**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 4383 |  | Гордеева Т.В. |
| Преподаватель |  | Племянников А.К. |

Санкт-Петербург

2017

**Цель работы:** исследовать алгоритмы создания и проверки цифровой подписи, алгоритмы генерации ключевых пар RSA, DSA, ECDSA и получить практические навыки работы с ними, в том числе и в программном продукте CrypTool 1.

**1 Генераторов ключевых пар**

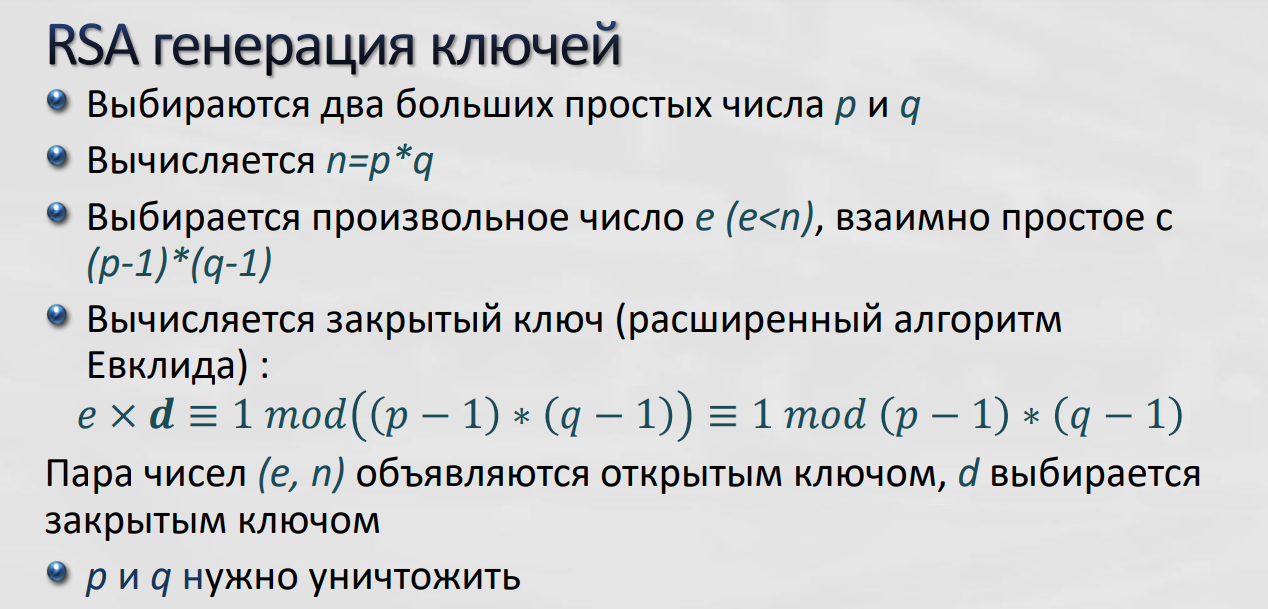
**1.1 Задание**

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.

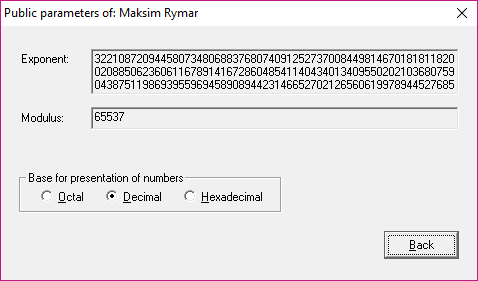
3. С помощью утилиты «Digital Signatures/PKI-> PKI/Display…» вывести сгенерированный открытый ключ и сохранить соответствующий скриншот.

**1.2 Описание алгоритмов генерации (как в лекции).**

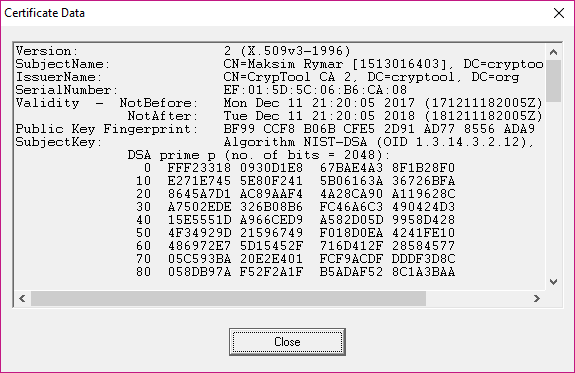
****

**1.3 Таблица с фактическими временем генерации ключевых пар.**

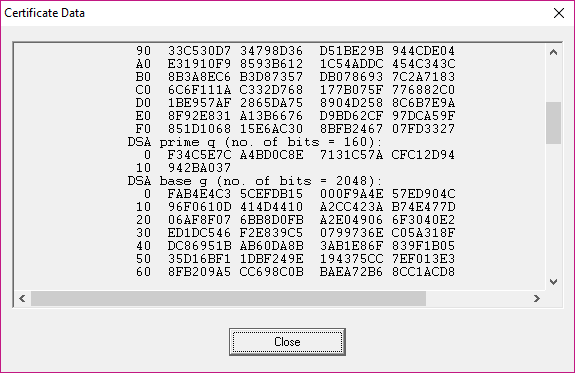
Перейдем к утилите «Digital Signatures/PKI->PKI/Generate…». Сгенерируем ключевые пары по алгоритмам RSA-2048, DSA2048, EC-239. Зафиксируем время генерации в таблице.



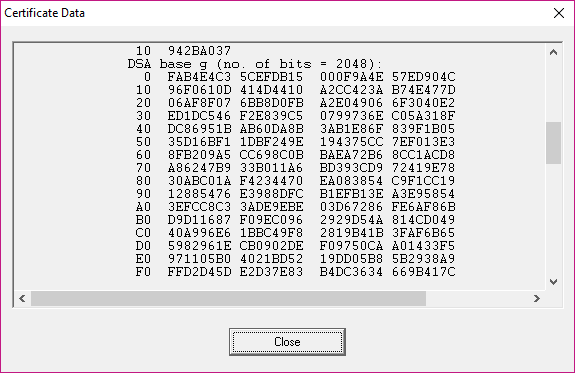
Пара RSA-2048



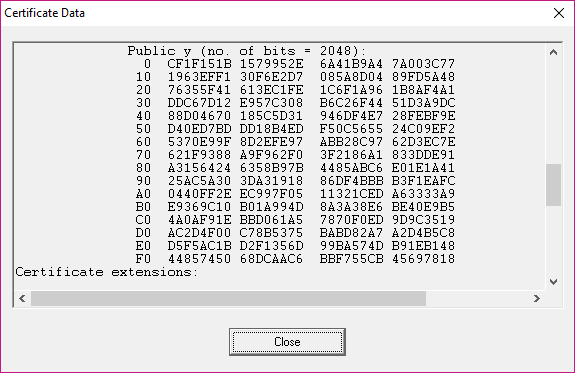
Открытая часть (параметр p) DSA-2048



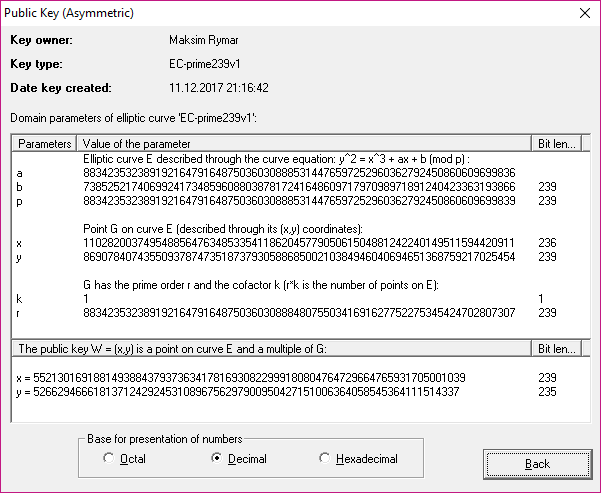
Открытая часть (параметр q) DSA-2048



Открытая часть (параметр ) DSA-2048



Открытая часть (параметр ) DSA-2048



Открытая часть EC-239

|  |  |  |  |
| --- | --- | --- | --- |
| **Алгоритм** | **RSA-2048** | **DSA-2048** | **EC-239** |
| **Время генерации ключа** | 1,142с | 1,717c | 0,076с |

Таблица сравнения времени генерации ключей

Из таблицы можно сделать вывод, что быстрее всего сгенерировался ключ алгоритмом EC-239.

