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

ОТЧЕТ

по лабораторной работе №8

по дисциплине «Криптография и защита информации»

Тема: Изучение цифровой подписи

Студент гр. 8383 _____ Ларин 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.

1. Генераторов ключевых пар

Генерация ключевых пар для алгоритма RSA

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

1. Вычисление n = p * q
2. Выбор произвольного e (e <n), взаимно простого с φ(n).
3. Вычисление d: e ∗ d = 1 mod φ(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) = 0 mod q.
3. Выбирается $e_1 : e_1^q = 1 \, mod \, p$.
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

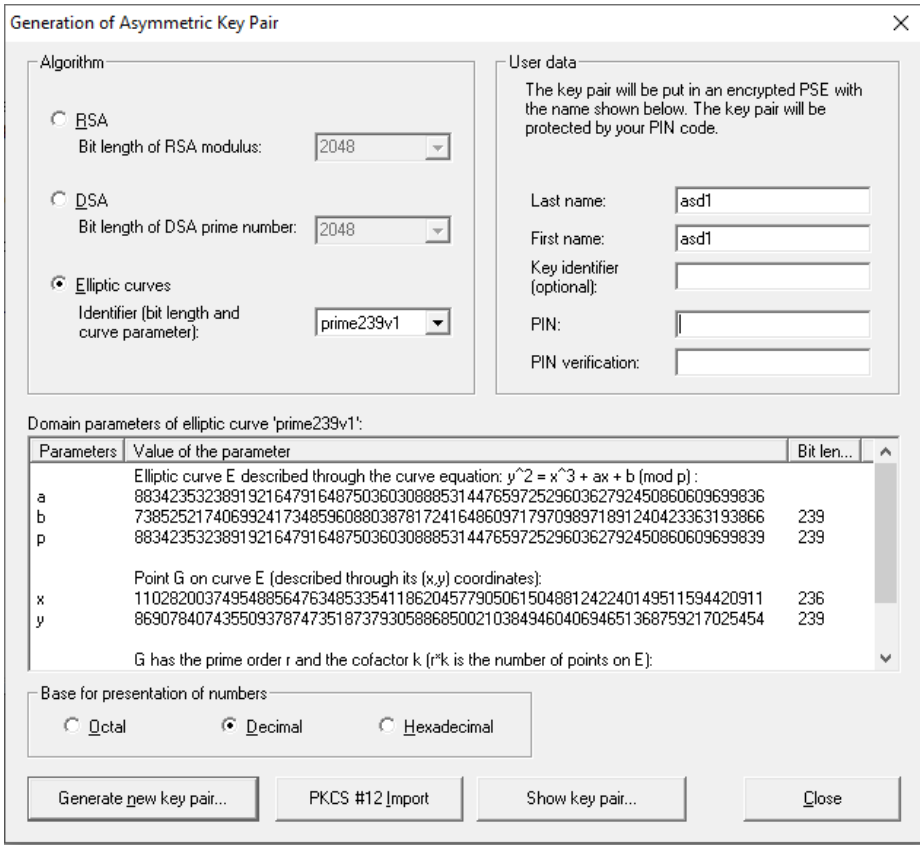
1. Выбирается эллиптическая кривая $E_p(a,b)$, p − простое число.
2. Выбирается точка на кривой $e_1 = (x_1, y_1)$
3. Выбирается простое число q – порядок одной из циклических подгрупп группы точек эллиптической кривой: $q \times (x_1, y_1) = O$.
4. Выбирается закрытый ключ d.
5. Вычисляется точка на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

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

Выполнение

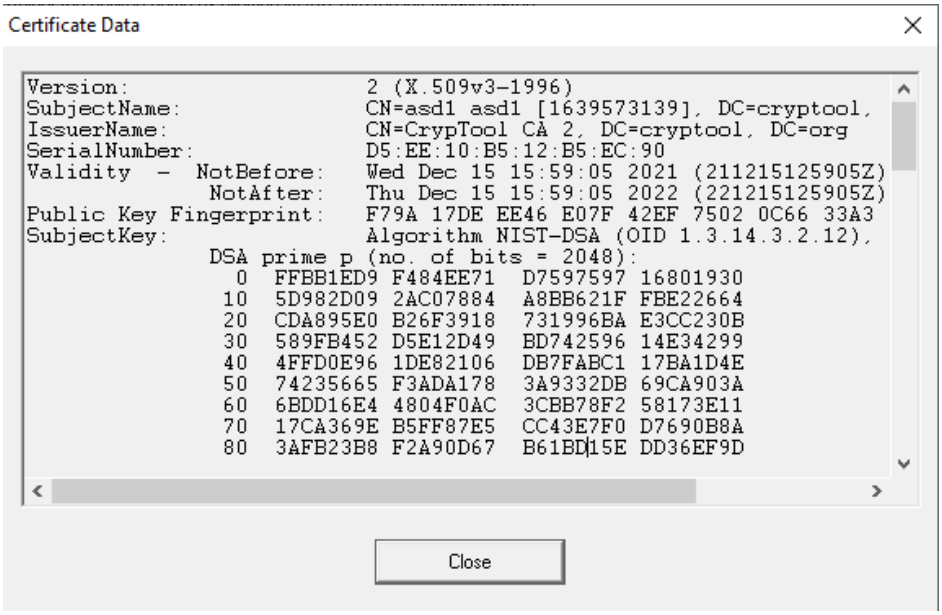
1. Перешли



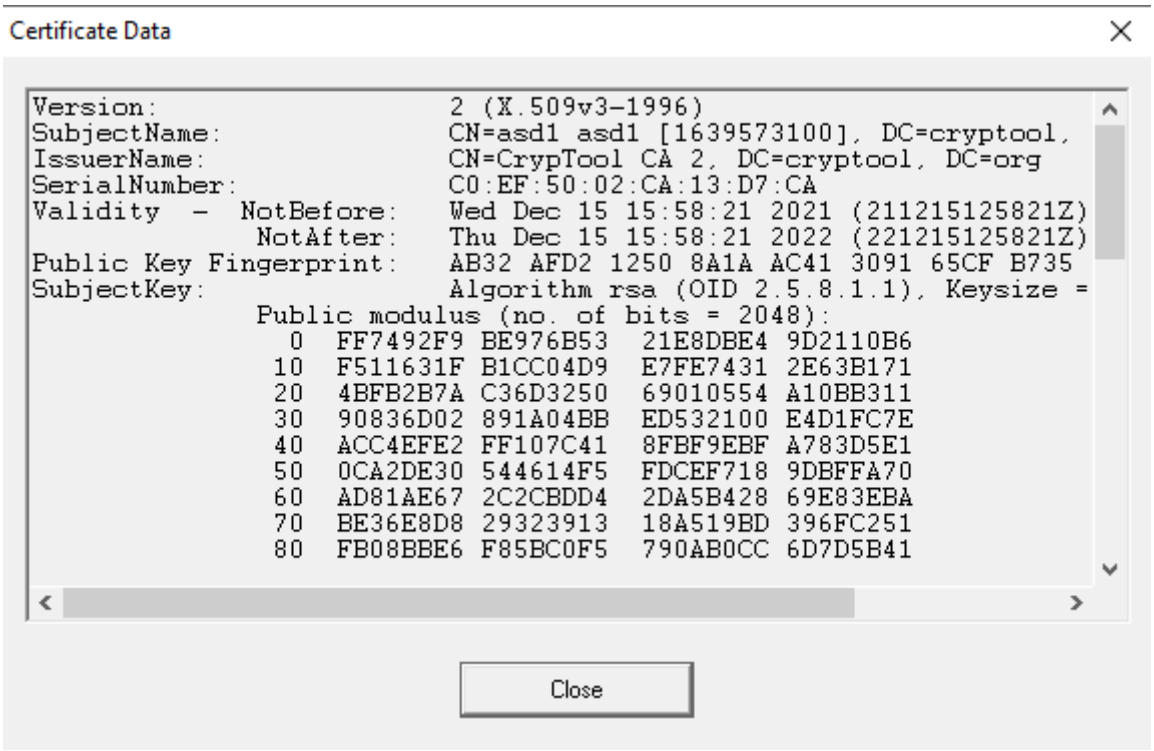
2. Сгенерированы ключи методами . Время генерации в таблице

