

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 цифровой 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 p и q . Вычисляется $n = p * q$. Выбирается произвольное число e ($e < n$), взаимно простое с $(p - 1) * (q - 1)$. Вычисляется закрытый ключ (расширенный алгоритм Евклида):

$$e \times d \equiv 1 \mod ((p - 1) * (q - 1)) \equiv 1 \mod (p - 1) * (q - 1)$$

Пара чисел (e, n) объявляются открытым ключом, d выбирается закрытым ключом, p и q нужно уничтожить.

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

$$p - 1 = 0 \mod q$$

Выбирается e_1 , такое, что $e_1 : q = 1 \mod p$ путем вычисления $e_1 = e_0^{p-\frac{1}{q}} \mod p$, где $e_0 \in Z_p$ (теорема Ферма). Выбирается целое $d < q$ и вычисляется $e_2 = e_1^d \mod p$. Объявляется открытый ключ (e_1, e_2, p, q) . Назначается закрытый ключ d .

ECDSA: Выбирается эллиптическая кривая $E_p(a, b)$, p – простое. Выбирается точка на кривой $e_1 = (x_1, y_1)$. Для дальнейших вычислений выбирается другое простое число q - порядок циклической подгруппы группы точек эллиптической кривой : $q \times (x_1, y_1) = O$. Выбирается целое число d ,

$1 < d < q - 1$ и назначается закрытым ключом. Вычисляется другая точка на кривой $e_2 = d \times e_1$. Объявляется открытый ключ (a, b, p, q, e_1, e_2) .

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

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

Полученные ключи представлены на рис. 1 – 3 соответственно.

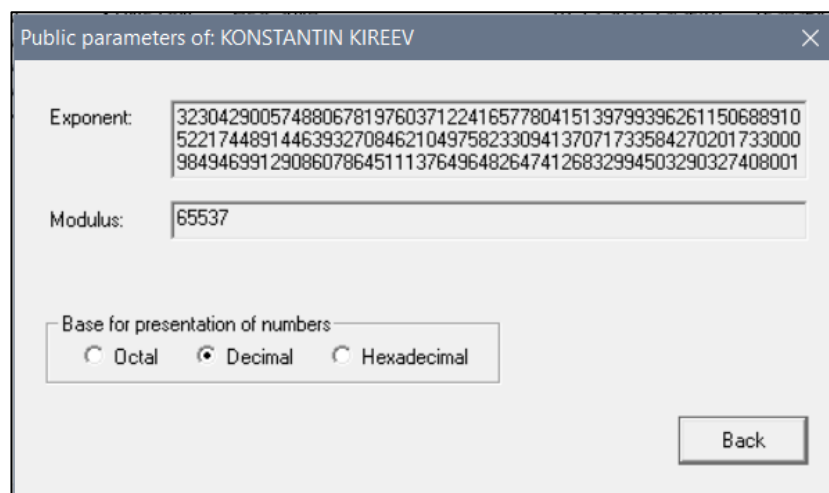


Рисунок 1 – Открытый ключ для RSA-2048

SubjectKey: Algorithm NIST-DSA (OID 1.3.14.3.2.12).

DSA prime p (no. of bits = 2048):

```

0 FFE59E07 2198AC6E C8A2D48F CC2D5C45
10 3FBD868B BECCF3B4 5FFC892A 383BF27F
20 36C33EC7 39CC8641 FA31F6C8 355D466E
30 7E247939 824C1A33 D21A1D86 82103F17
40 CB872664 A1A63F4D C1715848 EC0F8632
50 4EB8C338 724F241D 59C8C806 AC316203
60 A50464A4 D097C703 D181B697 B1662985
70 A130AA6E B0E540FD 73D84437 7B8C1F66
80 7442DCC8 E5FB952A 1799ED49 E3395567
90 5D02645F 61061D45 55EA5759 C7060575
A0 2442B04D 013D24F1 B5E55D6B EA4078B6
B0 CFDC6544 EF348739 C8D3CACA F9932C10
C0 190A97A2 C4F29CF9 DBCEFB82 9368B80B
D0 9442FD0C 920BF9D1 BA821702 63F2AC22
E0 7910A1B0 131919CD 9F08E797 AD05FEF
F0 2BFE8210 6AFA95D8 18A335FF 3AB7C749

```

DSA prime q (no. of bits = 160):

```

0 9491F5FE 9C61317B 4D0D706D 81F14C80
10 833501B5

```

DSA base g (no. of bits = 2048):

```

0 2255F1A9 A689F8F3 E21D0C95 340E9404
10 EE376E0B DF4FB349 D6E6BF77 C05D2084
20 6ECD8888 6E6FF56A 6599329C D64E7F04
30 AC33C33B 5086A264 8D238D46 2E599DBF
40 999E1699 FCC50C34 203DE0BC 550FAD3C
50 B65841F8 E17BF300 83EBE1AC A8C2F6C5
60 163B32C4 AAAEA770 1A3738C3 594A2FB8
70 EB1512B0 C9175B3E 2B840FCF CFCBFFF6
80 673EE99E 4733C4F2 ACCA699F 4A8EB9AB
90 B08F4878 80F41614 195E21DF D3EB5C97
A0 576C2B18 B940E562 730BC725 54392262
B0 349860BA 11D1233B 45A7A888 11789F91
C0 0A67A582 F70DEEC3 837DB527 97243D3B
D0 237E4A7A F19FAF24 1BD8C9C9 3F8857F3
E0 84ED5FBE A982FA58 F34F6E08 351D351A
F0 C7F3ABAB 0A31A3D5 E0A7218B 2A3F24A2

```

Public y (no. of bits = 2047):

```

0 60CB4246 3FF69913 4FC2E0D2 1F72D1C9
10 798A2E90 3FBE3629 F5E39F1A 81186C9C
20 02EDEA87 93B6A89D 516D115C 71E265EA
30 3D5CC35B 9F886C6A 704D31BB 56E321D6
40 ED73F650 EA224611 BC76F81F 0B5676F1
50 68FD5004 1DA0D93E A8614FC8 1D9BE638
60 CC9DD2B0 665CD34E F3421DB2 35BB0BCC
70 4D5B8D55 C586B3CA 78460A0F B718BEAA
80 6851F442 AD50E228 4232EB1A 9FFF2DF8
90 C943B5A4 CB10128E 3A24F3C5 B0C03483
A0 68C828FB CB137F89 D885FCEB C076C01F
B0 72E7519C EA950EA2 1EF880A7 50C79E17
C0 F21281C1 9F27E87F 02A791BC 766FF26B
D0 8BA84756 32610BF5 C6AC999B 963B8587
E0 202D0DD3 4E94D5B0 89BFBFEC C0C2D8EC
F0 4BC05787 9C36DC70 E5AE2813 D43A66C9

```

Рисунок 2 – Открытый ключ для DSA-2048

Public Key (Asymmetric)

