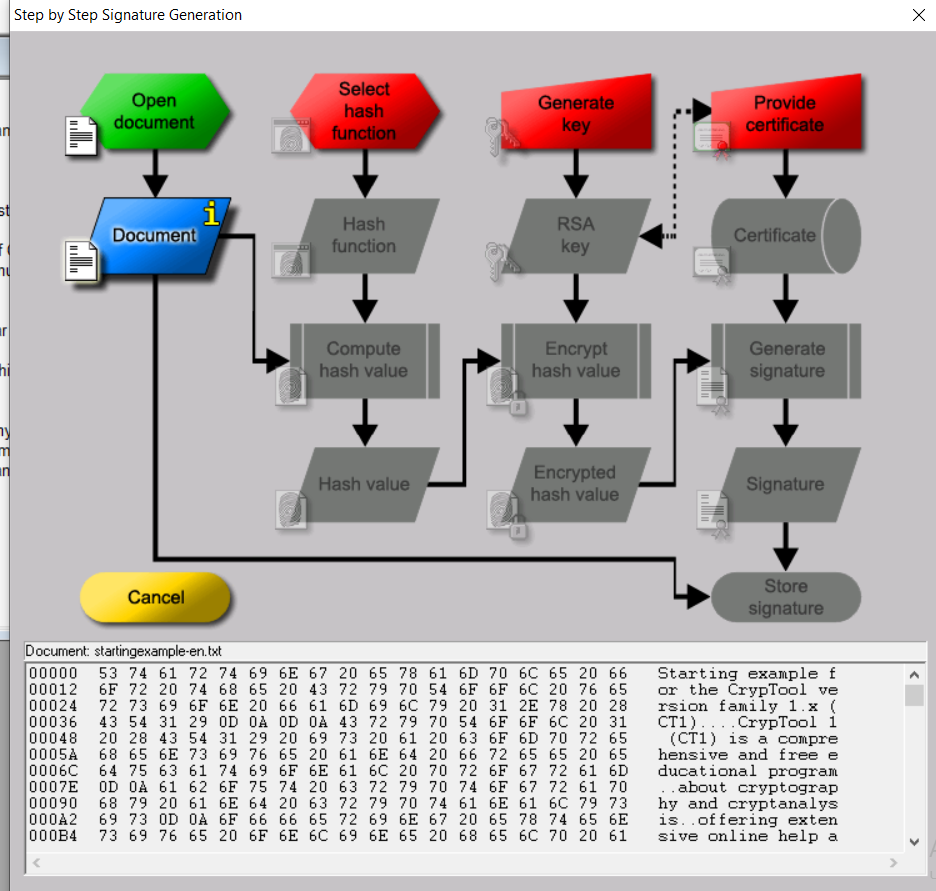
• в сертификате из Cryptool отсутствует поле «Назначение»;

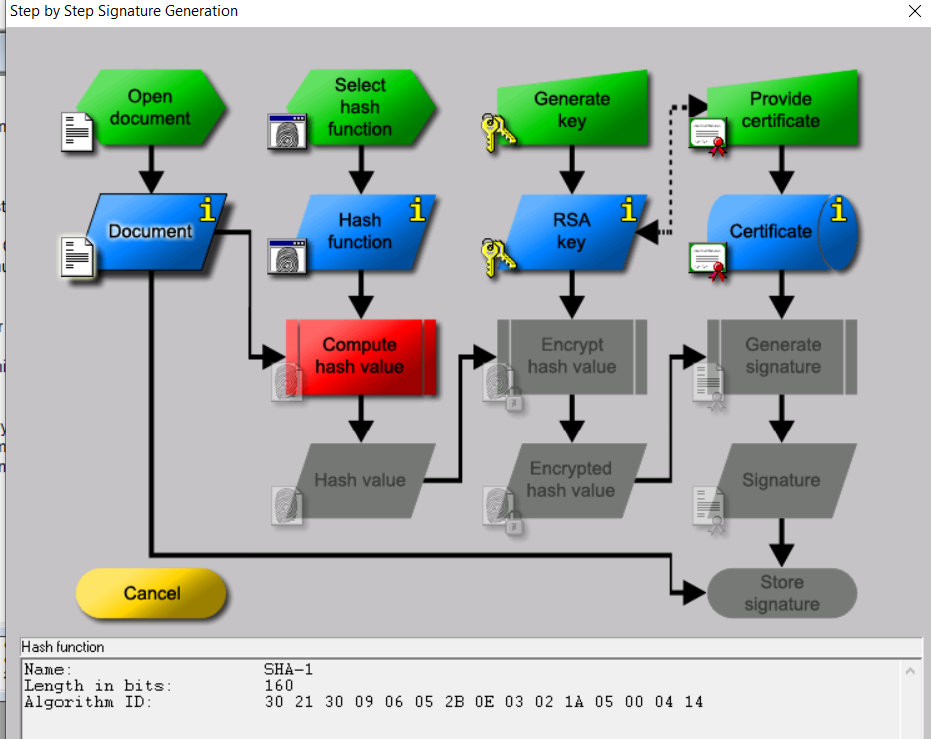
• в сертификате из Cryptool есть поле с дайджестом открытого ключа;

