**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. 8383 |  | Киреев К.А. |
| 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 .

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 | 2.021 секунд |
| DSA-2048 | 1.905 секунд |
| EC-239 | 0.010 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.

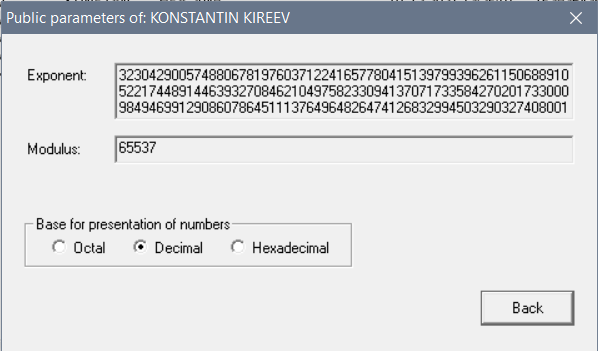
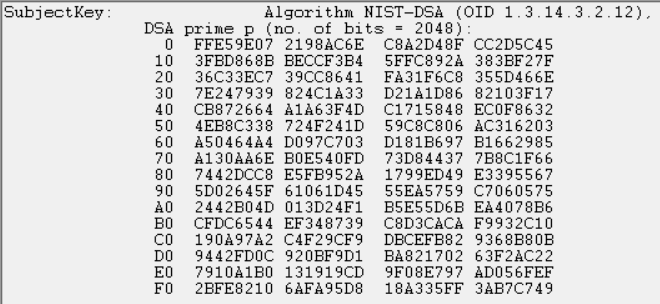
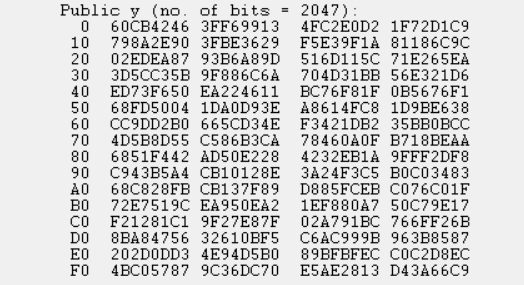
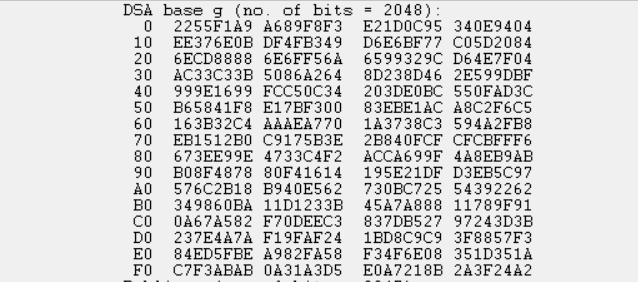


Рисунок 1 – Открытый ключ для RSA-2048







Рисунок 2 – Открытый ключ для DSA-2048

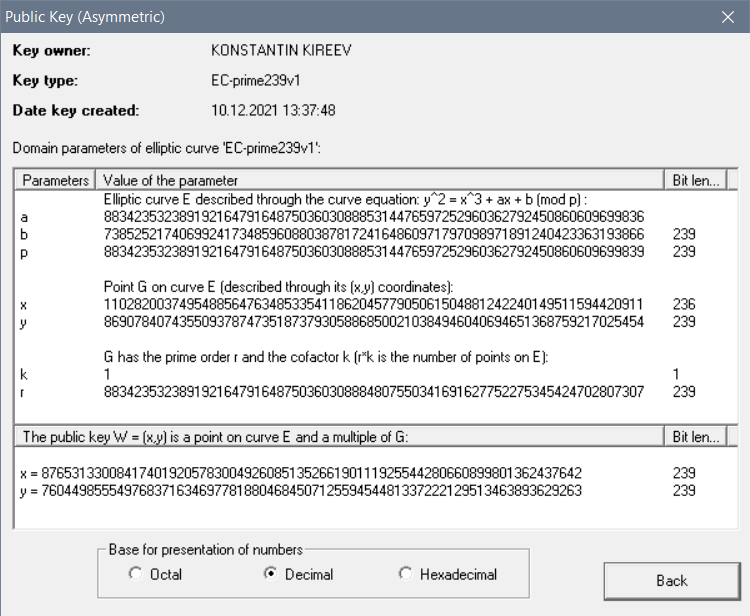


Рисунок 3 – Открытый ключ для EC-239

**Процессы создания и проверки цифровой подписи**

***Обобщенная схема создания и проверки цифровой подписи***

Обобщенная схема представлена на рис. 4.

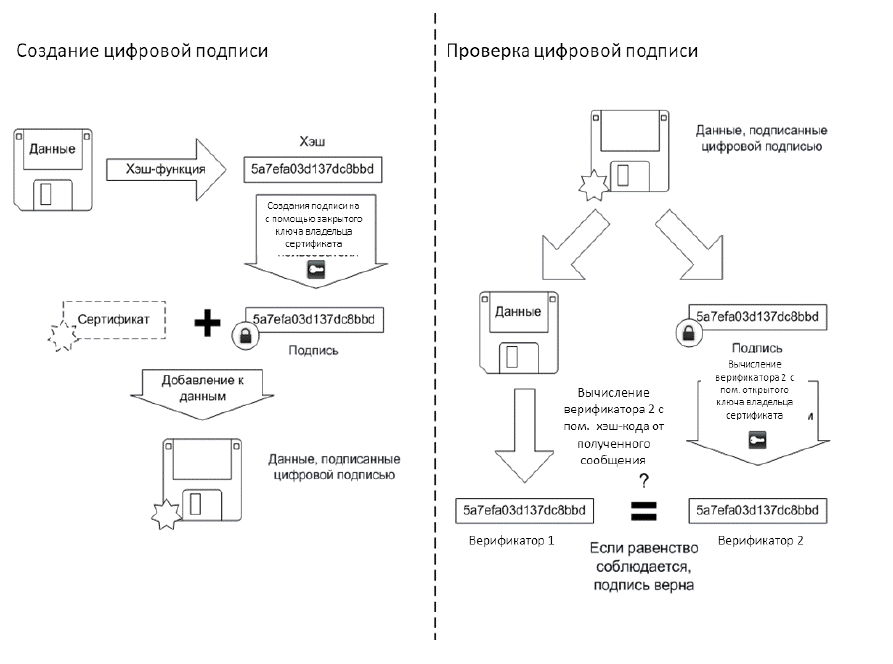


Рисунок 4 – Обобщенная схема

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

Исходный текст представлен на рис. 5.