Key owner: KONSTANTIN KIREEV

Key type: EC-prime239v1

Date key created: 10.12.2021 13:37:48

Domain parameters of elliptic curve 'EC-prime239v1':

Parameters	Value of the parameter	Bit len...
Elliptic curve E described through the curve equation: $y^2 = x^3 + ax + b \pmod{p}$:		
a	883423532389192164791648750360308885314476597252960362792450860609699836	239
b	738525217406992417348596088038781724164860971797098971891240423363193866	239
p	883423532389192164791648750360308885314476597252960362792450860609699839	239
Point G on curve E (described through its (x,y) coordinates):		
x	110282003749548856476348533541186204577905061504881242240149511594420911	236
y	86907840743509378747351873793058868500210384946040694651368759217025454	239
G has the prime order r and the cofactor k (r*k is the number of points on E):		
k	1	1
r	883423532389192164791648750360308884807550341691627752275345424702807307	239
The public key W = (x,y) is a point on curve E and a multiple of G:		
x	876531330084174019205783004926085135266190111925544280660899801362437642	239
y	76044985549768371634697781880468450712559454481337222129513463893629263	239

Base for presentation of numbers

☐ Octal
 ☒ Decimal
 ☐ Hexadecimal

Back

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.

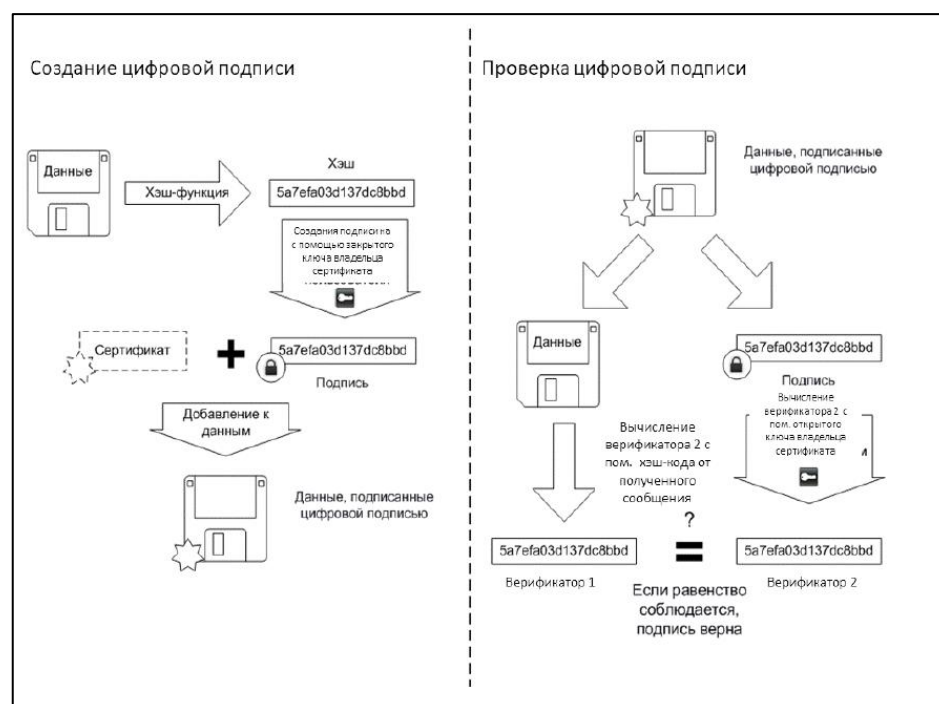


Рисунок 4 – Обобщенная схема

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

Исходный текст представлен на рис. 5.

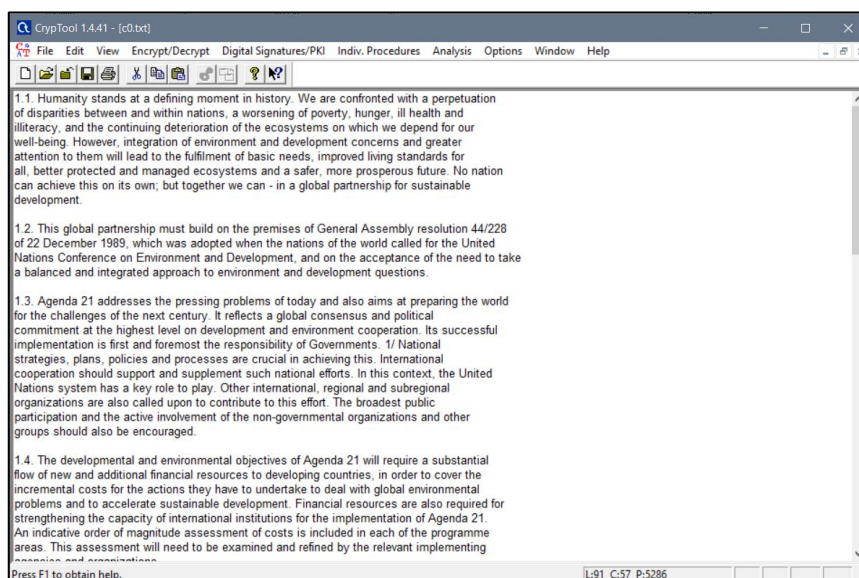


Рисунок 5 – Исходный текст

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.