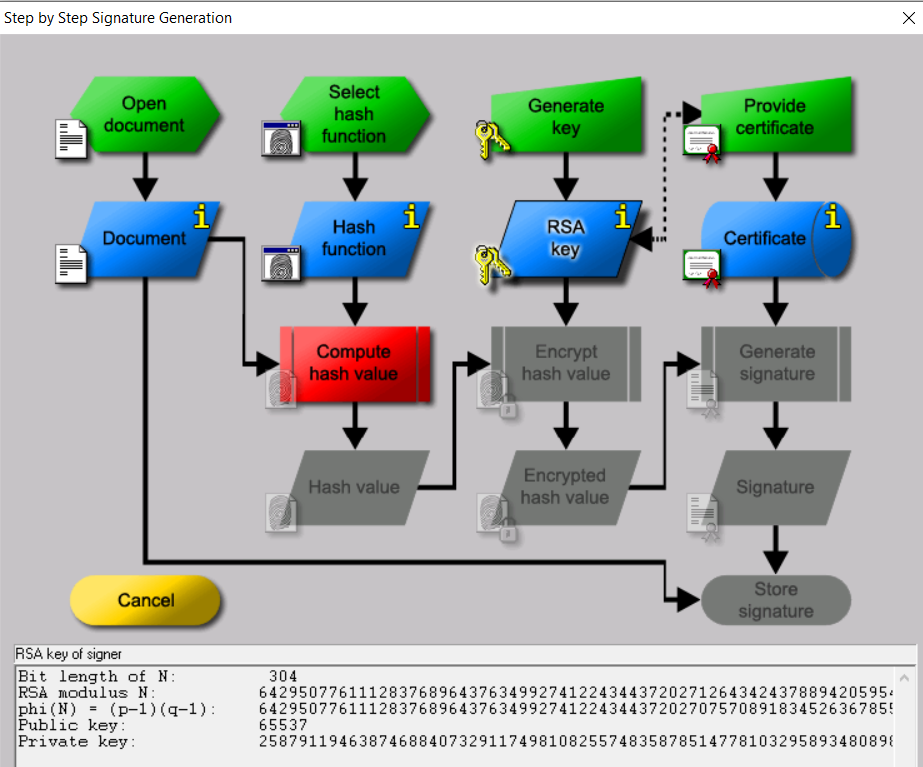
• в сертификате из Cryptool в расширениях есть только поле с идентификатором;

