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 электронной подписи

Студент гр. 0303 _____ Болкунов В.О.

Преподаватель _____ Племянников А. К.

Санкт-Петербург
2023

Цель работы.

исследовать алгоритмы создания и проверки электронной подписи, алгоритмы генерации ключевых пар для алгоритмов электроной подписи RSA, DSA, ECDSA и получить практические навыки работы с ними.

Порядок выполнения работы

1. **Генерация ключевых пар**

   1. Перейти к утилите «Digital Signatures/PKI –> PKI/Generate…».

   2. Сгенерировать ключевые пары по алгоритмам RSA-2048, DSA-2048, EC-239. Зафиксировать время генерации в таблице.

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

2. **Процессы создания и проверки электронной подписи**

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

   2. Задать хеш-функцию и другие параметры электронной подписи.

   3. Создать подписи, используя закрытые ключи, сгенерированные в предыдущем задании. Зафиксировать время создания электронной подписи для каждого ключа (опция Display signature time должна быть включена)

   4. Сохранить скриншот любой электронной подписи с помощью приложения Digital Signatures/PKI –> Extract Signature.

   5. Выполнить процедуру проверки любой подписи Digital Signatures/ PKI –> Verify Signature для случаев сохранения и нарушения целостности исходного текста. Сохранить скриншоты результатов

3. **Создание и проверка электронной подписи на основе эллиптических кривых**

   1. Выполнить процедуру создания подписи Digital Signatures/PKI –> Sign Document… алгоритмом ECSP-DSA в пошаговом режиме (Display

inter. results = ON). Зафиксировать скриншоты последовательности шагов.

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

3. Проверить лекционный материал по ECDSA, создав и проверив подпись сообщения M (принять M = h(M)) приложением Indiv.Procedures –> Number Theory… –> Point Addition on EC.

4. Демонстрация процесса подписи в среде PKI

1. Запустить демонстрационную утилиту «Digital Signatures/PKI –> Signature Demonstration…».

2. Получить сертификат ключа проверки электронной подписи (открытого ключа) на ранее сгенерированную ключевую пару RSA-2048.

3. Выполнить и сохранить скриншоты всех этапов создания электронной подписи документа.

4. Сохранить скриншот полученного сертификата ключа проверки этой электронной подписи

5. Подписание своего отчета

1. Сконвертировать отчет в формат pdf.

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

3. Открыть pdf-версию отчета и попытаться подписать с использованием этого сертификата.

4. Создать собственный самоподписанный сертификат в среде Adobe Reader и использовать его для подписи отчета.

5. Сохранить скриншоты свойств подписи и сертификата.

6. Внести изменения (маркеры, комментарии) в отчет и проверить подпись.

Выполнение работы.

1. Генерация ключевых пар

С помощью утилиты Digital Signatures/PKI в среде CrypTool (рис. 1) были сгенерированы ключевые пары для алгоритмов RSA-2048, DSA-2048 и EC-239. Результаты генераций для данных алгоритмов представлены соответственно на рисунке 2, листинге 1 и рисунке 3.