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

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

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

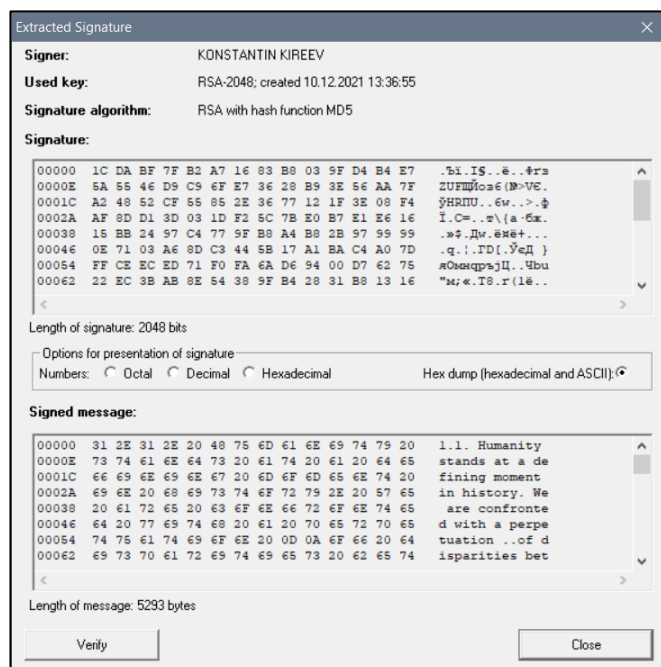


Рисунок 6 – Цифровая подпись RSA с MD5

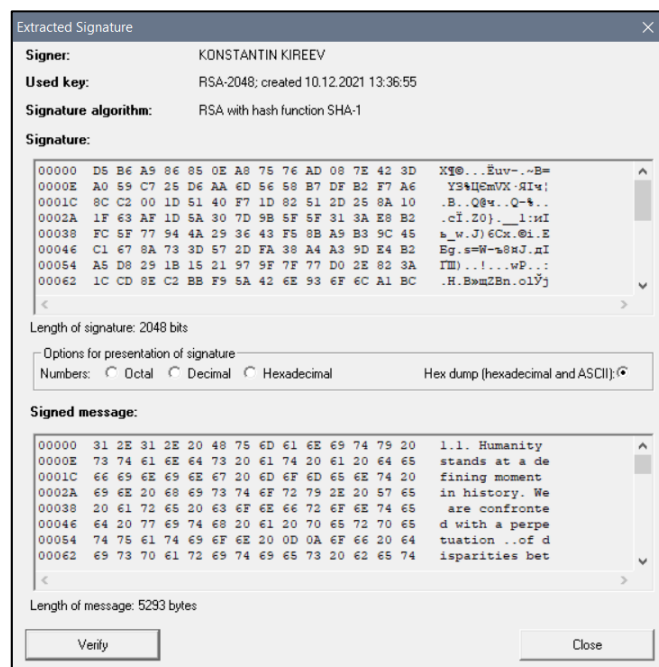


Рисунок 7 – Цифровая подпись RSA с SHA-1

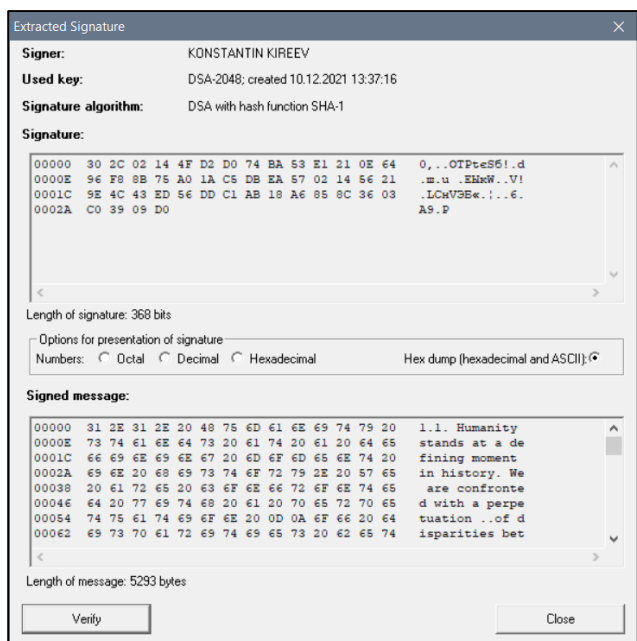


Рисунок 8 – Цифровая подпись DSA с SHA-1

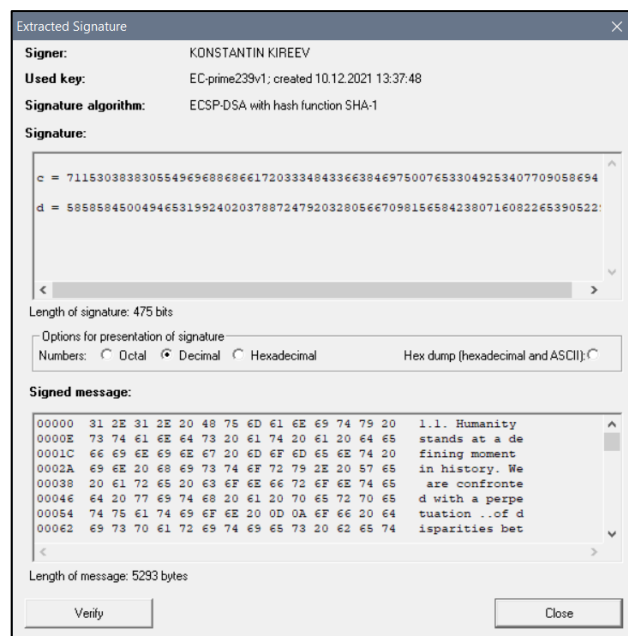



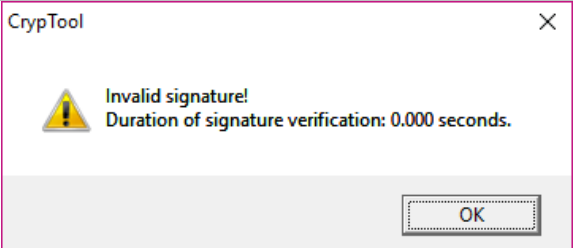
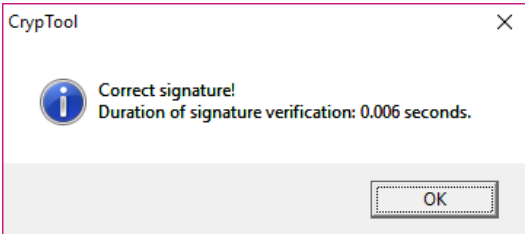
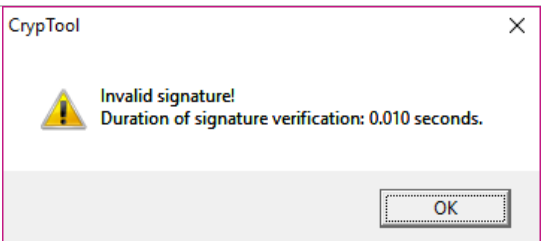

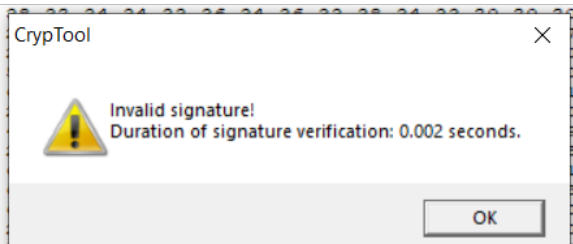
Рисунок 9 – Цифровая подпись ECDSA с SHA-1

5. 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.

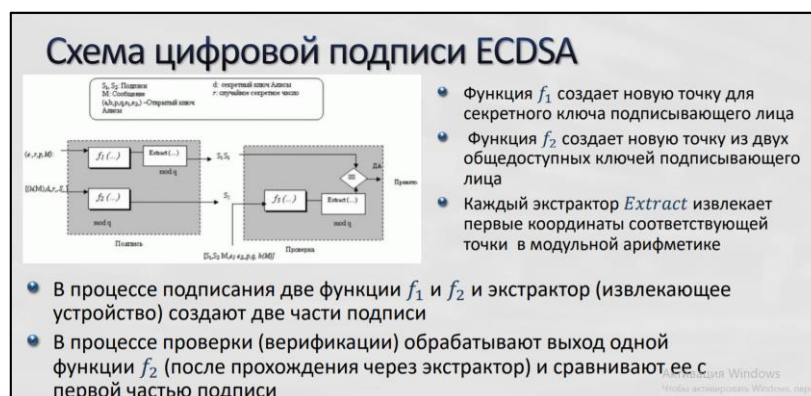
Таблица 3 – Проверка подписей на сохранение и нарушение целостности