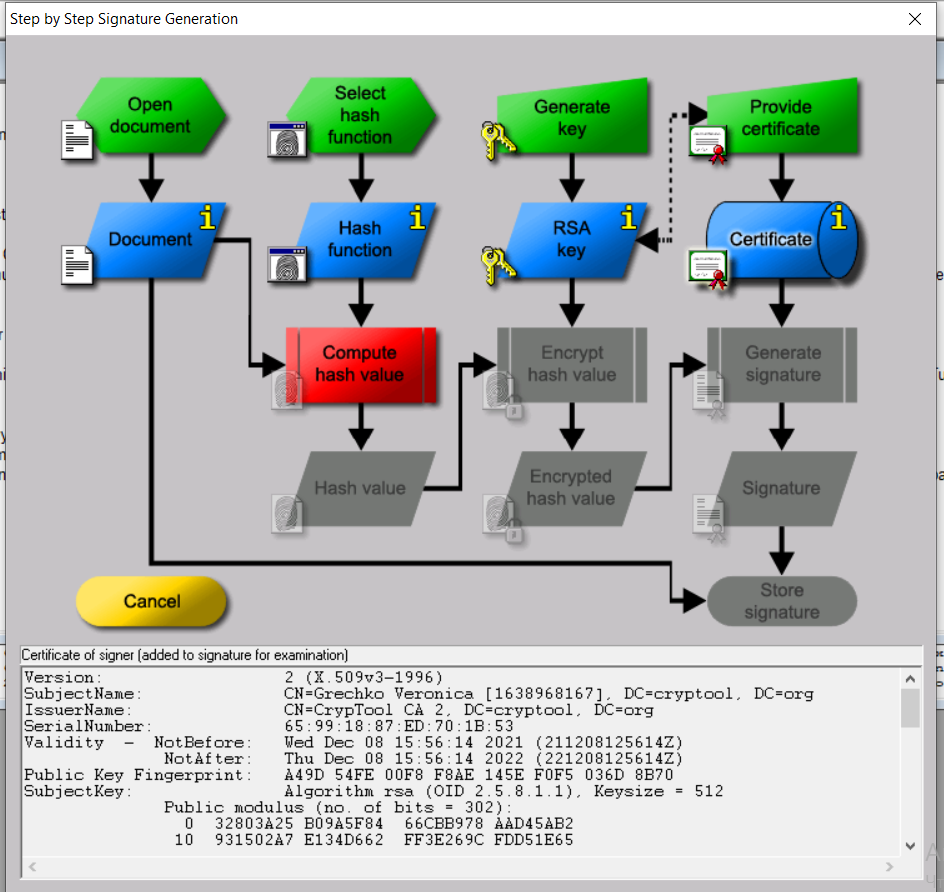
• в сертификате из Cryptool присутствуют поля с дайджестом сертификата, полученным с помощью MD5 и SHA-1.

