

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

ОТЧЕТ
по практической работе №8
по дисциплине «Криптографические методы защиты информации»
Тема: «Изучение электронной подписи»

Студент гр. 9361

Кисляков Н.

Преподаватель

Племянников А.К.

Санкт-Петербург

2023

Цель работы

Исследовать алгоритмы создания и проверки электронной подписи, алгоритмы генерации ключевых пар для алгоритмов электронной подписи RSA, DSA, ECDSA и получить практические навыки работы с ними, в том числе с использованием приложения CrypTool версий 1 и 2.

1. Генерация ключевых пар

1.1 Задание

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.

1.2 Описание алгоритмов генерации ключевых пар

Генерация ключевых пар для алгоритма RSA:

1. Выбираются два больших простых числа p и q ;
2. Вычисляется $n = p \cdot q$;
3. Выбирается произвольное число e ($e < n$), взаимно простое с $\varphi(n)$ ($\varphi(n) = (p - 1) \cdot (q - 1)$);
4. Вычисляется $d: e \cdot d = 1 \bmod \varphi(n)$;
5. Пара чисел (e, n) объявляются открытым ключом, а d – закрытым ключом.
6. Числа p и q уничтожаются.

Генерация ключевых пар для алгоритма DSA:

1. Выбирается простое число pp , длиной между 512 и 1024 битами, число битов в p должно быть кратно 64;
2. Выбирается другое простое число q , которое имеет тот же самый размер, что и дайджест – 160 битов, такое, что $(p - 1) = 0 \bmod q$;

3. Выбирается e_1 , такое, что $e_1^q = 1 \bmod p$, путем вычисления $e_1 = e_0^{p-1/q} \bmod p$, где $e_0 \in Z_p$ (теорема Ферма);

4. Выбирается целое $d < q$ и вычисляется $e_2 = e_1^d \bmod pp$;

5. Числа (e_1, e_2, p, q) объявляются открытым ключом, а d – закрытым ключом.

Генерация ключевых пар для алгоритма ECDSA:

1. Выбирается эллиптическая кривая $E_p(a, b)$, p – простое число;

2. Выбирается точка на эллиптической кривой $e_1 = (x_1, y_1)$;

3. Для дальнейших вычислений выбирается другое простое число q – порядок циклической подгруппы группы точек эллиптической кривой: $q \cdot (x_1, y_1) = O$;

4. Выбирается целое число d , $1 < d < q - 1$ и назначается закрытым ключом;

5. Вычисляется другая точка на кривой $e_2 = d \cdot e_1$;

6. Числа (a, b, p, q, e_1, e_2) объявляются открытым ключом.

1.3 Таблица с фактическими временем генерации ключевых пар

Используем в Cryptool 1 утилиту «Digital Signatures/PKI -> PKI/Generate...» для генерации ключевых пар по алгоритмам RSA-2048, DSA-2048, EC-239. В таблице 1 кажем время, которое тратится для генерации ключа.

Таблица 1

Алгоритм	Время генерации ключевых пар
RSA-2048	1,721
DSA-2048	1,207
EC-239 (prime239v1)	0,014

Как видно из таблицы 1, EC-239 занимает минимальное количество времени для генерации ключей.

1.4 Скриншоты со значениями открытых ключей

На рисунке 1, 2, 3 представлены сгенерированные открытые ключи. Которые мы получили в прошлой под главе.

Public Parameters of: 1 2

Modulus: 318799803176851384728506824118771264461552729180602604410512
537319631840130470243849346825492374005600797950679835163324
106586040281176418094421162760587174744334879486462857863329

Exponent: 65537

Base for presentation of numbers
☐ Octal ☒ Decimal ☐ Hexadecimal

Back

Рисунок 1 – Открытый ключ RSA-2048

Certificate Data

Version: 2 (X.509v3-1996)
 SubjectName: CN=1 1 [1685195915], DC=cryptool, DC=org
 IssuerName: CN=CrypTool CA 2, DC=cryptool, DC=org
 SerialNumber: EF:C8:97:AD:FF:A4:71:85
 Validity - NotBefore: Sat May 27 16:58:36 2023 (230527135836Z)
 NotAfter: Mon May 27 16:58:36 2024 (240527135836Z)
 Public Key Fingerprint: 69AD 4446 1D64 8180 748C 67E4 0B09 6EC5
 SubjectKey: Algorithm NIST-DSA (OID 1.3.14.3.2.12),
 DSA prime p (no. of bits = 2048):
 0 FFF9ADDA B13A906A ADDD3213 823D8B94
 10 32ECFDC2 5848E942 A1AA962A 009C6C51
 20 4D794A82 88ED430C 9292D420 83F74194
 30 66CE9166 953C2654 4EEE9419 2FC3D1F7
 40 1FCD768D 329641AF EA4B9295 A9970D48
 50 9E78343E 705C1E4B 21241026 B34431A9
 60 9DF7695B 217EF400 071C5D53 AD456507
 70 260B270A 37627B0C B698BA9A 4FABF948
 80 57393E24 D676CE68 127A518A 7ECC8C72

Close

Рисунок 2 – Открытый ключ DSA-2048

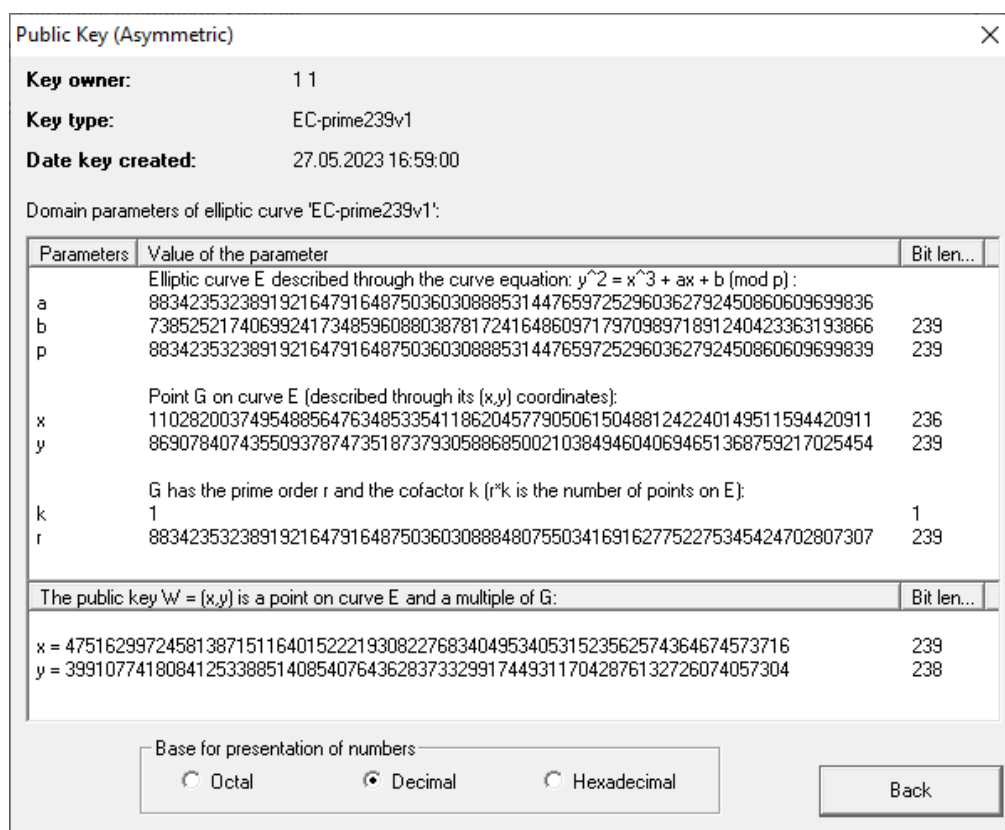


Рисунок 3 – Открытый ключ ЕС-239

2. Процессы создания и проверки электронной подписи

2.1 Задание

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.