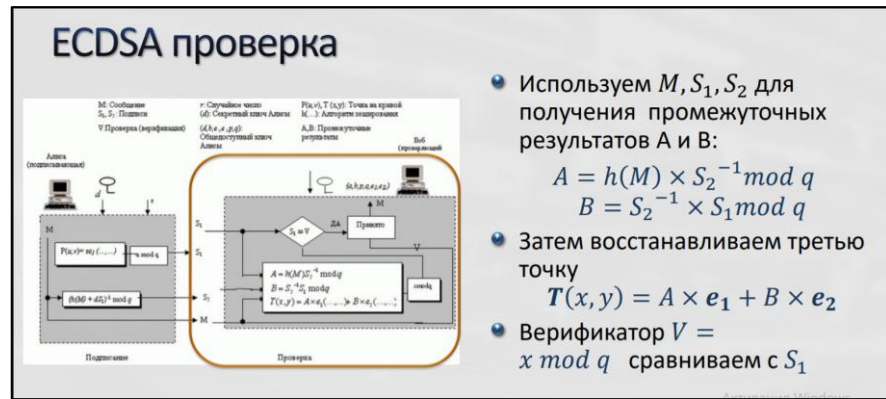
Алгоритм	Сохранение целостности	Нарушение целостности
RSA-2048		

		
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

```
Signature originator: KONSTANTIN KIREEV
Domain parameters to be used 'EC-prime239v1':

a = 8834235323891921647916487503603088853144765972529603627924508
b = 7385252174069924173485960880387817241648609717970989718912404
Gx = 1102820037495488564763485335411862045779050615048812422401495
Gy = 8690784074355093787473518737930588685002103849460406946513687
k = 1
r = 8834235323891921647916487503603088848075503416916277522753454

Secret key s of the signature originator:

s = 1102425865215131318869838233632016573492122662528950860364229
```

```
Chosen signature algorithm: ECSP-DSA with hash function SHA-1
Size of message M to be signed: 5293 bytes
Continue ...
Calculate a 'hash value' f (message representative) from message M,
f = 9317987353558204788077231260038817576602152843
Continue ...
```

```
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 on
the curve)
u = 2025222728998582744464655553612427028435551515436275826926752
Vx = 786487695916772261562337716762368798791877120294438204548101
Vy = 663125697911488862625289837160836197220300288799486576471638
Continue ...
```

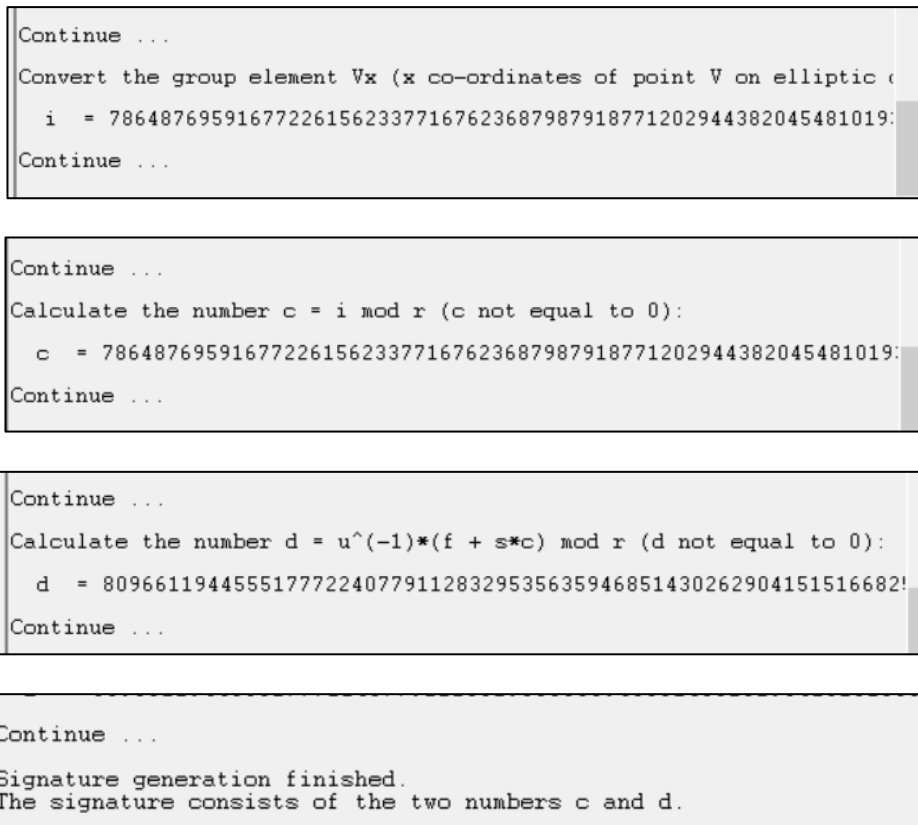


Рисунок 10 – Этапы создания подписи

Выполним сравнение реализаций алгоритма в Cryptool и в лекционных материалах.

Cryptool	a	b	(Gx, Gy)	r	s	f	u	i	(Vx, Vy)	c	d
Лекция	a	b	$e_1 = (x, y)$	q	d	$h(M)$	r	u (абсцисса)	$P(u, v)$	S_1	S_2

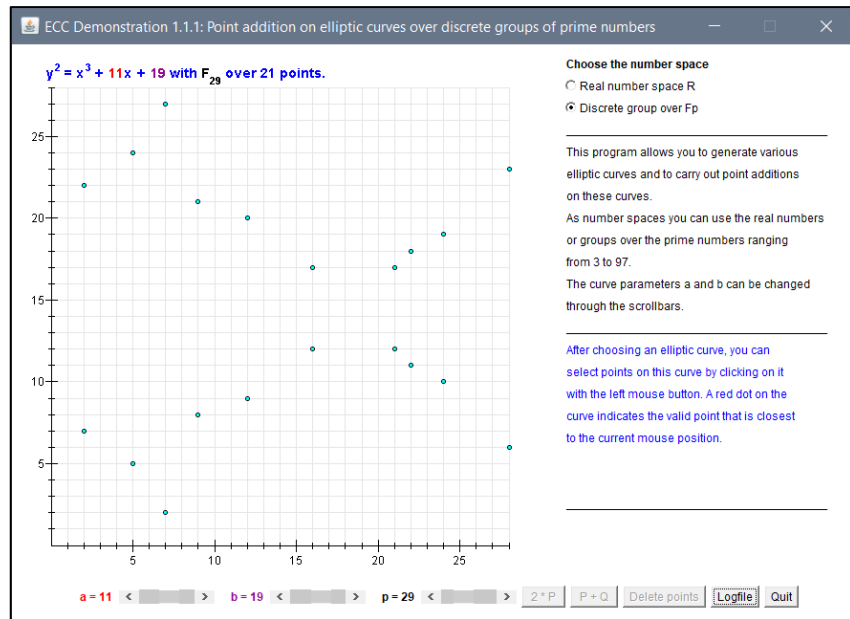
2. Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

Данная проверка уже выполнялась в пункте 8.1.