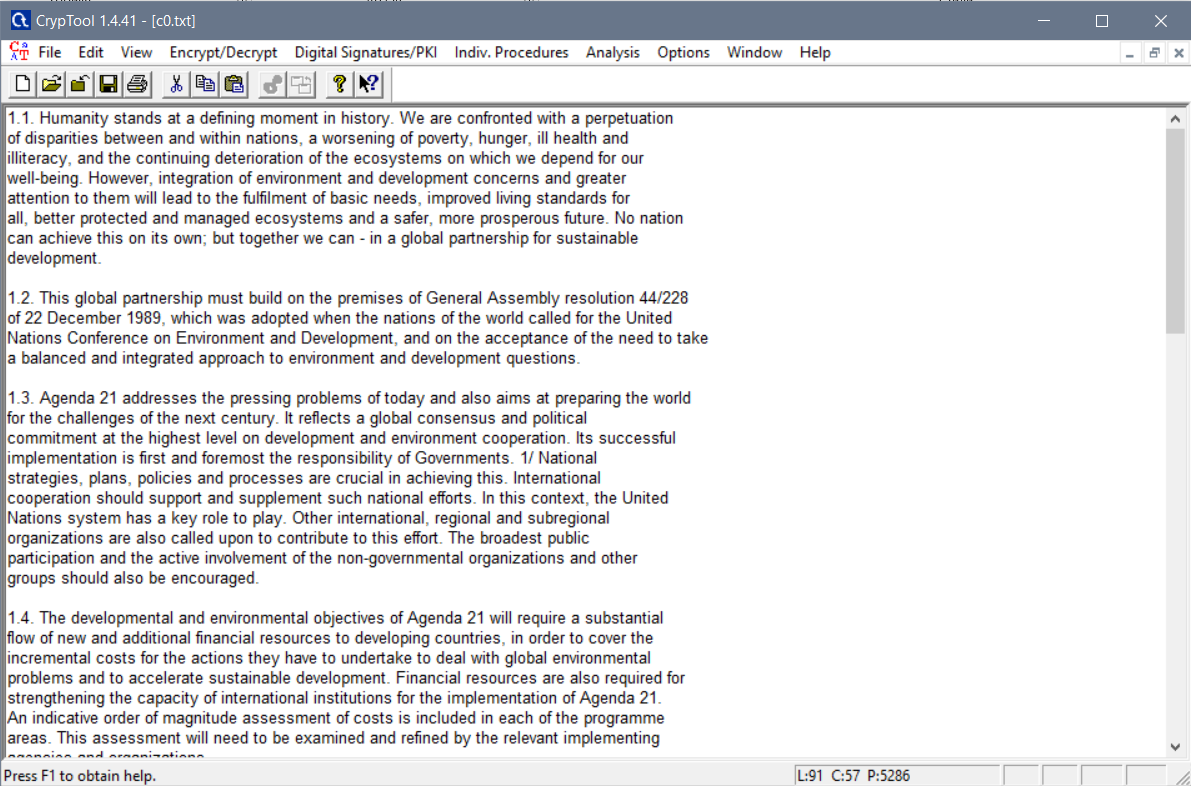
**

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 | 0.006 секунд |
| SHA-1 | 0.006 секунд |
| DSA-2048 | MD5 | - |
| SHA-1 | 0.000 секунд |
| EC-239 | MD5 | - |
| SHA-1 | 0.002 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 Рисунок 9 – Цифровая подпись ECDSA  с SHA-1 с SHA-1 | |
|  | |

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

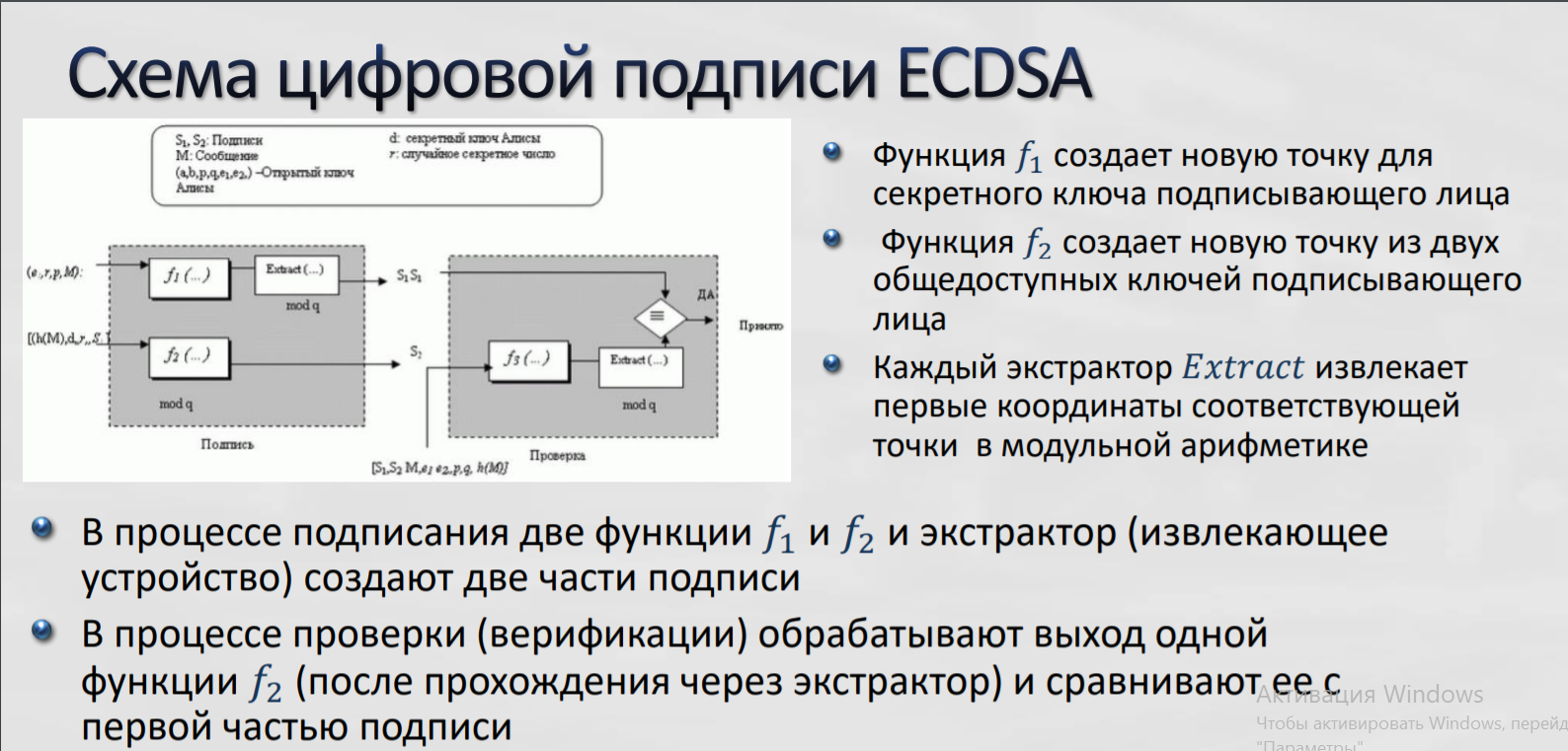
Выполним проверку и сведем полученные значения в таблицу 3.

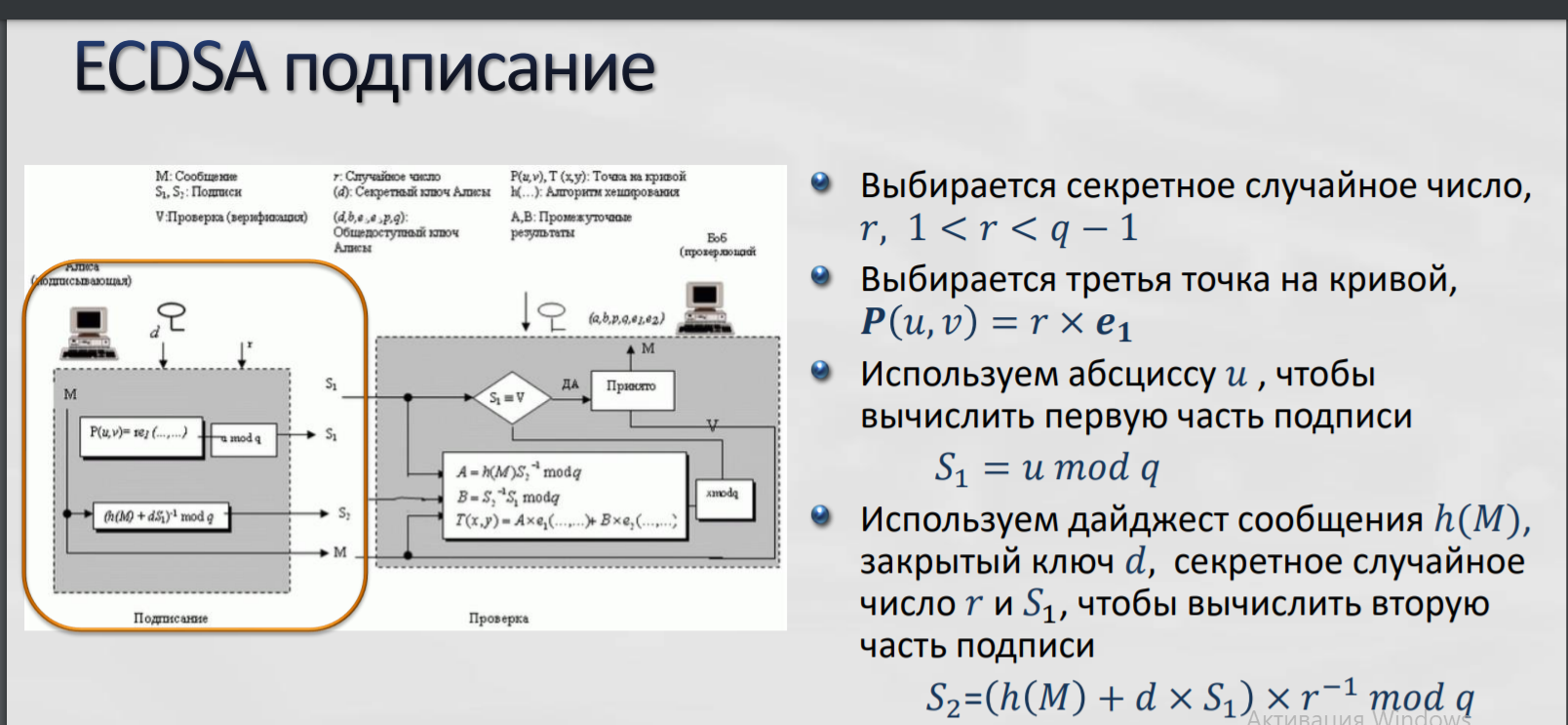
*Таблица 3 – Проверка подписей на сохранение и нарушение целостности*