2.2 Обобщенная схема создания и проверки электронной подписи

Электронная подпись – это некоторая информация в электронной форме (код), которая присоединена к другой информации (файлу данных) с целью

подтверждения авторства и контроля целостности файла данных. Обобщенная схема, поясняющая работу протокола подписания документа и проверки электронной подписи, продемонстрирована на рисунке 4.



Рисунок 4 – Обобщенная схема создания и проверки электронной подписи

Следует отметить, что в самом общем случае на стороне отправителя запускается процедура создания электронной подписи (процедура подписания), а на стороне получателя – процедура проверки электронной подписи (процедура верификации). Подпись создается на основе хэш-кода отправляемого файла данных и закрытого ключа отправителя, а при проверке подписи используется хэш-код от полученного файла данных, сопровождающая его подпись и открытый ключ отправителя, доставленный получателю.

2.3 Таблица с фактическими временами создания электронной подписи различными алгоритмами

На рисунке 5 представлен исходный текст, который состоит из 143265 знаков с пробелами.

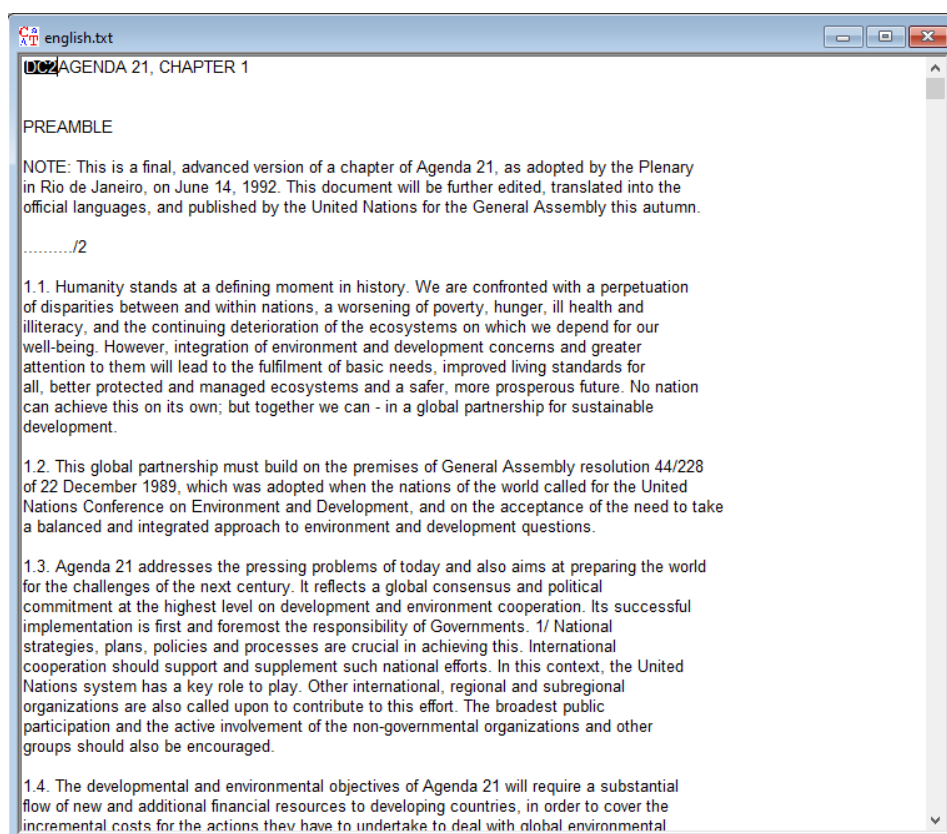


Рисунок 5 – Исходный текст

С помощью сгенерированных ключевых пар, создадим цифровые подписи и зафиксируем время генерации в таблице 2.

Таблица 2

Алгоритм	Хэш-функция	Время генерации цифровой подписи
RSA	SHA-1	0,012 секунды
DSA	SHA-1	0,004 секунды
ECSP-DSA	SHA-1	0,004 секунды
ECSP-NR	SHA-1	0,004 секунды

2.4 Скриншот со значением электронной подписи

Сгенерированная цифровая подпись алгоритмом RSA и хэш-функцией SHA-1 представлена на рисунке 6.