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*.

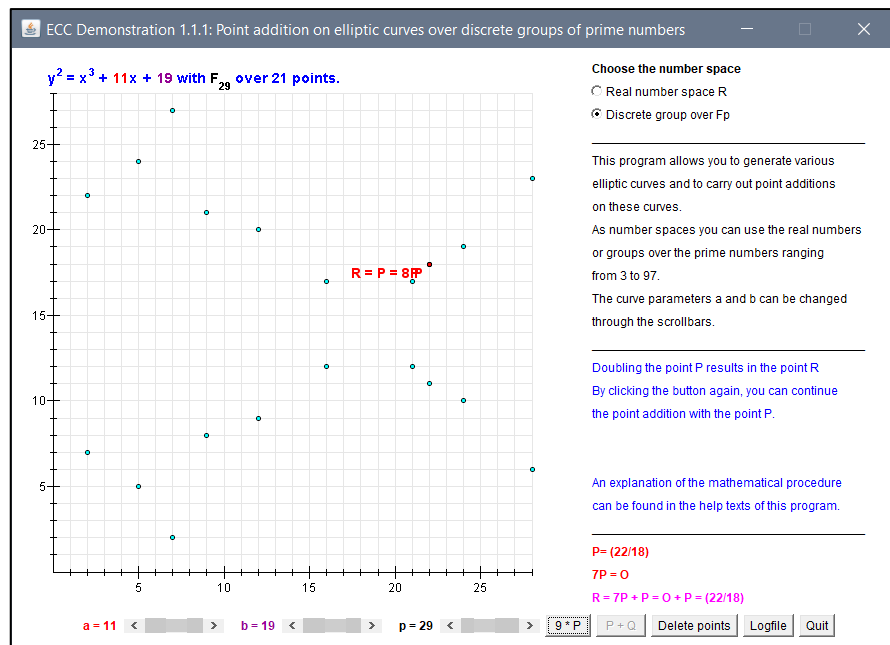
Подписание:

Выбрана эллиптическая кривая с параметрами $a = 11, b = 19, p = 29$.



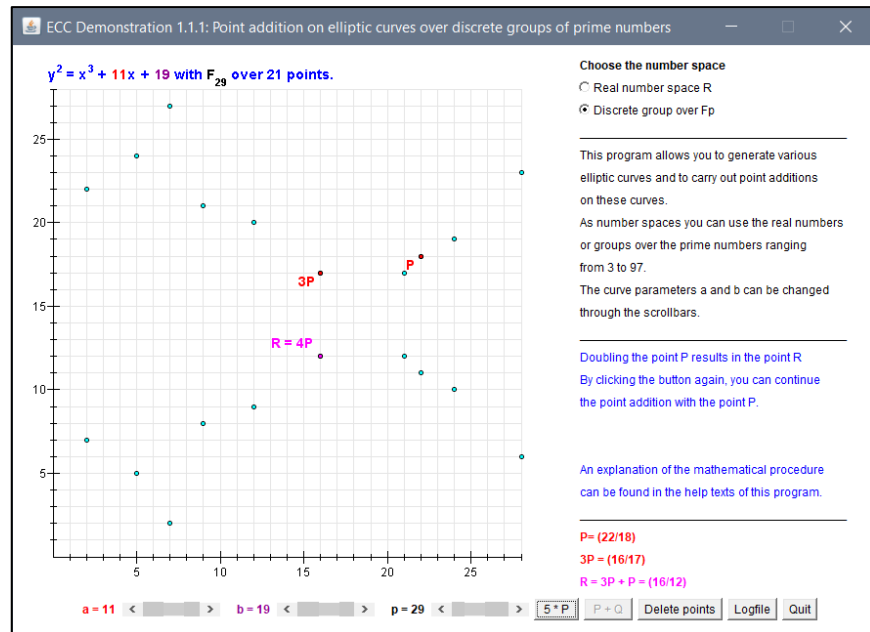
Выбрана точка P (в лекции $e_1 = (x_1, y_1) = (22, 18)$) на плоскости.

Определено q перебором, пока не будет $q \times (x_1, y_1) = 0$ (q – порядок циклической подгруппы группы точек эллиптической кривой). $q = 7$.



Выбрано число $d = 4$.

Найдена точка $e_2 = d \times e_1 = (16, 12)$



Полученный открытый ключ: $(11, 19, 29, 7, (22, 18), (16, 12))$.

Пусть $r = 3$

$$P = r \times e_1 = 3 \times (22, 18) = (16, 17)$$

$$S_1 = u \bmod q = 16 \bmod 7 = 2$$

Пусть $h(M) = M = 70$, тогда:

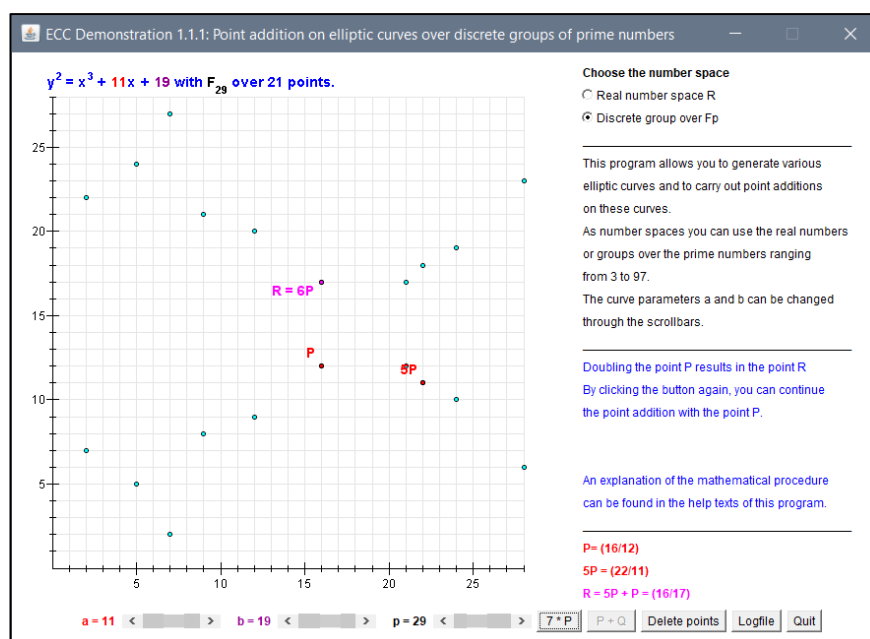
$$S_2 = (h(M) + d \times S_1) \times r^{-1} \bmod q = (70 + 4 \times 2) \times 5 \bmod 7 = 5$$