**2 Процессы создания и проверки цифровой подписи**

**2.1 Задание**

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

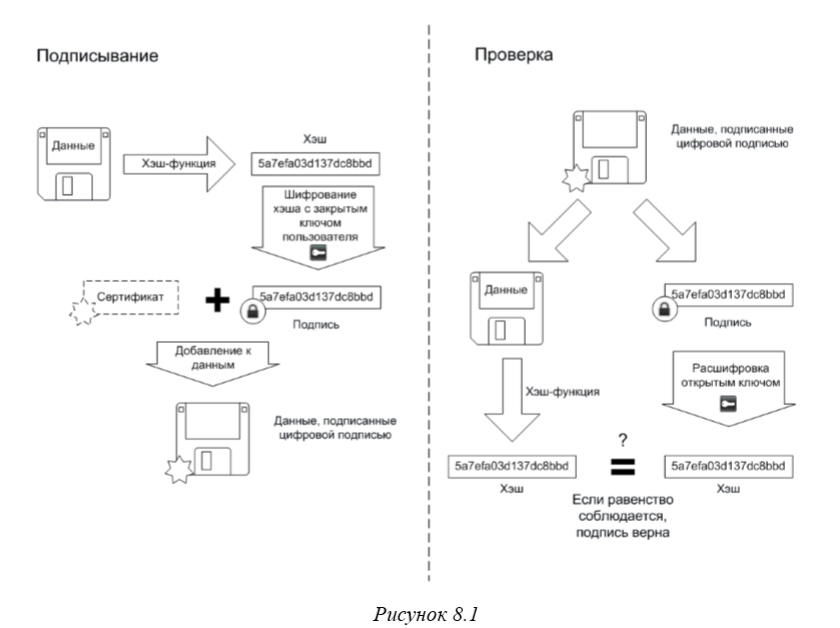
2. Задайте хэш-функцию, и другие параметры цифровой подписи.

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.

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



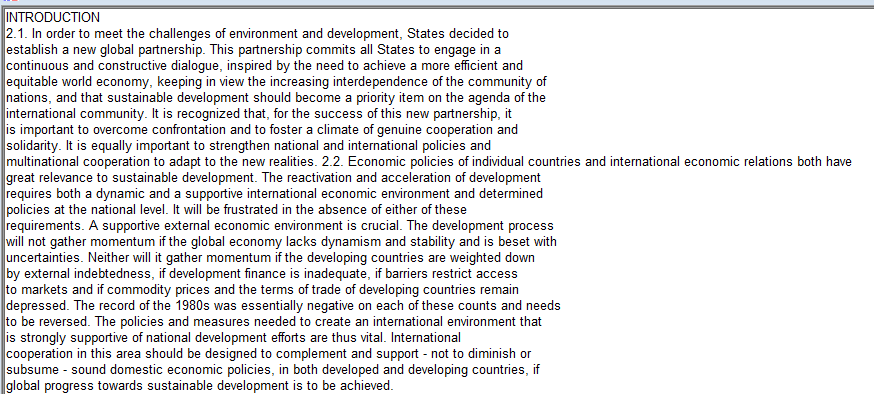
**2.3 Таблица с фактическими временем генерации цифровой подписи.**

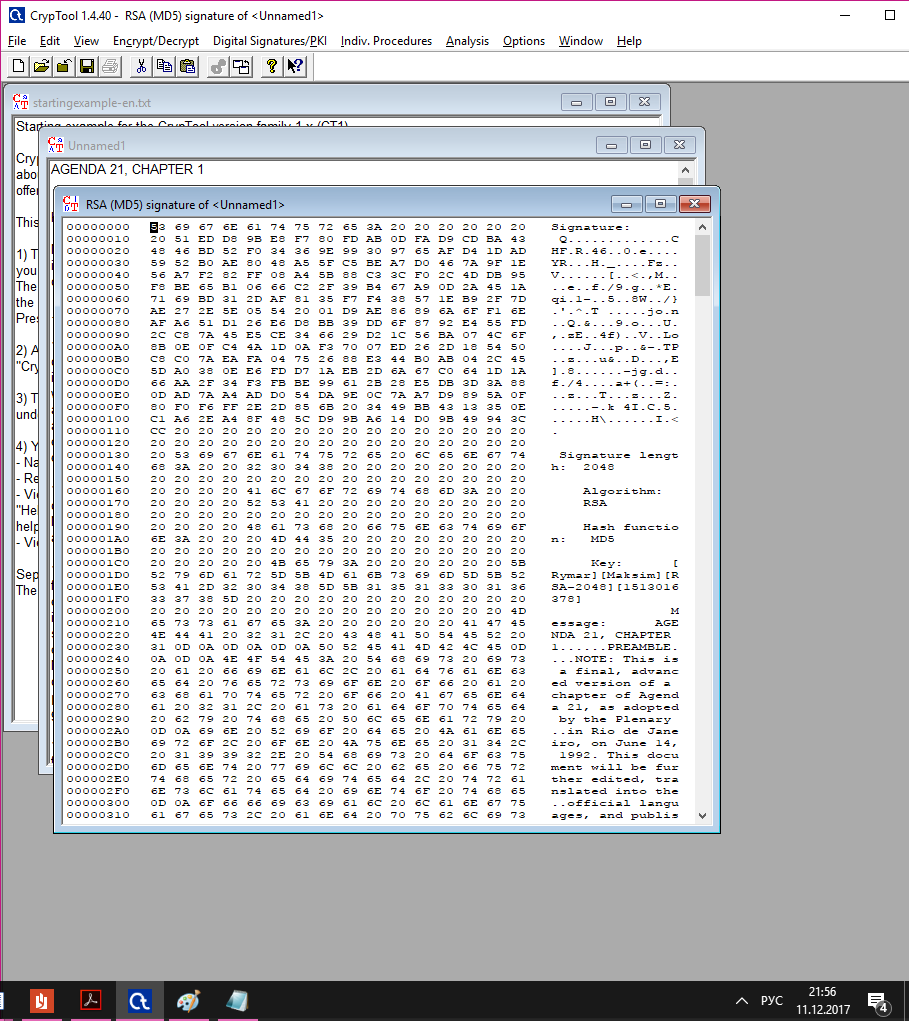
Откроем текст не менее 5000 знаков. Перейдем к приложению Digital Signatures/PKI-> Sign Document…

Зададим хэш-функцию, и другие параметры цифровой подписи.

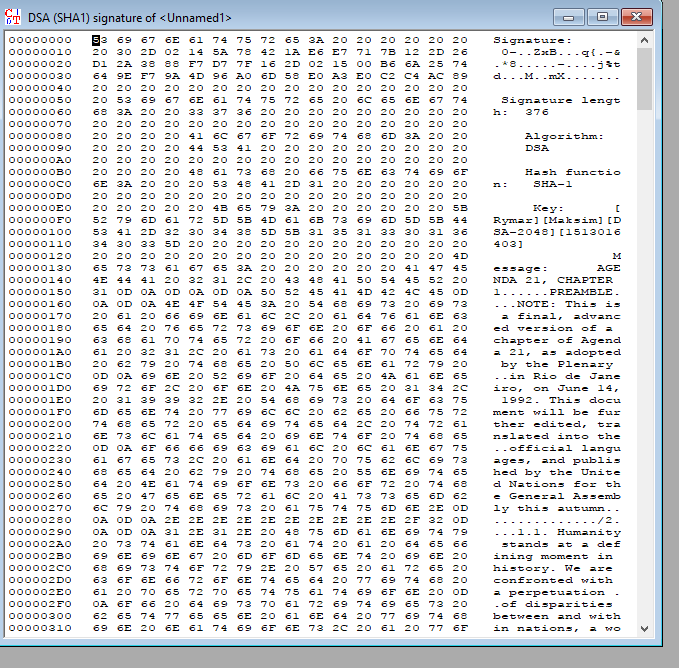
Создадим подпись ключами, сгенерированными в предыдущем задании. Зафиксируем время создания цифровой подписи для каждого ключа.

Исходный текст:

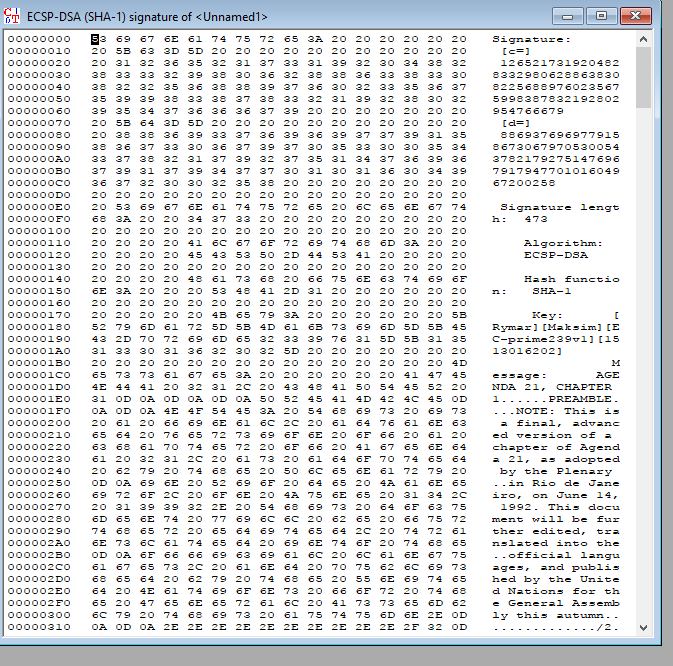
****



RSA with MD5



DSA with SHA-1



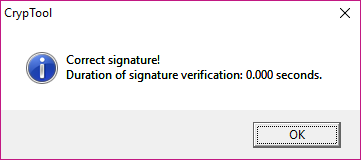
EC-293 with SHA1

|  |  |  |  |
| --- | --- | --- | --- |
| **Алгоритм** | **RSA-2048** with **MD5** | **DSA-2048** with **SHA-1** | **EC-239** with **SHA-1** |
| **Время генерации подписи** | 0,010c | 0,002c | 0,032с |

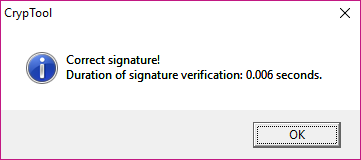
Быстрее всего был проведен процесс подписания парой DSA-2048 и SHA-1.