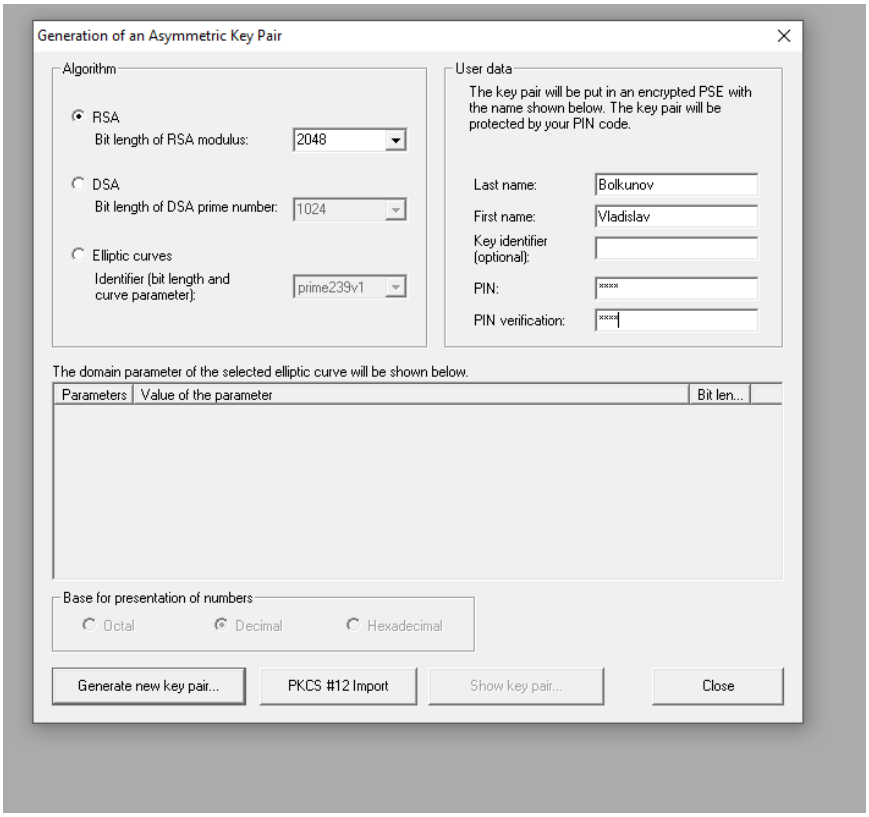

Рисунок 1: генерация ключевых пар

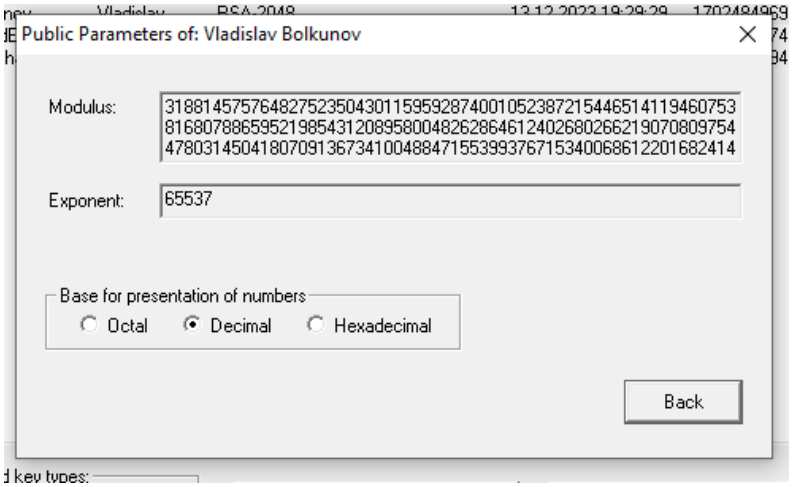

Рисунок 2: Результаты генерации ключей RSA

Листинг 1. Результат генерации ключей DSA

```
DSA prime p (no. of bits = 2048):
  0  FFDC0771 72D4DA49  F1F27DFA 42BC9CFB
 10  F8D6269D 95291C4B  847DBE89 75D50513
 20  A4E123CE 16E279F4  25DE071C 8663035D
 30  2273718E F83EE1A4  E120ED03 BD29F2DA
 40  7F6D3FB2 55F6E01E  F30690FE 2F21C022
 50  E2F91BD4 C15CE04E  1E2D2173 D61A1B59
 60  979D46C2 FDBA14BB  560202DD 7065983C
 70  F4BEA7C7 B37B0587  37F703CA 3555090B
 80  667B90B8 FB64BFDF  7D1A82A1 CA6F089E
 90  A85151A9 61A1576C  E2C73DD4 5ABEE67B
 A0  A475DADF 831CBF5B  304E8CA4 867C141A
 B0  0F4332B2 EDED7767  A2699D63 A3820FCF
 C0  C56E09B9 3468AC26  C4379DDF B19C3BBD
 D0  F46A5E87 69A6ABB2  13F18A0B 88594528
 E0  062FFF4F 124F3AFD  8CD622CC 5E1806BF
 F0  9592F224 85A78604  8DE8D0EC 3735CF3D
DSA prime q (no. of bits = 160):
  0  BD8365E1 132A64F2  A8433927 D0927D6F
 10  D8C8C17D
DSA base g (no. of bits = 2048):
  0  B2252E0C 2B8C67F2  040B3263 CDFA9CCF
 10  C2D3EADA 4F8C5A33  C54D8490 CF4C6240
 20  3DDAE0CB 1B65795B  2E7ABF3B B627CE2F
 30  60D56CFA 4BF9908C  A12B0BA4 FEB8F1F3
 40  01FBE27F 17465CA7  983A15E2 7FA5AFC9
 50  482C7DE8 B4AA254F  7AD193E0 04E96219
 60  466EFB90 81FF70C9  5D9F6655 347EBA28
 70  3FF19807 F6418B1C  64BAD8B9 A0DB4012
 80  E4E4BB7C FDE0439E  3F30B922 E31D03D7
 90  DC827C44 A01DA5D4  7A43DABB DAC57AE5
 A0  898C1216 AE3AC5C4  E150C781 2A520000
 B0  535CEF9C 6929F3AD  DC52EE48 F364827E
 C0  8306ADEB 96788371  B4778562 9B7C9CCA
 D0  9835CA38 65F61DA1  47E51E93 0F1F3D7A
 E0  7569A65E 89CC14C3  A7D90D6B 6EDA5CAA
 F0  D7120653 A170737D  055E0CF8 18D2ABDC
Public y (no. of bits = 2048):
  0  97F700E9 3C888BF6  0EB89742 C7BDD7B9
 10  EAF2C89E CFFBB280  7D9F2017 8A7FAE3C
 20  49FE1F93 A4543C9A  BD483D53 F1A45ACA
 30  76150F2E 50C1D725  A357F338 8B4BDB89
 40  3438BCEE 8EC01EA7  F37E852B 2C14D603
 50  D9738538 DC968D51  2FE90176 267F1625
 60  2C4295F1 804D1D4A  70895D85 0833A3FE
 70  C209C561 609D6983  AC3F6FEA 5A383A7A
 80  E73ABC89 EF029239  78B0BEA5 B4935B9B
 90  2D336B95 F63EE9EC  51A37F5C CD7ADA09
 A0  633B191B 17F37CAA  8CB8F7A2 CC23EAF9
 B0  76A2CB5C AC1BA106  D3BF2EC9 D5E9CA0F
```

```
C0  FB2A0845 B2262933  3EDC1D87 D30A7B85
D0  475A8D2D 9C64CD0B  5DED8EB1 3A028233
E0  35C35485 6CBB13F4  CE360230 C149B1A9
F0  88872B7B 4558B407  1A5676ED 019663D8
```

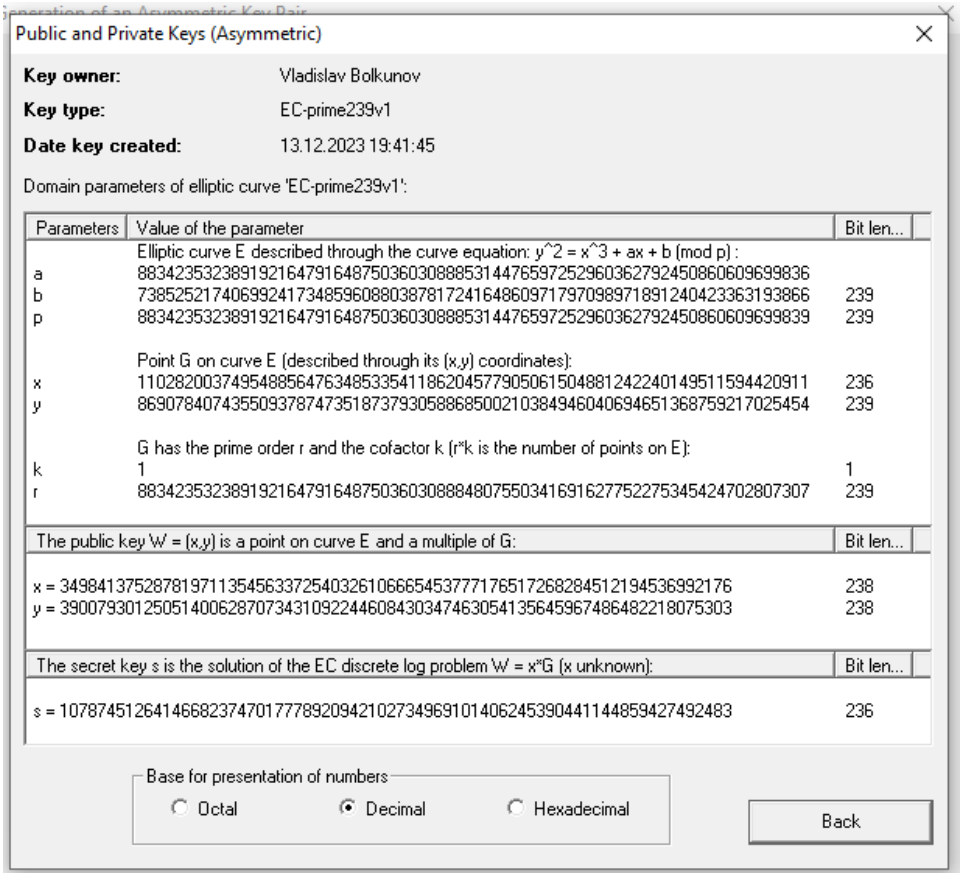

Рисунок 3: Результаты генерации EC-239

Итого программой было затрачено следующее время для генерации ключевых пар рассматриваемых алгоритмов:

Алгоритм	**Время, с**
RSA-2048	0.644
DSA-2048	3.694
EC-239	0.010

2. Процессы создания и проверки электронной подписи

В среде CrypTool был введён достаточно большой текст (около 6000 знаков). Для данного текста были выбраны настройки цифровой подписи (рис.

4), после чего с помощью полученных на предыдущем шаге ключей были созданы цифровые подписи. Полученные подписи и их время создания для ключей сгенерированных с помощью алгоритмов RSA-2048, DSA-2048 и EC-239 представлены соответственно на рисунках 5-7.