Проверка:

$$A = h(M) \times S_2^{-1} \bmod q = 70 \times 3 \bmod 7 = 0$$

$$B = S_2^{-1} \times S_1 \bmod q = 3 \times 2 \bmod 7 = 6$$

$$\begin{aligned} T(x, y) &= A \times e_1 + B \times e_2 = 0 \times (22, 18) + 6 \times (16, 12) \\ &= (16, 17) \end{aligned}$$



$V = 16 \bmod 7 = 2 = S_1$ — подпись действительна

Схемы цифровой подписи на эллиптических кривых

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

Структура сертификата



Сертификат, полученный в CrypTool

```

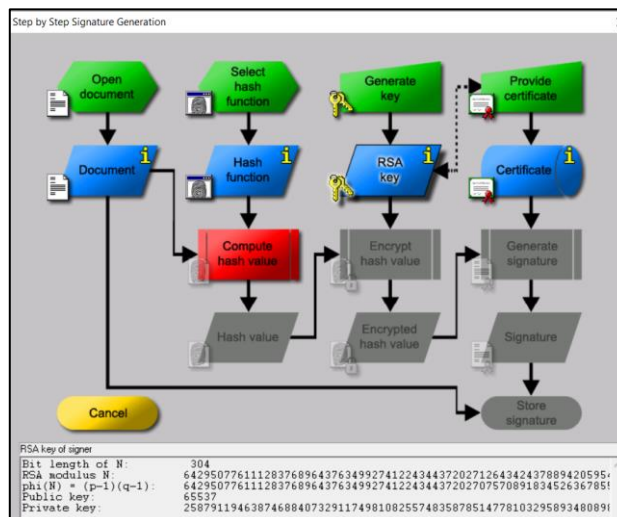
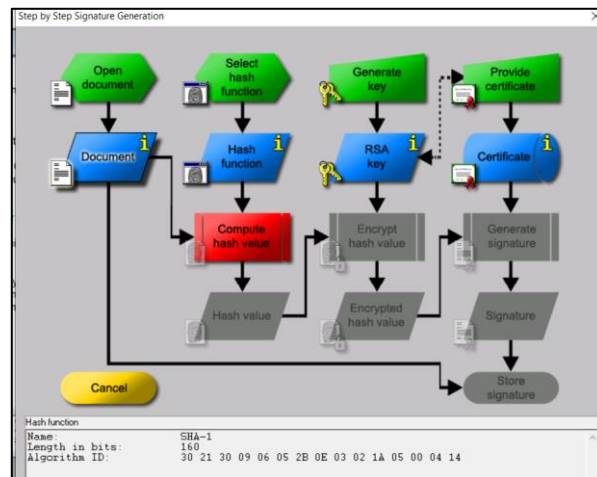
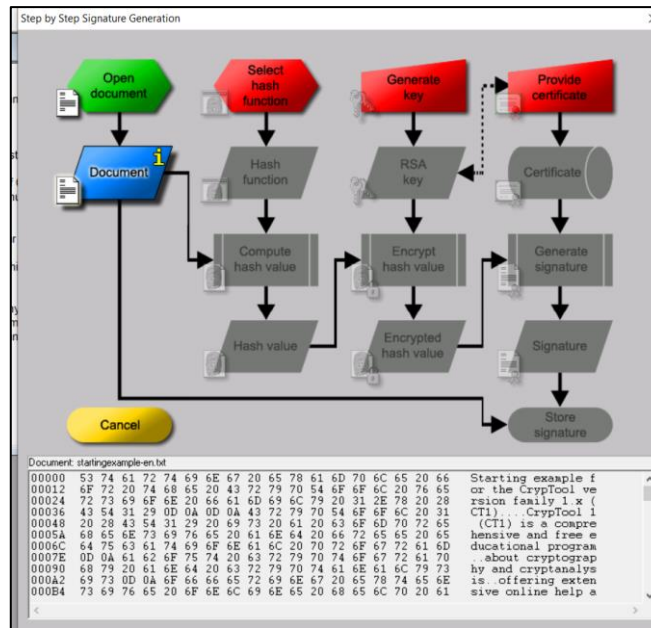
*Безымянный – Блокнот
Файл Правка Формат Вид Справка
Version: 2 (X.509v3-1996)
SubjectName: CN=KONSTANTIN KIREEV [1639132615], DC=cryptool, DC=org
IssuerName: CN=CryptTool CA 2, DC=cryptool, DC=org
SerialNumber: 58:09:F6:39:EC:87:5E:81
Validity - NotBefore: Fri Dec 10 13:36:57 2021 (211210103657Z)
NotAfter: 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 FFE63688 2130CED2 08F8E3CA D0E6826F
10 68875FC4 171D249D 63782C98 64D86D3D
20 459D0547 E35F601C 9EC219E5 38F15F10
30 3ABEEBA9 879D133E BE23DE6C 87D96805
40 88859988 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 70EFC42 1CEE8EDF D1AD0878 C9E18E10
C0 52053344 9E2605F6 4D27D46A 2B798B2E
D0 A77242D0 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 C5735B87 80C964AF B5C99CAE 39590584
20 30F68F48 58464F7A 00EA5956 A36D9A24
30 5167851B 8162E18E 3885A851 0024DAF5
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 A080A45F 73800016 880835B0
A0 408068DC 7360F903 456D8964 9D18DA8B
B0 6E8806A8 A863CD14 C49A31CF 048FFA68
C0 E89A7DC6 96E70E39 1A6F133C 6D69D4BD
D0 FF09F587 58FE30F7 3832CA4A 98E8B7EA
E0 57775EE8 326FAAEE 5C06801D D067500C
F0 48D95F39 72741718 0E01D0D0 0A826DC4
Certificate Fingerprint (MD5): 42:77:08: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

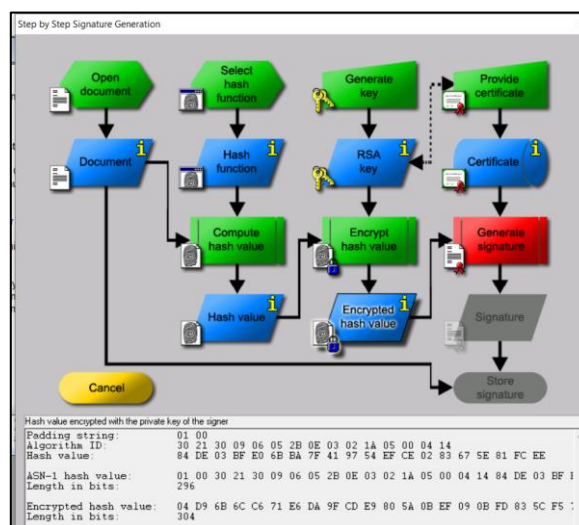
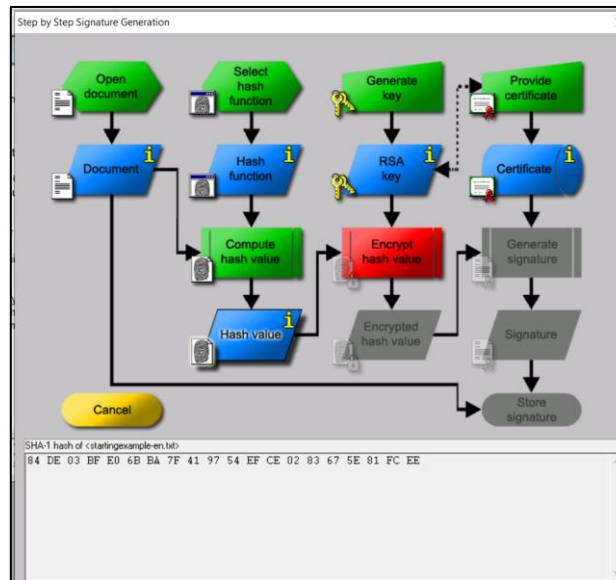
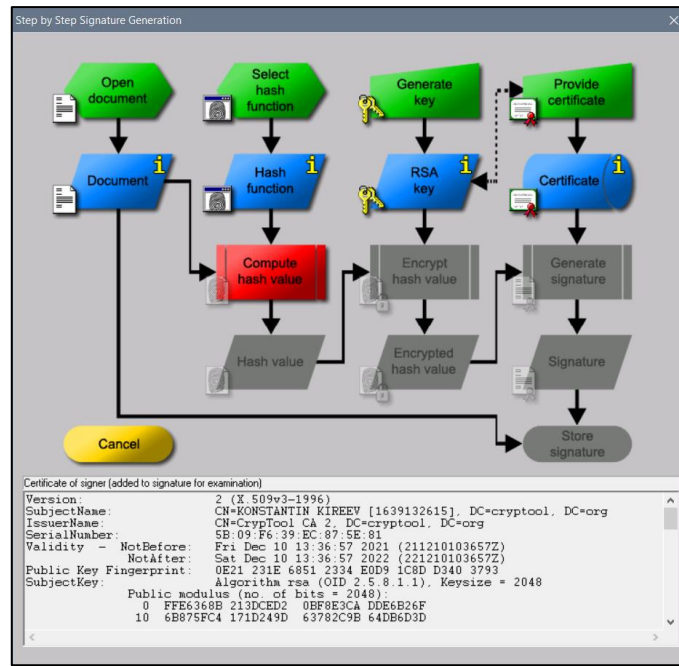
```

