1. **Подготовительный этап**

Для использования api crypto pro, первым делом было необходимо установить тестовый сертификат, инструкция по его установки была взята с официального сайта (Рисунки 1 - 2)

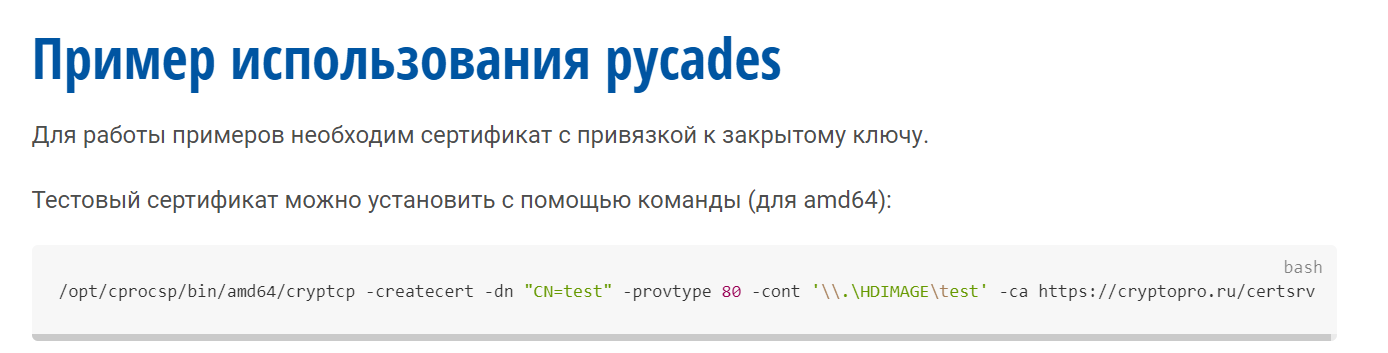


Рисунок 1 – Команда установки временного сертификата

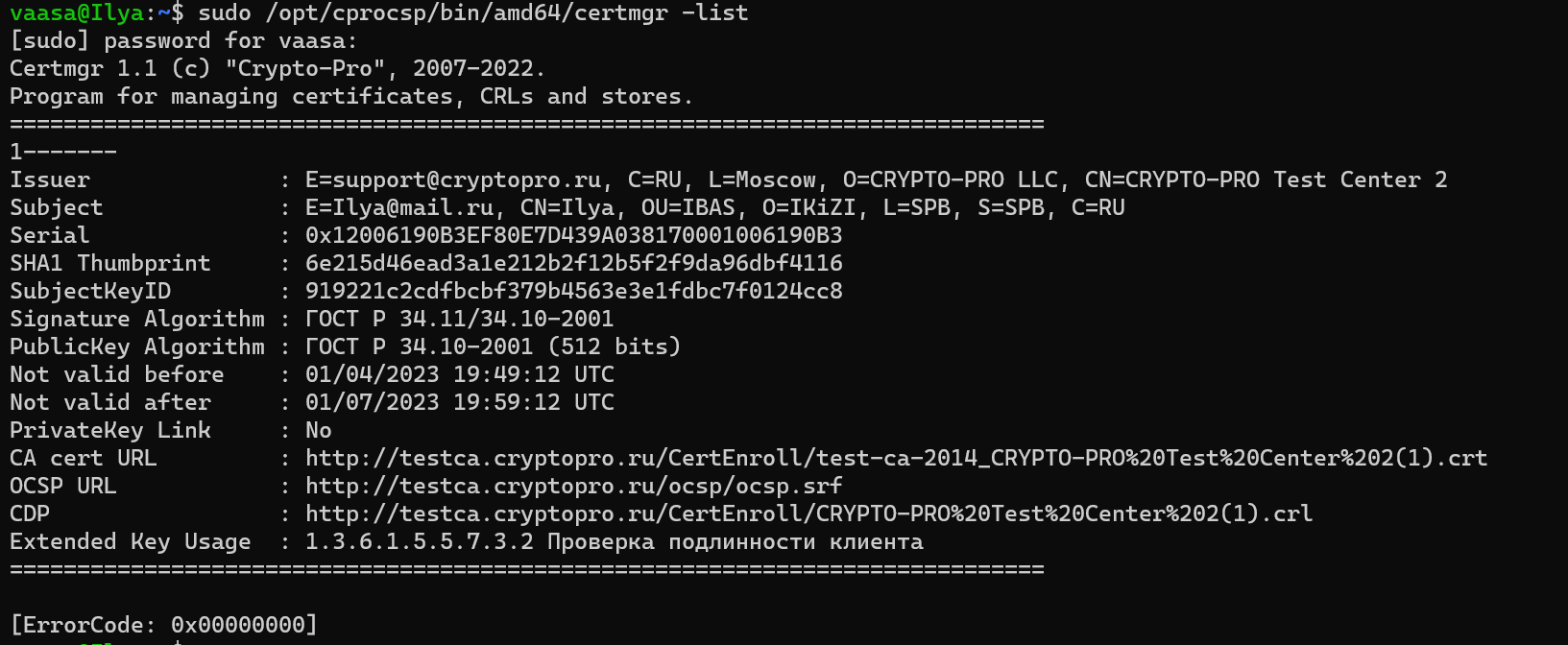


Рисунок 2 – Вывод всех установленных сертификатов

Для выполнения лабораторной работы был использован язык Python.