RSA-2048	DSA-2048	EC-239
0.937	6.638	0.033

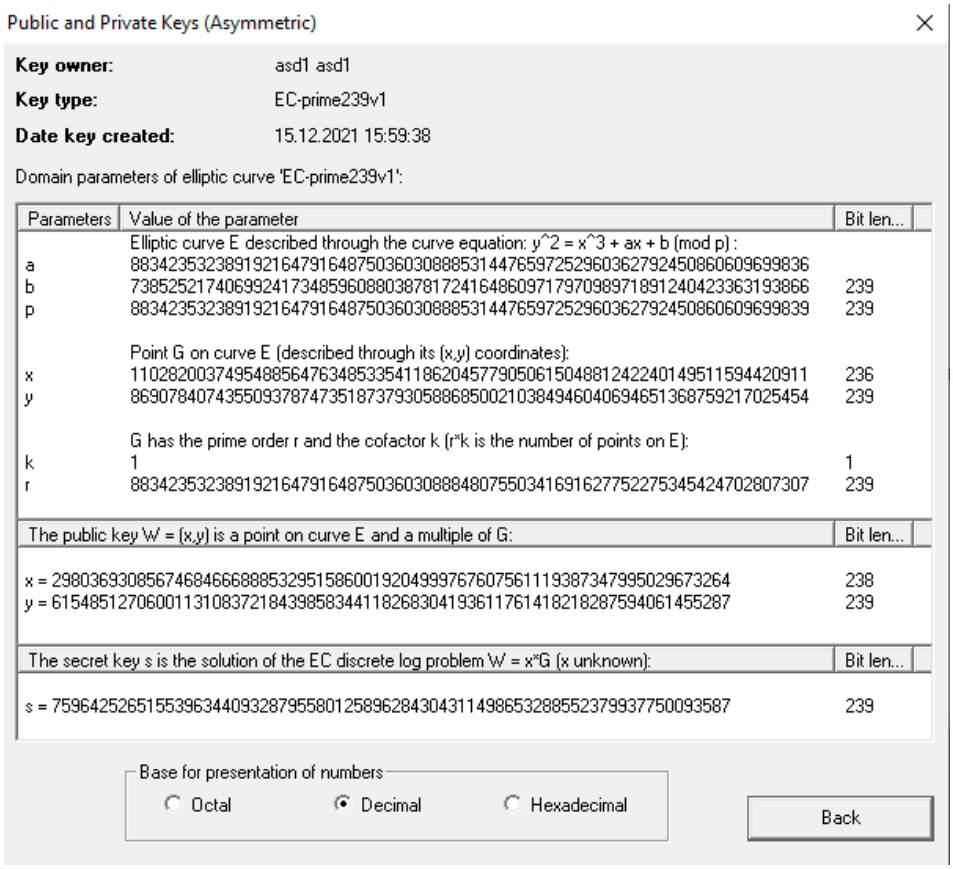
3. DSA-2048



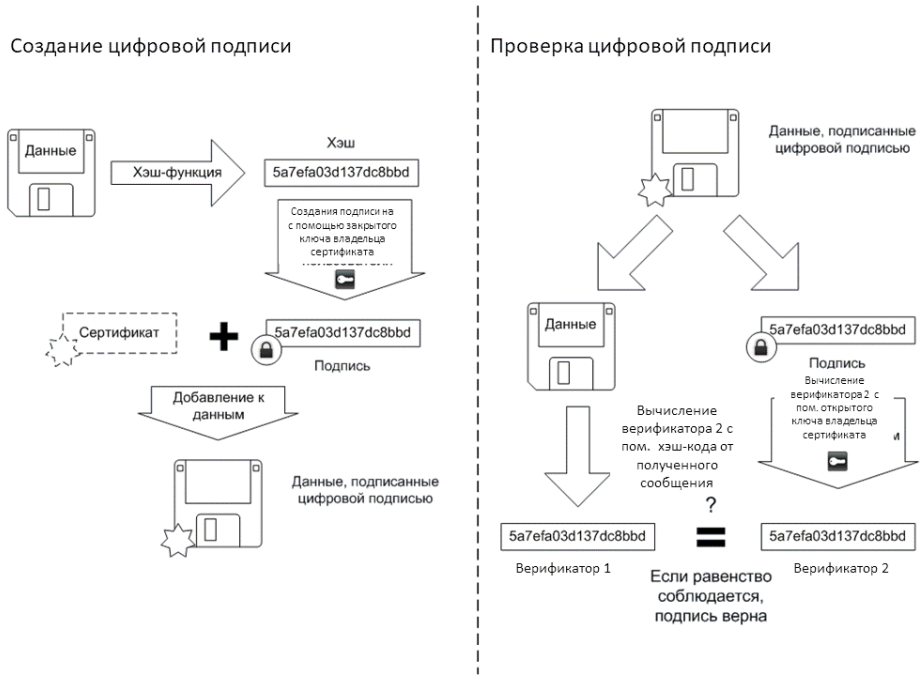
4. RSA-2048



5. EC-239



2. Процессы создания и проверки цифровой подписи



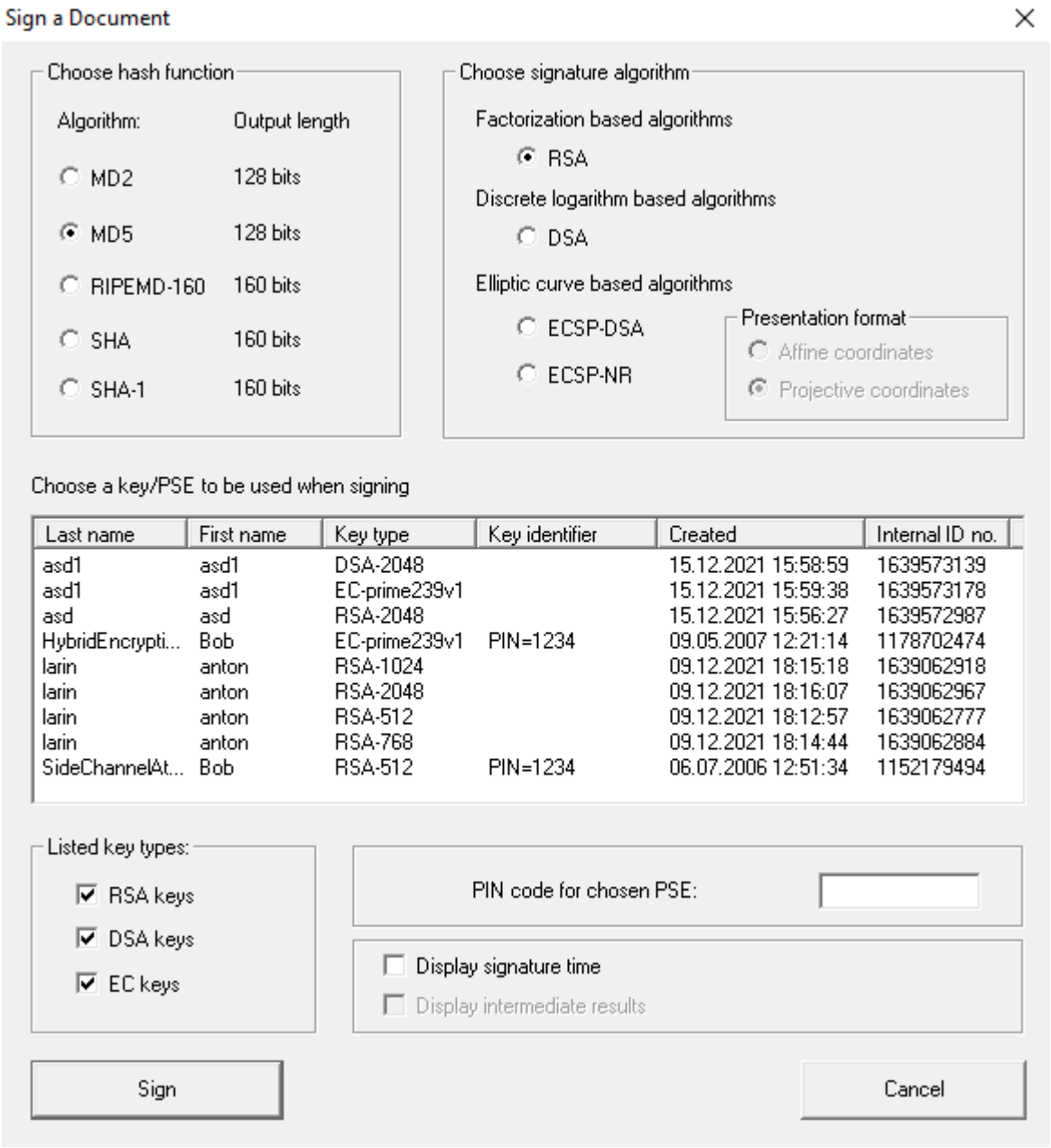
Задание

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.

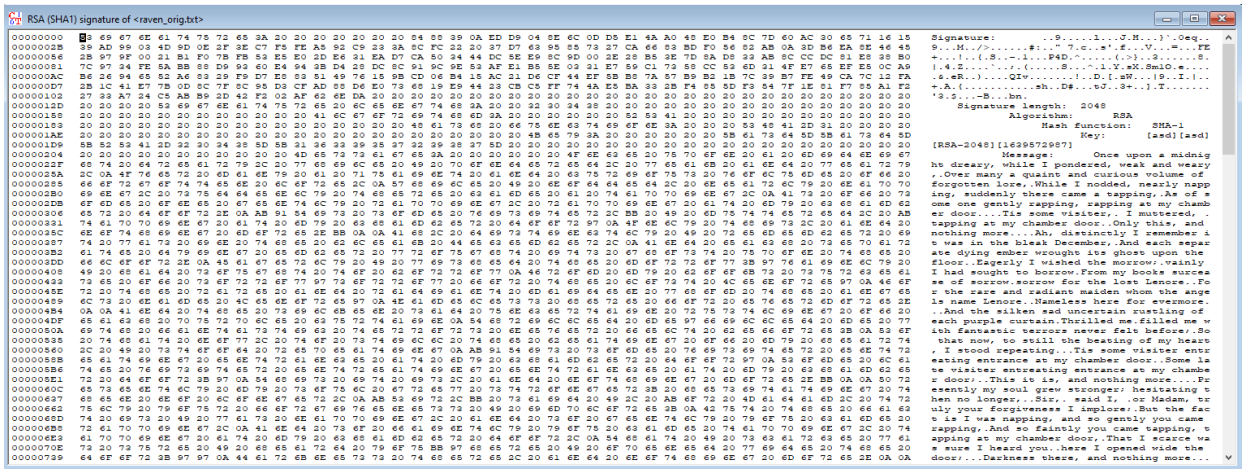
Выполнение

1. Взят полный текст поэмы Raven.