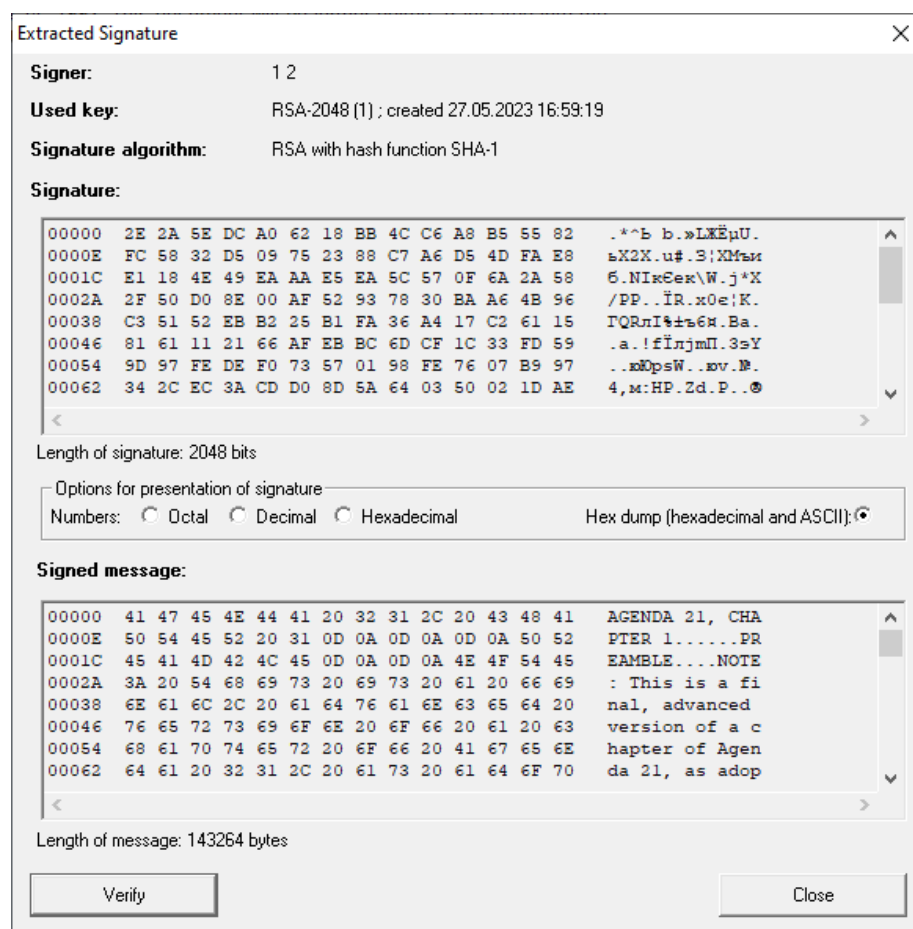


Рисунок 6 – Цифровая подпись

2.5 Скриншоты с результатами проверки электронной подписи

Выполним процедуру проверки цифровой подписи, сгенерированной алгоритмом RSA и хэш-функцией SHA-1, для случаев сохранения и нарушения целостности исходного текста. Результаты проверки продемонстрированы на рисунках 7 и 8.



Рисунок 7 – Проверка при сохранении

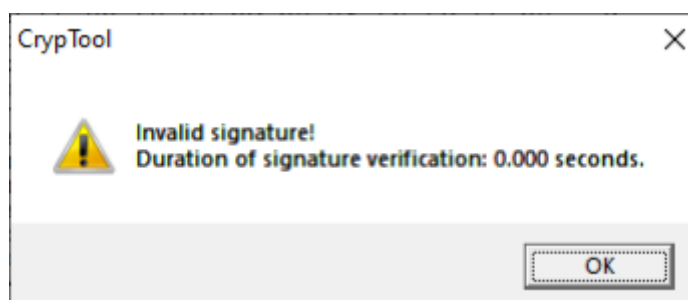


Рисунок 8 – Проверка при нарушении

3. Создание и проверка электронной подписи на основе эллиптических кривых

3.1 Задание

1. Выполнить процедуру создания подписи Digital Signatures/PKI → Sign Document... алгоритмом ECSP-DSPA в пошаговом режиме (Display inter. results = ON). Зафиксировать скриншоты последовательности шагов.

2. Выполнить процедуру проверки подписи ECSP-DSPA 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.

3.2 Описание алгоритма формирования и проверки подписи ECDSA

Схема цифровой подписи ECDSA показана на рисунке 9.

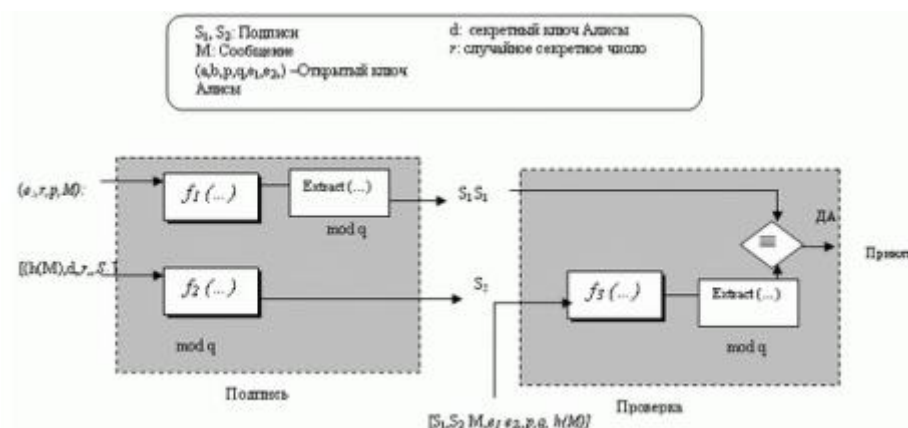


Рисунок 9 – Схема цифровой подписи ECDSA

Функция f_1 создает новую точку для секретного ключа подписывающего лица. Функция f_2 создает новую точку из двух общедоступных ключей подписывающего лица. Каждый экстрактор *Extract* извлекает первые координаты соответствующей точки в модульной арифметике.

В процессе подписания две функции f_1 и f_2 и экстрактор (извлекающее устройство) создают две части подписи. В процессе проверки (верификации) обрабатывают выход одной функции f_2 (после прохождения через экстрактор) и сравнивают ее с первой частью подписи.

С использованием ключевой пары (закрытый ключ – d , и открытый ключ – (a, b, p, q, e_1, e_2)) осуществляется подписание документа, а затем на принимающей стороне – верификация подписи.

Алгоритм создания электронной подписи ECDSA состоит из следующих операций:

1. Выбирается секретное случайное число r , $1 < r < q - 1$;
2. Выбирается третья точка на кривой $P(u, v) = r \times e_1$;
3. Используем абсциссу u , чтобы вычислить первую часть подписи $S_1 = u \bmod q$;
4. Используем дайджест сообщения $h(M)$, закрытый ключ d , секретное случайное число r и S_1 , чтобы вычислить вторую часть подписи $S_2 = (h(M) + d \times S_1) \times r^{-1} \bmod q$.

Алгоритм проверки цифровой подписи ECDSA включает следующие операции:

1. Используем M, S_1, S_2 для получения промежуточных результатов A и B : $A = h(M) \times S_2^{-1} \bmod q$, $B = S_2 - 1 \times S_1 \bmod q$;
2. Затем восстанавливаем третью точку $T(x, y) = A \times e_1 + B \times e_2$;
3. Верификатор $V = x \bmod q$ сравниваем с S_1 .

Схема протокола ECDSA представлена на рисунке 10.

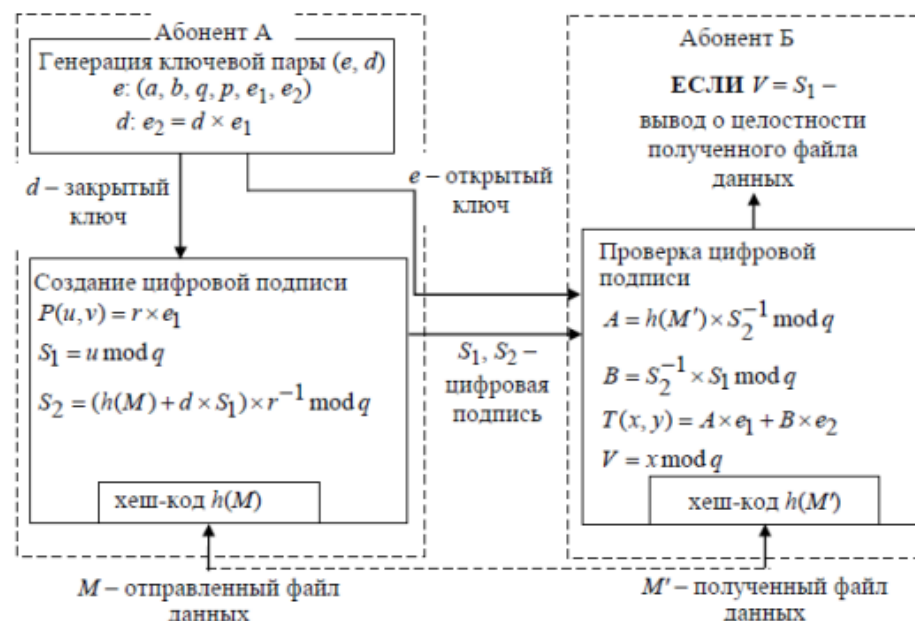


Рисунок 10 – Схема протокола ECDSA

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

На рисунке 11, 12, 13, 14, 15, 16 и 17 изображена процедура создания подписи алгоритмом ECSP-DSA в пошаговом режиме при помощи CrypTool 1.