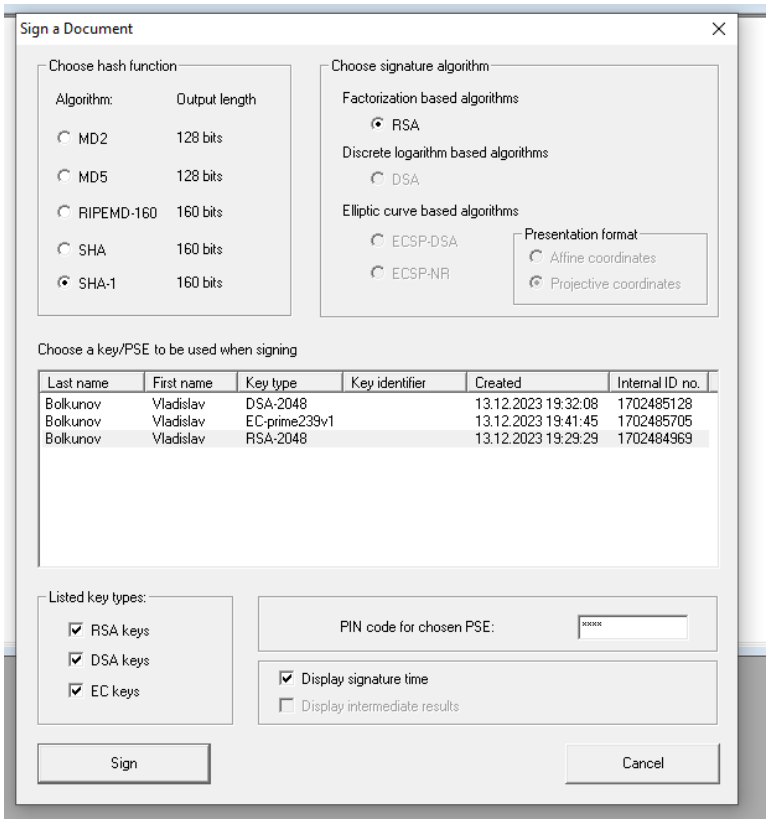

Рисунок 4: параметры цифровой подписи

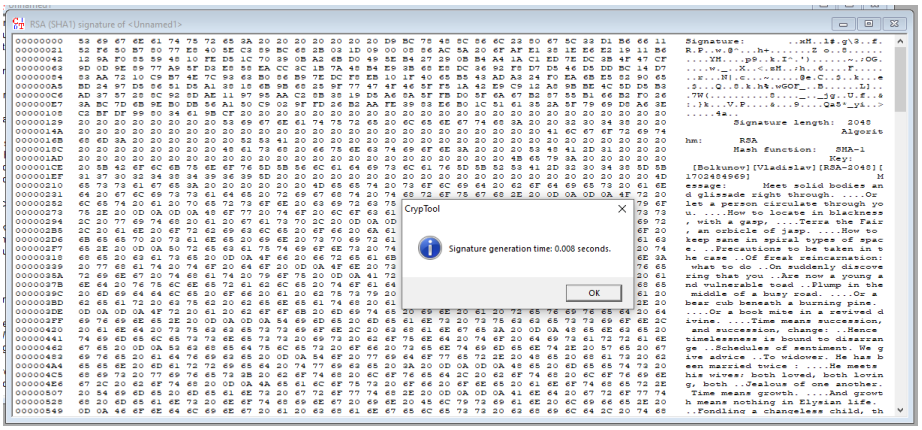

Рисунок 5: подпись с ключом RSA-2048

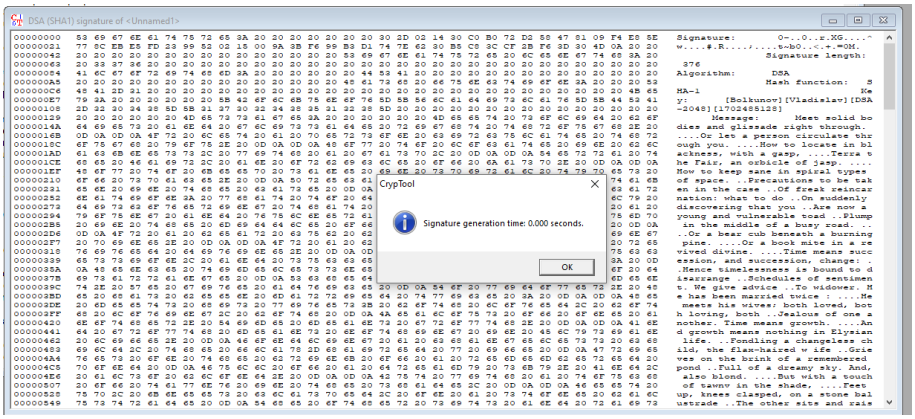

Рисунок 6: подпись с ключом DSA-2048

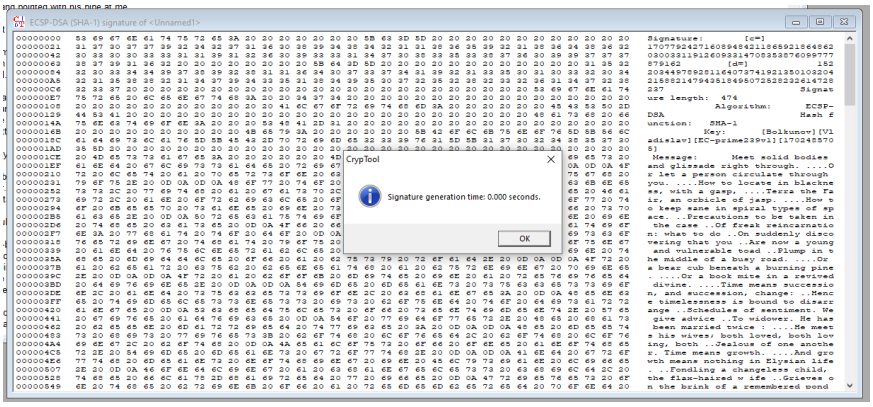

Рисунок 7: подпись с ключом EC-239

Итого программой было затрачено следующее время для создания электронной подписи с помощью ключей для RSA-2048, DSA-2048 и EC-239:

Алгоритм	**Время, с**
RSA-2048	0.008
DSA-2048	0.000
EC-239	0.000

Сигнатура подписи, полученной с помощью ключей RSA-2048 представлена на рисунке 8.

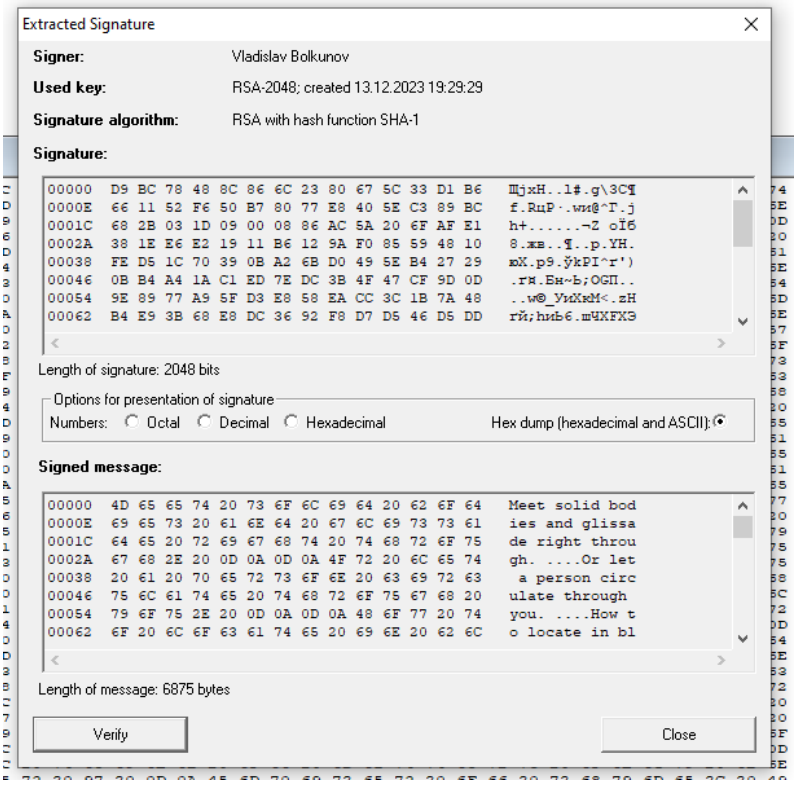

Рисунок 8: подпись для RSA-2048

На рисунках 9, 10 представлены результаты проверки электронной подписи для исходного подписанного документа, и для модифицированного документа. Изменение документа привело к ошибке при подтверждении подписи.