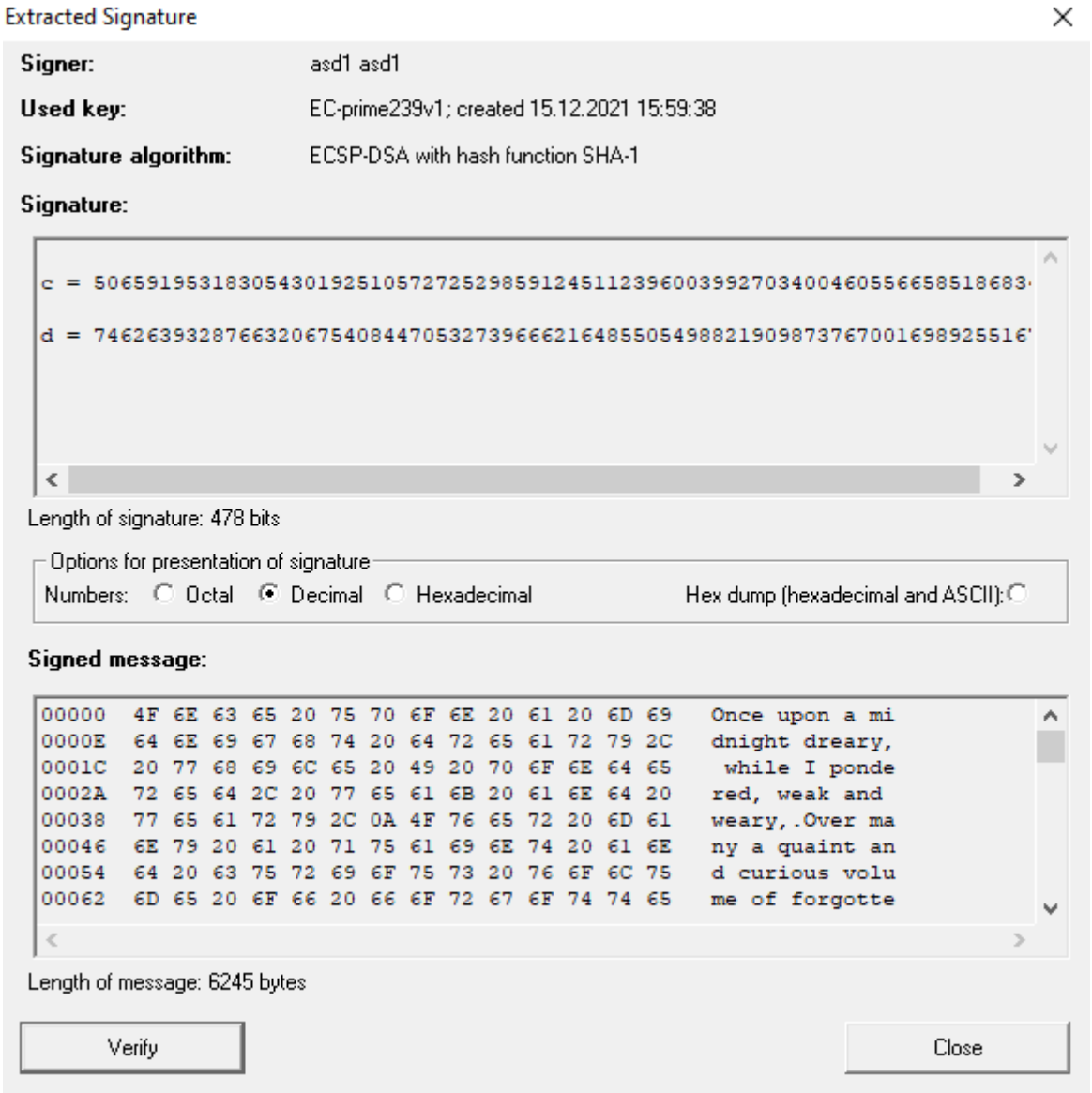


2. Задана функция SHA-1
3.

RSA-2048	DSA-2048	EC-239
0.014	0	0

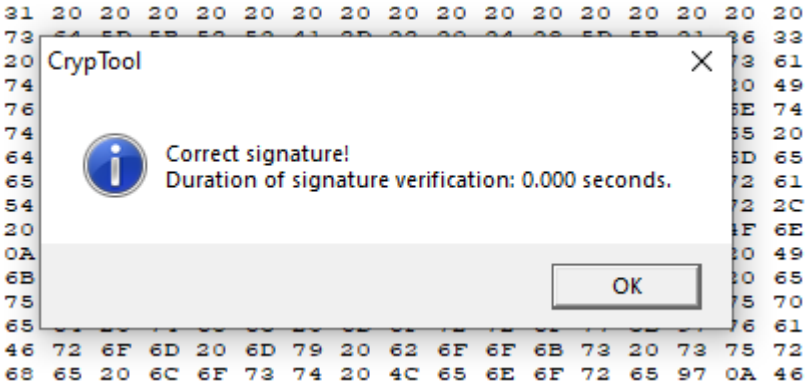


4. Извлечённая подпись



5. Результат проверки подписи

Проверка без изменений



Проверка после фальсификации

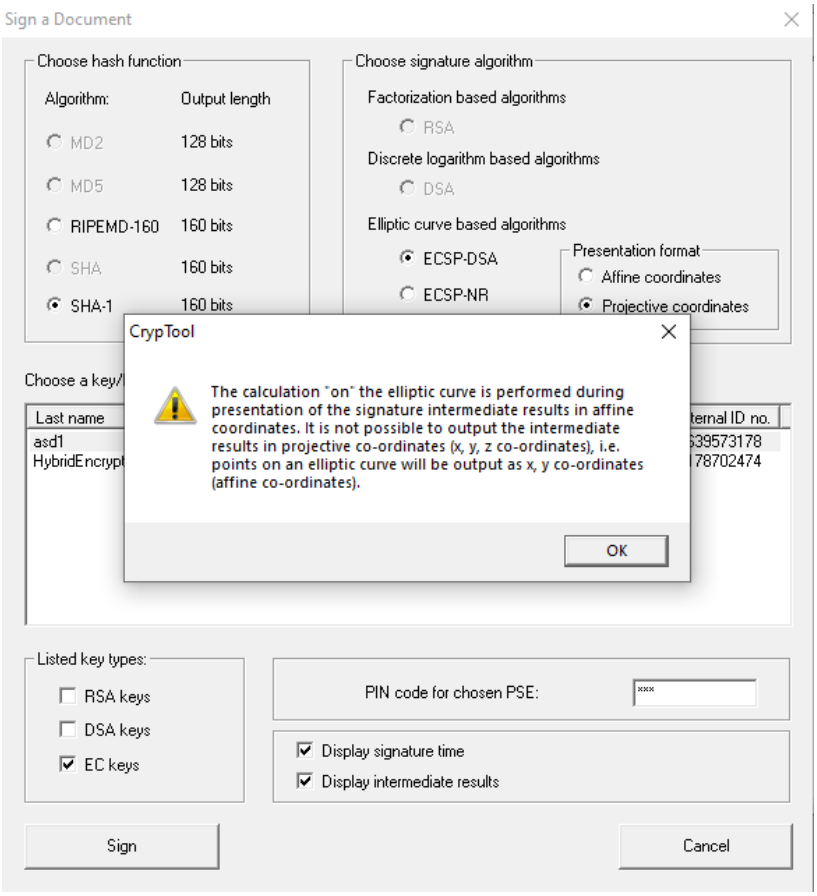
3. Схемы цифровой подписи на эллиптических кривых

Задание

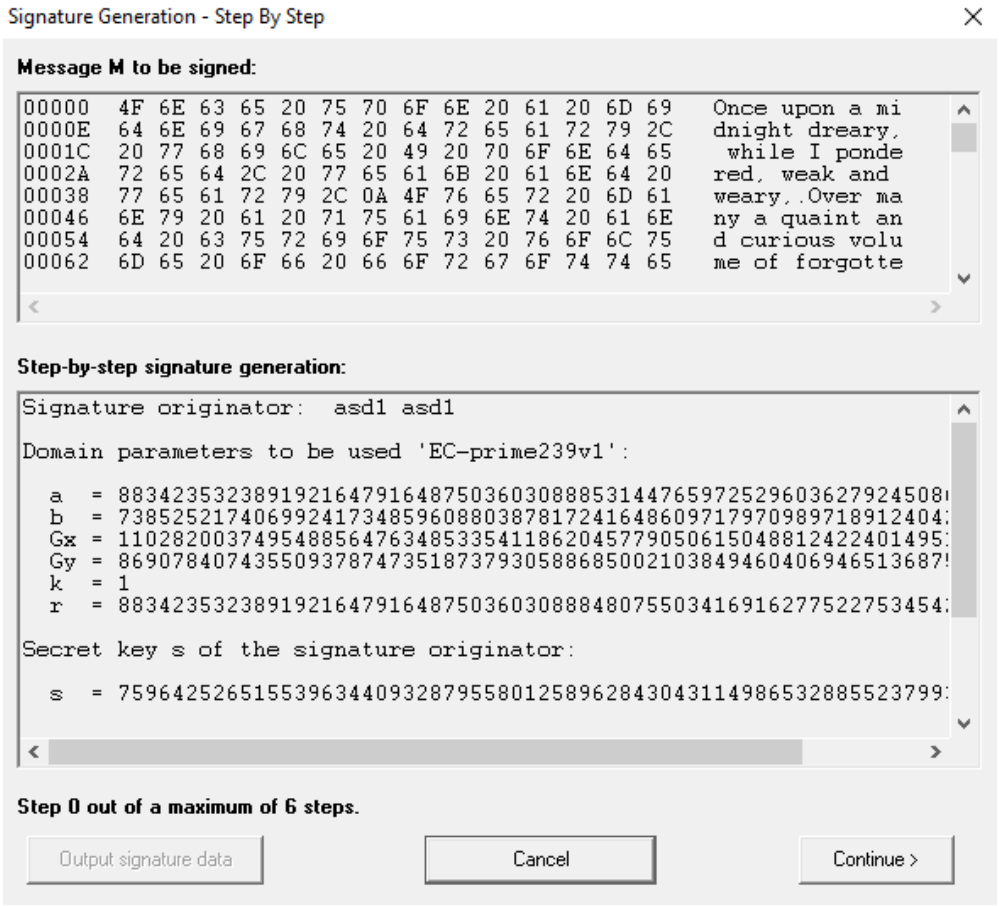
1. Выполните процедуру создание подписи «Digital Signatures/PKI→ Sign Document...» алгоритмом ECSP-DSA в пошаговом режиме (Display inter. results=ON). Зафиксируйте скриншоты последовательности шагов.
2. Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного 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.