1. **Указание используемых алгоритмов**

**Для выполнения лабораторной работы требовалось использовать следующие алгоритмы:** ГОСТ Р 34.11-94, ГОСТ 28147-89 и ГОСТ Р 34.10-2001. ГОСТ Р 34.10-2001– алгоритм цифровой подписи, используется по умолчанию (Рисунок 2). ГОСТ Р 34.11-94 – алгоритм хэширования данных, указывается при использовании хэш функции (Рисунок 3)

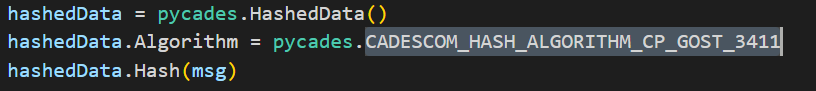


Рисунок 3 – Выбор алгоритма хэширования

ГОСТ 28147-89 – Алгоритм шифрования данных, данный алгоритм используется по умолчанию (Рисунок 4)

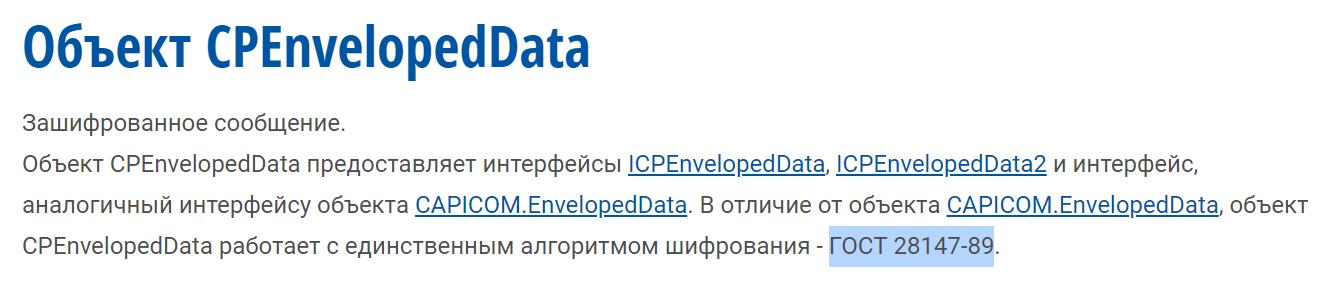


Рисунок 4 – Алгоритм шифрования

1. **Реализация приложения**

**Так как в лабораторной работе требовалось использовать все три компонента: Шифрования, хэш и ЭЦП, то они были использованы в следующем порядке: Вычисление Хэша (Рисунок 5) => Шифрование сообщения (Рисунок 6) => Вычисление ЭЦП от шифртекста и хэша (Рисунок 7)**

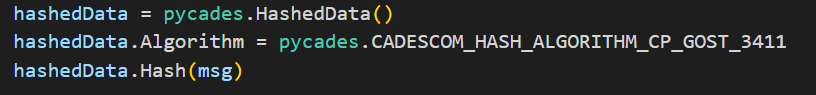


Рисунок 5 – Вычисление хэш значения

Для вычисления хэш значения, был использован класс pycades.HashedData(), в данном классе нужно указать алгоритм шифрования, после чего вызвать саму функцию шифрования.

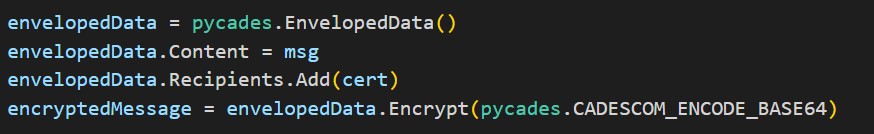


Рисунок 6 – Шифрования данных

Для шифрования данных использовался класс pycades.EnvelopedData(), в данном классе нужно указать само сообщения, сертификат и способ кодирования данных.

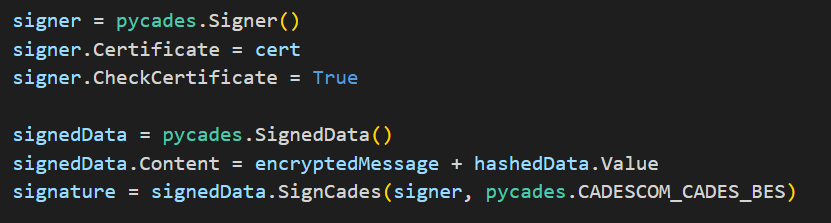


Рисунок 7 – Вычисление ЭЦП

Для вычисления ЭЦП использовались классы pycades.Signer и pycades.SignedData. Первый класс используется для описания использованного сертификата, а именно сам сертификат и его проверка, а второй уже используется для самой подписи.