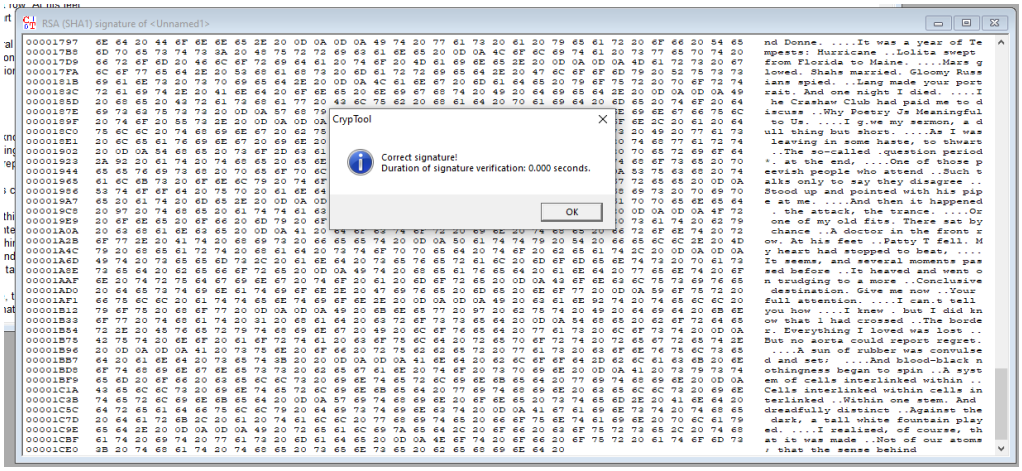

Рисунок 9: результат проверки цифровой подписи для исходного текста

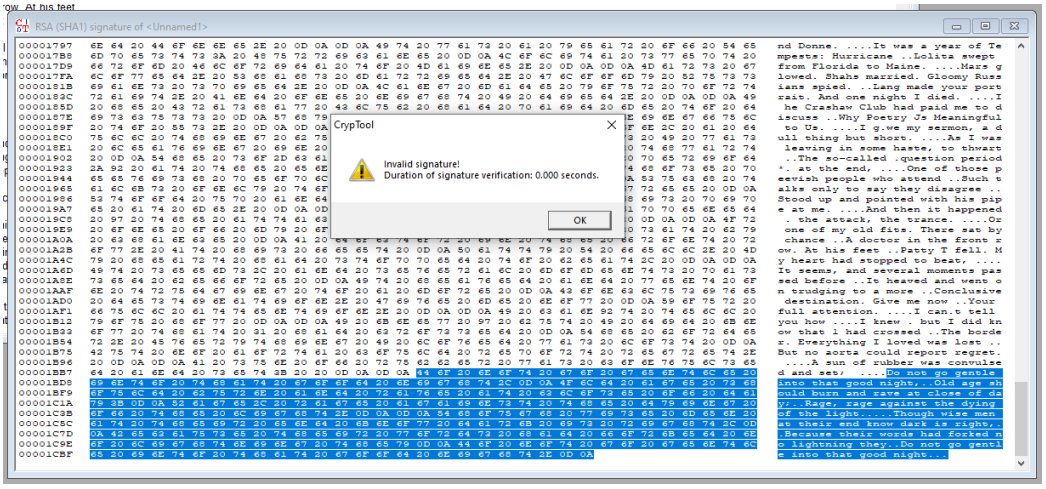

Рисунок 10: результат проверки цифровой подписи для модифицированного текста

3. Создание и проверка электронной подписи на основе эллиптических кривых

В пошаговом режиме была создана электронная подпись ECDSA на основе сгенерированного ключевой пары для алгоритма EC-239. Результаты выполнения шагов представлены в листингах 2-7.

Листинг 2. Параметры алгоритма и секретный ключ.

```
Signature originator:  Vladislav Bolkunov

Domain parameters to be used 'EC-prime239v1':

 a  = 883423532389192164791648750360308885314476597252960362792450860609699836
 b  = 738525217406992417348596088038781724164860971797098971891240423363193866
 Gx = 110282003749548856476348533541186204577905061504881242240149511594420911
 Gy = 869078407435509378747351873793058868500210384946040694651368759217025454
 k  = 1
 r  = 883423532389192164791648750360308884807550341691627752275345424702807307

Secret key s of the signature originator:

 s  = 107874512641466823747017778920942102734969101406245390441144859427492483

Chosen signature algorithm: ECSP-DSA with hash function SHA-1

Size of message M to be signed: 6875 bytes
```

Листинг 3. Значение хэш-функции сообщения

```
Calculate a 'hash value' f (message representative) from message M, using the chosen hash function SHA-1.

 f  = 1449303291998965672148937174666272885277250115845
```

Листинг 4. Генерация секретного ключа и точки на кривой

```
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 elliptic curve):

 u  = 875801560903898206384914426884088839128919098279093895849187907930691880
 Vx  = 362010413484323404277780207556081300510646835676468540156439158276563417
 Vy  = 774518145395845626655029452127437863826542301905094750891555260979673760
```

Листинг 5. Вычисление x-координаты точки на кривой

```
Convert the group element Vx (x co-ordinates of point V on elliptic curve) to the number i:

 i  = 362010413484323404277780207556081300510646835676468540156439158276563417
```

Листинг 6. Вычисление первой части подписи S1

```
Calculate the number c = i mod r (c not equal to 0):

 c  = 362010413484323404277780207556081300510646835676468540156439158276563417
```

Листинг 7. Вычисление второй части подписи S2

```
Calculate the number d = u^(-1)*(f + s*c) mod r (d not equal to 0):

 d  = 1308729950379414386882555367795697712256199859035098132452633986241263 1
```

Аналогично в пошаговом режиме была запущена проверка созданной электронной подписи. Результаты выполнения проверки представлены в листингах 8-13.

Листинг 8. Полученные данные

```
Signature originator:  Vladislav Bolkunov

Domain parameters to be used 'EC-prime239v1':

 a  = 883423532389192164791648750360308885314476597252960362792450860609699836
 b  = 738525217406992417348596088038781724164860971797098971891240423363193866
 Gx = 110282003749548856476348533541186204577905061504881242240149511594420911
 Gy = 869078407435509378747351873793058868500210384946040694651368759217025454
 k  = 1
```

```
 r  = 883423532389192164791648750360308884807550341691627752275345424702807307

Public key W=(Wx,Wy) (W is a point on the elliptic curve) of the signature originator:

 Wx  = 34984137528781971135456337254032610666545377717651726828451219453699217 6
 Wy  = 39007930125051400628707343109224460843034746305413564596748648221807530 3

Chosen signature algorithm: ECSP-DSA with hash function SHA-1

Size of message M to be signed: 6875 bytes

Bit length of c + bit length of d = 471 bits
```

Листинг 9. Вычисление хэш-функции

```
Calculate a 'hash value' f (message representative) from message M, using the chosen hash function SHA-1.

 f  = 144930329199896567214893717466627288527725011584 5
```

Листинг 10. Проверка на корректности значений c, d

```
If c or d does not fall within the interval [1, r-1] then the signature is invalid:

 c and d fall within the required interval [1, r-1].
```

Листинг 11. Вычисление h, h1, h2

```
Calculate the number h = d^(-1) mod r:

 h  = 113624498853029396660569355428753753323961896798945532297239772703635639

Calculate the number h1 = f*h mod r:

 h1 = 753962013956949361236203862995877865852417974654016038543489509150441928

Calculate the number h2 = c*h mod r:

 h2 = 365797842077243264209568228574614715199428816360354320631078115570380901
```

Листинг 12. Нахождение точки на кривой

```
Calculate the elliptic curve point P = h1 G + h2 W
(If P = (Px, Py) = (inf, inf) then the signature is invalid):

 Px = 362010413484323404277780207556081300510646835676468540156439158276563417
 Py = 774518145395845626655029452127437863826542301905094750891555260979673760
```

Листинг 13. Сравнение координаты точки.

```
Convert the group element Px (x co-ordinates of point P on elliptic curve) to the number i:

 i  = 362010413484323404277780207556081300510646835676468540156439158276563417
```

Calculate the number c' = i mod r:

c' = 362010413484323404277780207556081300510646835676468540156439158276563417

If c' = c then the signature is correct; otherwise the signature is invalid:
Verify results by comparing the two numbers c' and c.

С помощью утилиты **Point Addition on EC** был произведён процесс формирования подписи.