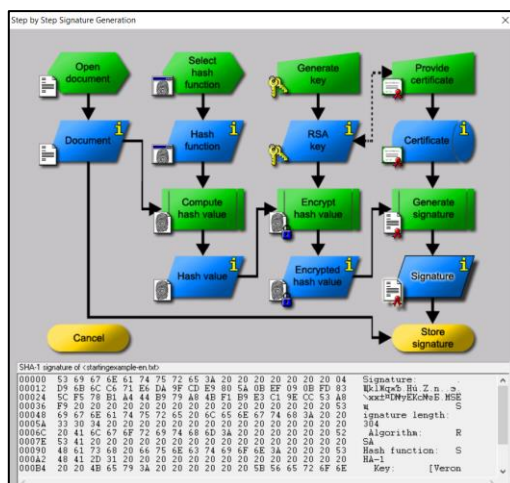
Между сертификатами из лекции и сгенерированным в программе Cryptool 1 есть несколько отличий в структуре:

- 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] |
```

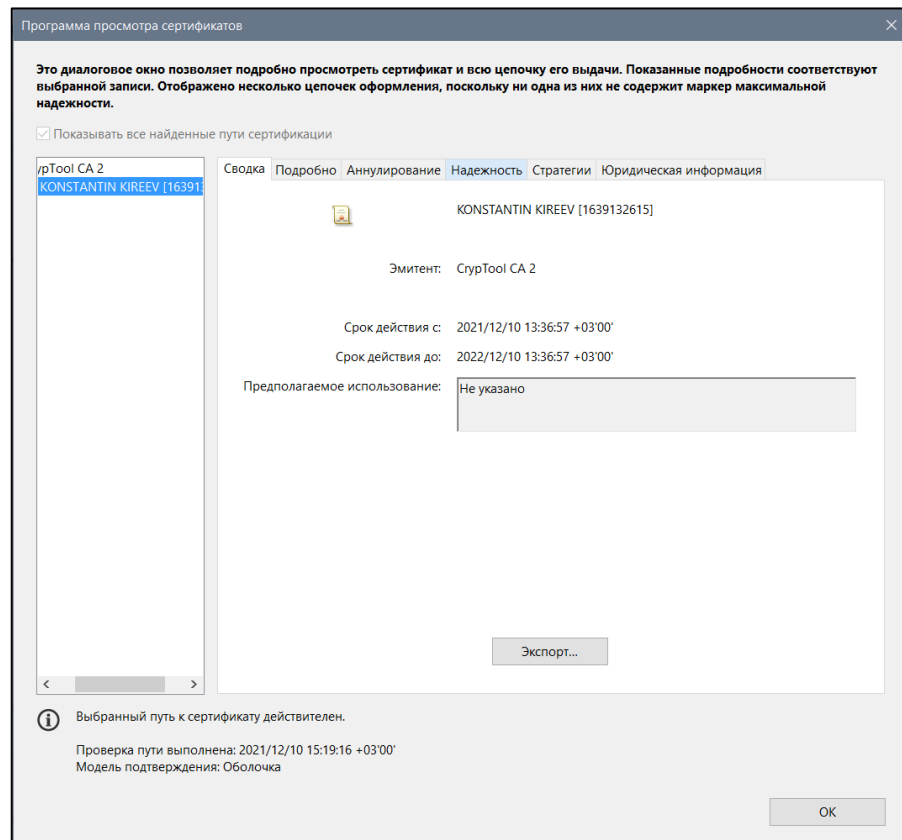
Signature: Algorithm sha1WithRSAEncryption (OID 1.3.14.3.2.29), NULL

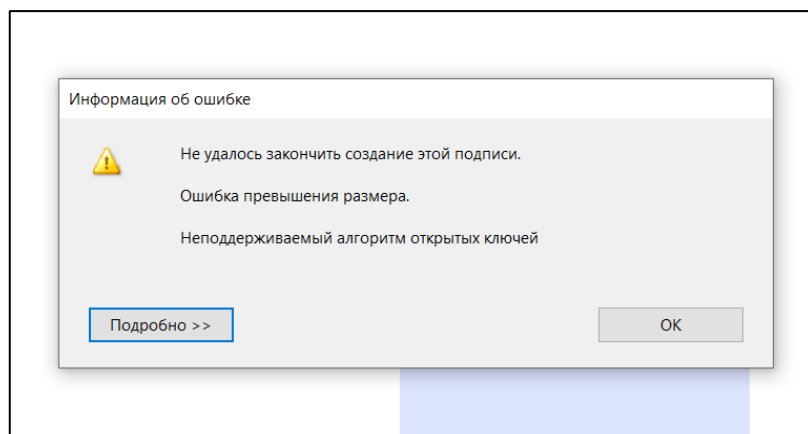
0	24AA9B69	EA33EC48	703A13BD	5318DF2C
10	C5735BB7	80C964AF	B5C99CAE	39590584
20	30F6BF48	58464F7A	00EA5956	A36D9A24
30	5167851B	B162E1BE	38B5A851	6024DAF5
40	D7692CF7	2624A27F	0A88C84E	F7B31938
50	D54882F8	30DCC8F3	13349042	733BAC64
60	303322F1	1F244273	C7286D51	9FA1CAE3
70	FA8DEA17	94CA5528	A9E9C8CC	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

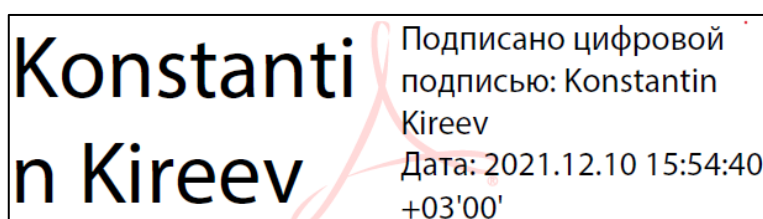
1. Сконвертируйте отчет в формат pdf.
2. Экспортируйте ранее созданный сертификат ключевой пары *RSA Digital Signatures/PKI->PKI/Generate...->Export PSE(#PKCS12)*.
3. Откройте pdf-версию отчета и попытайтесь подписать с использованием этого сертификата.



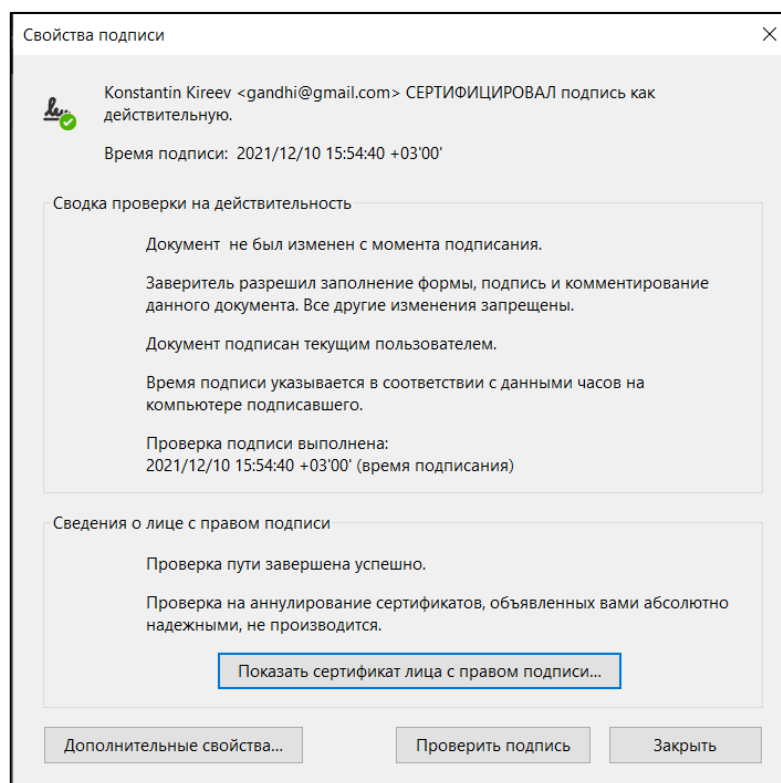


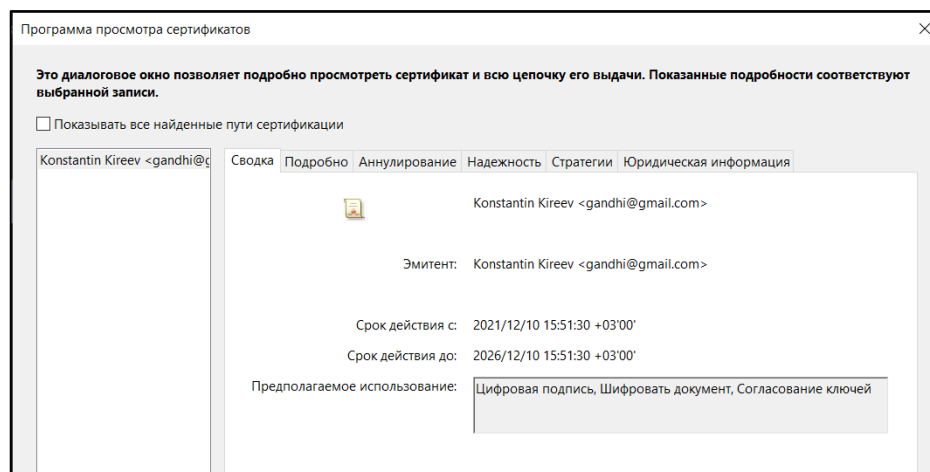
Подписать отчет сертификатом RSA не удалось.

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

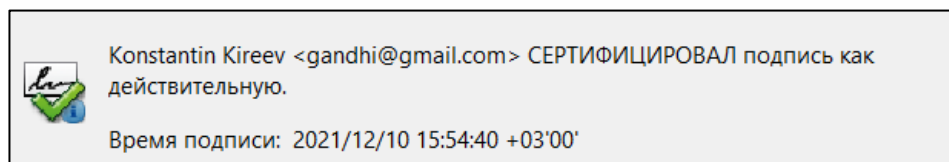


5. Сохраните скриншоты свойств подписи и сертификата.





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