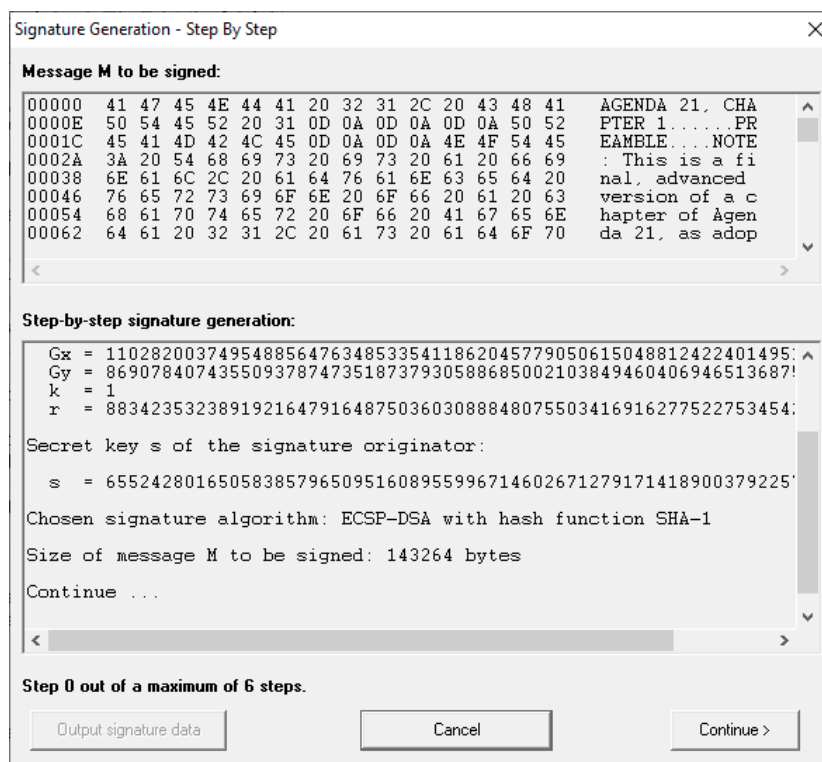


Рисунок 11 – Нулевой шаг создания подписи

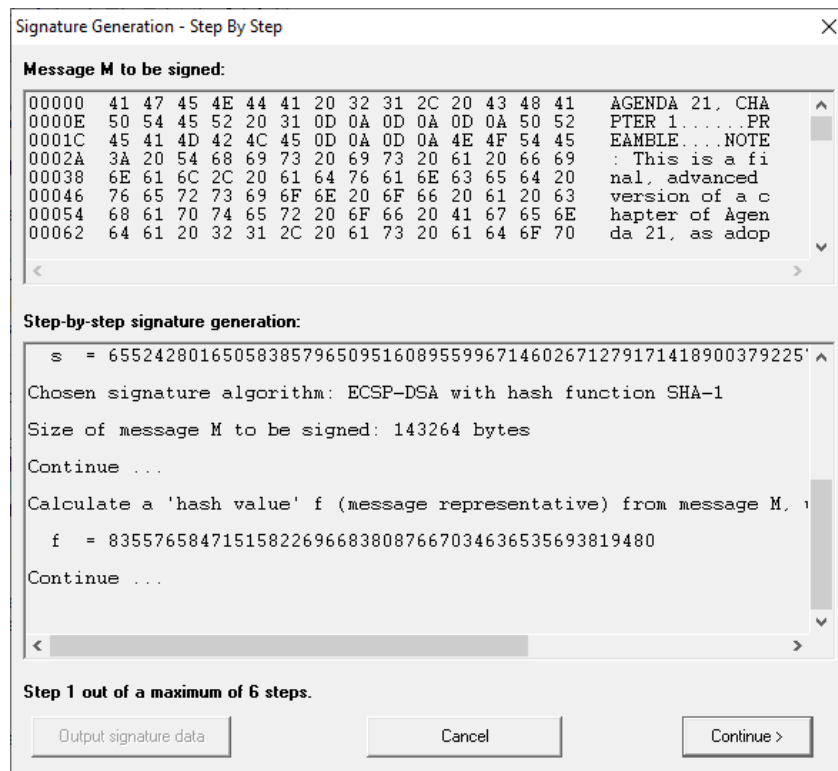


Рисунок 12 – Первый шаг создания подписи

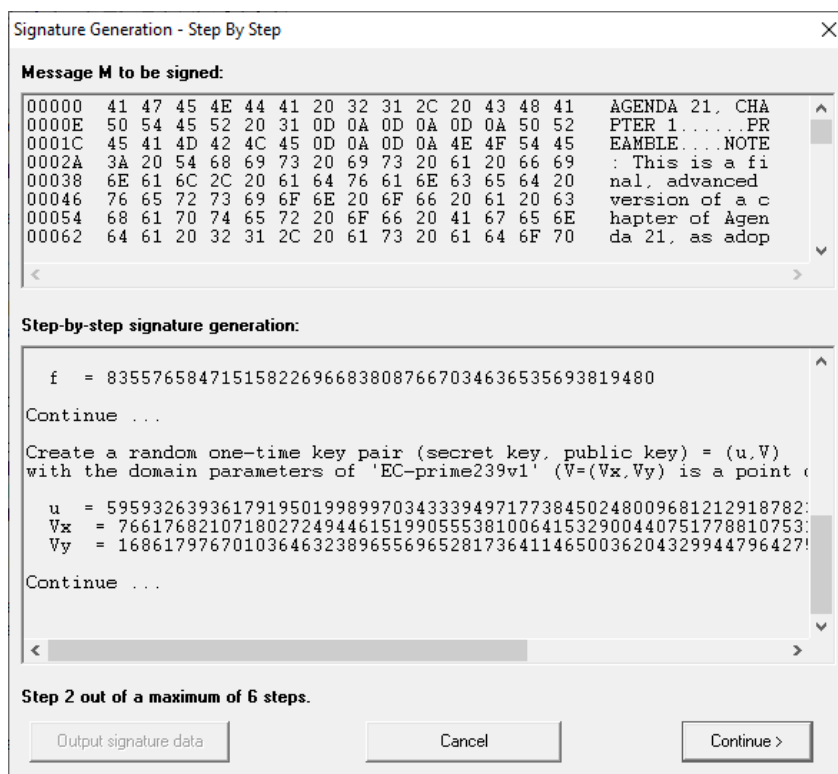


Рисунок 13 – Второй шаг создания подписи

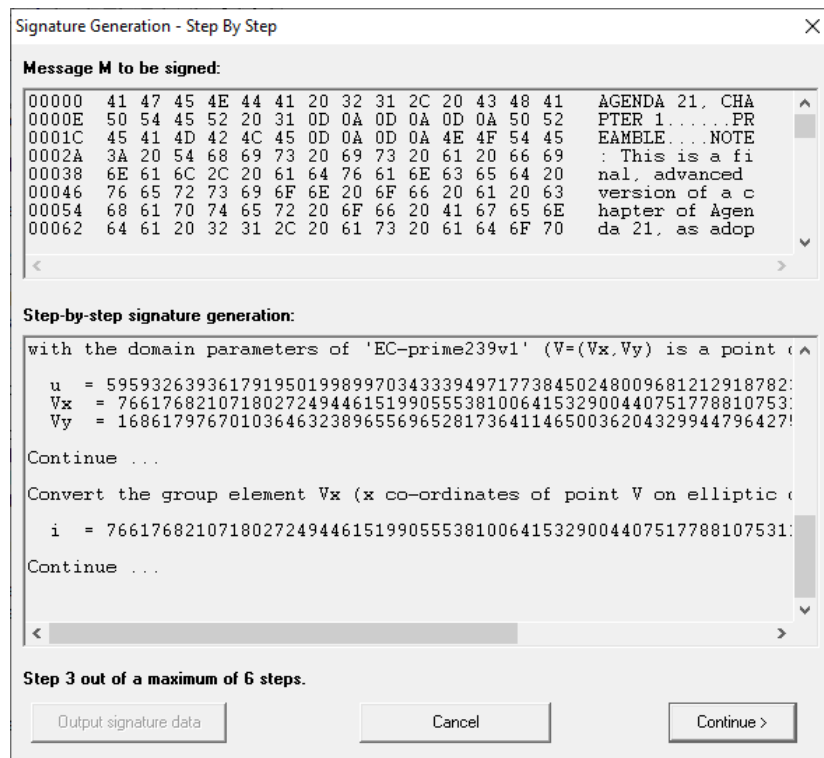


Рисунок 14 – Третий шаг создания подписи

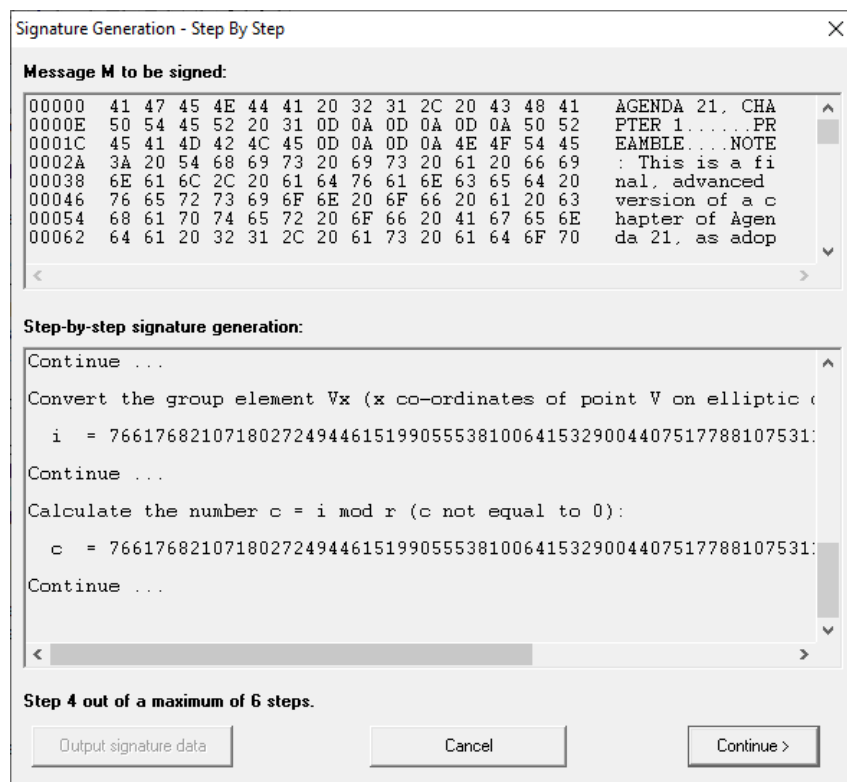


Рисунок 15 – Четвертый шаг создания подписи

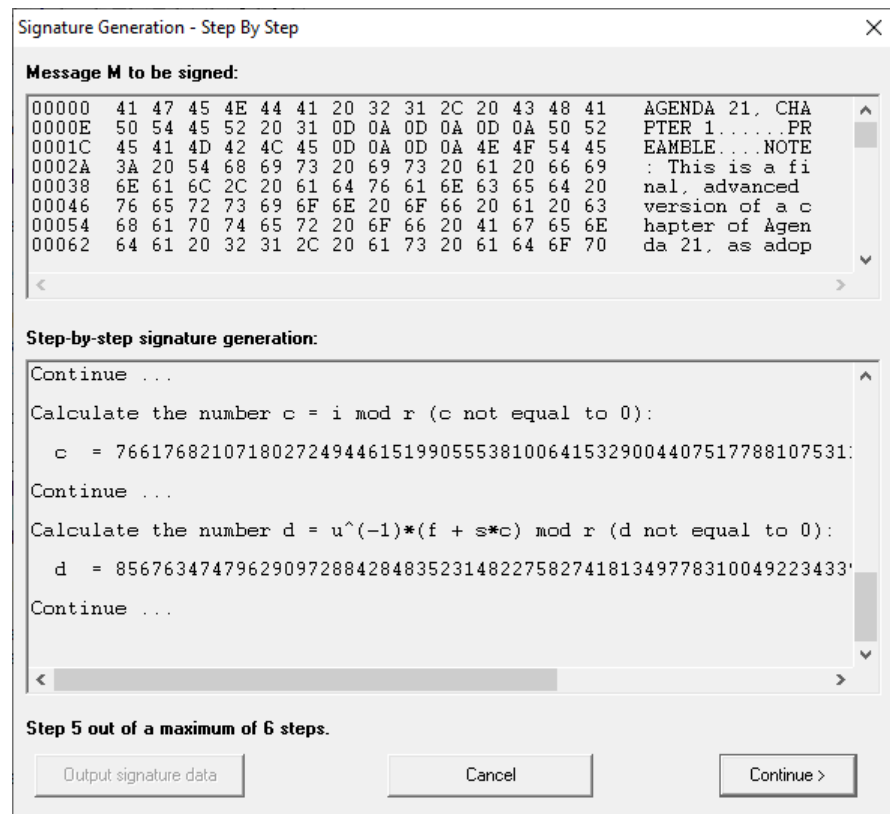


Рисунок 16 – Пятый шаг создания подписи

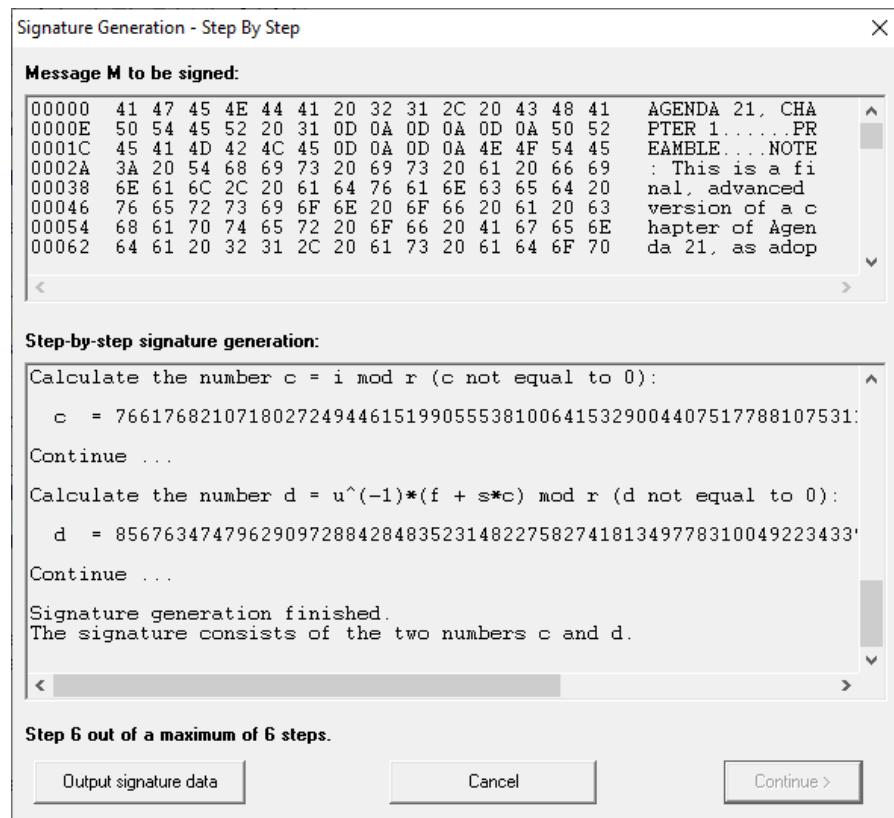