Выполнение

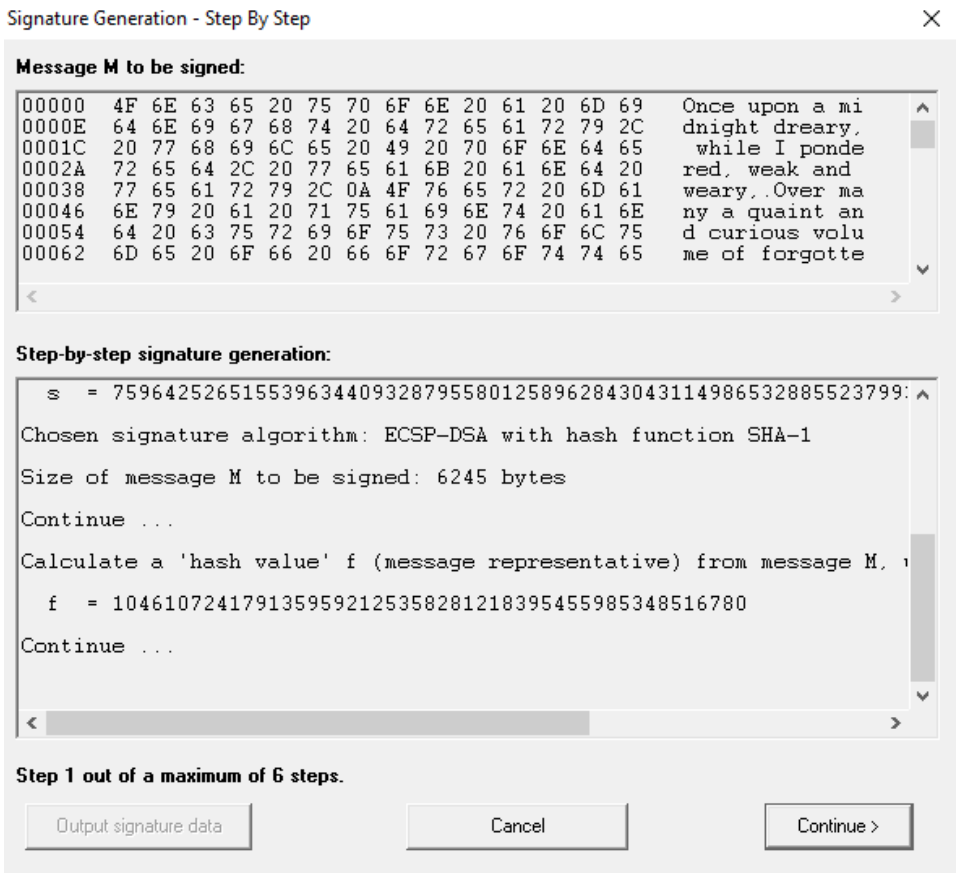
1. Начало работы:


Sign a Document
Choose hash function
Algorithm:
Output length
MD2
128 bits
MD5
128 bits
RIPEMD-160
160 bits
SHA
160 bits
SHA-1
160 bits
Choose signature algorithm
Factorization based algorithms
RSA
Discrete logarithm based algorithms
DSA
Elliptic curve based algorithms
ECSP-DSA
ECSP-NR
Presentation format
Affine coordinates
Projective coordinates
CrypTool
The calculation "on" the elliptic curve is performed during presentation of the signature intermediate results in affine coordinates. It is not possible to output the intermediate results in projective co-ordinates (x, y, z co-ordinates), i.e. points on an elliptic curve will be output as x, y co-ordinates (affine co-ordinates).
OK
Choose a key/
Last name
asd1
HybridEncrypt
ternal ID no.
639573178
78702474
Listed key types:
RSA keys
DSA keys
EC keys
PIN code for chosen PSE:
***
Display signature time
Display intermediate results
Sign
Cancel

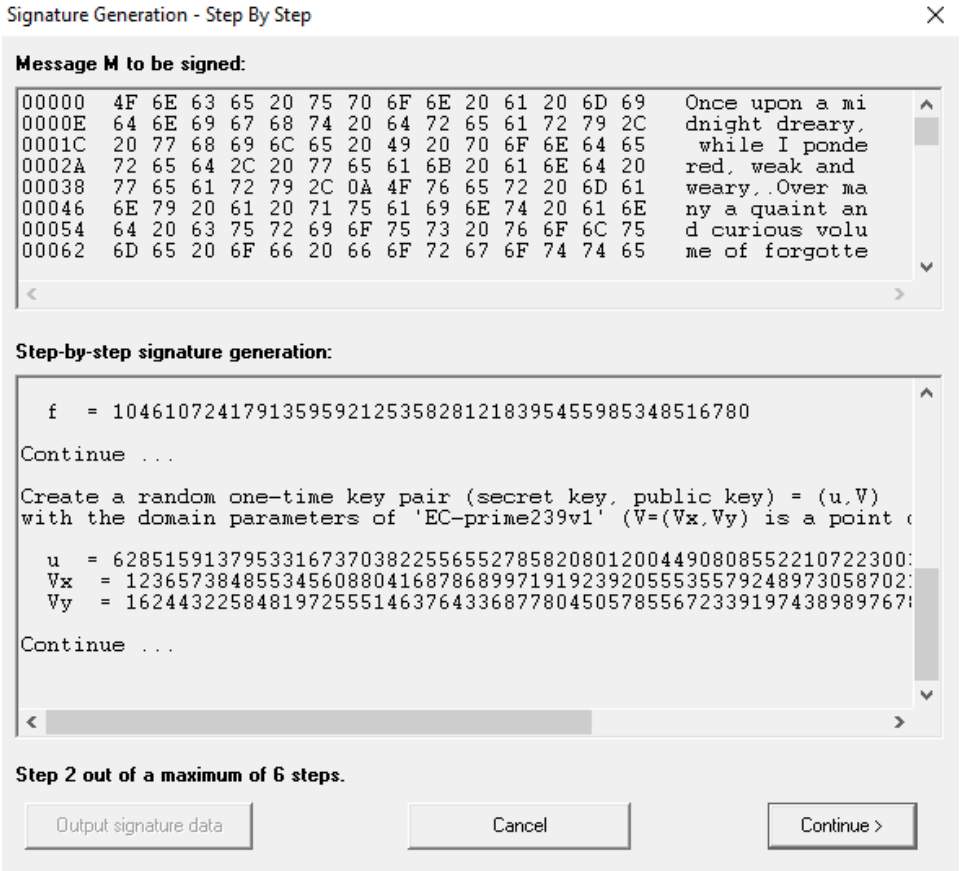
Шаг 0:


Signature Generation - Step By Step
Message M to be signed:
00000 4F 6E 63 65 20 75 70 6F 6E 20 61 20 6D 69 Once upon a mi
0000E 64 6E 69 67 68 74 20 64 72 65 61 72 79 2C dnight dreary,
0001C 20 77 68 69 6C 65 20 49 20 70 6F 6E 64 65 while I ponde
0002A 72 65 64 2C 20 77 65 61 6B 20 61 6E 64 20 red, weak and
00038 77 65 61 72 79 2C 0A 4F 76 65 72 20 6D 61 weary,.Over ma
00046 6E 79 20 61 20 71 75 61 69 6E 74 20 61 6E ny a quaint an
00054 64 20 63 75 72 69 6F 75 73 20 76 6F 6C 75 d curious volu
00062 6D 65 20 6F 66 20 66 6F 72 67 6F 74 74 65 me of forgotte
Step-by-step signature generation:
Signature originator: asd1 asd1
Domain parameters to be used 'EC-prime239v1':
a =
b =
Gx =
Gy =
k = 1
r =
Secret key s of the signature originator:
s =
Step 0 out of a maximum of 6 steps.
Output signature data
Cancel
Continue >

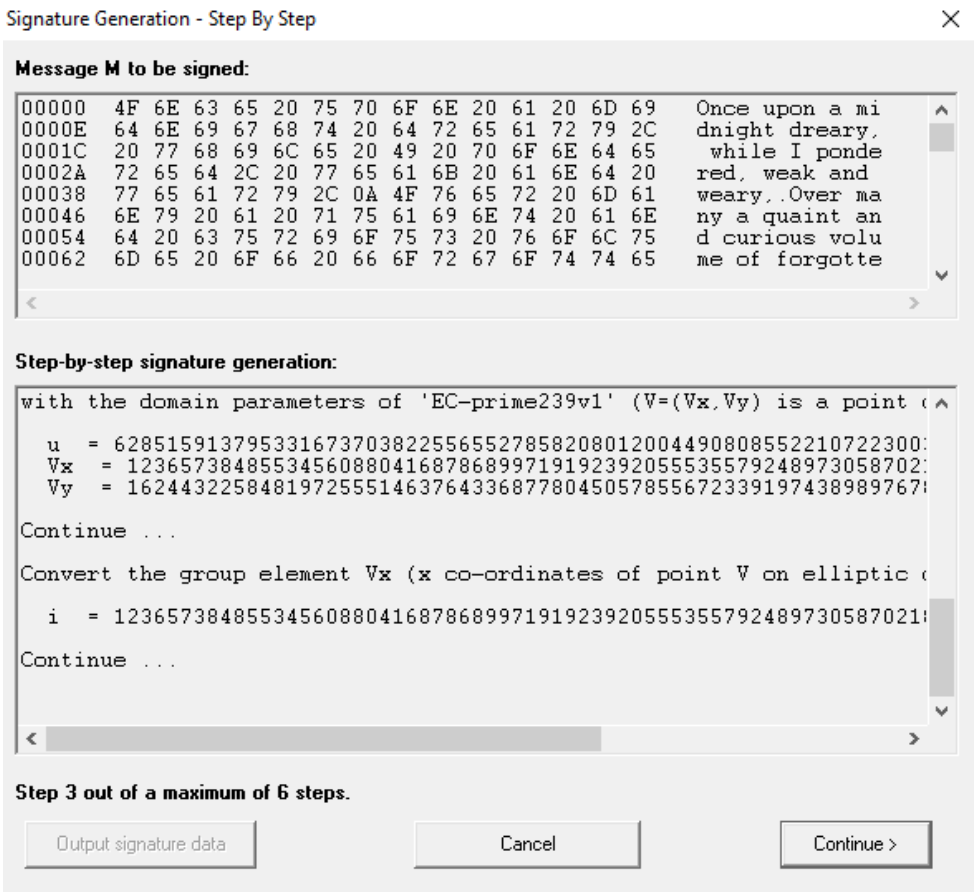
Шаг 1:


Signature Generation - Step By Step
Message M to be signed:
00000 4F 6E 63 65 20 75 70 6F 6E 20 61 20 6D 69 Once upon a mi
0000E 64 6E 69 67 68 74 20 64 72 65 61 72 79 2C dnight dreary,
0001C 20 77 68 69 6C 65 20 49 20 70 6F 6E 64 65 while I ponde
0002A 72 65 64 2C 20 77 65 61 6B 20 61 6E 64 20 red, weak and
00038 77 65 61 72 79 2C 0A 4F 76 65 72 20 6D 61 weary,.Over ma
00046 6E 79 20 61 20 71 75 61 69 6E 74 20 61 6E ny a quaint an
00054 64 20 63 75 72 69 6F 75 73 20 76 6F 6C 75 d curious volu
00062 6D 65 20 6F 66 20 66 6F 72 67 6F 74 74 65 me of forgotte
Step-by-step signature generation:
s = 75964252651553963440932879558012589628430431149865328855237993
Chosen signature algorithm: ECSP-DSA with hash function SHA-1
Size of message M to be signed: 6245 bytes
Continue ...
Calculate a 'hash value' f (message representative) from message M,
f = 1046107241791359592125358281218395455985348516780
Continue ...
Step 1 out of a maximum of 6 steps.
Output signature data
Cancel
Continue >

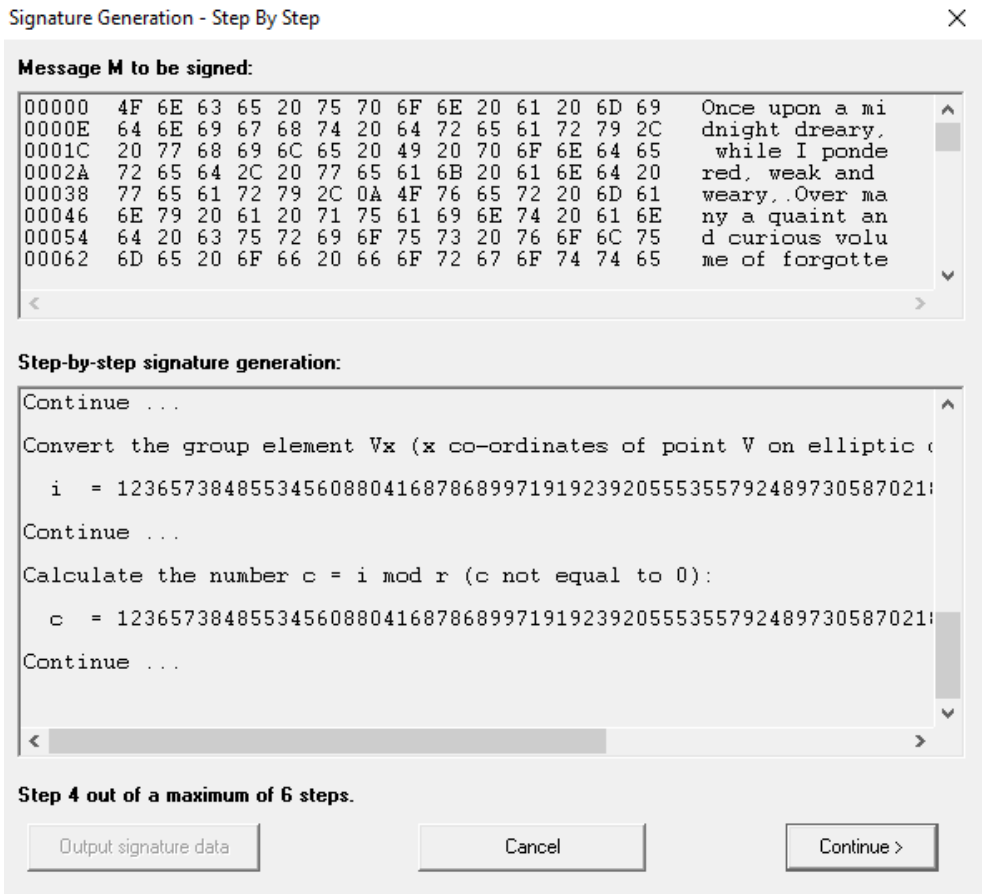
Шаг 2:


Signature Generation - Step By Step
Message M to be signed:
00000 4F 6E 63 65 20 75 70 6F 6E 20 61 20 6D 69 Once upon a mi
0000E 64 6E 69 67 68 74 20 64 72 65 61 72 79 2C dnight dreary,
0001C 20 77 68 69 6C 65 20 49 20 70 6F 6E 64 65 while I ponde
0002A 72 65 64 2C 20 77 65 61 6B 20 61 6E 64 20 red, weak and
00038 77 65 61 72 79 2C 0A 4F 76 65 72 20 6D 61 weary,.Over ma
00046 6E 79 20 61 20 71 75 61 69 6E 74 20 61 6E ny a quaint an
00054 64 20 63 75 72 69 6F 75 73 20 76 6F 6C 75 d curious volu
00062 6D 65 20 6F 66 20 66 6F 72 67 6F 74 74 65 me of forgotte
Step-by-step signature generation:
f = 1046107241791359592125358281218395455985348516780
Continue ...
Create a random one-time key pair (secret key, public key) = (u,V)
with the domain parameters of 'EC-prime239v1' (V=(Vx,Vy) is a point
u = 628515913795331673703822556552785820801200449080855221072230
Vx = 123657384855345608804168786899719192392055535579248973058702
Vy = 162443225848197255514637643368778045057855672339197438989767
Continue ...
Step 2 out of a maximum of 6 steps.
Output signature data
Cancel
Continue >

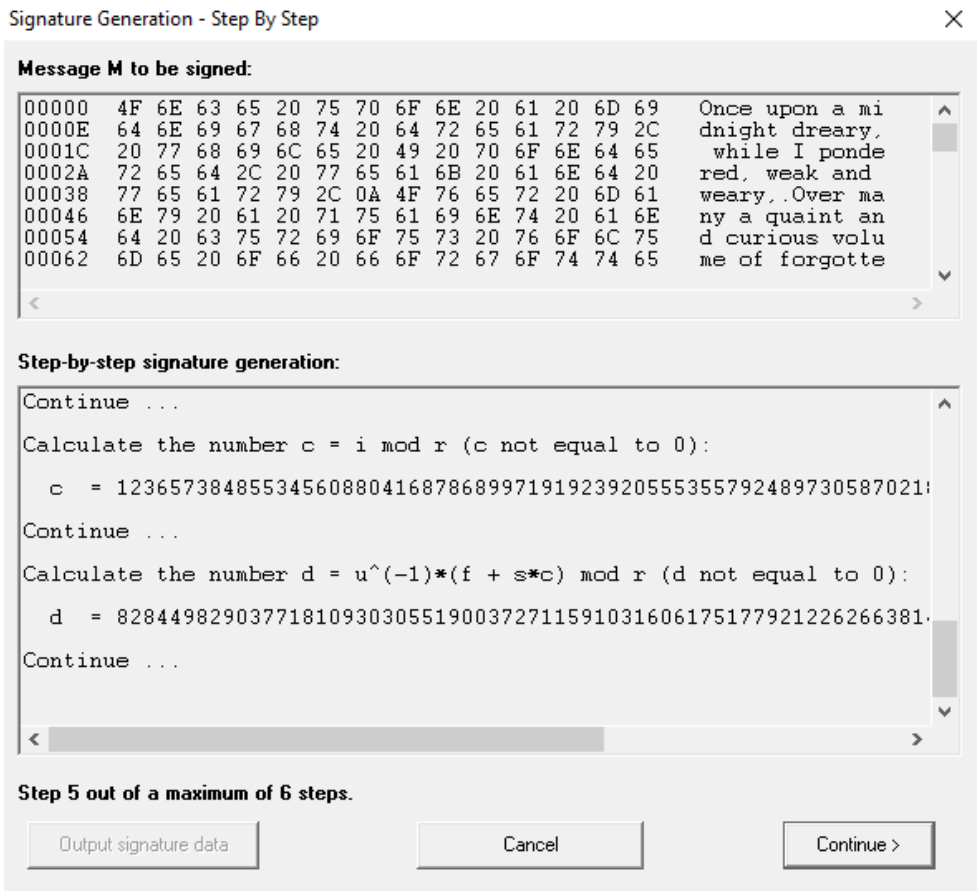
Шаг 3:


Signature Generation - Step By Step
Message M to be signed:
00000 4F 6E 63 65 20 75 70 6F 6E 20 61 20 6D 69 Once upon a mi
0000E 64 6E 69 67 68 74 20 64 72 65 61 72 79 2C dnight dreary,
0001C 20 77 68 69 6C 65 20 49 20 70 6F 6E 64 65 while I ponde
0002A 72 65 64 2C 20 77 65 61 6B 20 61 6E 64 20 red, weak and
00038 77 65 61 72 79 2C 0A 4F 76 65 72 20 6D 61 weary,.Over ma
00046 6E 79 20 61 20 71 75 61 69 6E 74 20 61 6E ny a quaint an
00054 64 20 63 75 72 69 6F 75 73 20 76 6F 6C 75 d curious volu
00062 6D 65 20 6F 66 20 66 6F 72 67 6F 74 74 65 me of forgotte
Step-by-step signature generation:
with the domain parameters of 'EC-prime239v1' (V=(Vx,Vy) is a point
u = 628515913795331673703822556552785820801200449080855221072230
Vx = 123657384855345608804168786899719192392055535579248973058702
Vy = 162443225848197255514637643368778045057855672339197438989767
Continue ...
Convert the group element Vx (x co-ordinates of point V on elliptic
i = 123657384855345608804168786899719192392055535579248973058702
Continue ...
Step 3 out of a maximum of 6 steps.
Output signature data
Cancel
Continue >