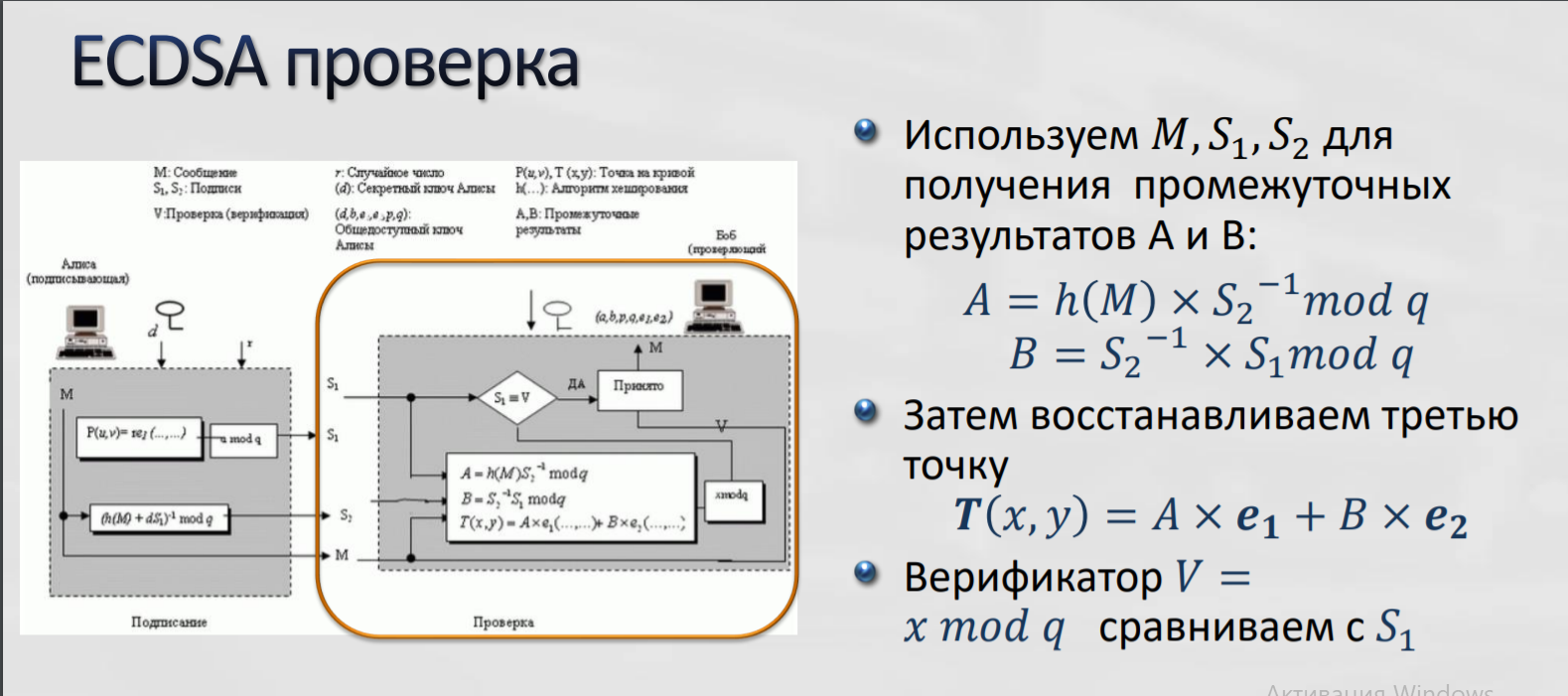
|  |  |  |
| --- | --- | --- |
| ***Алгоритм*** | ***Сохранение целостности*** | ***Нарушение целостности*** |
| RSA-2048 |  |  |
|  |  |
| DSA-2048 |  |  |
| EC-239 |  |  |

**Схемы цифровой подписи на эллиптических кривых**

***Описание алгоритма формирования и проверки подписи ECDSA***

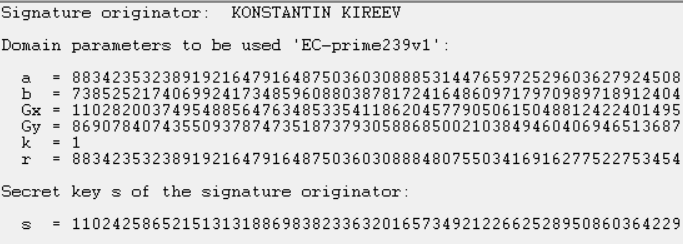




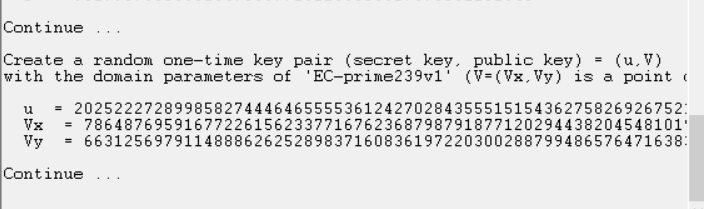


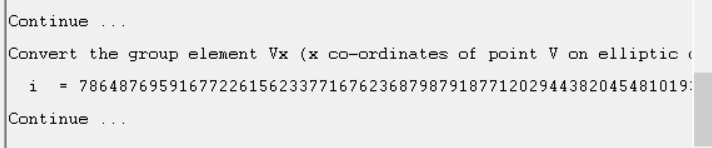
1. Выполните процедуру создание подписи *«Digital Signatures/PKI-> Sign Document…»* алгоритмом ECSP-DSA в пошаговом режиме (*Display inter. results=ON*). Зафиксируйте скриншоты последовательности шагов.

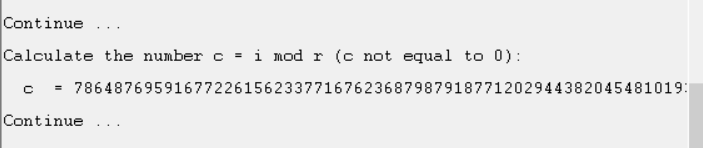
Шаги представлены на рис. 10

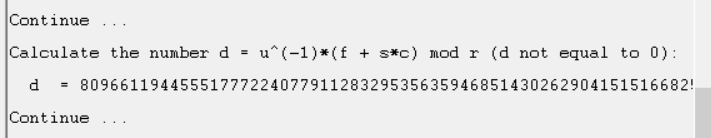












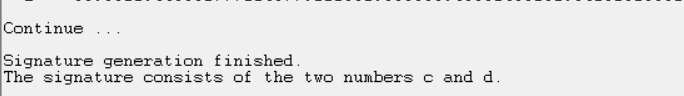


Рисунок 10 – Этапы создания подписи

Выполним сравнение реализаций алгоритма в CrypTool и в лекционных материалах.

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

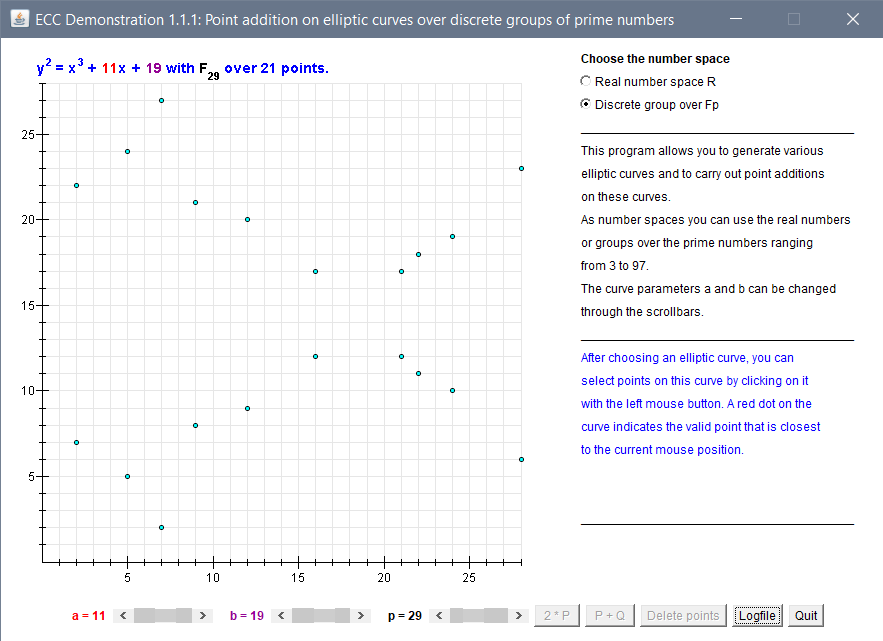
1. Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