Рисунок 17 – Шестой шаг создания подписи

Теперь сравним лекционную версию ECDSA и реализацию в CrypTool

1. Результаты сравнения представлены в таблице 3.

Таблица 3

Параметр из лекции	Параметр из CrypTool 1
a	a
b	b
$e_1 = (x, y)$	(Gx, Gy)
q	r
d	s
$h(M)$	f
r	u
u	i
$P(u, v)$	(Vx, Vy)
S_1	c
S_2	d

Выполним процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Результаты проверок показаны на рисунках 18 и 19.



Рисунок 18 – Проверка цифровой подписи

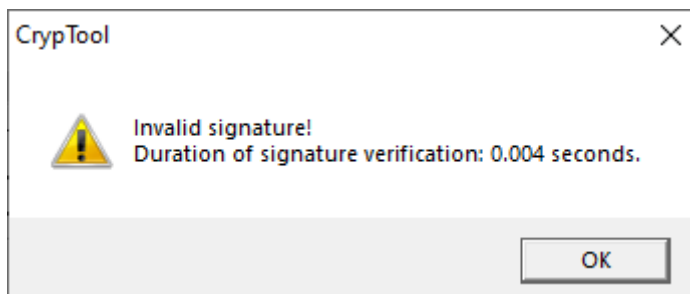


Рисунок 19 – Проверка цифровой подписи