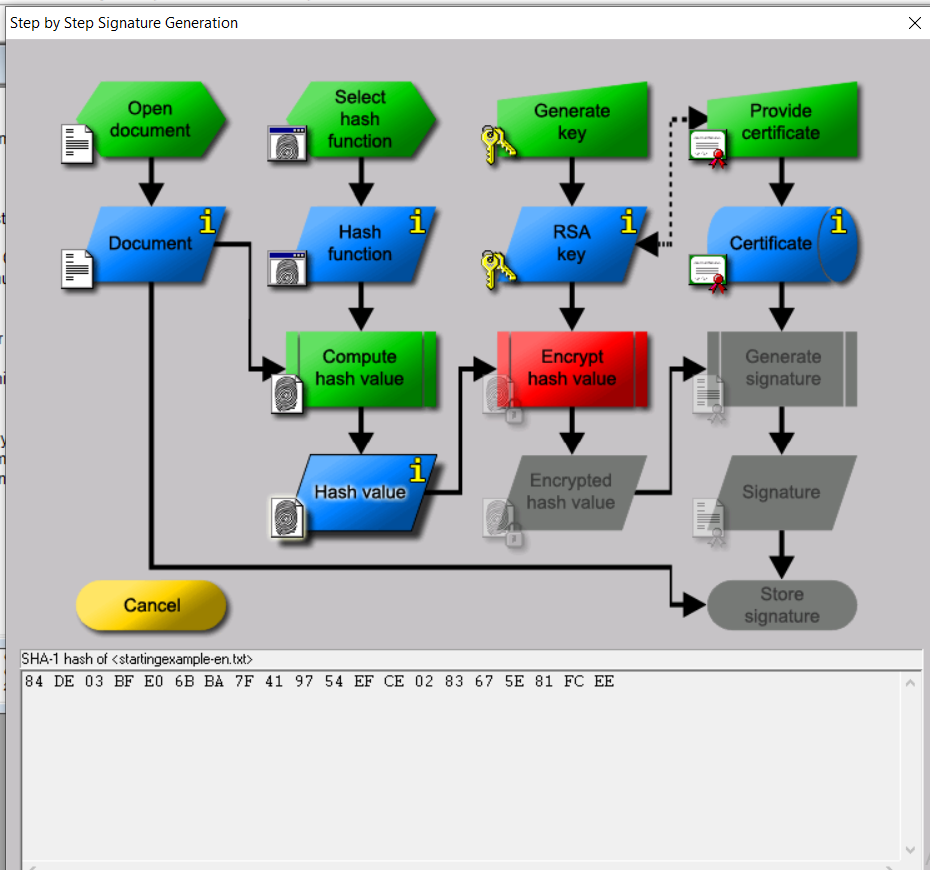
Схема процедуры подписания из CrypTool

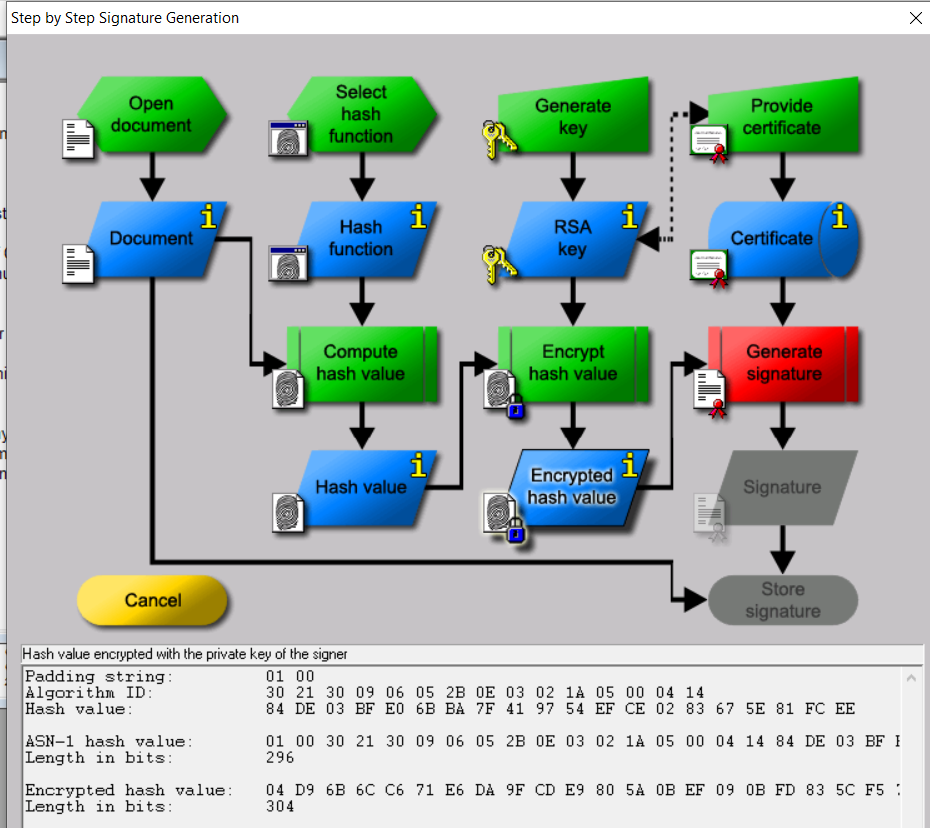


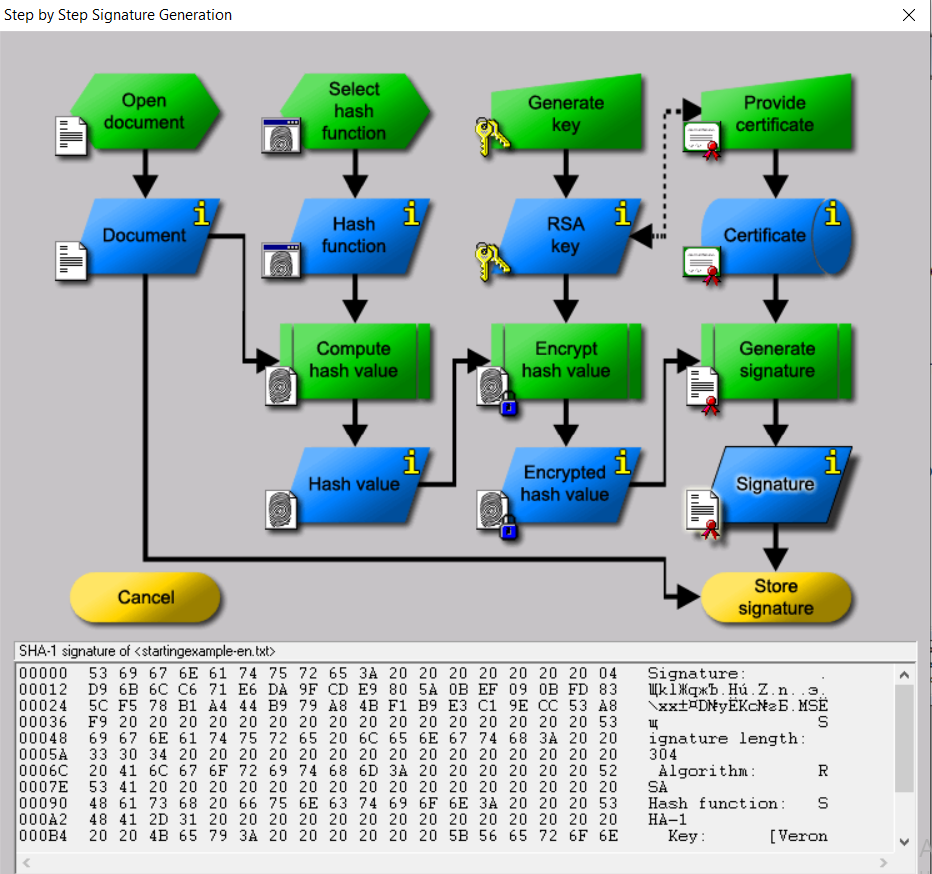












Задание.