**2.4 Скриншоты с результатами проверки цифрой подписи.**

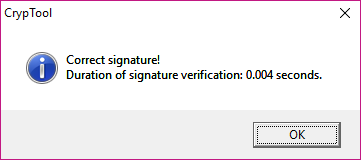
Процесс проверки подписи без искажений данных:



**RSA-2048** with **MD5**

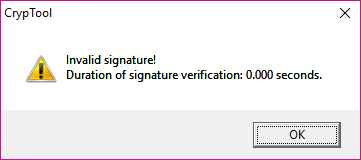


**DSA-2048** with **SHA-1**

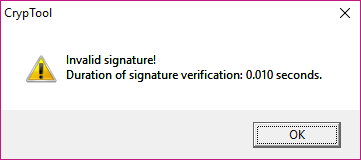


**EC-239** with **SHA-1**

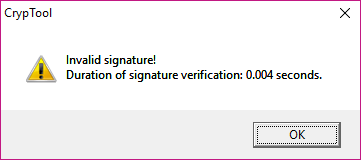
Процесс проверки подписи с искажением данных:



**RSA-2048** with **MD5**



**DSA-2048** with **SHA-1**



**EC-239** with **SHA-1**

Самое быстрое время проверки зафиксировано у подписи RSA-2048 with MD5.

**3 Процессы создания и проверки цифровой подписи**

**3.1 Задание**

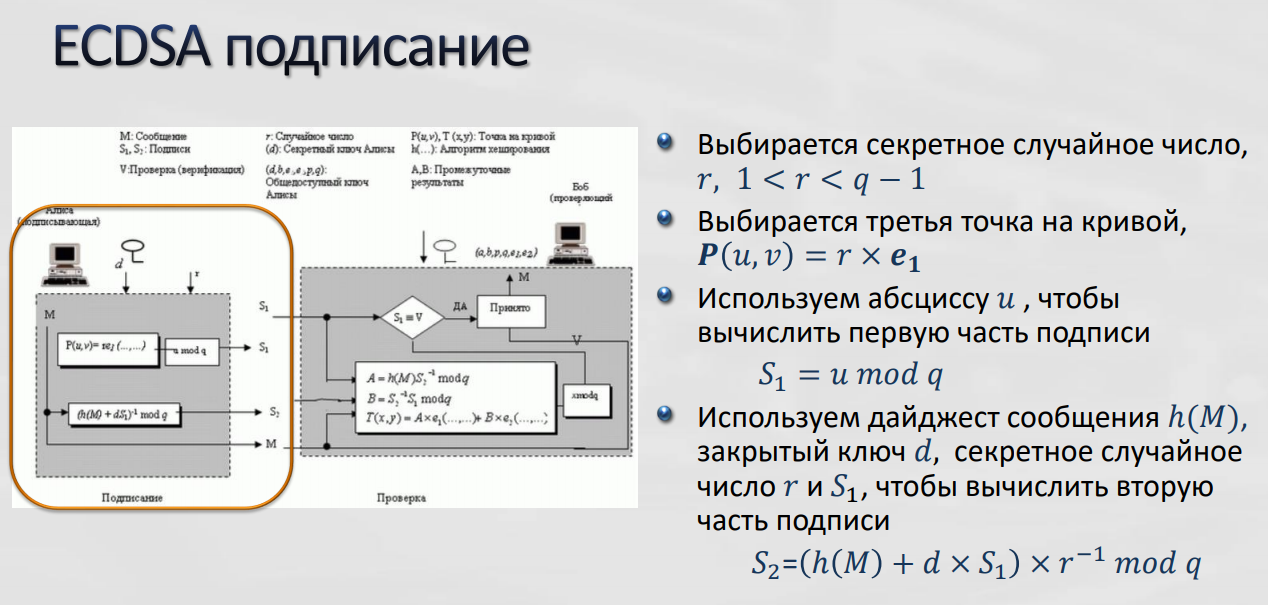
1. Выполните процедуру создание подписи «Digital Signatures/PKI->Sign Document…» алгоритмом ECSP-DSA в пошаговом режиме (Display inter. results=ON). Зафиксируйте скриншоты последовательности шагов.

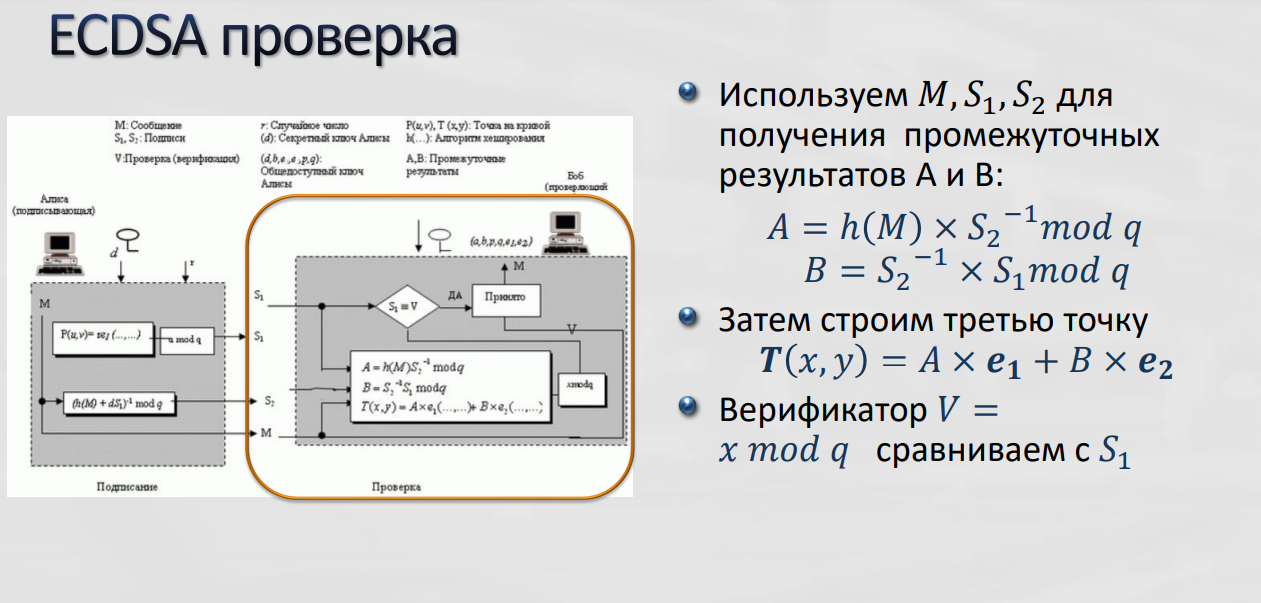
2. Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

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.

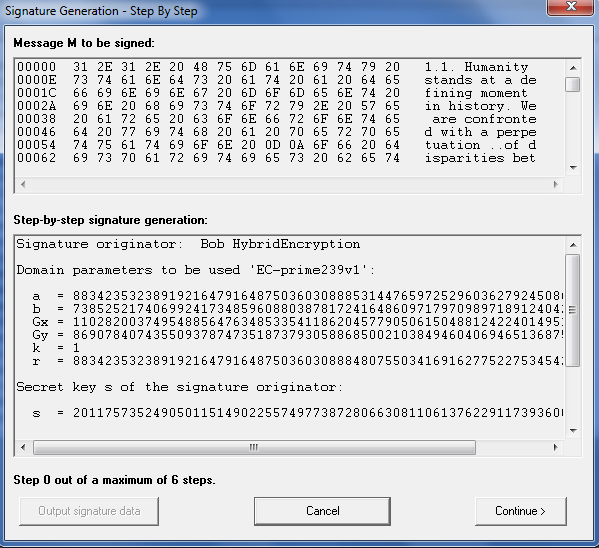
**3.2 Описание алгоритма формирования и проверки подписи ECDSA**

****

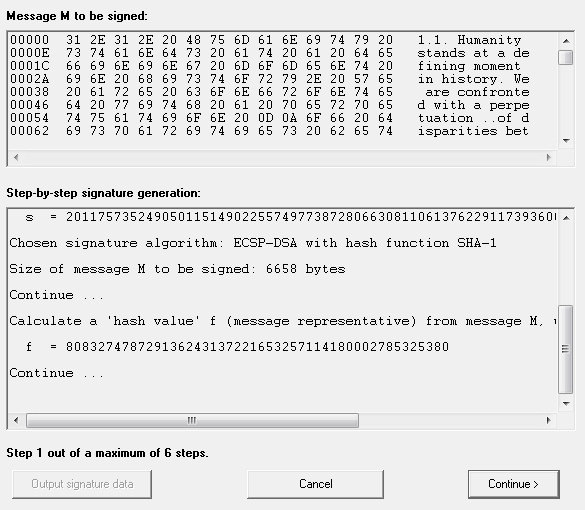
****

****

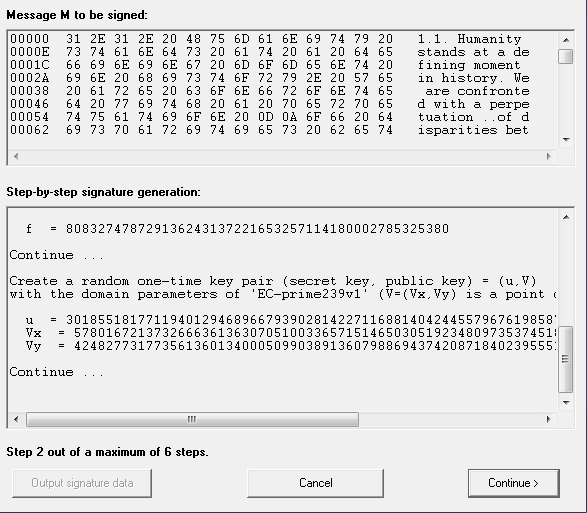
**3.3 Результаты (скриншоты) пошагового выполнения ECDSA в CrypTool 1. Сравнение лекционной версии и реализации.**