Соответственно для проверки сообщения, серверу необходимо получить сообщение, расшифровать его и вычислить хэш и цифровую подпись (Рисунок 8)

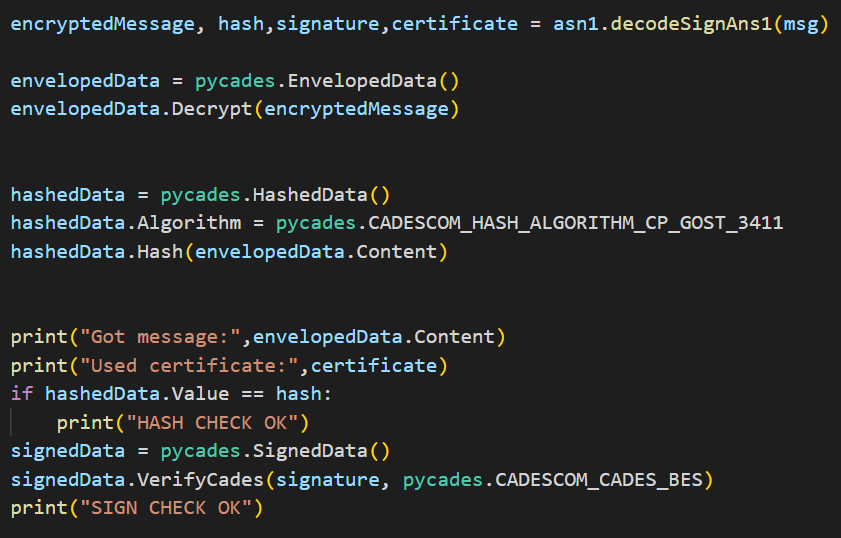


Рисунок 8 – Серверная сторона приложения

1. **ASN1 заголовок**

**Данный заголовок используется для передачи дополнительной информации, такой как публичные ключи, название сертификата и другая служебная информация. Так как в api crypto pro, после использования шифрования и ЭЦП, ASN заголовок создается автоматический, то для передачи данных все полученные данные, а именно хэш сообщения, шифр текст с ASN заголовком и ЭЦП с ASN заголовком, инкапсулировались в еще один ASN заголовок (Рисунок 9)**

****

**Рисунок 9 – Итоговый файл**

ASN1 заголовок шифртекста содержит информацию о сертификате, алгоритм шифрования, публичный ключ, и сам шифр текст