3.4 Результаты проверки лекционного материала по ECDSA с использованием приложения Indiv.Procedures → Number Theory... → Point Addition on EC

Выполним проверку лекционного материала по ECDSA с помощью утилиты из CrypTool 1.

1. Генерация ключей ECDSA.

Возьмем следующую эллиптическую кривую: $E_{23}(-18,75)$. Далее выберем точку $e_1 = (x_1, y_1) = (7, 4)$, как представлено на рисунке 20.

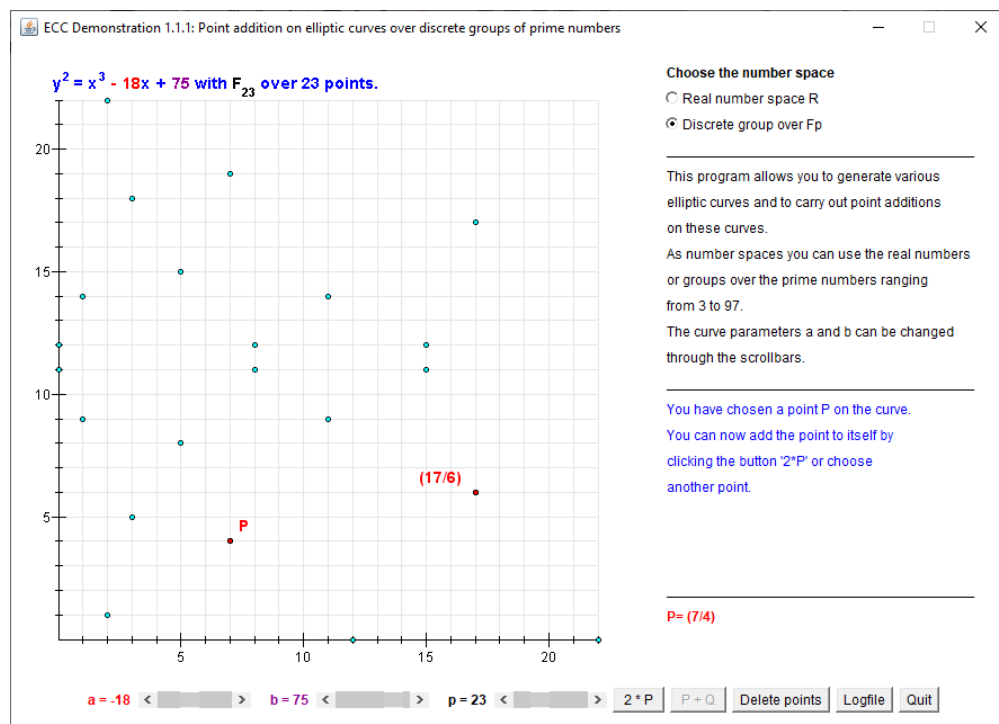


Рисунок 20 – Выбор точки e_1

Теперь подберем такое число q , что $q \times (x_1, y_1) = O$. Получилось, что $q = 11$, как продемонстрировано на рисунке 21.