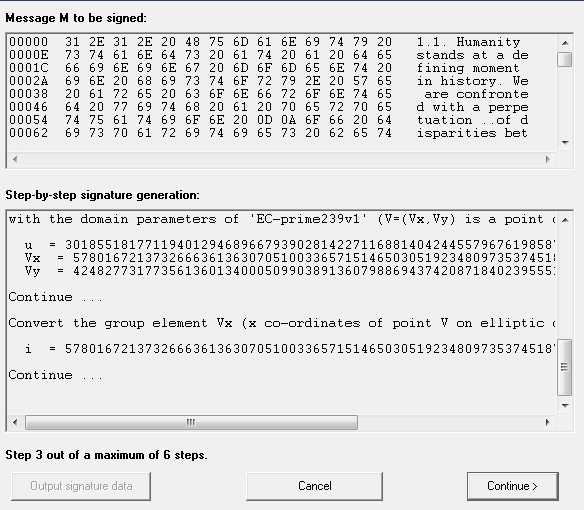
0 step



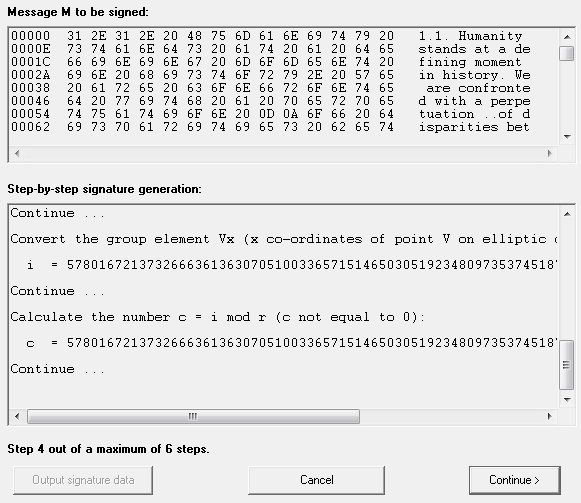
1 step



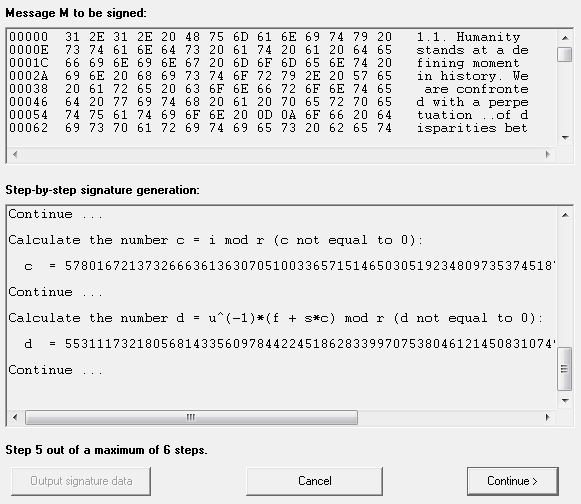
2 step



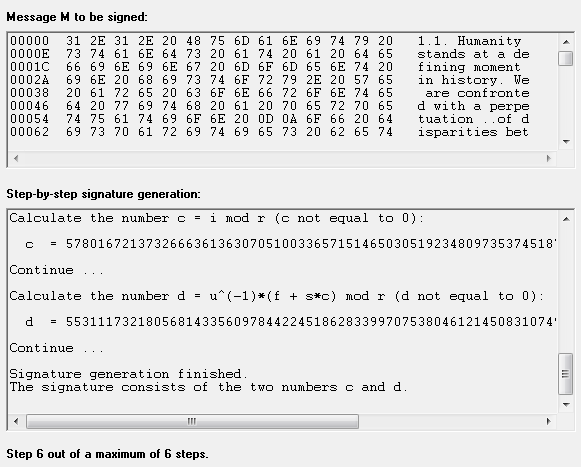
3 step



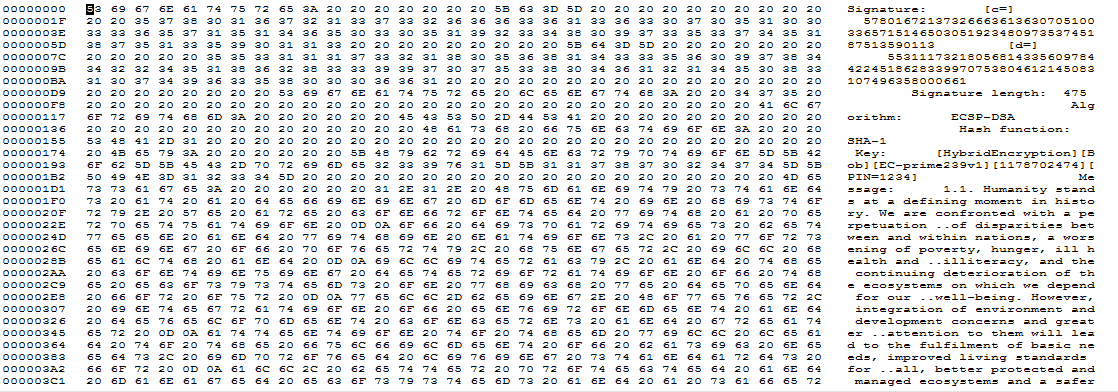
4 step



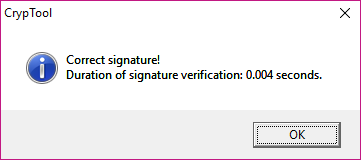
5 step



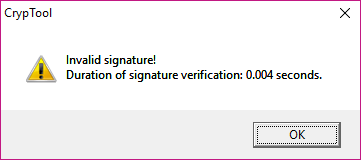
6 step



Результат подписания



Данные не были искажены

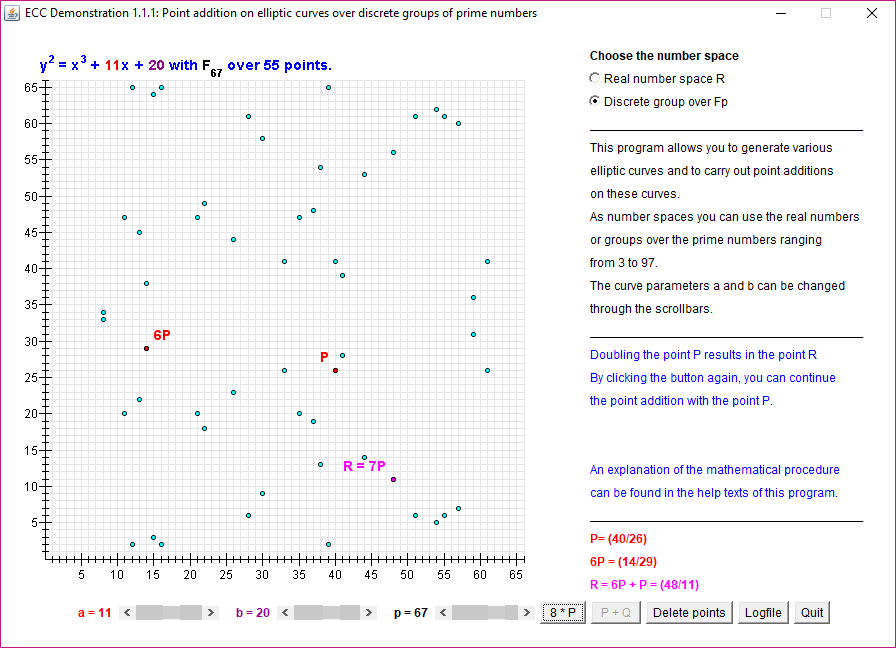


Данные были искажены

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cryptool 1** |  |  |  |  |  |  |  |  |  |  |  |
| **Лекция** |  |  |  |  |  |  |  |  |  |  |  |