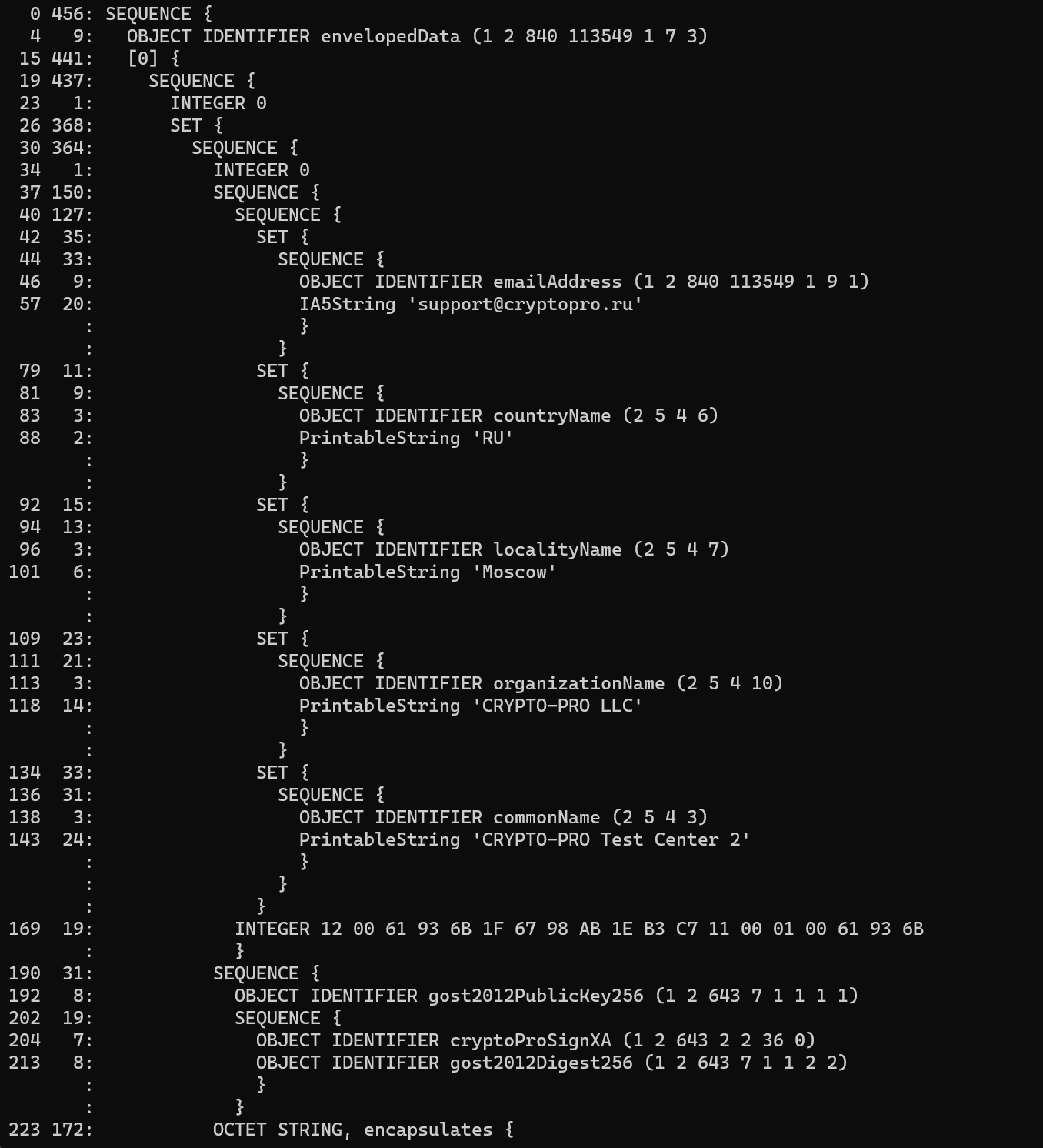


Рисунок 10 – Первая часть Вторая часть заголовка ASN1 шифртекста

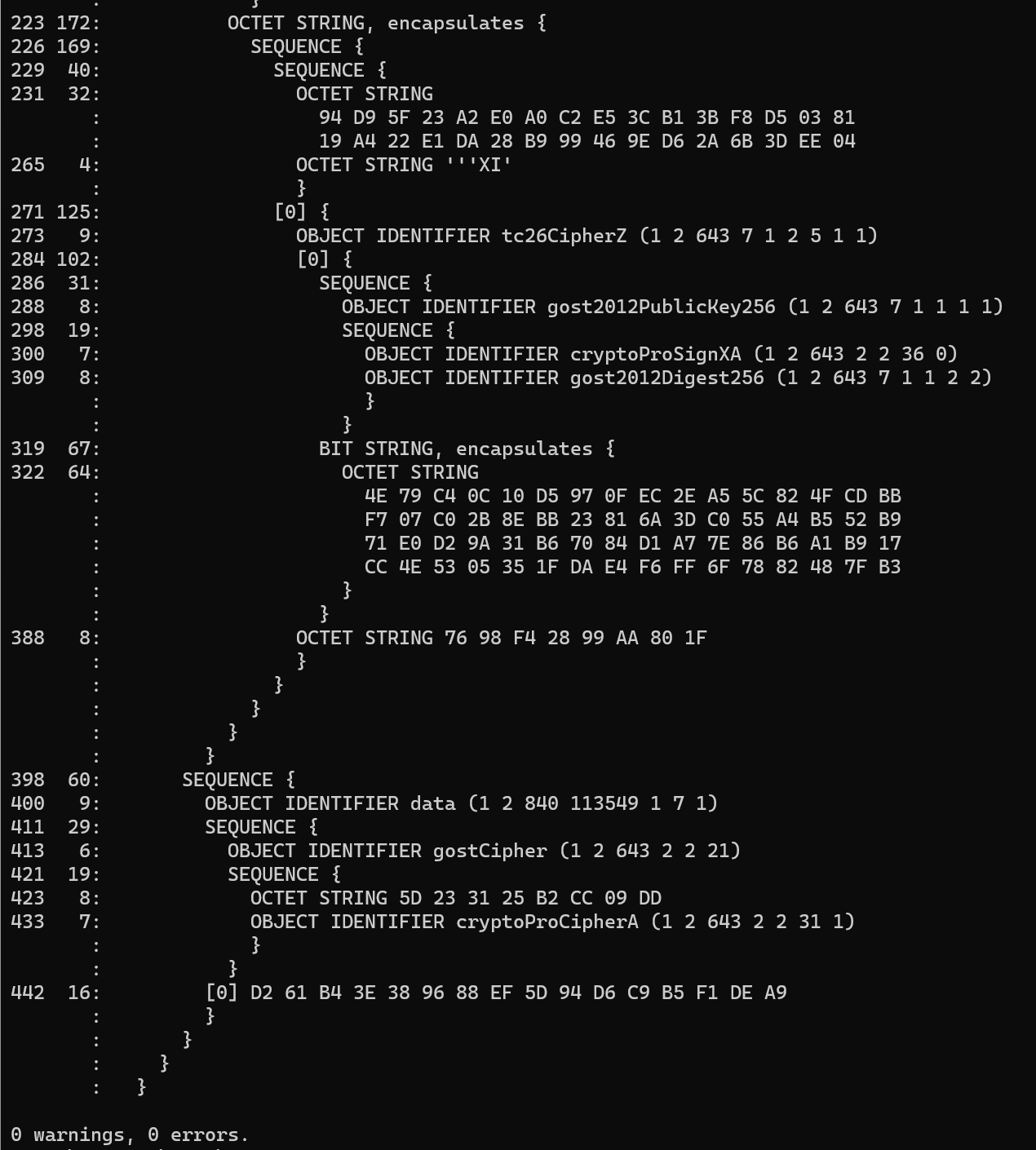


Рисунок 11 – Вторая часть заголовка ASN1 шифртекста

ASN1 заголовок ЭЦП содержит информацию о сертификате, дату и время подписи, используемый алгоритм, данные о ключах, дополнительную информацию о центре, который выдал сертификат (Рисунки 12 – 17)