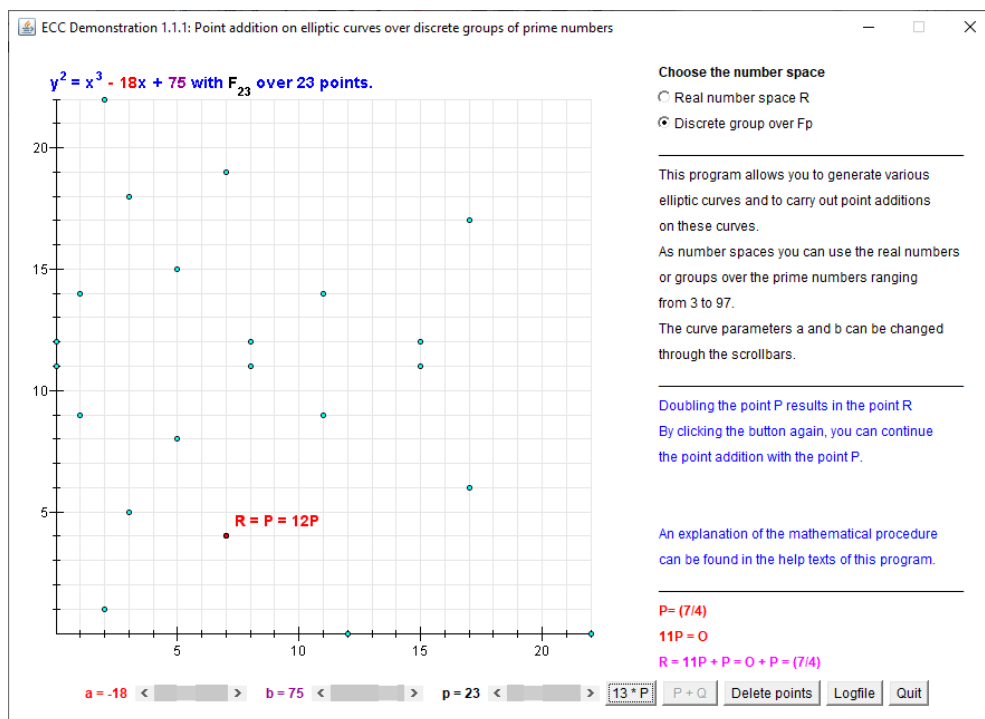


Рисунок 21 – Подбор числа q

Выбираем целое число d такое, что $1 < d < q - 1$. Выбрано $d = 5$ – закрытый ключ. После чего, вычисляем точку $e_2 = d \times e_1 = 5 \times (7, 4) = (1, 9)$, как показано на рисунке 22.

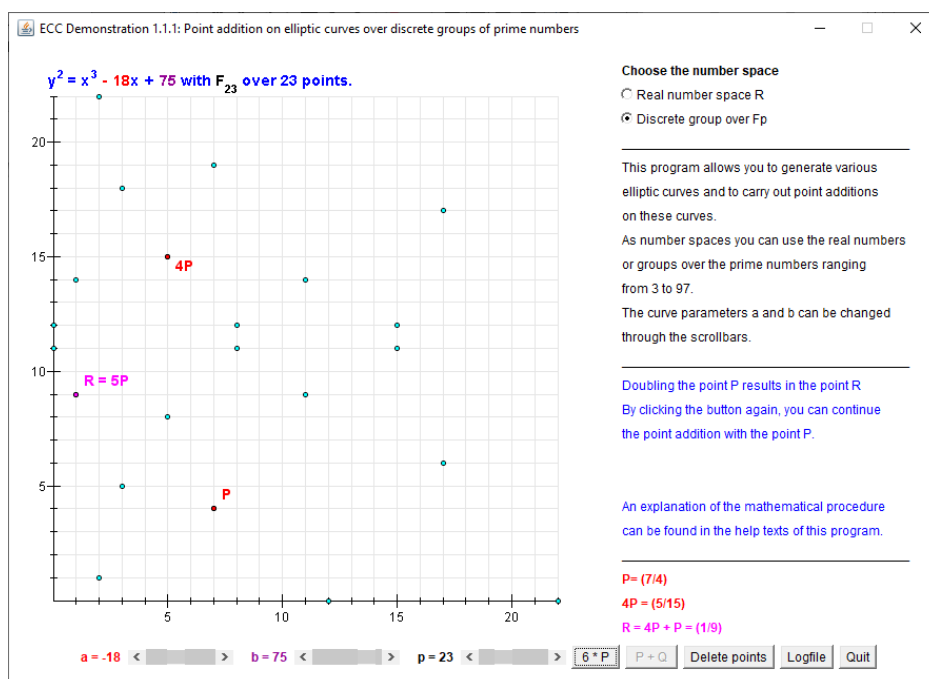


Рисунок 22 – Вычисление точки e_2

Открытый ключ – $(a = -18, b = 75, p = 23, q = 11, e_1 = (7, 4), e_2 = (1, 9))$.

2. ECDSA подписание.

Текст представляется $M = h(M) = 66$. Выбираем случайное $r = 3$.

Находим третью точку $P = r \times e_1 = 3 \times (7,4) = (15,11) \Rightarrow u = 15, v = 11$.

Вычисляем первую часть подписи: $S_1 = u \bmod q = 15 \bmod 11 = 4$.

Вычисляем вторую часть подписи: $S_2 = (h(M) + d \times S_1) \times r^{-1} \bmod q = (66 + 5 \times 4) \times 3^{-1} \bmod 11 = 3$.

Тогда получаем $M = h(M) = 66, SS1 = 4, SS2 = 3$.

3. ECDSA проверка.

Вычислим промежуточные результаты A и B :

$$A = h(M) \times S_2^{-1} \bmod q = 66 \times 3^{-1} \bmod 11 = 0;$$

$$B = S_2^{-1} \times S_1 \bmod q = 3^{-1} \times 4 \bmod 11 = 5.$$

Восстанавливаем третью точку $T(x,y)$: $T(x,y) = A \times e_1 + B \times e_2 = 0 \times (7,4) + 5 \times (1,9) = (15,11)$.

Вычисляем верификатор V : $V = x \bmod q = 15 \bmod 11 = 4$.

Так как $V = 4 = S_1$ следовательно, текст сохранил свою целостность

4. Демонстрация процесса подписи в среде РК

4.1 Задание

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.