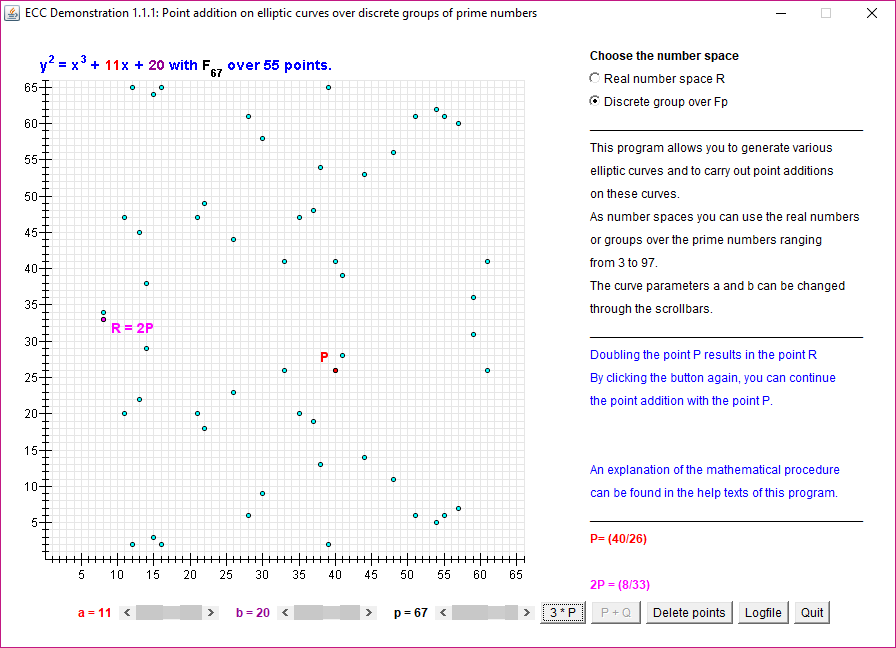
Таблица соответствия параметров с лекции и из Cryptool 1

**3.4 Проверка лекционного материала по эллиптическим кривым в демоприложении.**



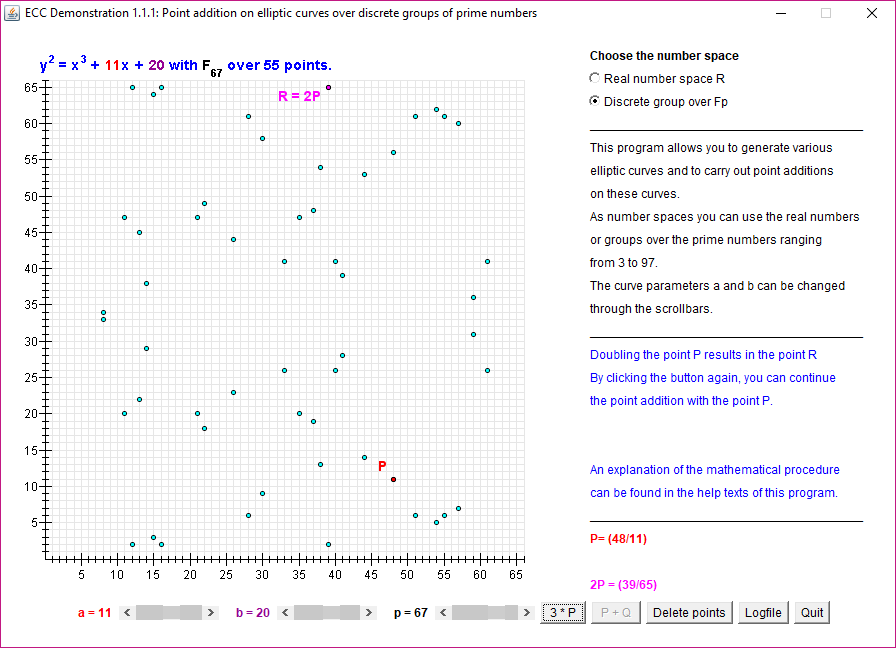
Генерация ключа

Здесь

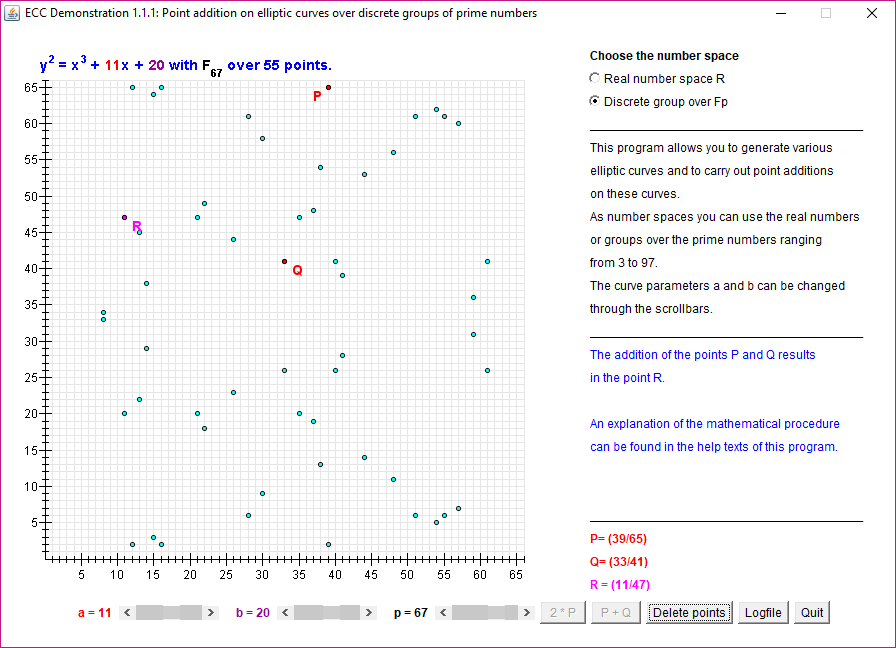


Зашифрование (1)

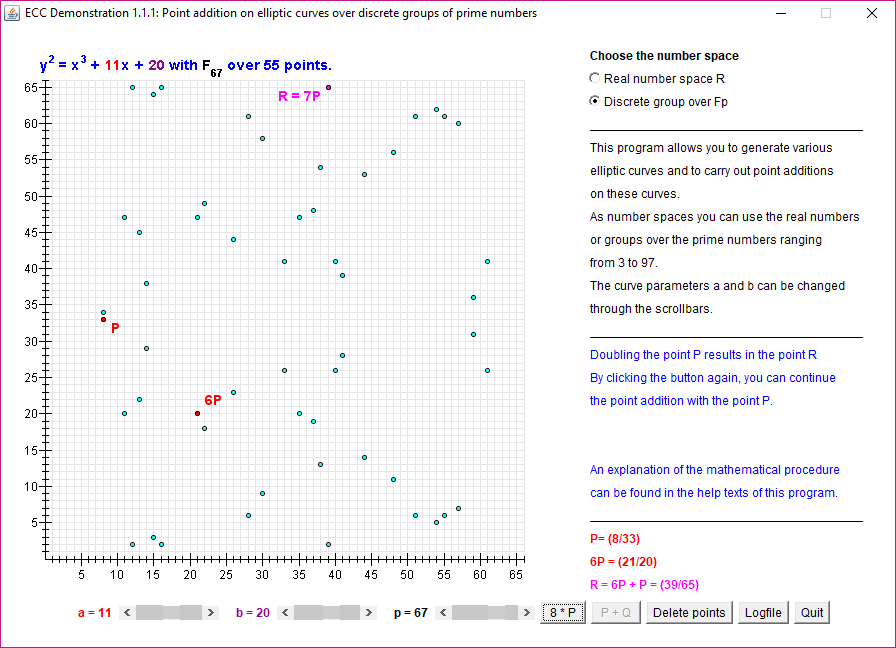
Точка P(33,41) – сообщение,



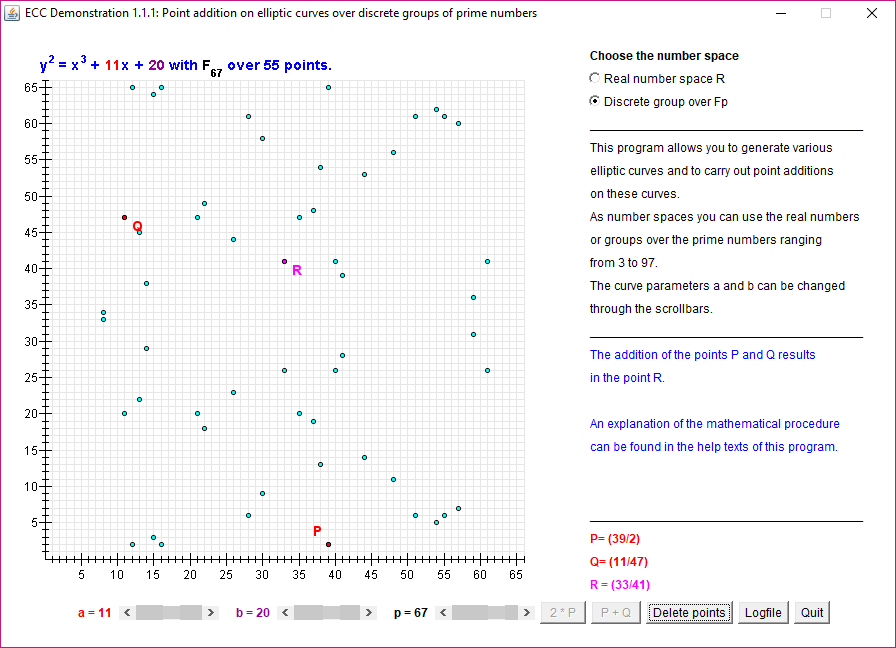
Вычисление



Зашифрование (2)



Вычисление



Окончательное расшифрование сообщения

Расшифрование удалось.