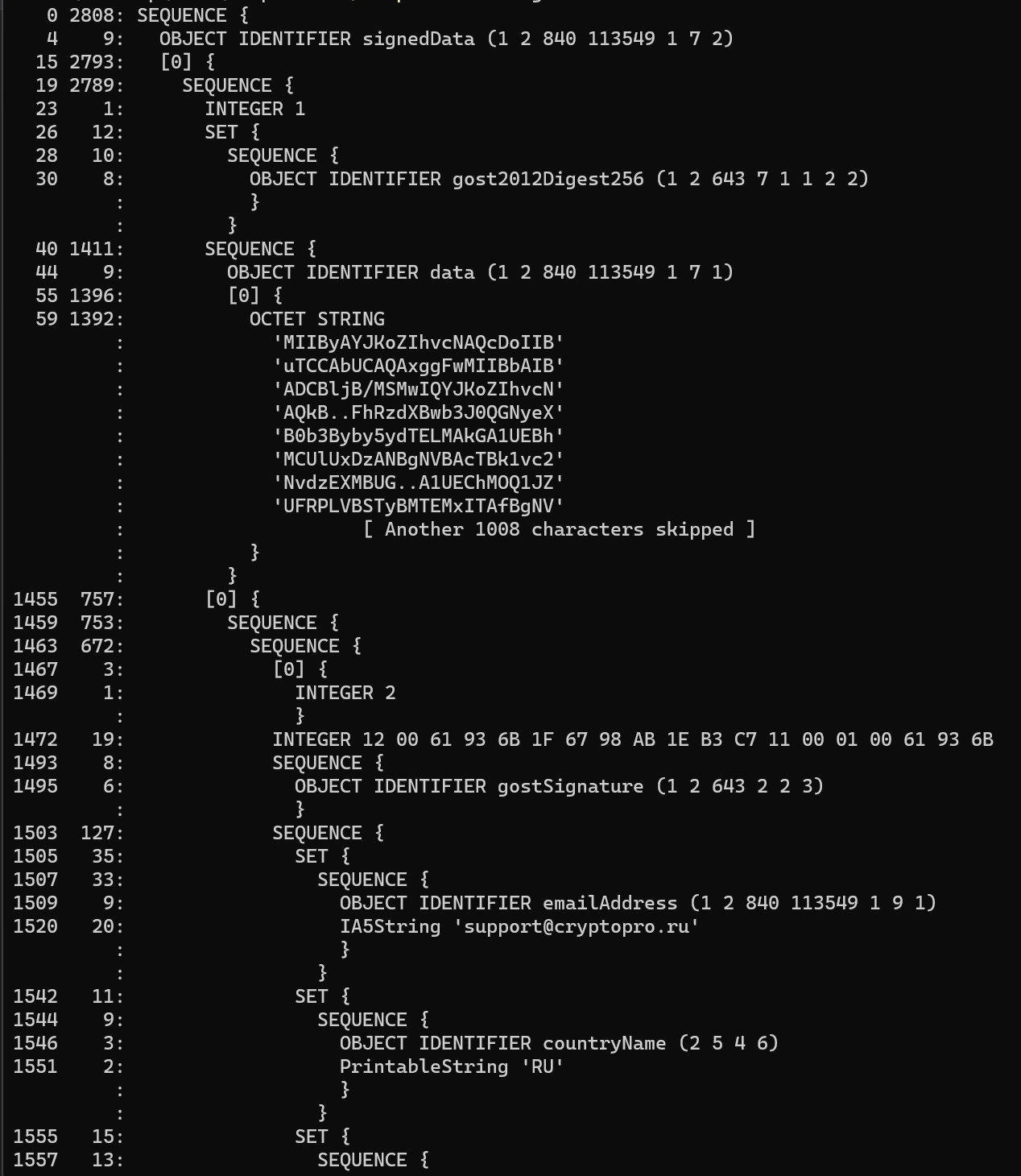


Рисунок 12 – Первая часть ASN1 заголовка ЭЦП

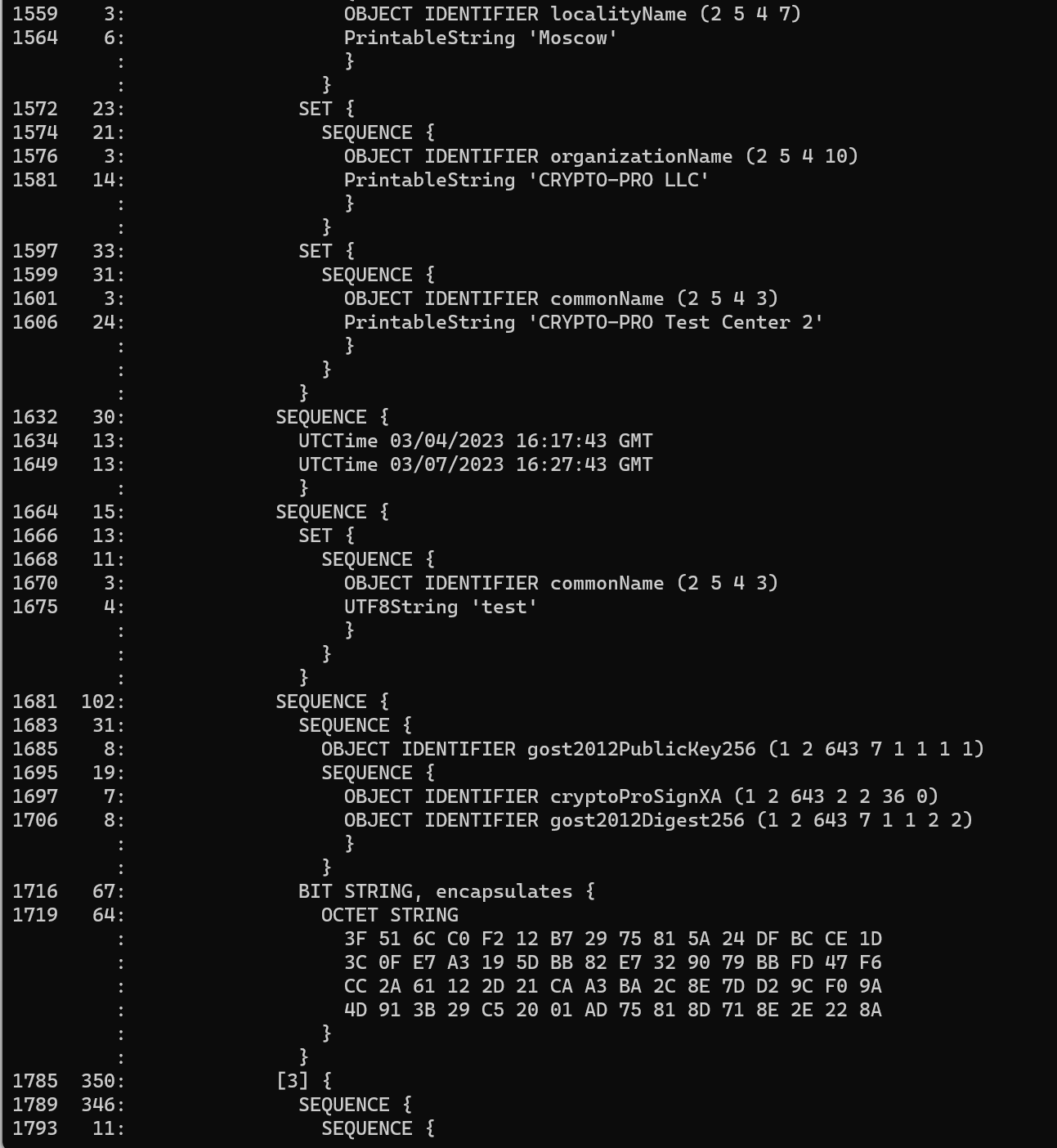