4.2 Описание структуры сертификата из CrypTool 1

На рисунке 23 представлен сертификат ключа проверки электронной подписи.

```

Version:                2 (X.509v3-1996)
SubjectName:            CN=Sergey Belenkov [1683988511], DC=cryptool, DC=org
IssuerName:             CN=CrypTool CA 2, DC=cryptool, DC=org
SerialNumber:          AD:7A:84:2B:DB:EE:D9:F4
Validity - NotBefore:   Sat May 13 17:35:12 2023 (230513143512Z)
                      - NotAfter: Mon May 13 17:35:12 2024 (240513143512Z)
Public Key Fingerprint: 3370 5936 D9E8 F7F2 FFDD 4C68 4E2F 9998
SubjectKey:             Algorithm rsa (OID 2.5.8.1.1), Keysize = 2048
Public modulus (no. of bits = 2048):
|0 FF8F1AA1 E1EA5A65 86B9250E 27749770
|10 CFFC5346 17B557C5 DEA45616 143972FA
|20 0C227B3B 05B23D21 05397276 24468834
|30 37C462EB 35BF38FC 6CB9BA8F 380F5ADD
|40 174EA162 FF783A14 13D65406 967A34A7
|50 A91B8D4A 8E7D8DF7 C937E5EB 7126B261
|60 4F55E271 CCA61DFC 17030268 B6342D43
|70 42682337 B5C53CF7 D02BB444 1E699D69
|80 4C65D2F6 8DF0847A 0224A77A 19F6604F
|90 547EF102 3267B70A D7AF57C1 44FB8FEE
|A0 955A322E 8579C2DE B05F2414 A9DC4C86
|B0 EB25CE73 B195EDD5 3B33418D 1805EC02
|C0 884A7865 0683D5D9 EA5722B1 006156BC
|D0 42AA839E FDA59949 902B3988 137E04FE
|E0 1B3ADC59 1EF9A457 9AACCF7E CC4D8058
|F0 A3BD9667 55AA5710 7586231E B5ED9F59
Public exponent (no. of bits = 17):
|0 010001
Certificate extensions:
Private extensions:
|OID 2.206.5.4.3.2:
|PrintableString:
|[[Belenkov]][Sergey][RSA-2048][168]
|3988511]
SHA1 digest of DER code of ToBeSigned:
|0 D9D5A589 19257FA6 184C8548 6B08DAFD
|10 C37C9F2F
Signature:             Algorithm sha1WithRSAEncryption (OID 1.3.14.3.2.29), NULL
|0 1DFA8817 BC85CDA3 CA37122C 0A03330F
|10 00F742A9 A9A16189 D139E422 F1BA5AE8
|20 CED420B 3CFC304C DA7E880F 937CCBDD
|30 6F70B81E 238C60DD 3AD2AC75 A6886368
|40 0199DAFD 222E0669 4C06A4E6 9EF2EE86
|50 EB8EA5EF 517D0669 5F6566D1 026B5020
|60 617FB443 C2420186 7B14D1E4 22868381
|70 96EB153C 1D689C69 44FC4A27 974580F7
|80 424A2535 9E0C50D4 5CBC033A 40A737EB
|90 56832DE6 F5D9DACB 9CD78AC9 1DC2F04A
|A0 9284F1E3 9DE51E3E 7BC316B0 57DF6196
|B0 1431A1B5 B944196F E4ACC9A9 0B42F7EA
|C0 92A57BEC DC438A6B F1ADD0A4 E7646A58
|D0 309BB261 4CBAF423 443C2FD8 94B5EEAA
|E0 FAA8B2A0 A68064A3 294642AE 996D8C69
|F0 644FC52F 01D0AA6F B3D17B18 5045C6F3
Certificate Fingerprint (MD5): 85:44:C3:96:6C:12:0C:42:B9:28:87:03:4C:61:F5:FD
Certificate Fingerprint (SHA-1): 7156 940B B7E2 3D98 9914 046C 3BF5 87E6 0A4B 01D7

```

Рисунок 23 – Сертификат из CrypTool 1

Как видно из рисунка выше, в структуре сертификата из CrypTool 1 имеется версия сертификата, имя субъекта, имя издателя, серийный номер, период действия, идентификатор алгоритма подписи, информация об открытом ключе (параметры и алгоритм генерации), уникальные идентификаторы издателя и субъекта, дополнительная информация об использовании ключа, а также электронная подпись сертификата.

4.2 Схема процедуры подписания из CrypTool 1

Запустим демонстрационную утилиту «Digital Signatures/PKI -> Signature Demonstration...» из CrypTool 1. Выполним все этапы создания цифровой подписи документа. На рисунках 24, 25, 26, 27 и 28 представлены все этапы.

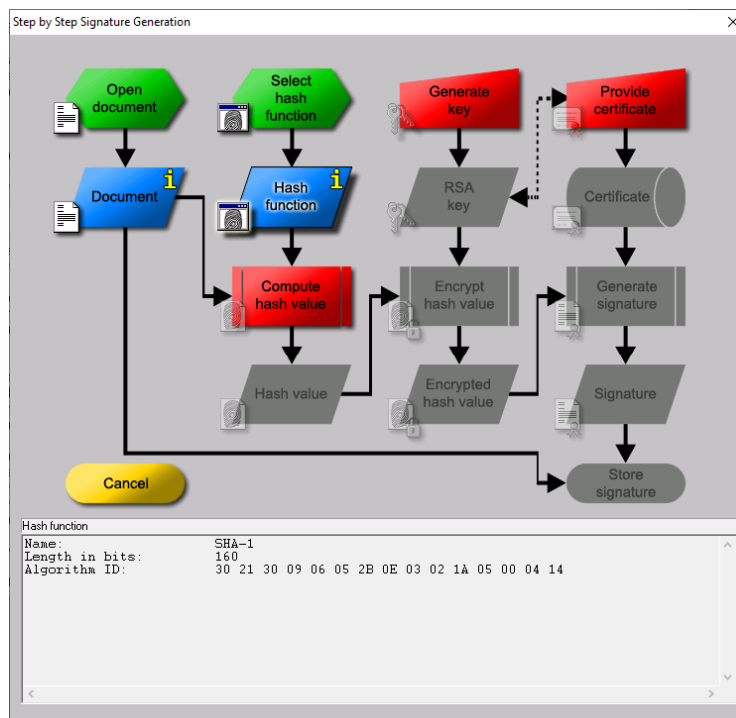


Рисунок 24 – Выбор хэш-функции

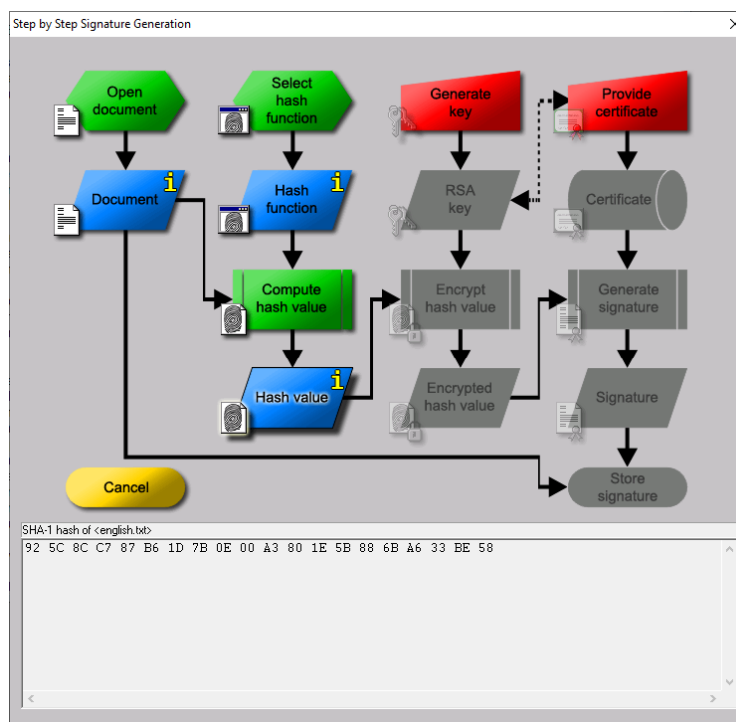


Рисунок 25 – Значение хэша документа

5.3 Скриншоты свойств подписи и сертификата

Рисунок 33 –

Рисунок 34 –

5.4 Скриншот результата проверки после внесения изменений в отчет

Рисунок 35 –

Вывод