Пример подписания и проверки подписи:

Подписание:

Пусть h(M)=M=50, тогда :

Проверка:

𝐵 = 𝑆2-1 × 𝑆1 𝑚𝑜𝑑 𝑞 =7 × 18 mod 31=3

**4 Демонстрация процесса подписи в среде PKI**

**4.1 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.

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

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 криптозадач на основе закрытого и открытого ключей, 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

Одним из решений криптозадач на основе открытого и закрытого ключей является использование сертификатов. Сертификат—это электронный документ, который содержит:

1. Открытый ключ пользователя (открытый ключ)

2. Информацию о пользователе, которому принадлежит сертификат

3. Информацию о сроке действия сертификата

4. Другие атрибуты

5. Цифровую подпись удостоверяющего центра, выдавшего сертификат

Существует несколько вариантов использования сертификатов электронной подписи:

1. Для зашифрования и расшифрования электронных документов

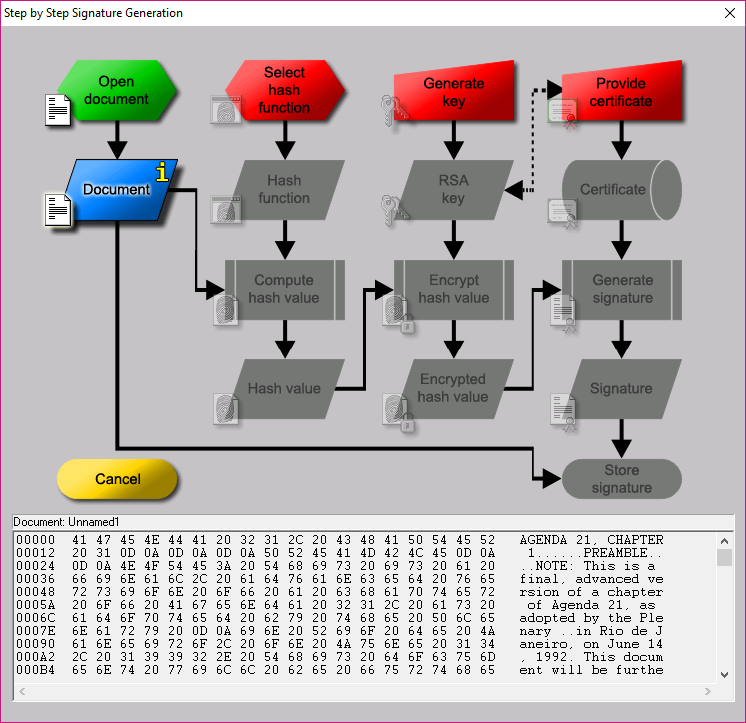
2. Для подписания электронного документа и проверки подписи

3. Для подписания и шифрования электронных документов.

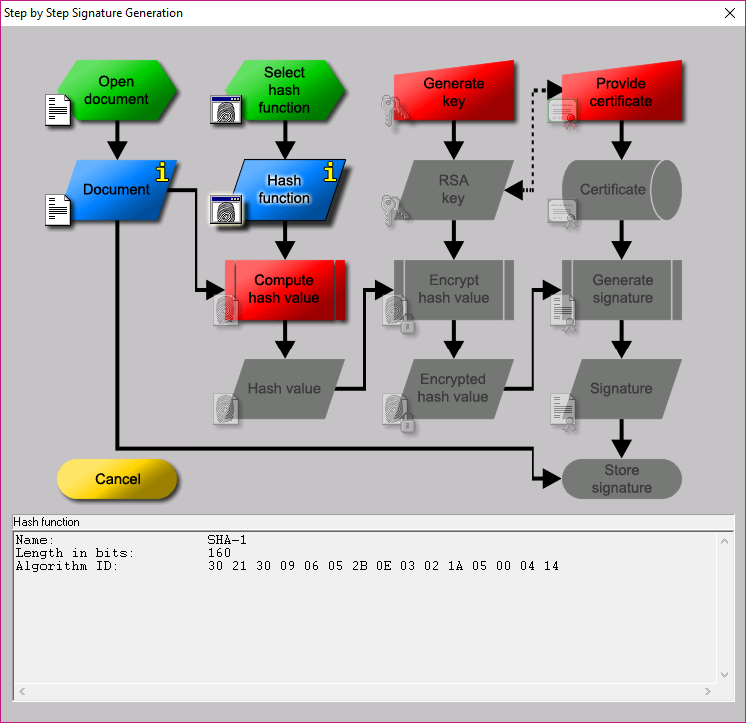


Структура сертификата из лекций

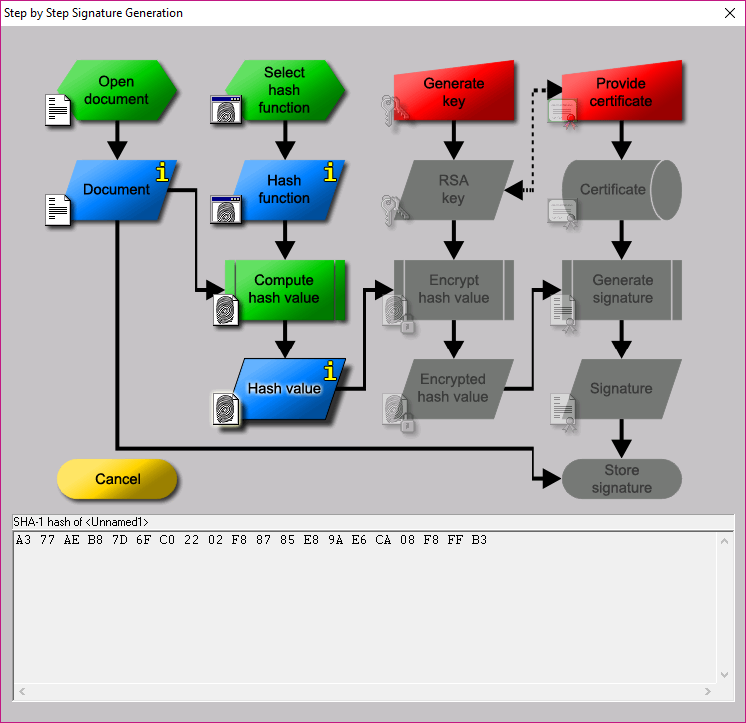
**4.3 Схема процедуры подписания из CrypTool.**