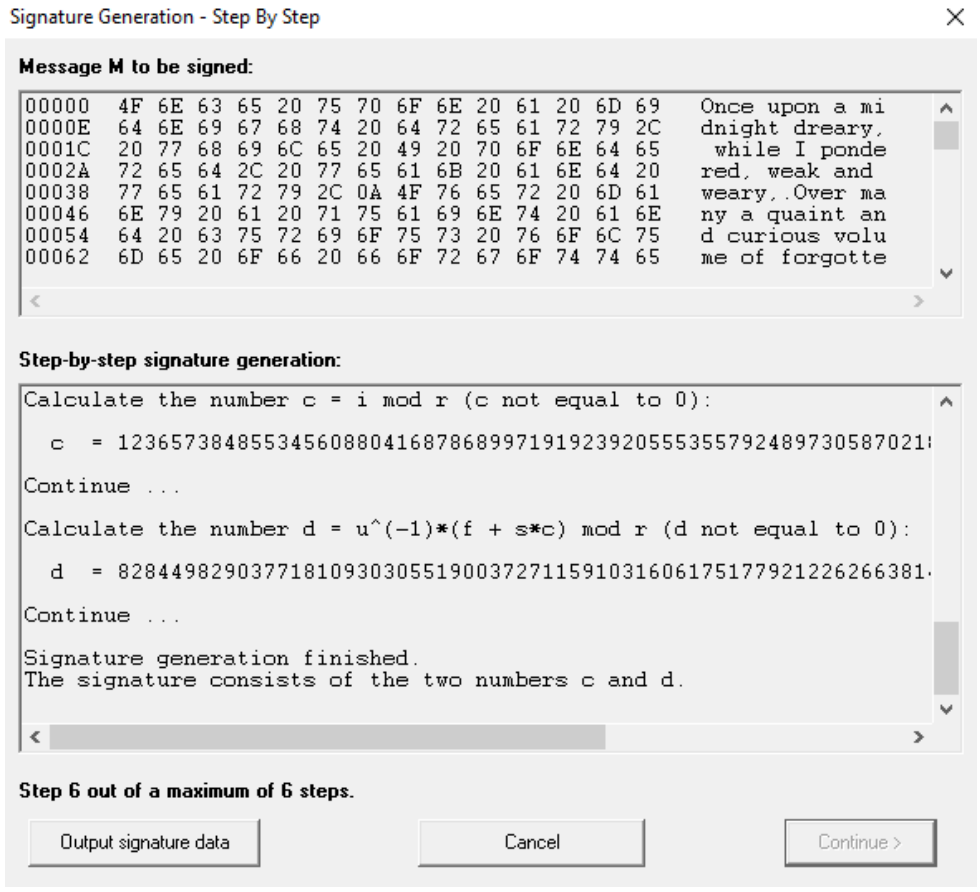
Шаг 4:


Signature Generation - Step By Step
Message M to be signed:
00000 4F 6E 63 65 20 75 70 6F 6E 20 61 20 6D 69 Once upon a mi
0000E 64 6E 69 67 68 74 20 64 72 65 61 72 79 2C dnight dreary,
0001C 20 77 68 69 6C 65 20 49 20 70 6F 6E 64 65 while I ponde
0002A 72 65 64 2C 20 77 65 61 6B 20 61 6E 64 20 red, weak and
00038 77 65 61 72 79 2C 0A 4F 76 65 72 20 6D 61 weary,.Over ma
00046 6E 79 20 61 20 71 75 61 69 6E 74 20 61 6E ny a quaint an
00054 64 20 63 75 72 69 6F 75 73 20 76 6F 6C 75 d curious volu
00062 6D 65 20 6F 66 20 66 6F 72 67 6F 74 74 65 me of forgotte
Step-by-step signature generation:
Continue ...
Convert the group element Vx (x co-ordinates of point V on elliptic
i = 123657384855345608804168786899719192392055535579248973058702
Continue ...
Calculate the number c = i mod r (c not equal to 0):
c = 123657384855345608804168786899719192392055535579248973058702
Continue ...
Step 4 out of a maximum of 6 steps.
Output signature data
Cancel
Continue >

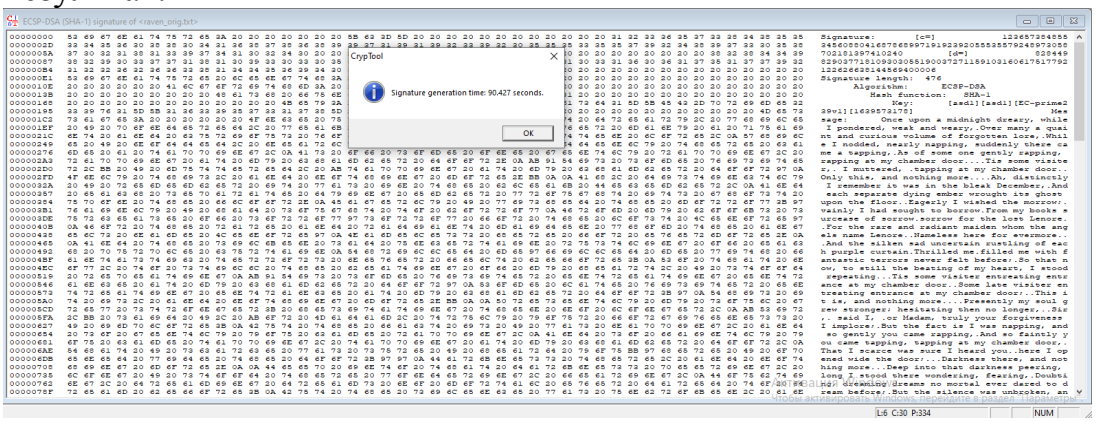
Шаг 5:


Signature Generation - Step By Step
Message M to be signed:
00000 4F 6E 63 65 20 75 70 6F 6E 20 61 20 6D 69 Once upon a mi
0000E 64 6E 69 67 68 74 20 64 72 65 61 72 79 2C dnight dreary,
0001C 20 77 68 69 6C 65 20 49 20 70 6F 6E 64 65 while I ponde
0002A 72 65 64 2C 20 77 65 61 6B 20 61 6E 64 20 red, weak and
00038 77 65 61 72 79 2C 0A 4F 76 65 72 20 6D 61 weary,.Over ma
00046 6E 79 20 61 20 71 75 61 69 6E 74 20 61 6E ny a quaint an
00054 64 20 63 75 72 69 6F 75 73 20 76 6F 6C 75 d curious volu
00062 6D 65 20 6F 66 20 66 6F 72 67 6F 74 74 65 me of forgotte
Step-by-step signature generation:
Continue ...
Calculate the number c = i mod r (c not equal to 0):
c = 123657384855345608804168786899719192392055535579248973058702
Continue ...
Calculate the number d = u^(-1)*(f + s*c) mod r (d not equal to 0):
d = 828449829037718109303055190037271159103160617517792122626638
Continue ...
Step 5 out of a maximum of 6 steps.
Output signature data
Cancel
Continue >

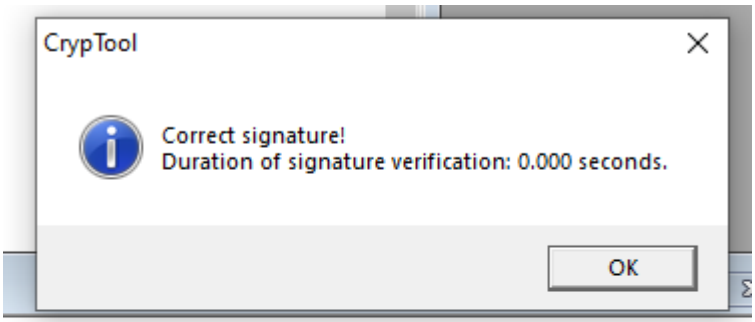
Шаг 6:


Signature Generation - Step By Step
Message M to be signed:
00000 4F 6E 63 65 20 75 70 6F 6E 20 61 20 6D 69 Once upon a mi
0000E 64 6E 69 67 68 74 20 64 72 65 61 72 79 2C dnight dreary,
0001C 20 77 68 69 6C 65 20 49 20 70 6F 6E 64 65 while I ponde
0002A 72 65 64 2C 20 77 65 61 6B 20 61 6E 64 20 red, weak and
00038 77 65 61 72 79 2C 0A 4F 76 65 72 20 6D 61 weary,.Over ma
00046 6E 79 20 61 20 71 75 61 69 6E 74 20 61 6E ny a quaint an
00054 64 20 63 75 72 69 6F 75 73 20 76 6F 6C 75 d curious volu
00062 6D 65 20 6F 66 20 66 6F 72 67 6F 74 74 65 me of forgotte
Step-by-step signature generation:
Calculate the number c = i mod r (c not equal to 0):
c = 123657384855345608804168786899719192392055535579248973058702
Continue ...
Calculate the number d = u^(-1)*(f + s*c) mod r (d not equal to 0):
d = 828449829037718109303055190037271159103160617517792122626638
Continue ...
Signature generation finished.
The signature consists of the two numbers c and d.
Step 6 out of a maximum of 6 steps.
Output signature data
Cancel
Continue >

Результат:

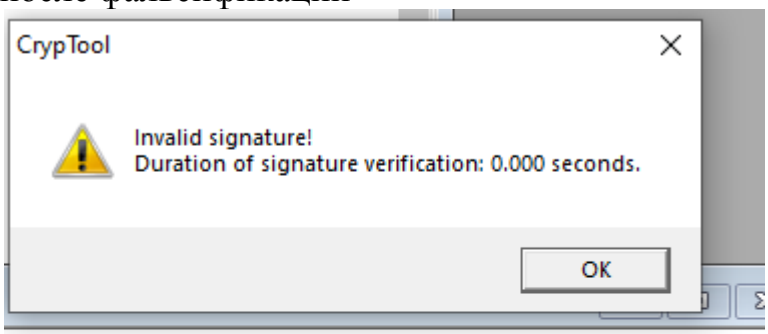


2. Проверка без изменений:




```
Signature:        [c=]              123657384855
345608804168786899719192392055535579248973058
702181397410240          [d=]              828449
829037718109303055190037271159103160617517792
122626638144569400006
Signature length:   476
      Algorithm:       ECSP-DSA
         Hash function:    SHA-1
            Key:        [asdl][asdl][EC-prime2
39v1][1639573178]                          Mes
sage:       Once upon a midnight dreary, while
```

Проверка после фальсификации




```
Signature:        [c=]              123657384855
345608804168786899719192392055535579248973058
702181397410240          [d=]              828449
829037718109303055190037271159103160617517792
122626638144569400006
Signature length:   476
      Algorithm:       ECSP-DSA
         Hash function:    SHA-1
            Key:        [asdl][asdl][EC-prime2
39v1][1639573178]                          Mes
sage:       Unce upon a midnight dreary, while
```

3. Генерация ключей

$y^2 = x^3 + ax + b$

a = 10

b = 20
p = 19
e1 = (11/13)
d = 5
e2 = d * e1 = (18/3)
q = 631

804291592781865767912555555060054978634942351405107451749311212
81330956526899
Подписание
M = h(M) = 42
r = 13
P(u,v) = 13 * e1 = (16/18)
S1 = u mod q = 16
S2 = r × h(M) + d × S1 = 13 * 42 + 5 * 16 mod q = 626
Проверка
A = h(m)^-1 * S1 mod q = 616 * 16 = 9856 mod q = 391
B = (q - s1) * m % q = 240
A * e1 = 14/15
B * e2 = 18/16
+ = 16/14
V = 16 = S1

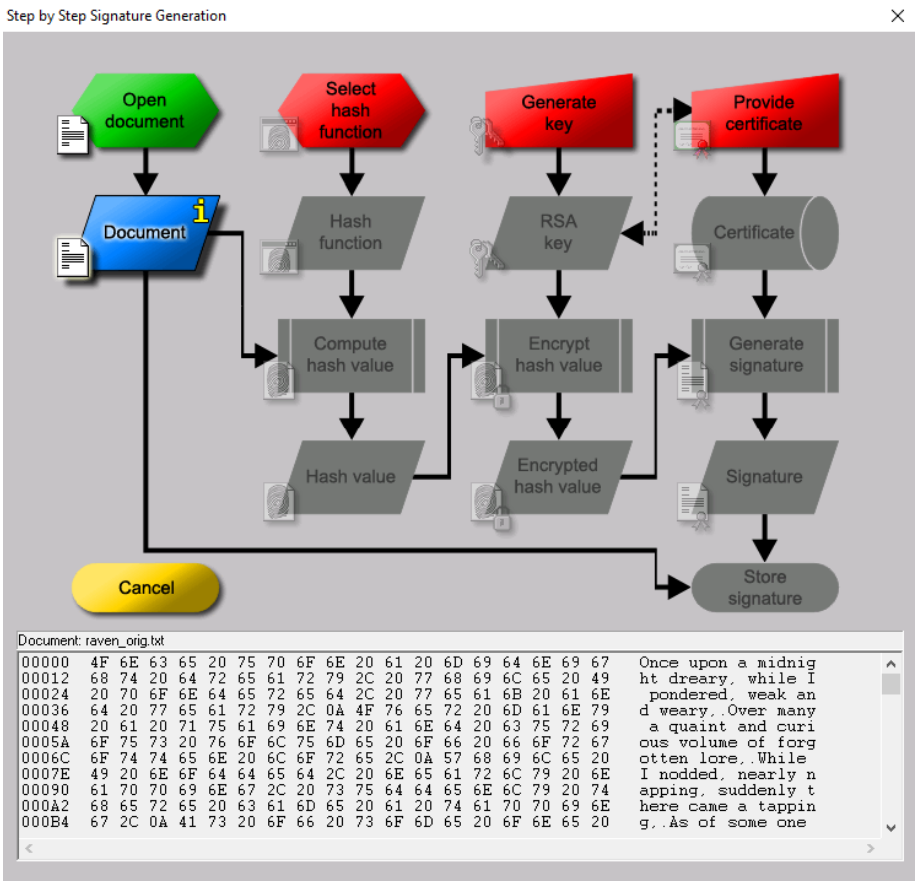
4. Демонстрация процесса подписи в среде PKI

Задание

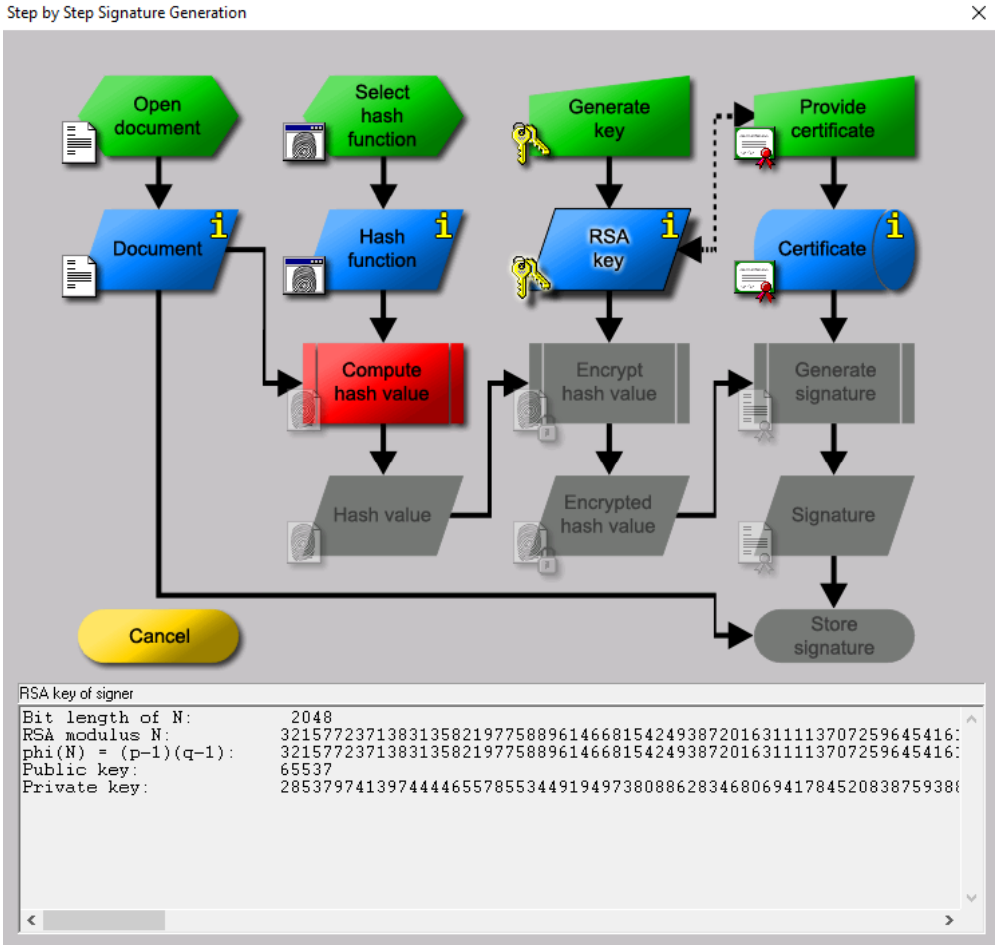
1. Запустить демонстрационную утилиту «Digital Signatures/PKI→Signature Demonstration...».
2. Получите сертификат на ранее сгенерированную ключевую пару RSA-2048.
3. Выполните и сохраните скриншоты всех этапов создания цифровой подписи документа.
4. Сохраните скриншот подписи.