Была выбрана кривая $\mathbf{E_{83}(10,\ 13)}$, с порядком группы $\mathbf{q = 73}$, и точка e_1 = **(14, 65)**, и закрытый ключ d = **7**, на рисунке 11 показано вычисление точки $e_2 = d * e_1 = (\mathbf{3, 53})$

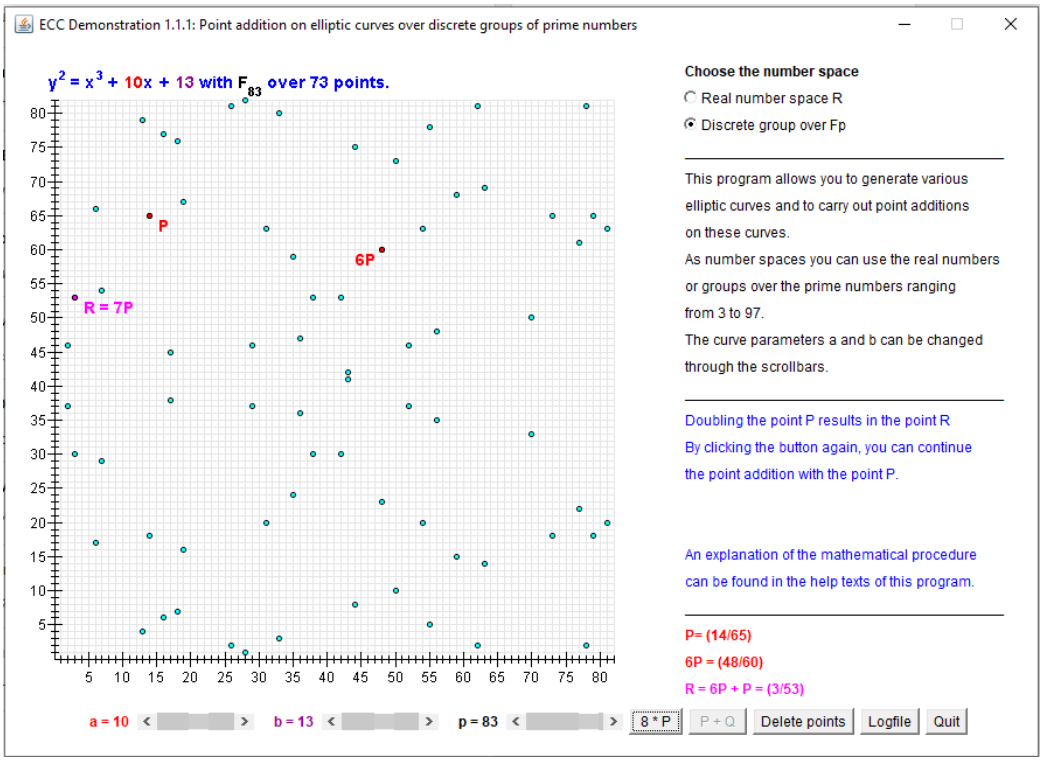

Рисунок 11. Вычисление e_2

Возьмём случайное $r = \mathbf{11}$ и вычислим точку $P(u, v) = r * e_1 = (\mathbf{17, 38})$ *(рис. 12)*

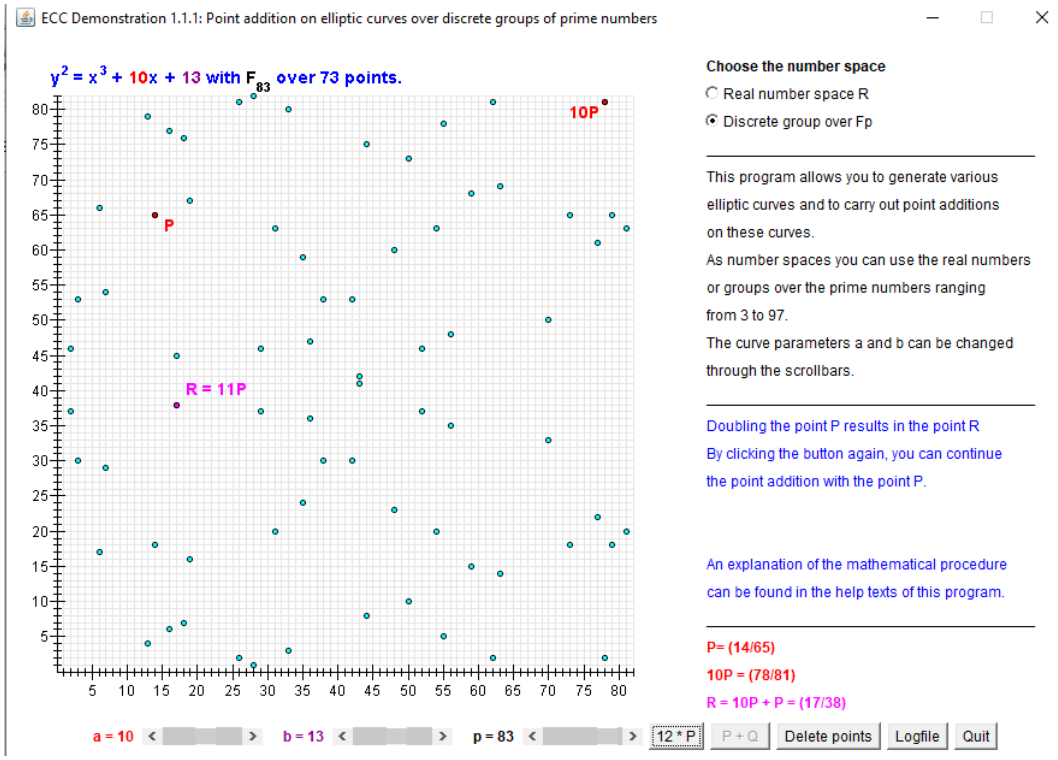

Рисунок 12. вычисление r * e_1

Вычислим $S_1 = u \ mod \ q = 17 \ mod \ 73 = \mathbf{17}$.

Допустим дайджест сообщения $h(M) = \mathbf{33}$.

Тогда вычислим

$$S_2 = r^{-1}(h(M) + d * S_1) \ mod \ 73 = 20 * (33 + 7 * 17) \ mod \ 73 = \mathbf{47}$$

Для проверки созданной подписи подписи найдём коэффициенты A, B.

$$A = h(M') * S_2^{-1} \ mod \ q = 33 * 14 \ mod \ 73 = \mathbf{24}$$

$$B = S_2^{-1} * S_1 \ mod \ 83 = 14 * 17 \ mod \ 73 = \mathbf{19}$$

Найдём третью точку: $P' = A * e_1 + B * e_2 =$

$= 24 * (14, 65) + 19 * (3, 53) = (70, 33) + (36, 36) = (\mathbf{17}, \mathbf{38})$

$P = P'$ (рис. 13) следовательно сообщение не модифицировано.