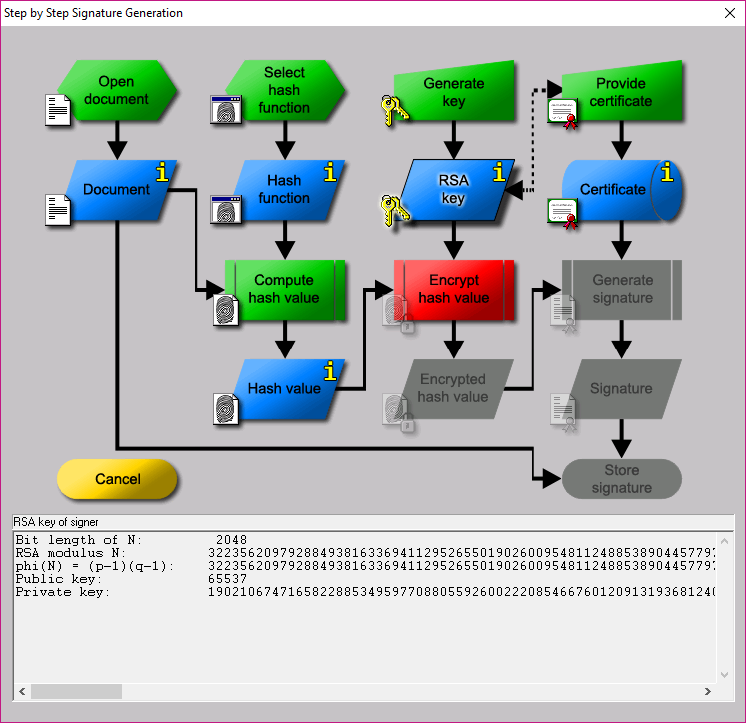
Исходный текст



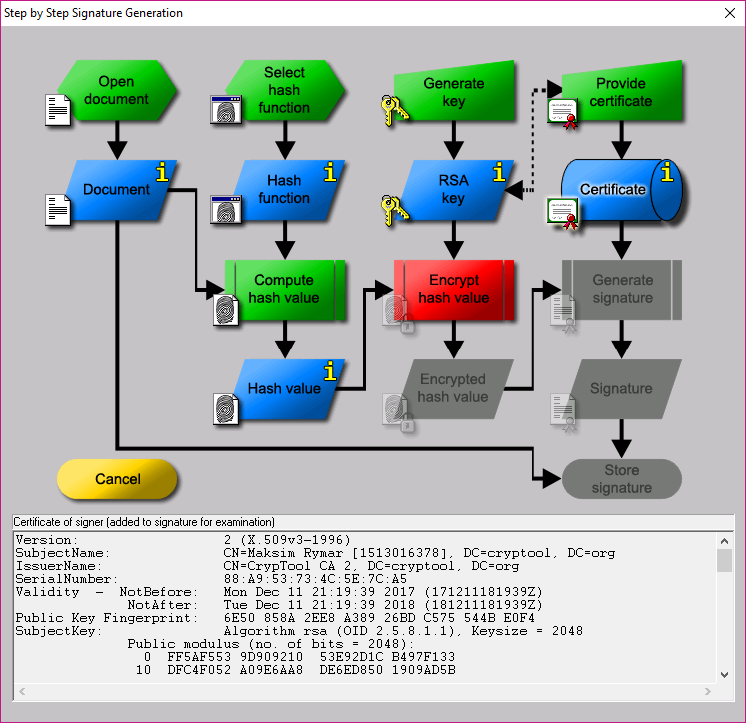
Хеш-функция



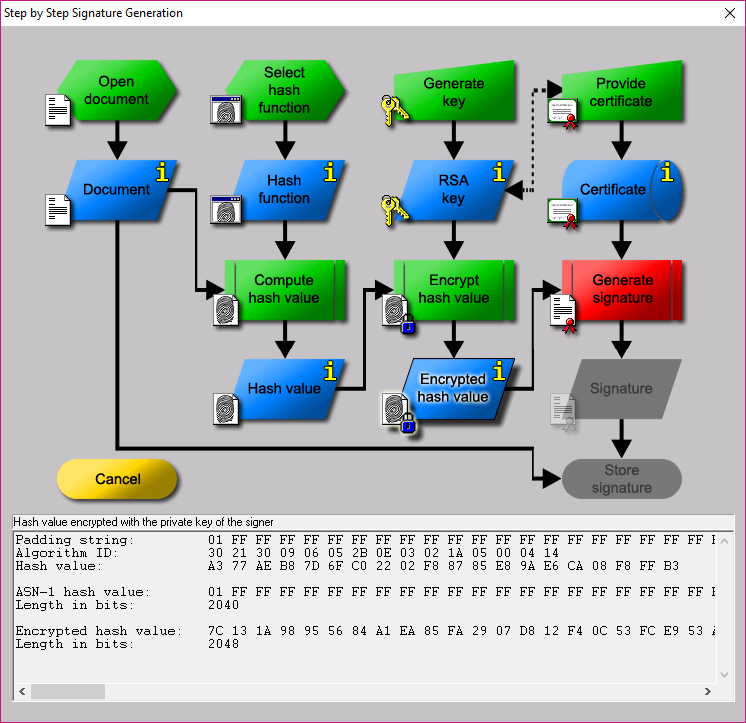
Хеш-значение текста



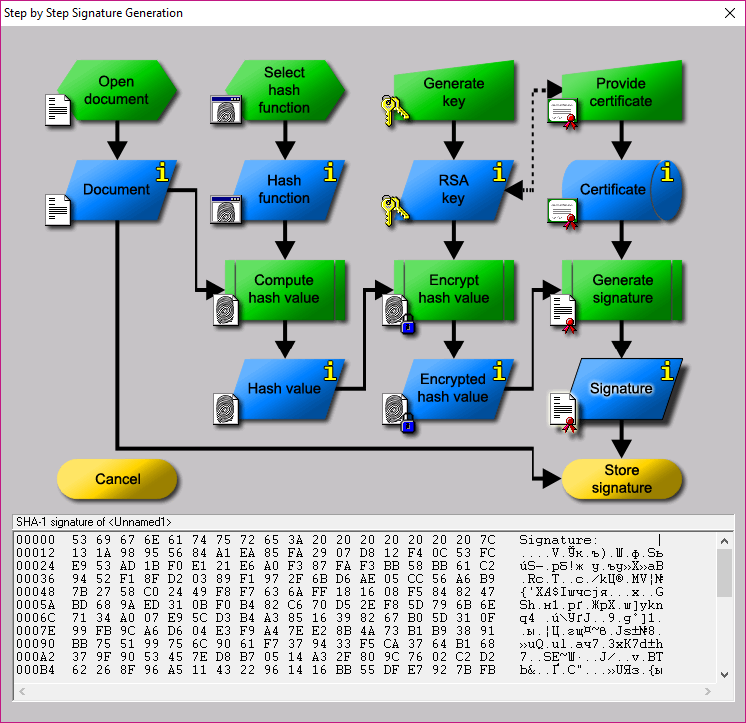
RSA-ключ



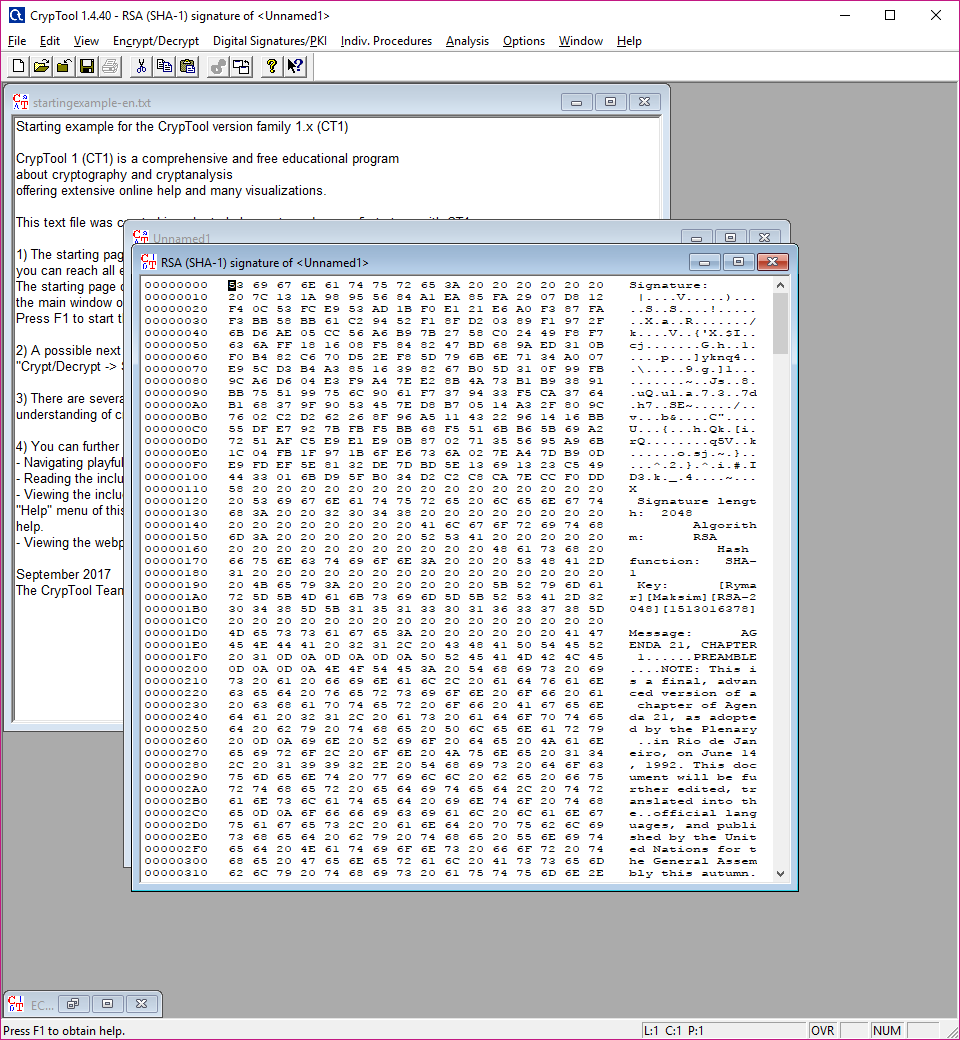
Сертификат



Зашифрованный хеш



Подписанный документ



Результат подписания

|  |
| --- |
| Version: 2 (X.509v3-1996)  SubjectName: CN=Tatsiana Gordeeva [1513016378], DC=cryptool, DC=org  IssuerName: CN=CrypTool CA 2, DC=cryptool, DC=org  SerialNumber: 88:A9:53:73:4C:5E:7C:A5  Validity - NotBefore: Mon Dec 11 21:19:39 2017 (171211181939Z)  NotAfter: Tue Dec 11 21:19:39 2018 (181211181939Z)  Public Key Fingerprint: 6E50 858A 2EE8 A389 26BD C575 544B E0F4  SubjectKey: Algorithm rsa (OID 2.5.8.1.1), Keysize = 2048  Public modulus (no. of bits = 2048):  0 FF5AF553 9D909210 53E92D1C B497F133  10 DFC4F052 A09E6AA8 DE6ED850 1909AD5B  20 493779B6 FA8B9E99 FB2940ED 16404057  30 005CF394 96B1EC3D E7302284 80DD7FF9  40 AA9D72EE B3B9B62E 47692FAB DA4B03D0  50 8062D942 2CEA2498 2076B4AD 94601AF6  60 E2C21587 2CA64443 FACFF8B2 2BB35176  70 81E360F6 EBBEA8E3 E2C13BA5 4988B36D  80 34B30925 A669A6F5 719FBFAB E18E5263  90 726B1F41 B37E5A5A 1D1C9121 293826F6  A0 9911D550 2BA52D0A 843D0F46 C701A146  B0 DCAD98FC C2586716 50D1414E CB2198C3  C0 40B12DFB F855D13E 9C03BD56 5A9E27CE  D0 9ACE7A8C 8DCA0F16 9F230E75 1D4B2612  E0 BAC2F72E B30DF368 9DAEA503 8D316DD7  F0 18044A15 FE02DA10 06444AE4 8FE82367  Public exponent (no. of bits = 17):  0 010001  Certificate extensions:  Private extensions:  OID 2.206.5.4.3.2:  PrintableString:  |[Gordeeva][Tatsiana][RSA-2048][151301|  |6378] |    SHA1 digest of DER code of ToBeSigned:  0 85072C42 34A669B8 3DB86F66 1542CE8C  10 618F86D8  Signature: Algorithm sha1WithRSASignature (OID 1.3.14.3.2.29), NULL  0 C6F3E5E3 AB766C9D 2FC10553 E4DF836E  10 B409AAA0 34D582E2 9BE40C8D 35C6C084  20 5DA005C5 822F9F98 4A0EC400 64FC5BF4  30 DC5916AA E79D9AFC FA837D21 25C4EBEA  40 2F0B7945 8ABF135E 644702A4 B3730A67  50 2EBF47BE 7FA096C0 398123BF 0DC0DFBA  60 36DE7127 B2EC20C8 E898763E B9281EA0  70 D2332BC3 8616B769 0B800C86 BA2D544D  80 AF773202 A6BCF496 AD07D983 F32B85C8  90 F812B1A0 71240738 5F960FBF 25093737  A0 22C73243 340B1512 0368EC6C 25C239FC  B0 5DDC42AC 213851A3 6C2B62D6 DED002FC  C0 C6ABA9D0 823CB966 819D7FE0 9FB0DABD  D0 3431437D 08DB9460 1AC72D30 1FCF1536  E0 EF66B882 5F95EAFD 0F931E45 0F84EC7C  F0 81E7A254 76D6CF0F 3D36EFD4 406D0E1D  Certificate Fingerprint (MD5): 88:00:80:2D:9B:CC:66:A2:9B:F4:CE:EC:34:AC:B2:40  Certificate Fingerprint (SHA-1): E4F5 9D1A 5561 E496 E2C9 3350 BB43 DF32 6221 C8EB |