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.

Полученный сертификат:

Version: 2 (X.509v3-1996)

SubjectName: CN=VERONICA GRECHKO [1638907721], DC=cryptool, DC=org

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

SerialNumber: 3C:1D:B2:54:9F:3B:30:DD

Validity - NotBefore: Tue Dec 07 23:08:43 2021 (211207200843Z)

NotAfter: Wed Dec 07 23:08:43 2022 (221207200843Z)

Public Key Fingerprint: 1FBA 7C65 FB17 EDE5 C278 B230 8C6F DCD9

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

Public modulus (no. of bits = 2048):

0 FF94C6EB A2078493 8D33D369 7A65B9A9

10 E45D9D9A 28412AF8 A42A80E2 E70707FA

20 9448493D D30F005E 35E3E253 1FE75C32

30 BDA3748C 65075C76 4A445FB9 F6FAFA9C

40 F56AF5A7 76C2C1FC 65F359F4 7C952F2F

50 866E2991 0FF20679 55A7F582 231F93AA

60 AC335BF0 AC7B70F3 C7C18C48 C32FF6E6

70 87C5365F 65CFC9F2 E3675C69 7A2B46B5

80 2C2501E3 25EE7274 EE8675AC C69769A5

90 5E095B99 FEAEEBD6 3459F30E B60A4459

A0 2B81B3BE 2FD6ADF8 BC78CC8C 0430B969

B0 C3BB1C6E 1B5AA2FB 7BF27B73 EC0F9334

C0 7E196351 C78929A8 08C2BD7F B1DA6175

D0 7E8C21D9 1A7187DC 918F7645 F3F1778B

E0 1F126108 8B64DCAC 9F0FE427 C43481B7

F0 6A91E961 6B0FBED0 CA9F98F5 8246D62B

Public exponent (no. of bits = 17):

0 010001

Certificate extensions:

Private extensions:

OID 2.206.5.4.3.2:

PrintableString:

|[GRECHKO][VERONICA][RSA-2048][16|

|38907721] |

Signature: Algorithm sha1WithRSASignature (OID 1.3.14.3.2.29), NULL

0 958B4A9B 4B9F0883 7F13BF61 DD522D3E

10 ED72DD52 D2DB6619 280F2033 D4D0089D

20 15682A05 BF99C2AD 6EDE9715 CDAA6759

30 2217349F 153F2F9C A3F6601C 5376870D

40 23AA1B5A 1ABC008D 68CD2DEB 0B7CDE3D

50 DA4BAF94 62413E95 262E04E3 7199480C

60 85410CB4 A0C0CC2D 5F08301C 24172709

70 E2C7486E 0D75D982 4D82AE38 BCCB9FE2

80 562DE15C 745C661B FC989208 794A514C

90 25828EF7 C8870D2F B46C1514 79A67FFD

A0 176489E2 CF8050C0 594AE8F1 7B356BF2

B0 7FEBCA8F C1246D99 1269DDEF 02A75038

C0 FBE85ACF 664649C3 3D0DD8C9 47C4474C

D0 3060E626 C6827656 72AFF059 21A41EBA

E0 8731B61F C0DCA9BB 2FAA5F71 4516A4F9

F0 85B2B5CE 8FA0024A 76AB3998 FF4466CA

Certificate Fingerprint (MD5): 20:66:66:E4:54:E7:EF:17:AA:83:39:6A:FA:B7:D6:EC

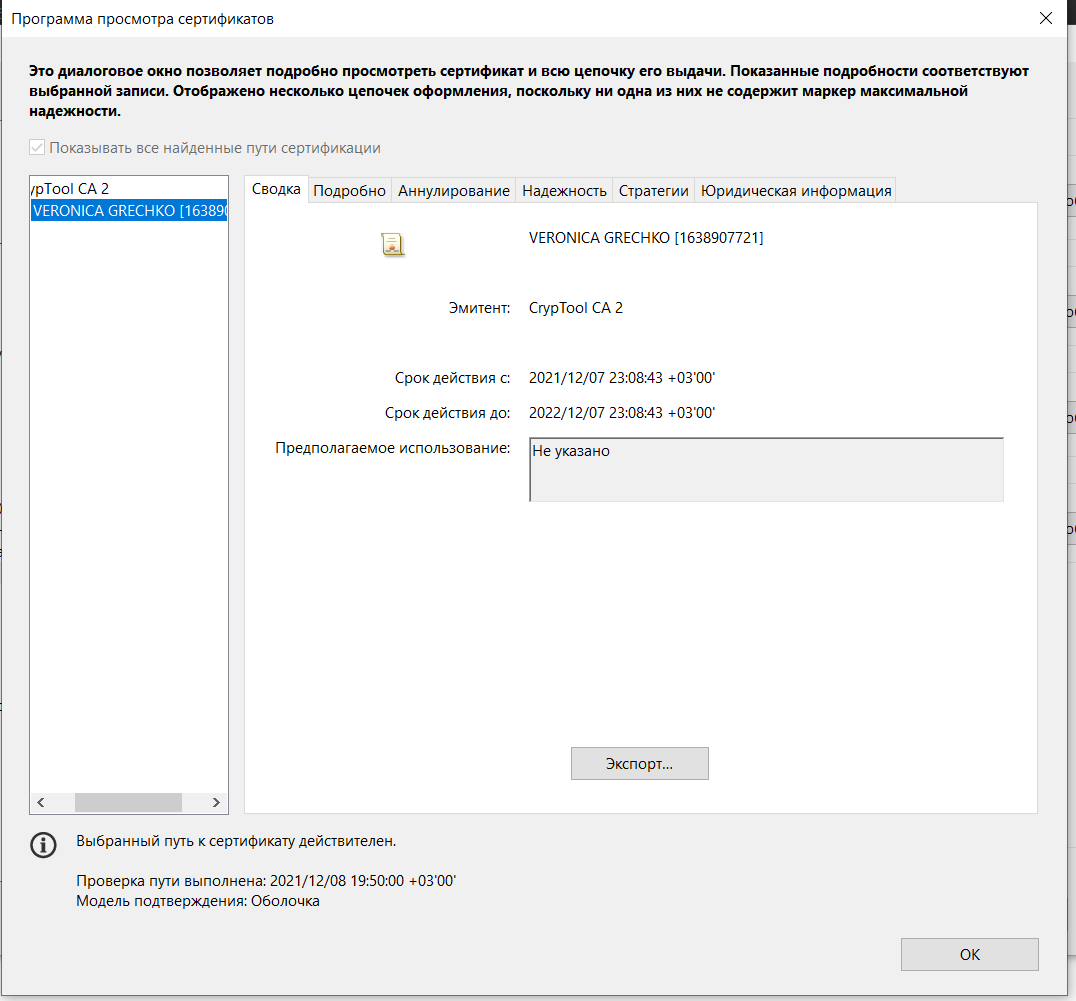
Certificate Fingerprint (SHA-1): 23F2 7657 42C7 2628 CB5F B24C 2D1A B35C A47F 6096

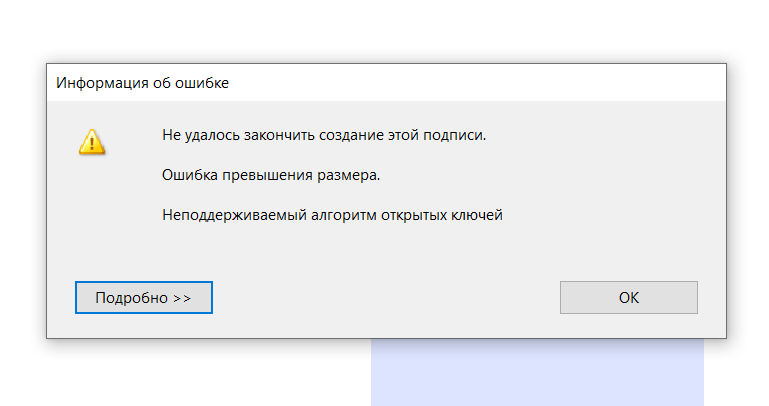
**Подписание своего отчета.**

1. Сконвертируйте отчет в формат pdf.