Выполнение

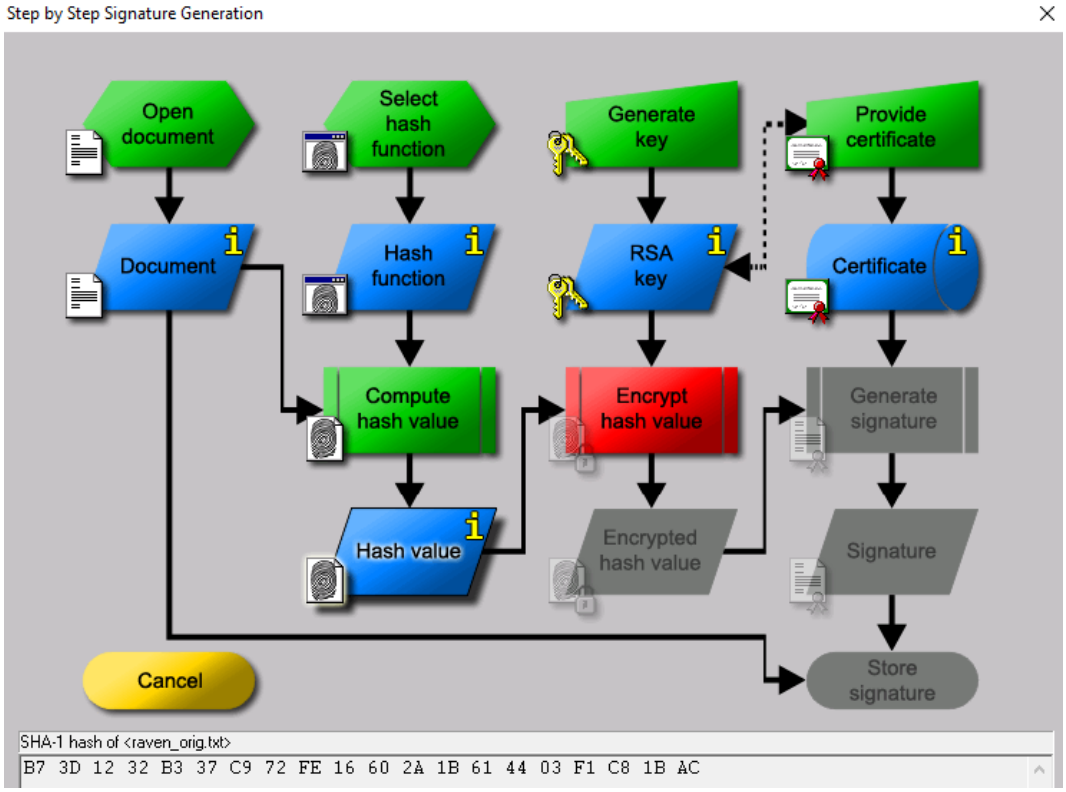
1. Начало работы, утилита



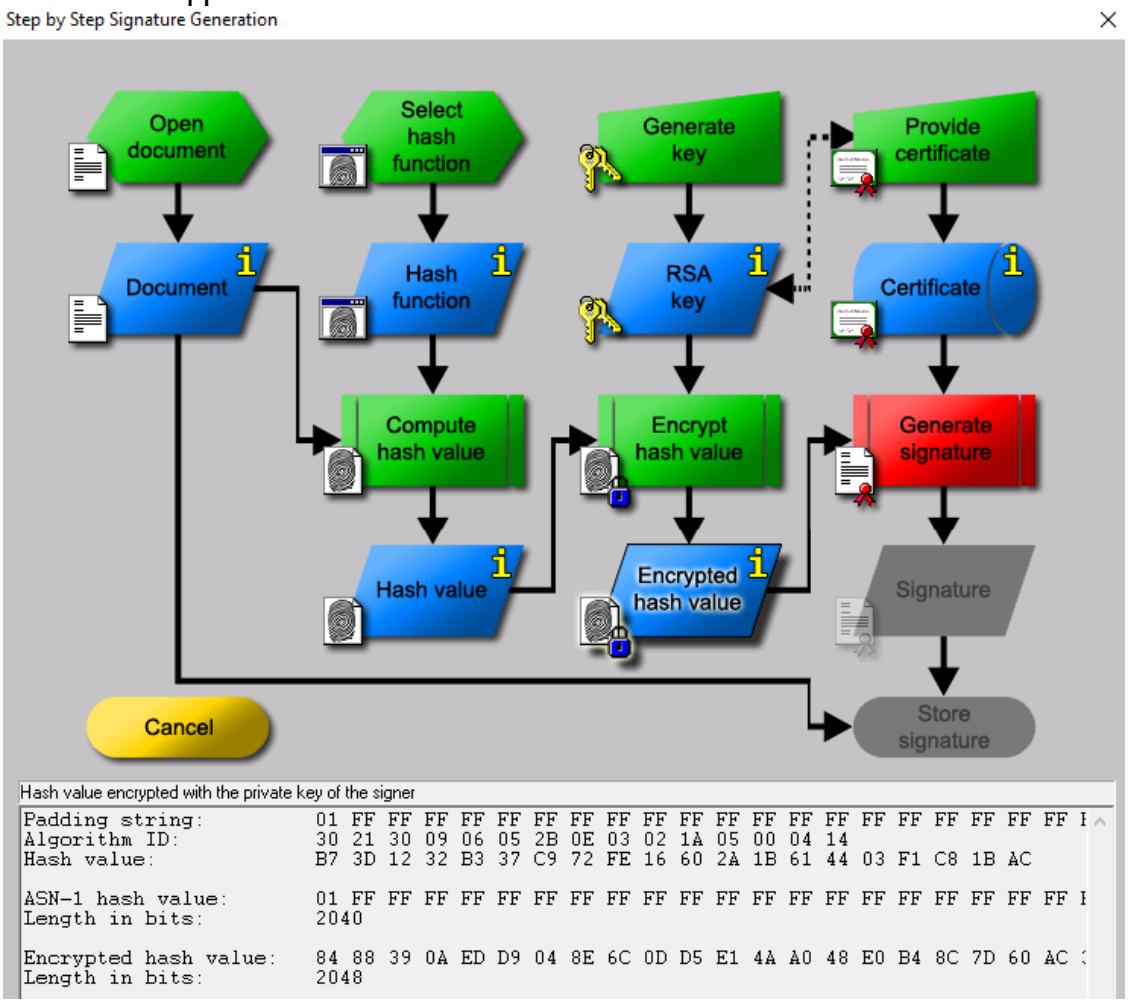
2. Използован сертификат для RSA-2048



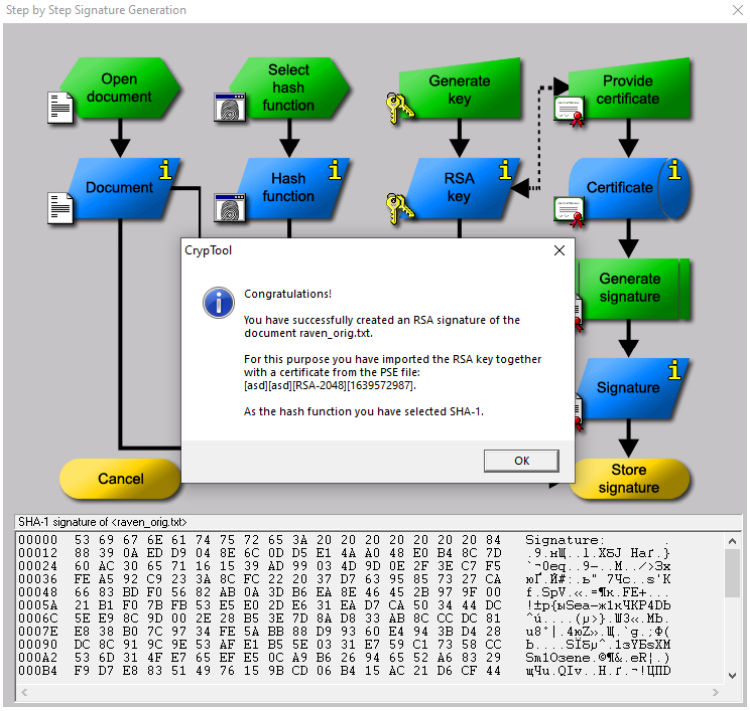
3. Выбрана функция SHA-1
   Расчитан хэш



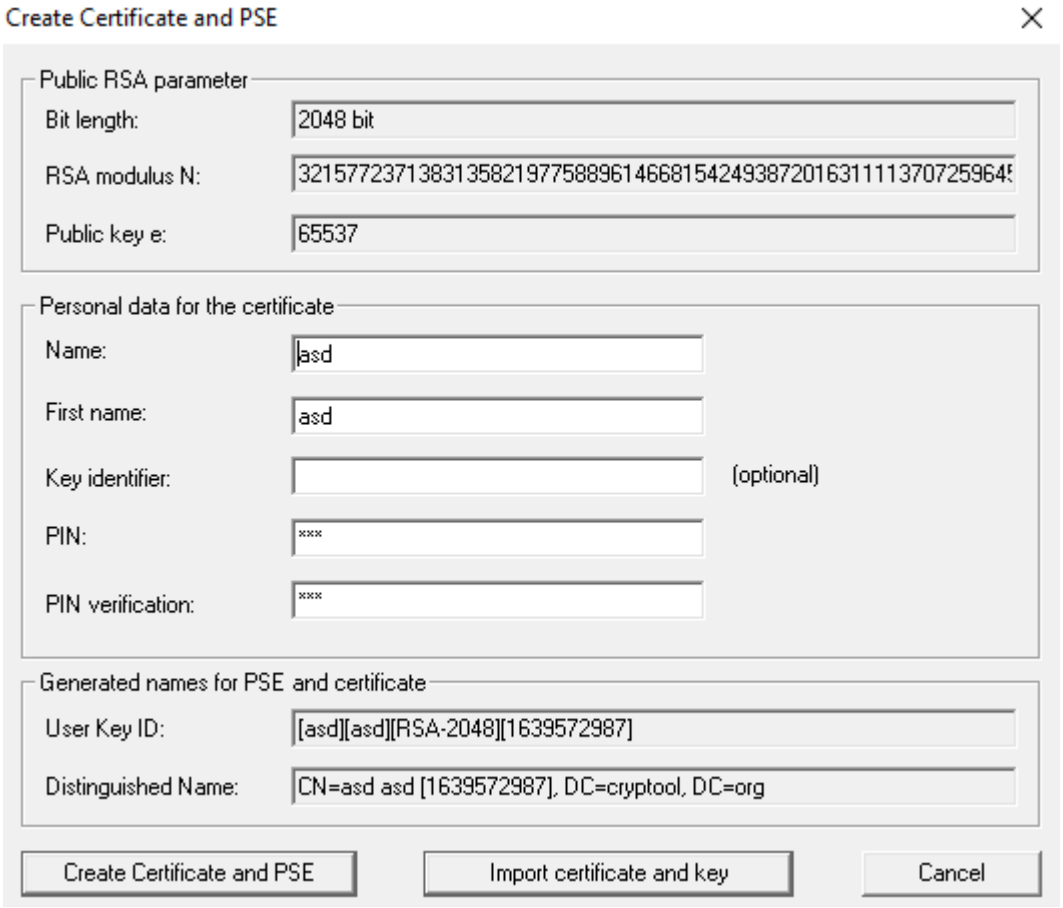
   Хэш зашифрован



Сгенерирована цифровая подпись



4. Сертификат:



5. Имитация атаки на гибридную криптосистему

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.

Выполнение

1. На этом этапе отчет сохранён в формате ODT и экспортирован в PDF

Выводы.

1. a
2. a
3. a
4.