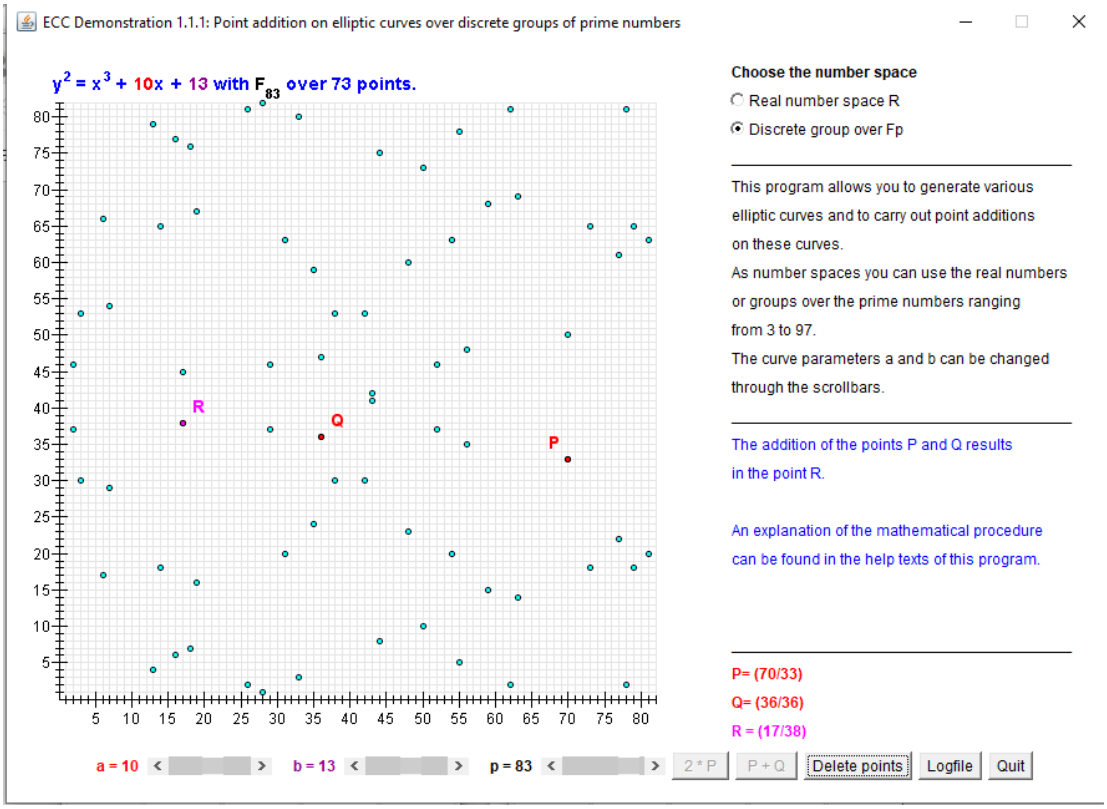

Рисунок 13. Восстановление точки

4. Демонстрация процесса подписи в среде PKI

В среде CrypTool1 был выполнен процесс генерации сертификата и подписи документа. Результататы выполнениея шагов представлены на рисунках 14-16 и в листингах 14-16.

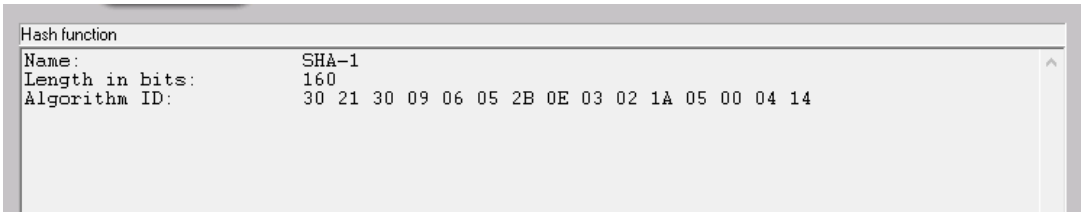

Рисунок 14: выбор хэш-функции

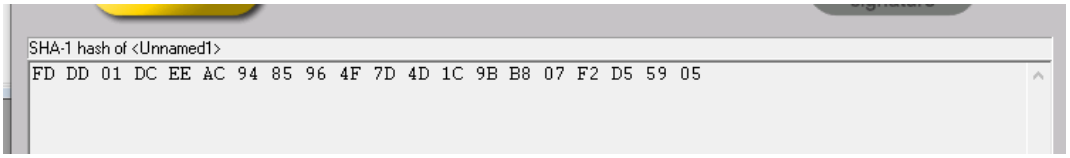

Рисунок 15: вычисление хэш-функции

Листинг 14. Параметры ключей RSA

```
Bit length of N:        2048
RSA modulus N:
318814575764827523504301159592874001052387215446514119460753816807886595219854312089580048262864612402680266219070809754478031450418070913673410048847155399376715340068612201682414440761351242473001646578722585173412237883981863153491093499505499691361408447366901212945821087957480011810747261720940360265523468603270451875252014186619219499800908212785488607500009903393091715272539989853587121138736758258314461383007056369256815762525731944651012021817692532372689298383063730712306163123495097666209045742594845009535383305261411145022039786828732395612722199866074487457644377801279308701581398024749733759178513208958004826286461240268026621907080975447803145041807091367341004884715539937671534006861220168241444076135124247300164657872258517341223788398186315349109349950549969136140844736690121294582108795748001181074726172094036026552346860327045187525201418661921949980090821278548860750000990339309171527253998985358712113873675825831446138300705636925681576252573194465101202181769253237268929838306373071230616312349509766620904574259484500953538330526141114502203978682873239561272219986607448745764437780127930870158139802474973375917851
phi(N) = (p-1)(q-1):
318814575764827523504301159592874001052387215446514119460753816807886595219854312089580048262864612402680266219070809754478031450418070913673410048847155399376715340068612201682414440761351242473001646578722585173412237883981863153491093499505499691361408447366901212945821087957480011810747261720940360265519897466999427491745792337625763324322502930606606414324839473637345346714675511595399963699370029908436250547114969929211135663647328772638684623116636473694090126885326504873391990046371427377083641811595723170483633917846861052790041661485826216202153020514484317950976141734468355503105156054074392306103044
Public key:             65537
Private key:
253589346781889527637147186298074809662631691319580306290639420729943700844039022764510403831345184856482896649738960307783165257470491762803930488675913816837950988944209918999993627759105221765950575011096474396215977976625273423079134123254378342172801027706321518068460678611097174659847780738375269546687927965229002788000921750585065174384054126212548419554443397183898798829505771456682556535765449884750191567672555463934072218903392001295176628750890988864293822182968176336192548470746237052962344385563474879139132894448006772066040279103325370773792434905466423706706673977693530433516466727846010597446 41
```

*Листинг 15. Полученный с*ертификат.

```
Version:                2 (X.509v3-1996)
SubjectName:            CN=Vladislav Bolkunov [1702484969], DC=cryptool, DC=org
IssuerName:             CN=CrypTool CA 2, DC=cryptool, DC=org
SerialNumber:           6A:B3:92:D7:61:0C:06:52
Validity  -  NotBefore:   Wed Dec 13 19:29:39 2023 (231213162939Z)
          NotAfter:   Fri Dec 13 19:29:39 2024 (241213162939Z)
Public Key Fingerprint:   C1DF 3FE6 4D64 0A1D 4650 DC6E 044E 92F4
SubjectKey:             Algorithm rsa (OID 2.5.8.1.1), Keysize = 2048
          Public modulus (no. of bits = 2048):
            0  FC8CBF1F D3523B3D  46DF0B8D 6BBC17DF
           10  2DED65ED FF282BC5  8E8B646E 632F4E4B
           20  C8BC6890 9A59069A  8423F4C7 3CF53475
           30  B776C01F 3DED769B  134B2D0A 84CFDAB8
           40  0EFE1027 EC3DEE51  C7F42366 CF37C57A
           50  B7A82FB2 ED069B9D  8C7CF735 A07E9D68
           60  5F0C8EC7 4F598EAA  BFE97F0A 5282B268
           70  FEE34B9C 46D5CDB8  B1AEEFA1 2696026D
           80  6A28D311 7E186DF6  34C01EFB 78556695
           90  A847568D 9F695B30  ED4F38E2 AA1F1C70
```

```
            A0  FE8810E5 7929CE6F  11D1FAC7 19651CD7
            B0  7D032F80 DCF958B0  2E06AB13 398B7257
            C0  5466646B D5E961E5  BBCE0407 157C10D7
            D0  36EED899 DC629F25  2754BACD F6C57361
            E0  6757677D 891FD118  1F1537E2 BE1CCD28
            F0  C2E01DAF A81EC471  37A55139 91426F1B
           Public exponent  (no. of bits = 17):
             0  010001
Certificate extensions:
Private extensions:
   OID 2.206.5.4.3.2:
      PrintableString:
             |[Bolkunov][Vladislav][RSA-2048][|
             |1702484969]                    |

SHA1 digest of DER code of ToBeSigned:
             0  B9228EE5 135ED9EC  32389B0D 0BE671B5
            10  1047E1C2
Signature:              Algorithm sha1WithRSASignature (OID 1.3.14.3.2.29), NULL
             0  3DECEFAC 1AB53CB9  2B2FD2A4 AB1B3702
            10  5FF1F23C 7AC2BA63  E4C5AAD8 E48048CC
            20  C0457B1C BDCAB221  C8F1C98F BF3EDF94
            30  EC5FC5C6 921D1A57  9CE3D5A0 C8253251
            40  1C92FEF3 B6747247  4C2441A1 D04F8C05
            50  FED59DF6 52EBA861  7ED2F3AC FC9E4259
            60  6425252F D80221B7  0701D4A9 111B9094
            70  F247F2BC A502DF3E  28E708F2 1E828A93
            80  F6BF86D2 6BB029AB  3940FFF6 342B50FA
            90  51388917 7DC56F00  3D52A39F 3BB8CB8F
            A0  84B00B6C 0E3B7081  F58D2C66 7CB82054
            B0  B9778D06 9816DD96  932D77FB 782AC76F
            C0  5E617BA5 4A952F90  0001C928 398B09C7
            D0  2B3F0812 56988B7B  337B5FF3 5230697D
            E0  D37221D8 859AF407  092CB4A5 9336B508
            F0  68933BD2 3959AD97  86AC9684 596419DC
Certificate Fingerprint (MD5):    CC:67:77:16:FA:FE:FA:26:C2:90:A9:05:C5:C5:DC:70
Certificate Fingerprint (SHA-1):  2075 2493 692E E253 ACB1 2589 436E 6C85 EB36 BE6C
```