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.*

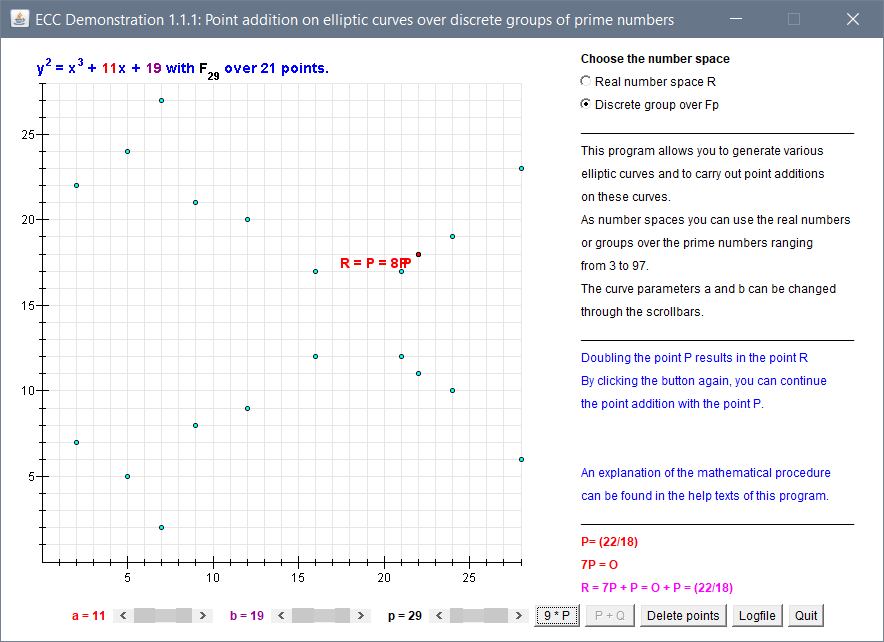
Подписание:

Выбрана эллиптическая кривая с параметрами .



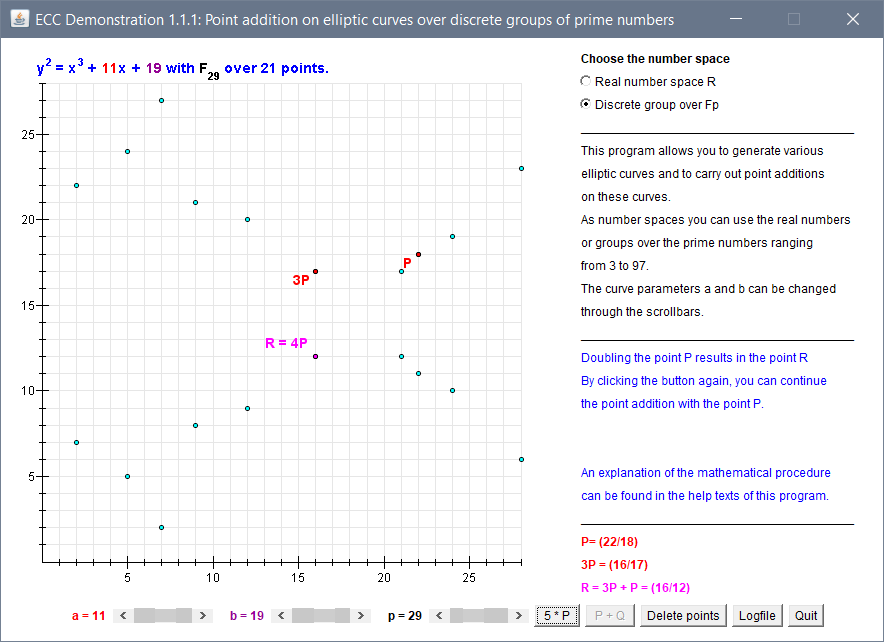
Выбрана точка P (в лекции ) на плоскости.

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



Выбрано число .

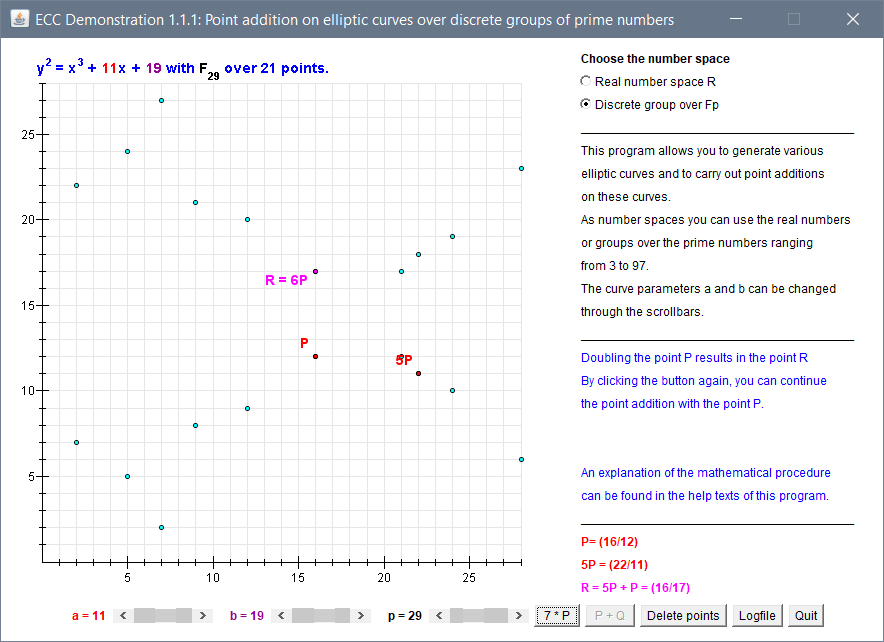
Найдена точка



Полученный открытый ключ: .

Пусть , тогда:

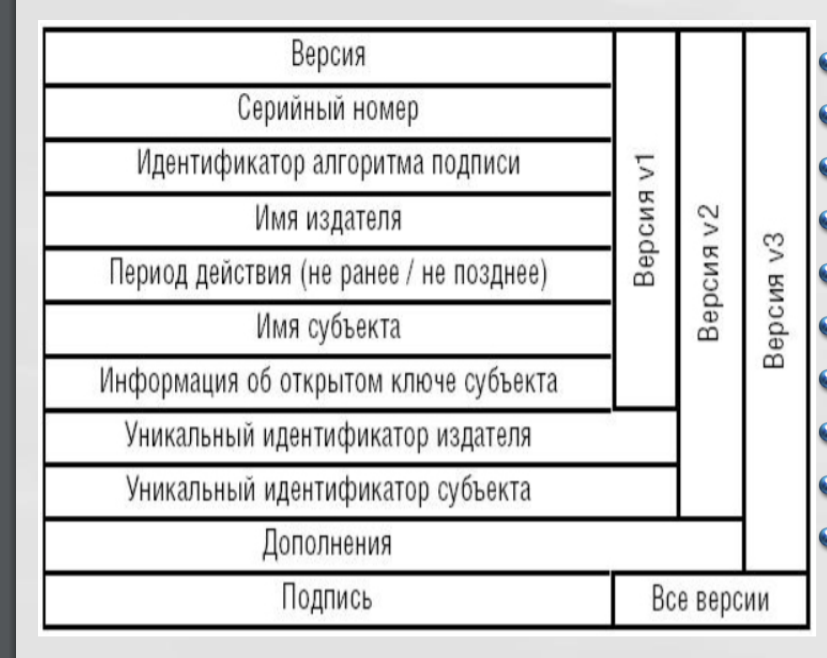
Проверка:



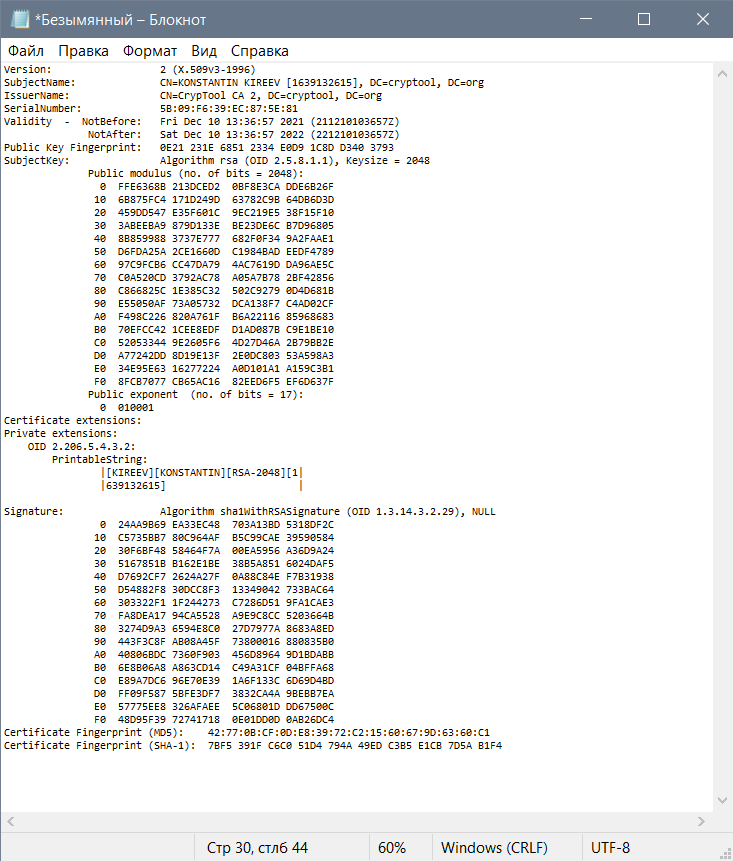
**Схемы цифровой подписи на эллиптических кривых**

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

*Структура сертификата*



*Сертификат, полученный в CrypTool*



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

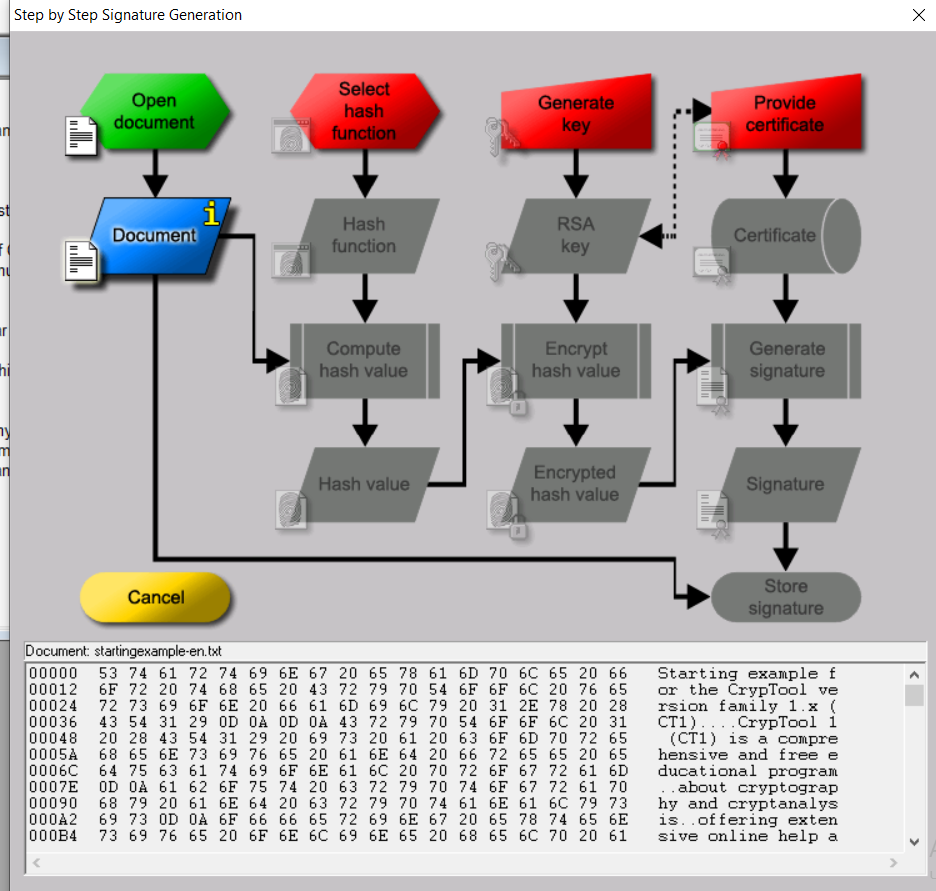
• в сертификате из Cryptool отсутствует поле «Назначение»;

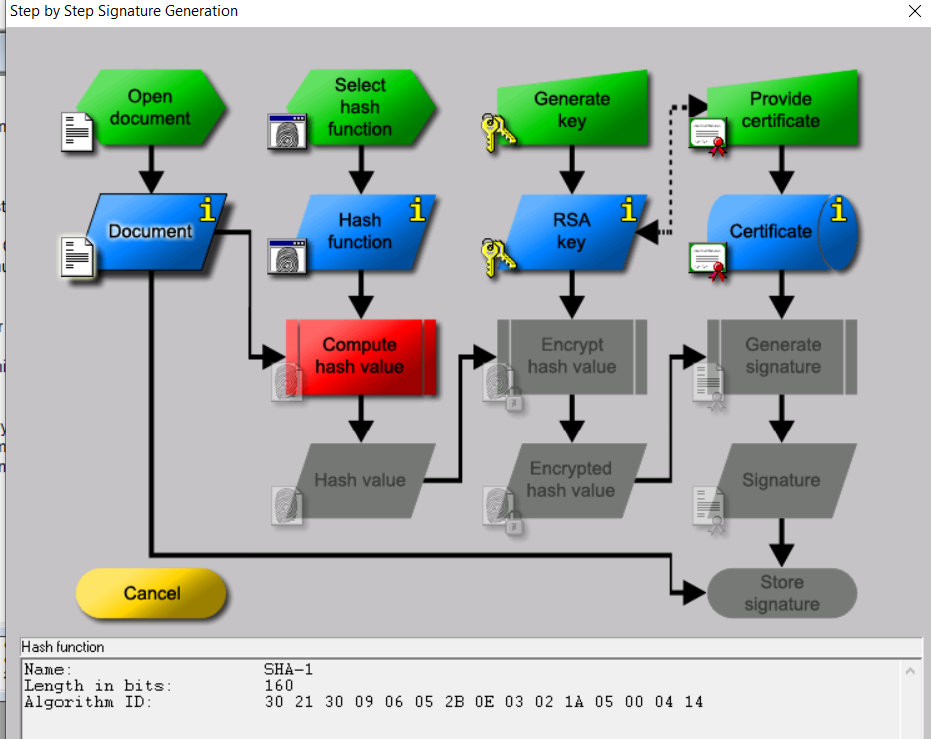
• в сертификате из Cryptool есть поле с дайджестом открытого ключа;

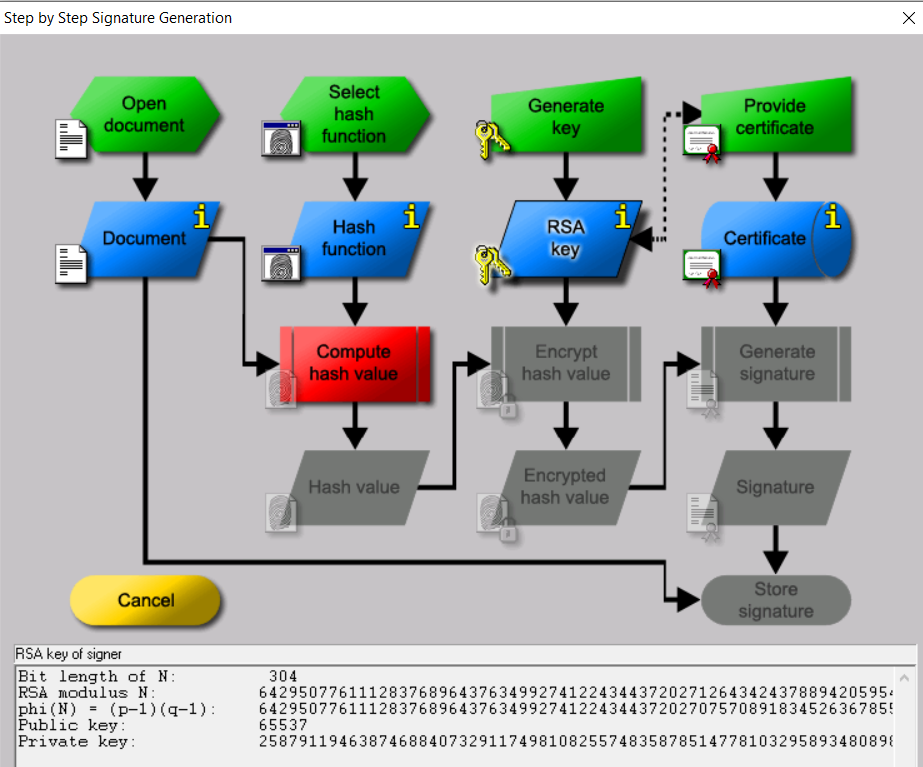
• в сертификате из Cryptool в расширениях есть только поле с идентификатором;