Сертификат из Cryptool 1

Версия их Cryptool не содержит уникального идентификатора издателя, а в остальном соответствует 3 версии из лекции.

**4 Демонстрация процесса подписи в среде PKI**

**4.1 Задание**

1. Сконвертируйте отчет в формат pdf.

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

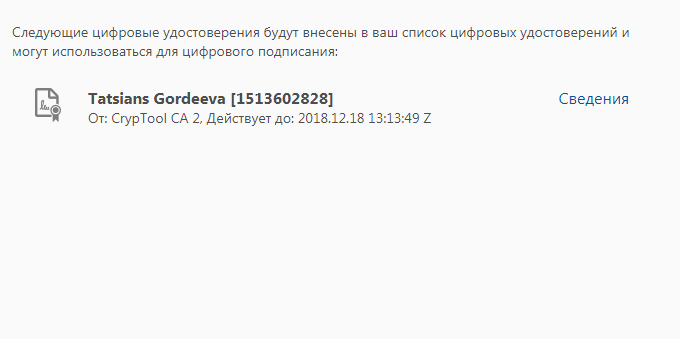
3. Откройте pdf-версию отчета и попытайтесь подписать с использованием этого сертификата.

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

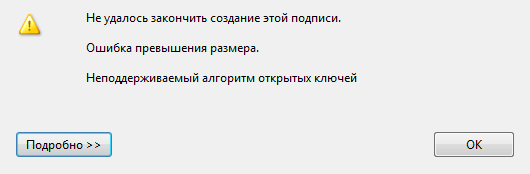
5. Сохраните скриншоты свойств подписи и сертификата.

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

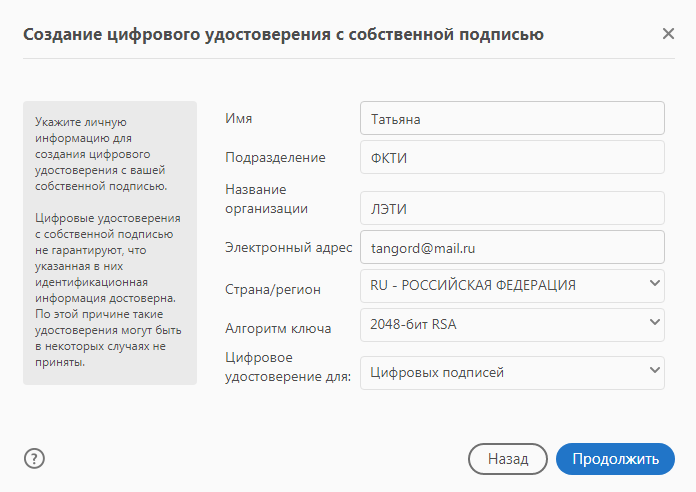
**4.2 Скриншот титульного листа с цифровой подписью.**



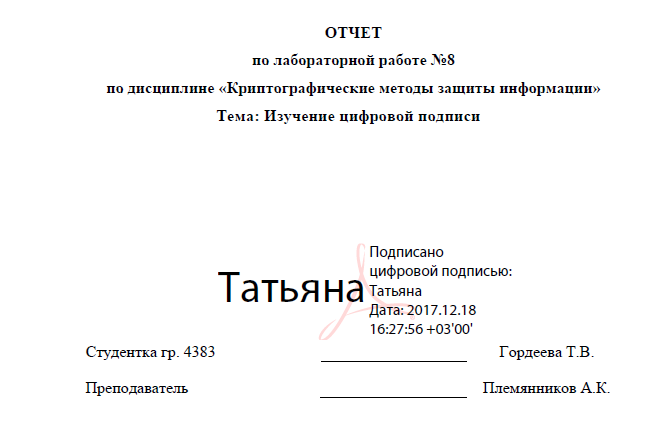
Попытка подписи сертификатом Cryptool



Ошибка в создании подписи на основе сертификата Cryptool

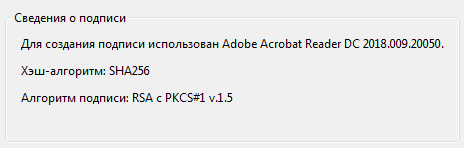


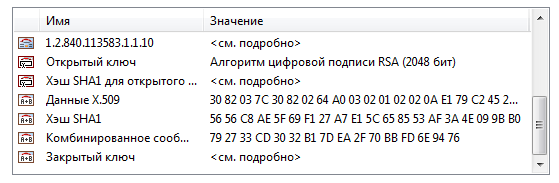
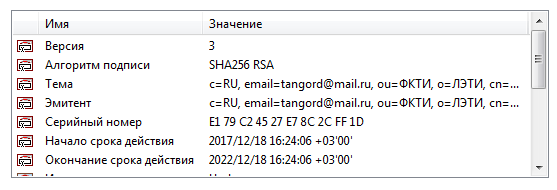
Создание цифровой подписи в Adobe Acrobat



Созданная цифровая подпись средствами Adobe

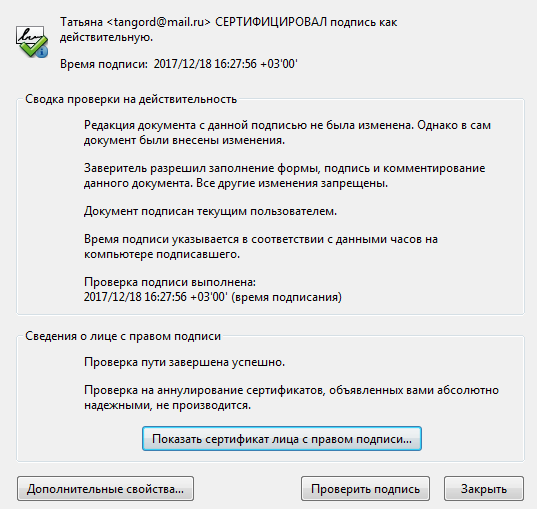
**4.3 Скриншоты свойств подписи и сертификата.**





Поля сертификата

**4.3 Скриншот результата проверки после внесения изменений в отчет.**



После изменения документа

**Вывод.**

* В ходе данной лабораторной работы были описаны алгоритмы генерации ключей для цифровой подписи. Также была приведена таблица с расчетами времени генерации ключей, из которой можно сделать вывод, что Наиболее быстрым оказался алгоритм **EC-239** – 0,076с.
* Также была изучена обобщенная схема создания и проверки цифровой подписи. Наиболее быстрым оказался алгоритм **DSA-2048** with **SHA-1** – 0,002с**.**
* Был изучен алгоритм формирования и проверки подписи ECDSA. При изучении алгоритма формировании подписи с помощью демоприложения параметры приложения были сопоставлены с параметрами из лекций. Также был рассмотрен пример шифрования и расшифрования с помощью эллиптических кривых.
* В данной лабораторной работе была изучена схема подписания в Cryptool на основе демоприложения. Документ был подписан при помощи сертификата Cryptool и раннее созданного сертификата на основе ключа RSA-2048. После сравнения структуры сертификатов Cryprtool и сертификата из лекции можно сделать вывод, что в сертификате Cryptool отсутствует уникальный идентификатор издателя.
* Также в ходе данной лабораторной работы были осуществлены две попытки подписания собственного отчета: сертификатом из Cryptoool и собственным сертификатом, созданным в Adobe. В первом случае была порождена ошибка размера данных и неиспользуемого алгоритма открытых ключей. Отчет был подписан с помощью собственного самоподписанного сертификата.