Листинг 16. Зашифрованный дайджест

```
Padding string:        01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00
Algorithm ID:          30 21 30 09 06 05 2B 0E 03 02 1A 05 00 04 14
Hash value:            FD DD 01 DC EE AC 94 85 96 4F 7D 4D 1C 9B B8 07 F2 D5 59 05
```

ASN-1 hash value: 01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 30 21 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 FD DD 01 DC EE AC 94 85 96 4F 7D 4D 1C 9B B8 07 F2 D5 59 05
Length in bits: 2040

Encrypted hash value: D9 BC 78 48 8C 86 6C 23 80 67 5C 33 D1 B6 66 11 52 F6 50 B7 80 77 E8 40 5E C3 89 BC 68 2B 03 1D 09 00 08 86 AC 5A 20 6F AF E1 38 1E E6 E2 19 11 B6 12 9A F0 85 59 48 10 FE D5 1C 70 39 0B A2 6B D0 49 5E B4 27 29 0B B4 A4 1A C1 ED 7E DC 3B 4F 47 CF 9D 0D 9E 89 77 A9 5F D3 E8 58 EA CC 3C 1B 7A 48 B4 E9 3B 68 E8 DC 36 92 F8 D7 D5 46 D5 DD BC 14 D7 83 AA 72 10 C9 B7 4E 7C 93 63 B0 86 B9 7E DC F8 EB 10 1F 40 65 B5 43 AD A3 24 F0 EA 6B E5 82 90 65 BD 24 97 D5 86 51 D5 A1 38 18 6B 9B 68 25 9F 77 47 4F 46 5F F5 1A 42 E9 C9 12 A8 9B BE 4C 5D D5 B3 AD 37 57 28 8C 92 8D AE 11 97 95 AA C2 8B 38 19 D5 A6 8A 5F FB D0 5F 6A 67 B2 87 55 B1 66 B2 F0 26 3A BC 7D 6B 9E B0 DB 56 A1 50 C9 02 9F FD 26 B2 AA FE 39 83 E6 B0 1C 51 61 35 2A 5F 79 69 D8 A6 3E C2 BF DF 99 80 34 61 9B CF
Length in bits: 2048

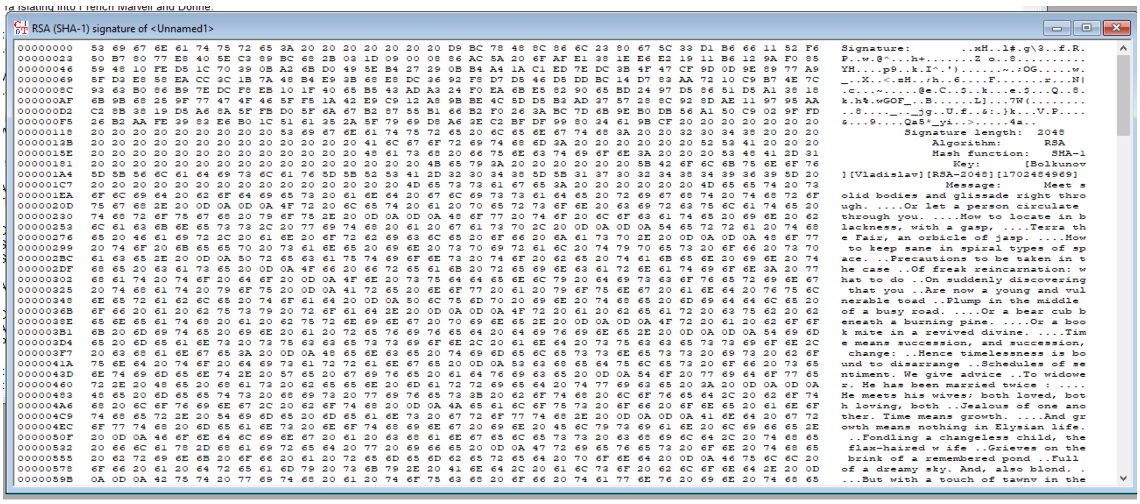

Рисунок 16: подписанный документ

5. Подписание отчета

Данный отчёт был подписан цифровой подписью в приложении Adobe Acrobat reader. На рисунке 17 показан процесс создания подписи, на рисунке 18 показана действующая электронная попдись, на рисунке 19 – её свойства.