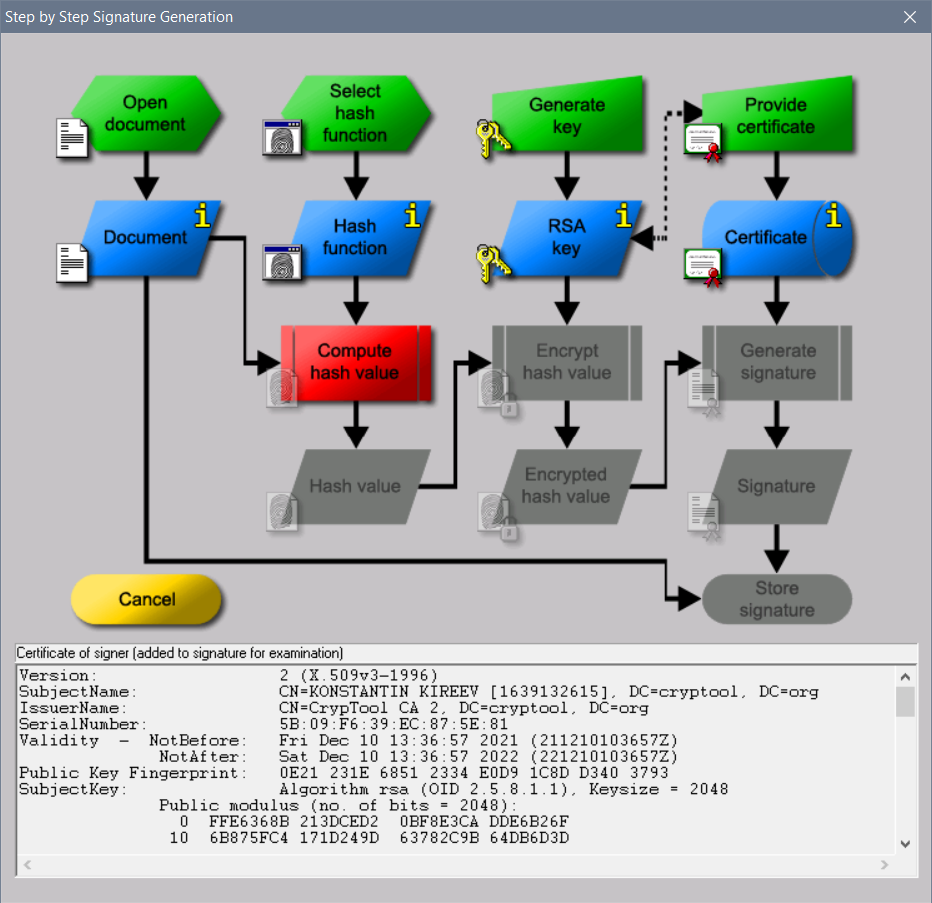
• в сертификате из Cryptool присутствуют поля с дайджестом сертификата, полученным с помощью MD5 и SHA-1.

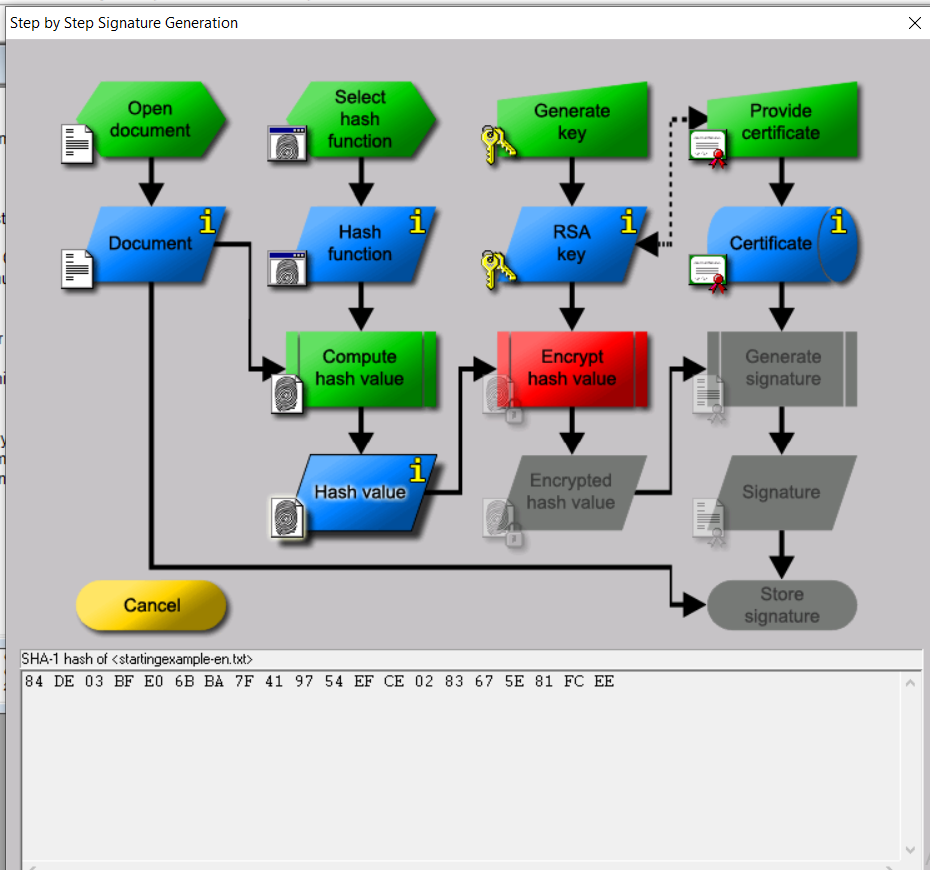
***Схема процедуры подписания из CrypTool***