Рисунок 13 – Вторая часть ASN1 заголовка ЭЦП

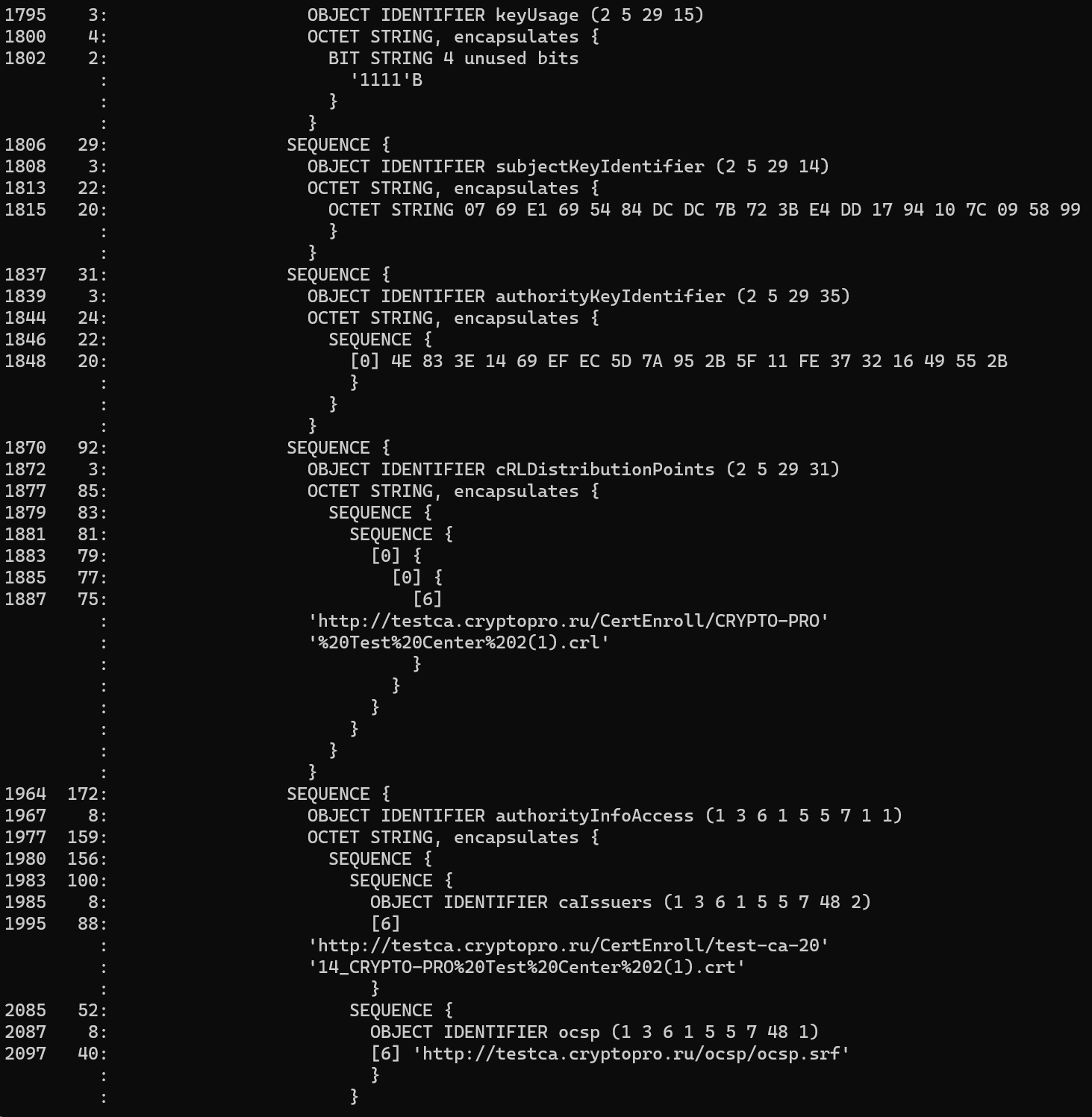