Выводы

- Изучены механизмы генерации ключевых пар для различных алгоритмов.
 - Алгоритм RSA генерирует пары (e, n) – открытый ключ и d – закрытый ключ на основе двух больших простых чисел p и q , которые впоследствии должны быть уничтожены.
 - Алгоритм DSA генерирует пары (e_1, e_2, p, q) – открытый ключ и d – закрытый ключ на основе простого числа p (длина от 512 до 1024 бит), q (такого, что $(p - 1) = 0 \bmod q$) и d .
 - Алгоритм ESDA генерирует пары (a, b, e_1, e_2, p, q) – открытый ключ и d – закрытый ключ на основе произвольно выбранной эллиптической кривой $E_p(a, b)$, где p – простое число, произвольно выбранной точки на данной кривой, d , простое число q (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 RSA, однако DSA требуется больше времени на проверку целостности.

- Изучен алгоритм формирования и проверки подписи ECDSA, основанный на эллиптических кривых.

Открытый ключ представляет собой пару (a, b, q, p, e_1, e_2) , где a, b, p – параметры, задающие определённую эллиптическую кривую, e_1 – произвольная точка на кривой, q – порядок циклической подгруппы группы точек

эллиптической кривой, такой, что для некоторой точки $e_1 = (x_1, y_1)$, лежащей на кривой, верно: $q \times (x_1, y_1) = 0$; $e_2 = d \times e_1$, где d – 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.