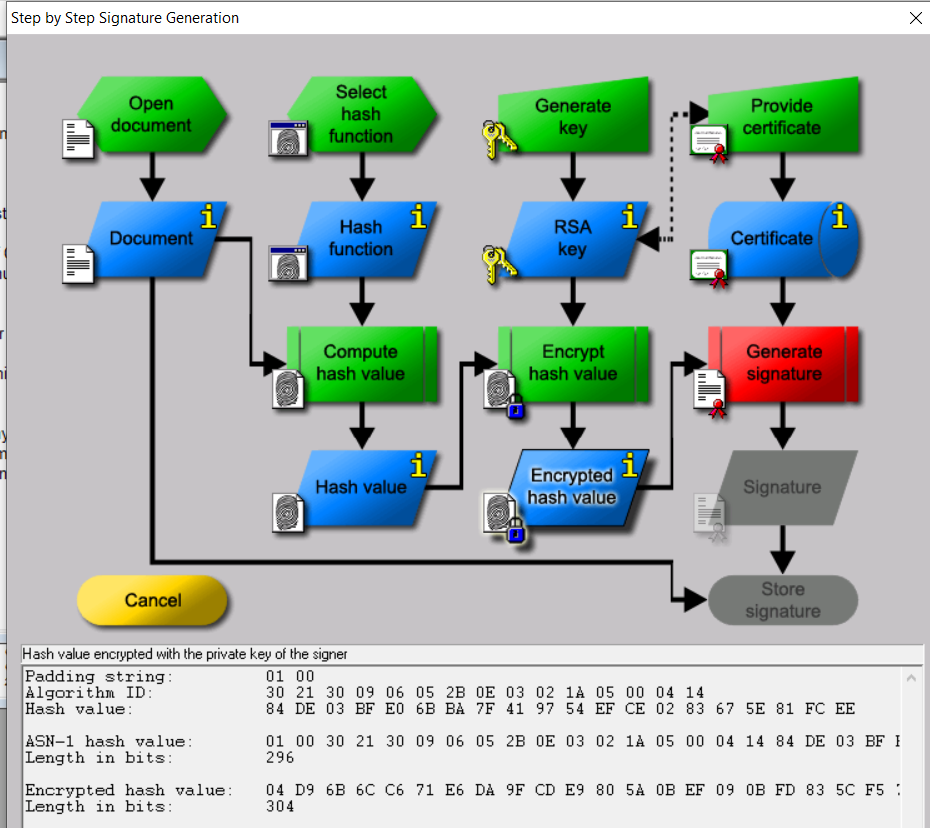


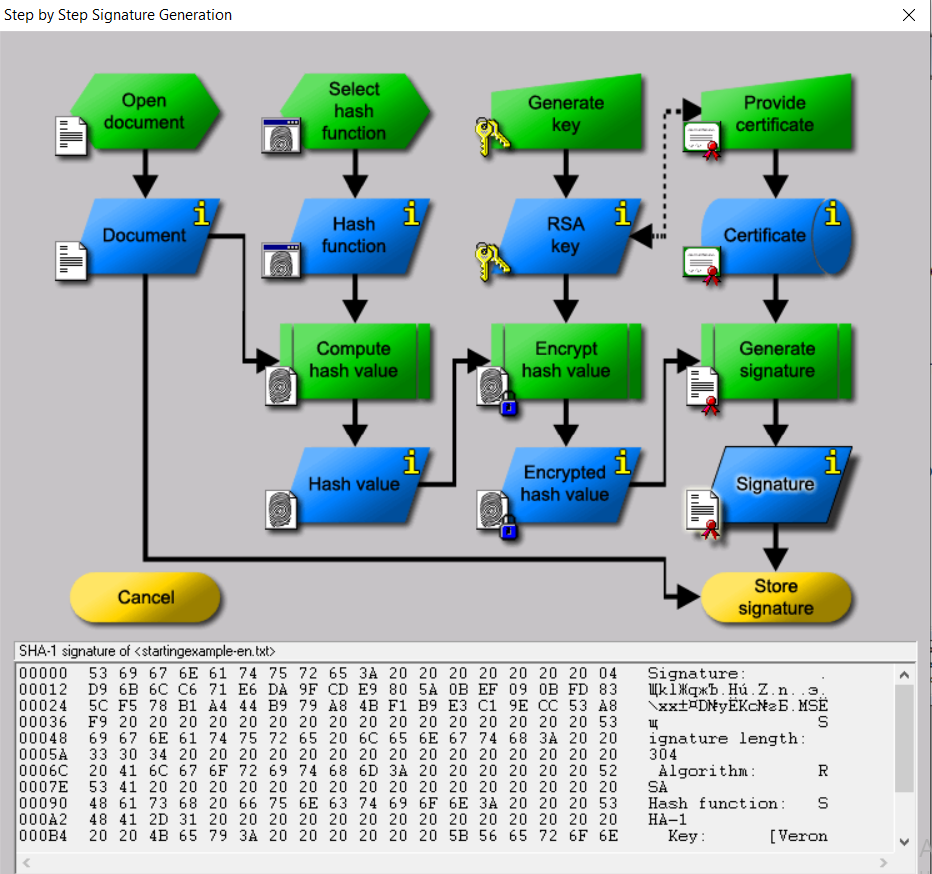












1. Запустить демонстрационную утилиту *«Digital Signatures/PKI-> Signature Demonstration…».*
2. Получите сертификат на ранее сгенерированную ключевую пару RSA-2048.
3. Выполните и сохраните скриншоты всех этапов создания цифровой подписи документа.
4. Сохраните скриншот сертификата для проверки этой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

Полученный сертификат:

Version: 2 (X.509v3-1996)

SubjectName: CN=KONSTANTIN KIREEV [1639132615], DC=cryptool, DC=org

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

SerialNumber: 5B:09:F6:39:EC:87:5E:81

Validity - NotBefore: Fri Dec 10 13:36:57 2021 (211210103657Z)

NotAfter: Sat Dec 10 13:36:57 2022 (221210103657Z)

Public Key Fingerprint: 0E21 231E 6851 2334 E0D9 1C8D D340 3793

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

Public modulus (no. of bits = 2048):

0 FFE6368B 213DCED2 0BF8E3CA DDE6B26F

10 6B875FC4 171D249D 63782C9B 64DB6D3D

20 459DD547 E35F601C 9EC219E5 38F15F10

30 3ABEEBA9 879D133E BE23DE6C B7D96805

40 8B859988 3737E777 682F0F34 9A2FAAE1

50 D6FDA25A 2CE1660D C1984BAD EEDF4789

60 97C9FCB6 CC47DA79 4AC7619D DA96AE5C

70 C0A520CD 3792AC78 A05A7B78 2BF42856

80 C866825C 1E385C32 502C9279 0D4D681B

90 E55050AF 73A05732 DCA138F7 C4AD02CF

A0 F498C226 820A761F B6A22116 85968683

B0 70EFCC42 1CEE8EDF D1AD087B C9E1BE10

C0 52053344 9E2605F6 4D27D46A 2B79BB2E

D0 A77242DD 8D19E13F 2E0DC803 53A598A3

E0 34E95E63 16277224 A0D101A1 A159C3B1

F0 8FCB7077 CB65AC16 82EED6F5 EF6D637F

Public exponent (no. of bits = 17):

0 010001

Certificate extensions:

Private extensions:

OID 2.206.5.4.3.2:

PrintableString:

|[KIREEV][KONSTANTIN][RSA-2048][1|

|639132615] |

Signature: Algorithm sha1WithRSASignature (OID 1.3.14.3.2.29), NULL

0 24AA9B69 EA33EC48 703A13BD 5318DF2C

10 C5735BB7 80C964AF B5C99CAE 39590584

20 30F6BF48 58464F7A 00EA5956 A36D9A24

30 5167851B B162E1BE 38B5A851 6024DAF5

40 D7692CF7 2624A27F 0A88C84E F7B31938

50 D54882F8 30DCC8F3 13349042 733BAC64

60 303322F1 1F244273 C7286D51 9FA1CAE3

70 FA8DEA17 94CA5528 A9E9C8CC 5203664B

80 3274D9A3 6594E8C0 27D7977A 8683A8ED

90 443F3C8F AB08A45F 73800016 880835B0

A0 40806BDC 7360F903 456D8964 9D1BDABB

B0 6E8B06A8 A863CD14 C49A31CF 04BFFA68

C0 E89A7DC6 96E70E39 1A6F133C 6D69D4BD

D0 FF09F587 5BFE3DF7 3832CA4A 9BEBB7EA

E0 57775EE8 326AFAEE 5C06801D DD67500C

F0 48D95F39 72741718 0E01DD0D 0AB26DC4

Certificate Fingerprint (MD5): 42:77:0B:CF:0D:E8:39:72:C2:15:60:67:9D:63:60:C1

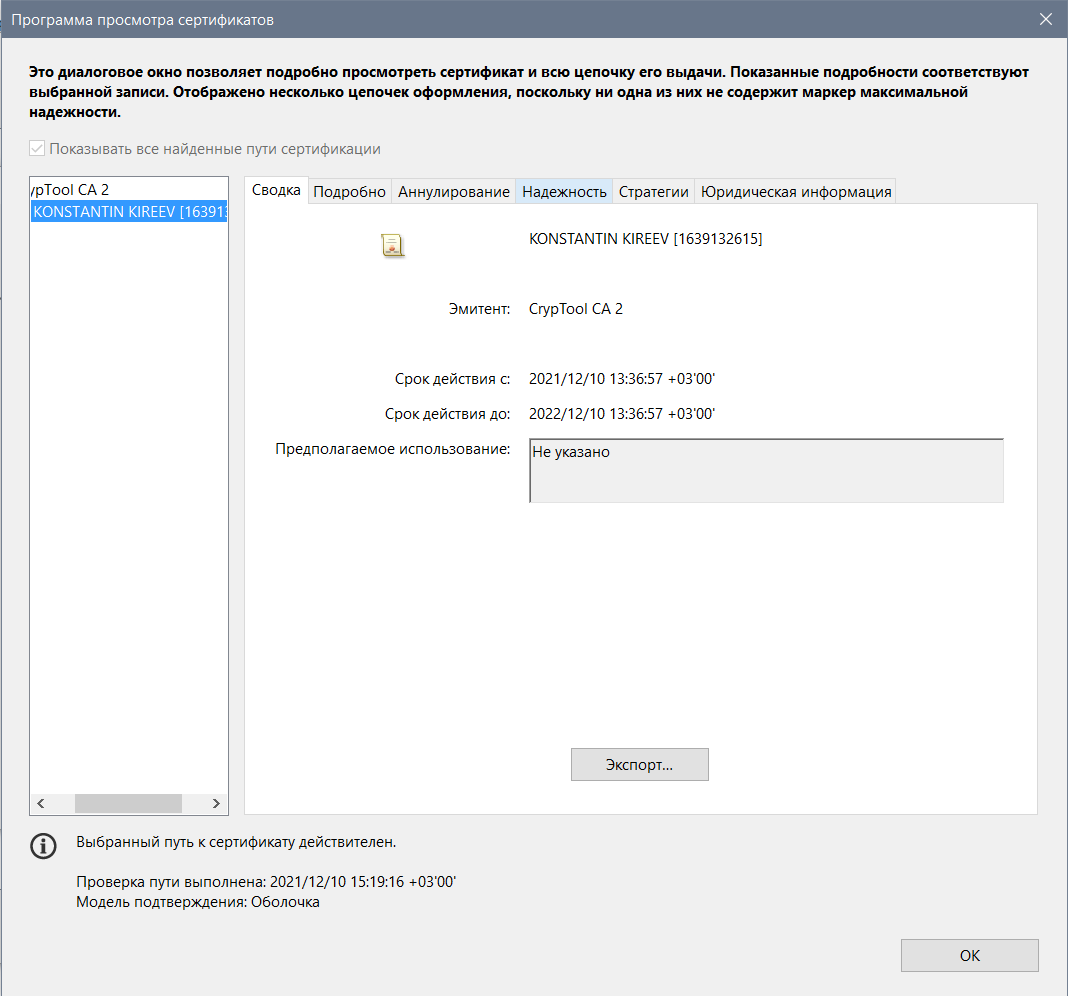
Certificate Fingerprint (SHA-1): 7BF5 391F C6C0 51D4 794A 49ED C3B5 E1CB 7D5A B1F4