Рисунок 14 – Третья часть ASN1 заголовка ЭЦП

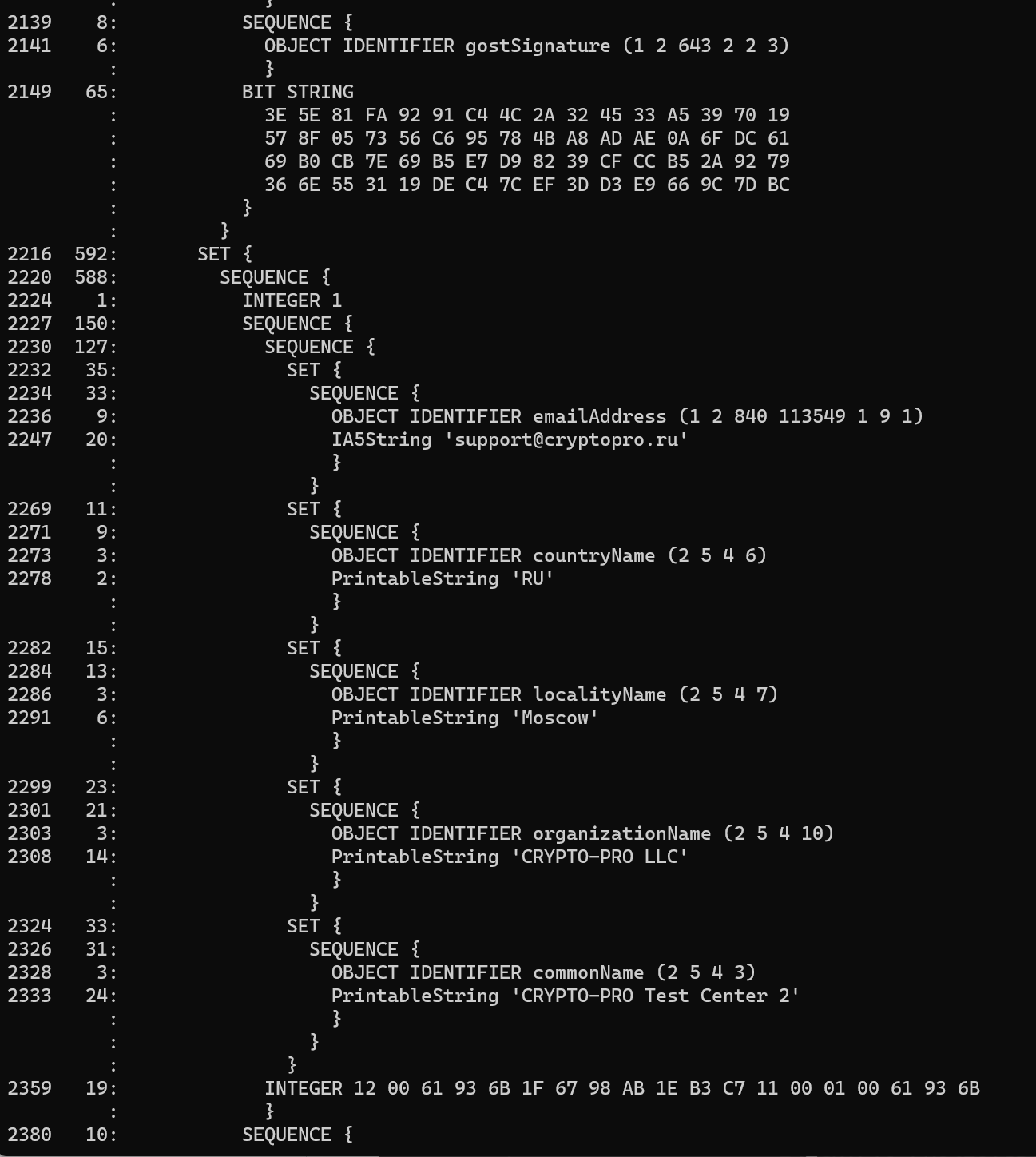


Рисунок 15 – Четвертая