2. Экспортируйте ранее созданный сертификат ключевой пары *RSA Digital Signatures/PKI->PKI/Generate…->Export PSE(#PKCS12).*

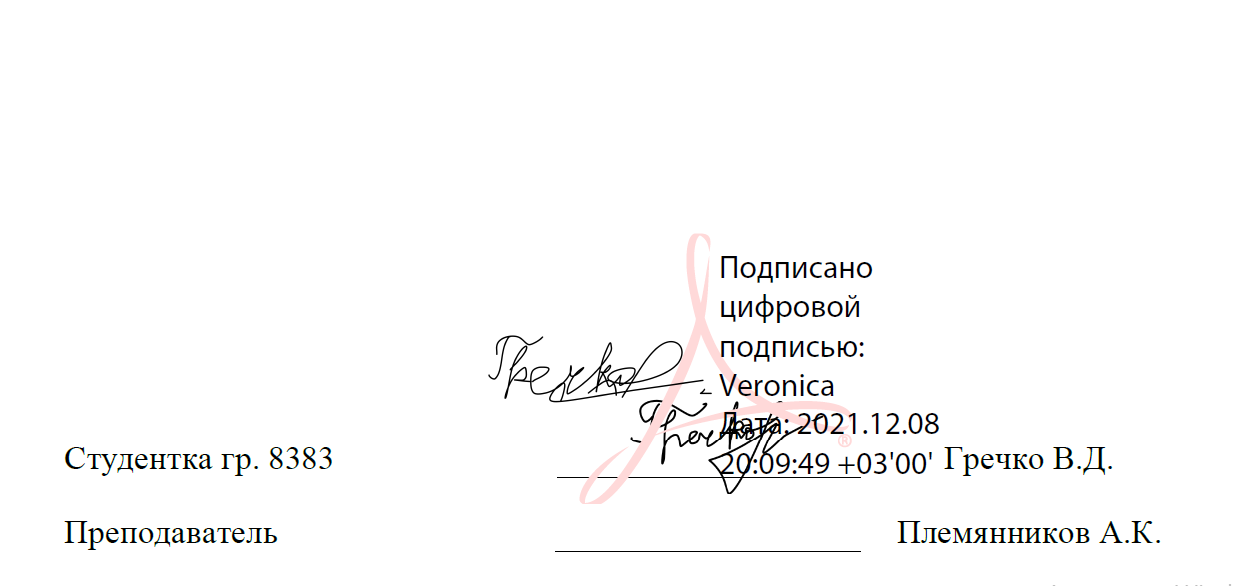
3. Откройте pdf-версию отчета и попытайтесь подписать с использованием этого сертификата.