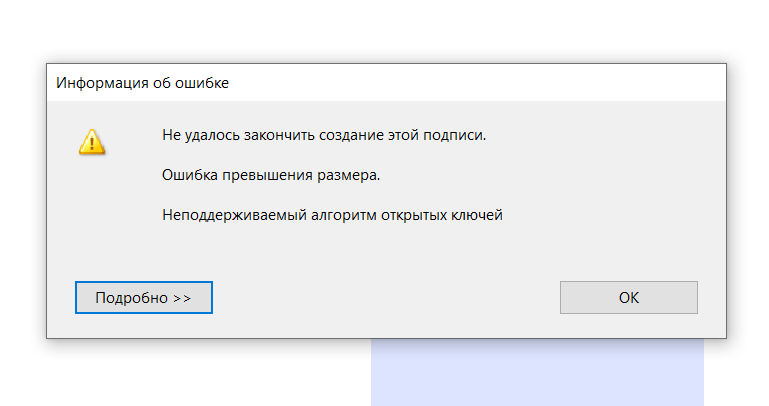
***Подписание своего отчета***

1. Сконвертируйте отчет в формат pdf.

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

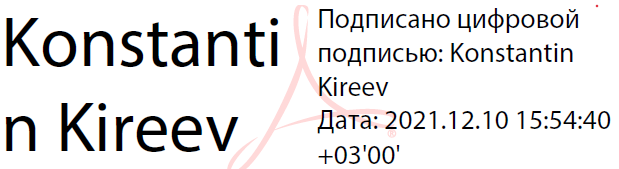
3. Откройте pdf-версию отчета и попытайтесь подписать с использованием этого сертификата.



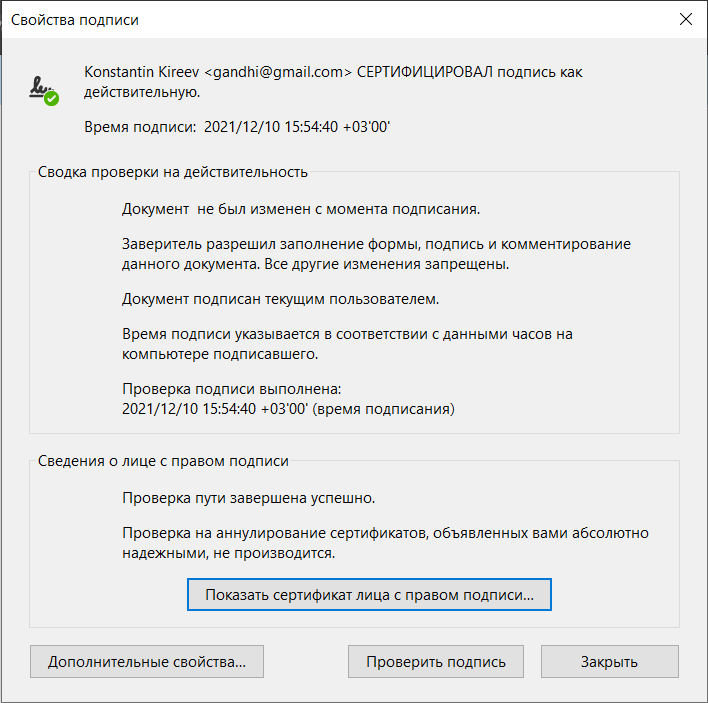


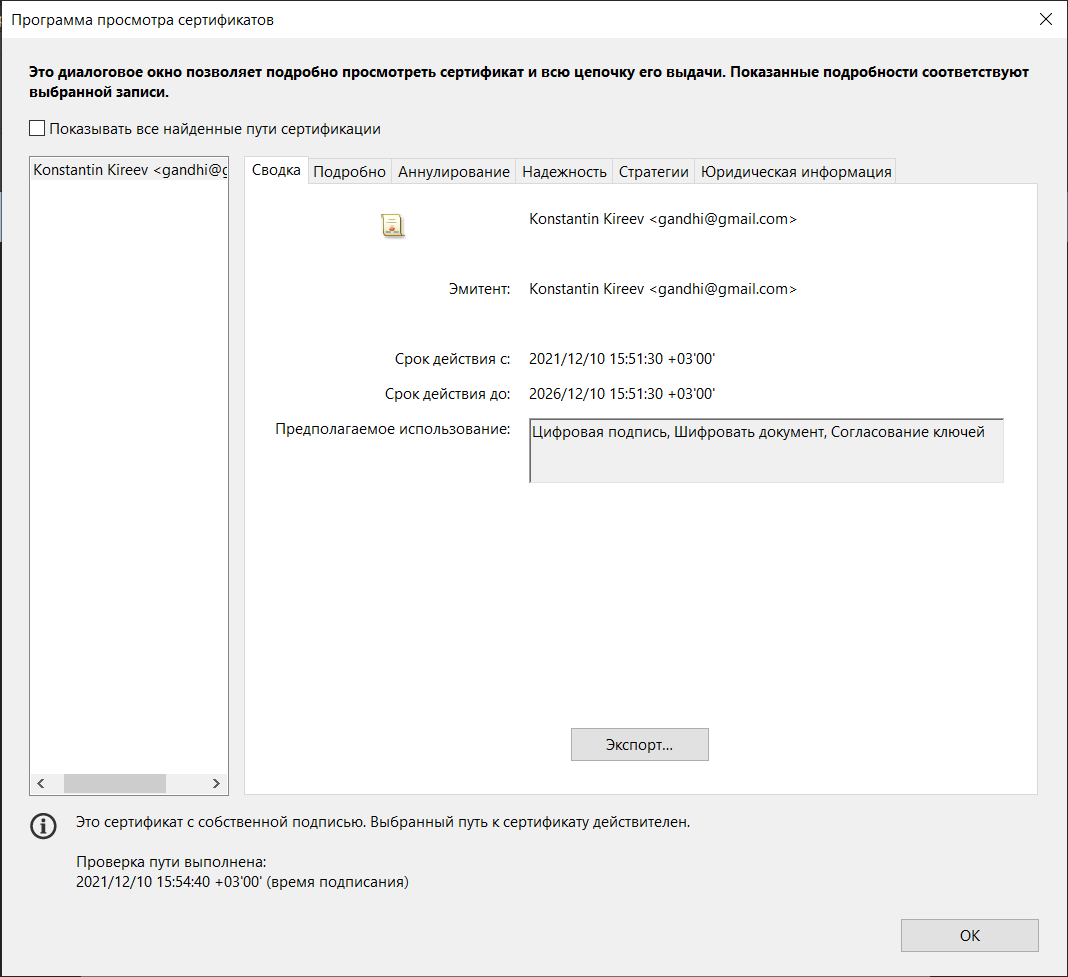
Подписать отчет сертификатом RSA не удалось.

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

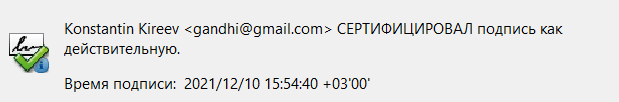


5. Сохраните скриншоты свойств подписи и сертификата.





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



**Выводы**

* 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 0.01 секунд.
* Изучен механизм создания цифровой подписи с различными ключами.

Лучше всего использовать ECDSA для создания и подтверждения подписи. 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.

* 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.

* Изучено создание сертификатов в среде PKI.

Сертификат — это электронный документ, который содержит: открытый ключ пользователя, информацию о пользователе, которому принадлежит сертификат, информацию о сроке действия сертификата, информацию об издателе сертификата и другие атрибуты, цифровую подпись удостоверяющего центра, выдавшего сертификат. Сертификат подтверждает электронную цифровую подпись и открытый ключ отправителя.

* Изучено создание подписи и проверка документа на целостность после внесения изменений средствами Adobe Acrobat Reader.