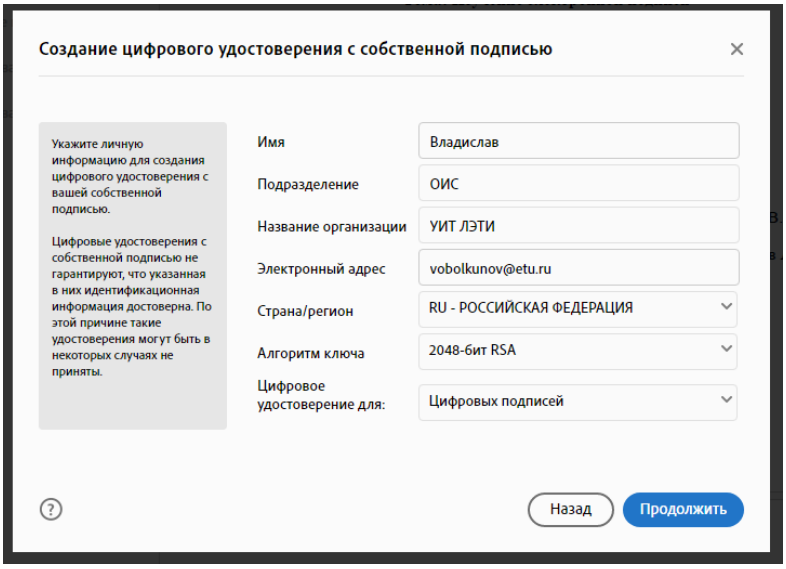

Рисунок 17: создание подписи

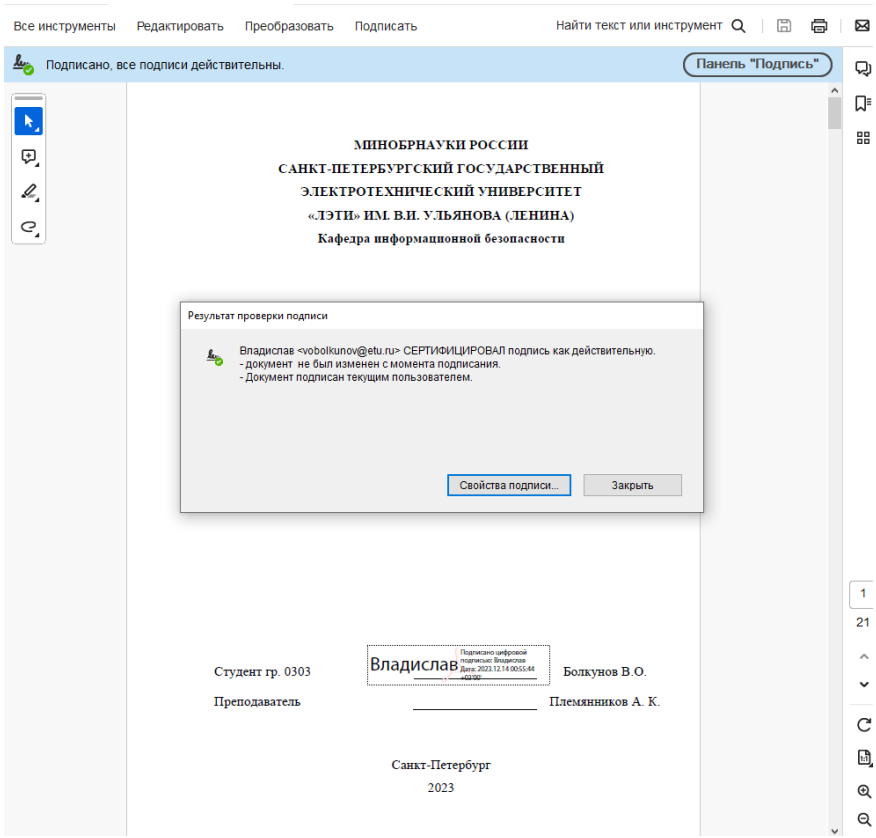

Рисунок 18: подписанный отчёт

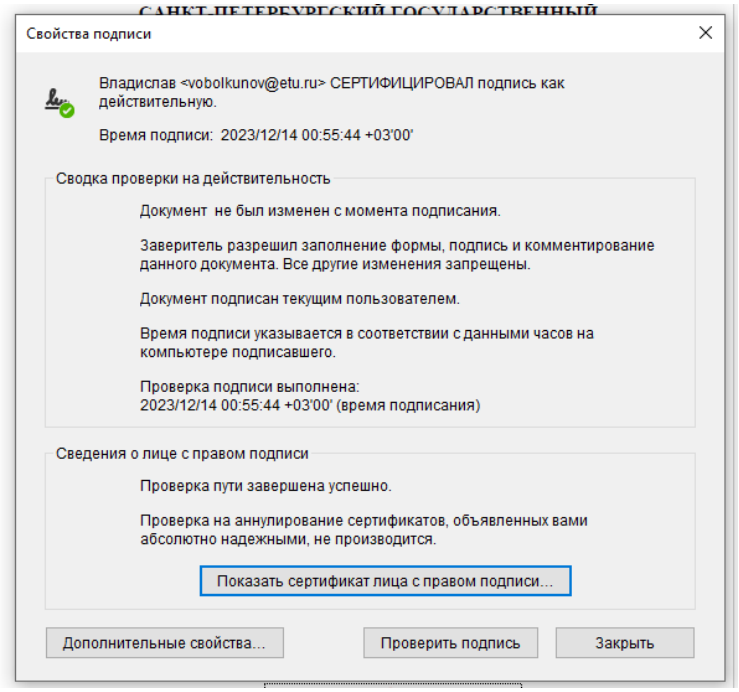

Рисунок 19: свойства подписи

Отчёт был модифицирован, после чего подпись была проверена повторно, приложение показывает соответствующее сообщение (рис. 20)

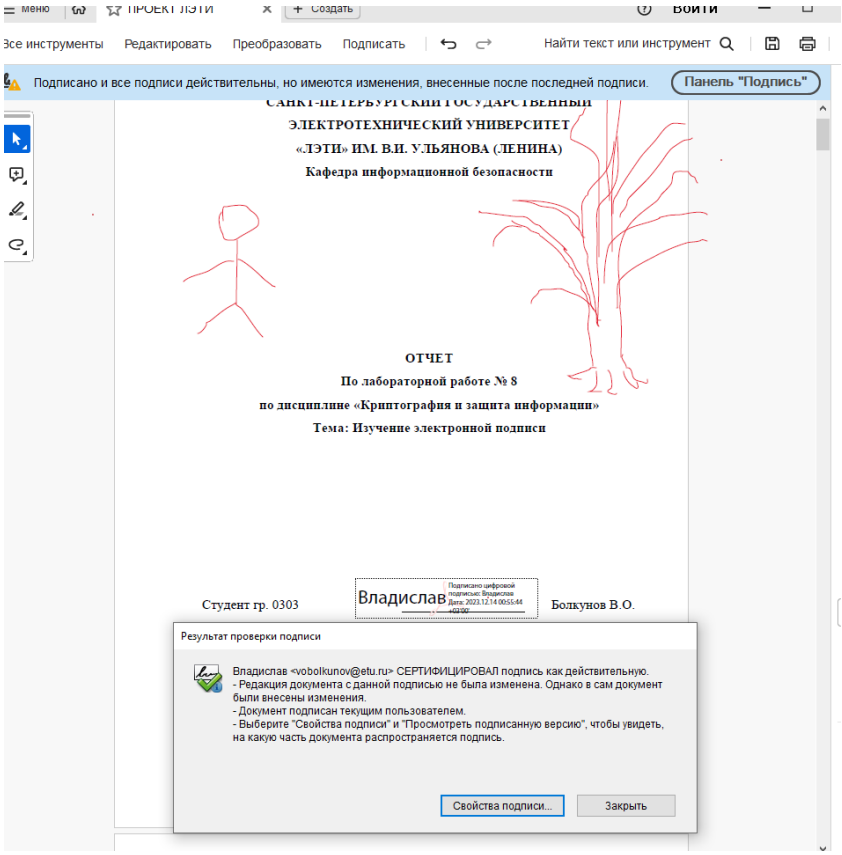

Рисунок 20: отчёт после модификации

Выводы:

1. С помощью среды CrypTool1 были сгенерированы ключевые пары для алгоритмов RSA-2048, DSA-2048, EC-239, и оценена их скорость работы: наилучший результат показал алгоритм EC-239
2. С помощью данных алгоритмов были созданы цифровые подписи на подготовленный документ (текст), и оценено время работы для каждого процесса: лучший результат показали подписи для алгоритмов EC-239 и DSA-2048.

   Для алгоритма EC-239 цифровая подпись была проверена для исходного документа и для модифицированного: в случае с модифицированным документом при проверке подписи была обнаружена модификация.
3. В среде CrypTool1 был исследован протокол электронной подписи ECSP-DSA, основанный на эллиптических кривых. С помощью данного протокола была создана цифровая подпись, после чего проверена. Также был рассмотрен пример формирования и проверки подписи с помощью утилиты Point Addition on EC.
4. Для полученного ключа RSA-2048 был создан сертификат в инфраструктуре открытых ключей по стандарту X.509, с помощью которого была создана электронная подпись.
5. В приложении Adobe Acrobat Reader была создана цифровая подпись, с которой успешно был подписан данный документ, что позволило определить его модификацию.