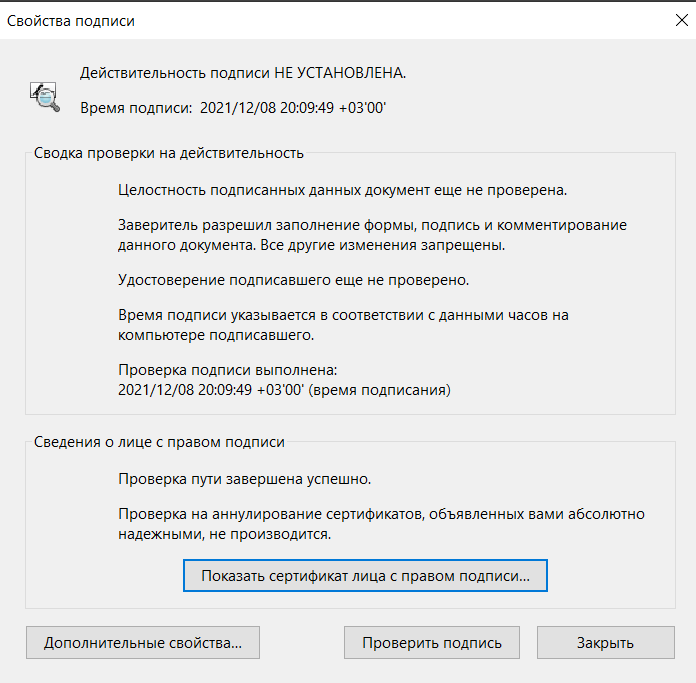


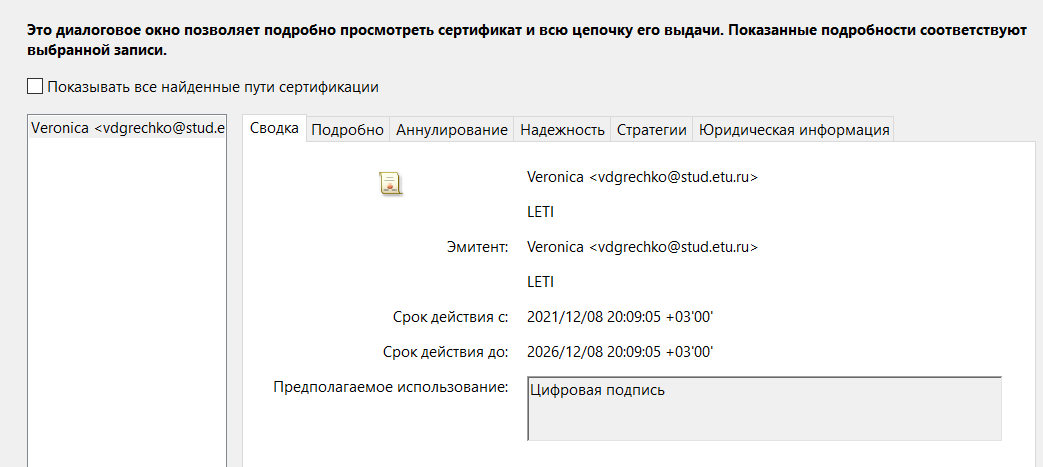
К сожалению, подписать отчет сертификатом RSA не удалось.

4. Создайте собственный самоподписанный сертификат в среде Adobe Reader и используйте его для подписи отчета.

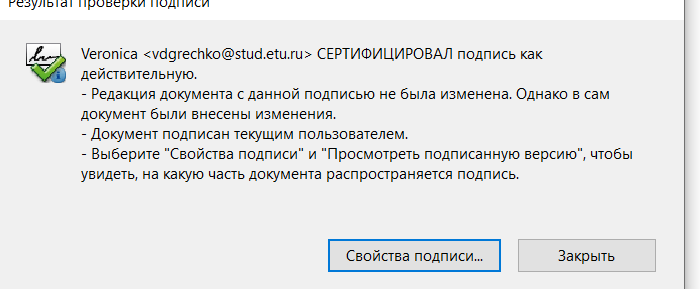


5. Сохраните скриншоты свойств подписи и сертификата.





6. Внесите изменения (маркеры, комментарии) в отчет и проверьте подпись.



**Выводы.**

1. Были изучены механизмы генерации ключевых пар для различных алгоритмов. Алгоритм RSA генерирует пары – открытый ключ и – закрытый ключ на основе двух больших простых чисел и , которые впоследствии должны быть уничтожены. Алгоритм DSA генерирует пары – открытый ключ и – закрытый ключ на основе простого числа (длина от 512 до 1024 бит), (такого, что ) и . Алгоритм генерирует пары – открытый ключ и – закрытый ключ на основе произвольно выбранной эллиптической кривой , где – простое число, произвольно выбранной точки на данной кривой, , простое число (порядок одной из циклических подгрупп группы точек эллиптической кривой). Наименьшая скорость генерация была у алгоритма EC-239 и составила 0.012 секунд.
2. Был изучен механизм создания цифровой подписи с различными ключами. Наиболее оптимальным будет использования алгоритма ESDSA для создания и подтверждения подписи – на большом тексте операция создания занимает 0 секунд, а процесс проверки 0.002 секунд. Вычисление DSA подписи быстрее, чем вычисление подписей RSА, однако DSA требуется больше времени на проверку целостности.
3. Был изучен алгоритм формирования и проверки подписи ECDSA, основанный на эллиптических кривых. Открытый ключ представляет собой пару , где – параметры, задающие определённую эллиптическую кривую, – произвольная точка на кривой, – порядок циклической подгруппы группы точек эллиптической кривой, такой, что для некоторой точки , лежащей на кривой, верно: , где – закрытый ключ.
4. Было изучено создание сертификатов в среде PKI. Данная инфраструктура решает основные задачи криптографии, а именно: обеспечение конфиденциальности информации; обеспечение целостности информации; обеспечение аутентификации пользователей и ресурсов, к которым обращаются пользователи; обеспечение возможности подтверждения совершенных пользователями действий. Сертификат - это электронный документ, который содержит: открытый ключ пользователя (открытый ключ) информацию о пользователе, которому принадлежит сертификат информацию о сроке действия сертификата информацию об издателе сертификата другие атрибуты цифровую подпись удостоверяющего центра, выдавшего сертификат (издателя сертификата). Сертификат подтверждает электронную цифровую подпись и открытый ключ отправителя.
5. Было изучено создание подписи и проверка документа на целостность после внесения изменений средствами Adobe Acrobat.