часть ASN1 заголовка ЭЦП

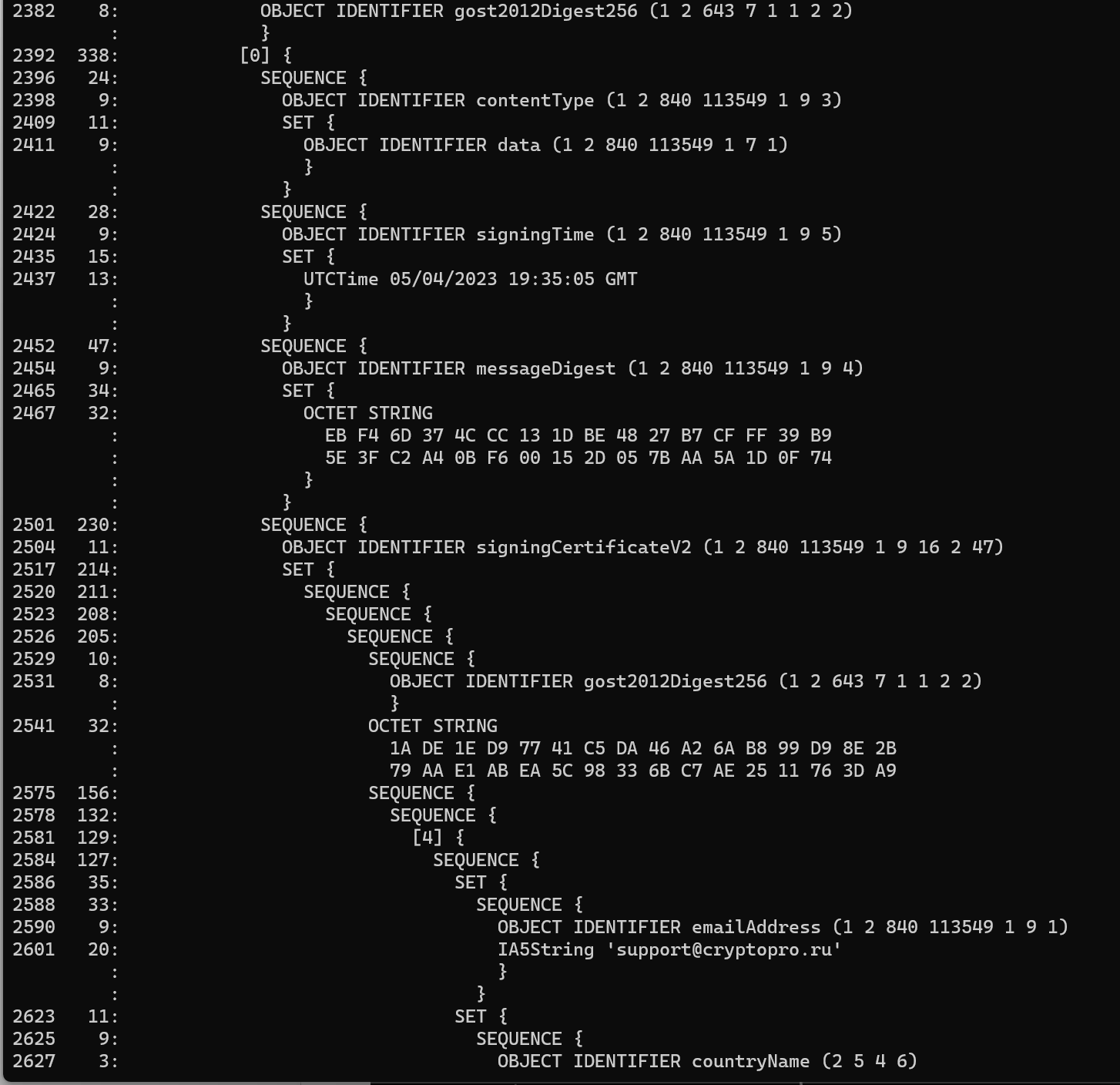


Рисунок 16 – Пятая часть ASN1 заголовка ЭЦП

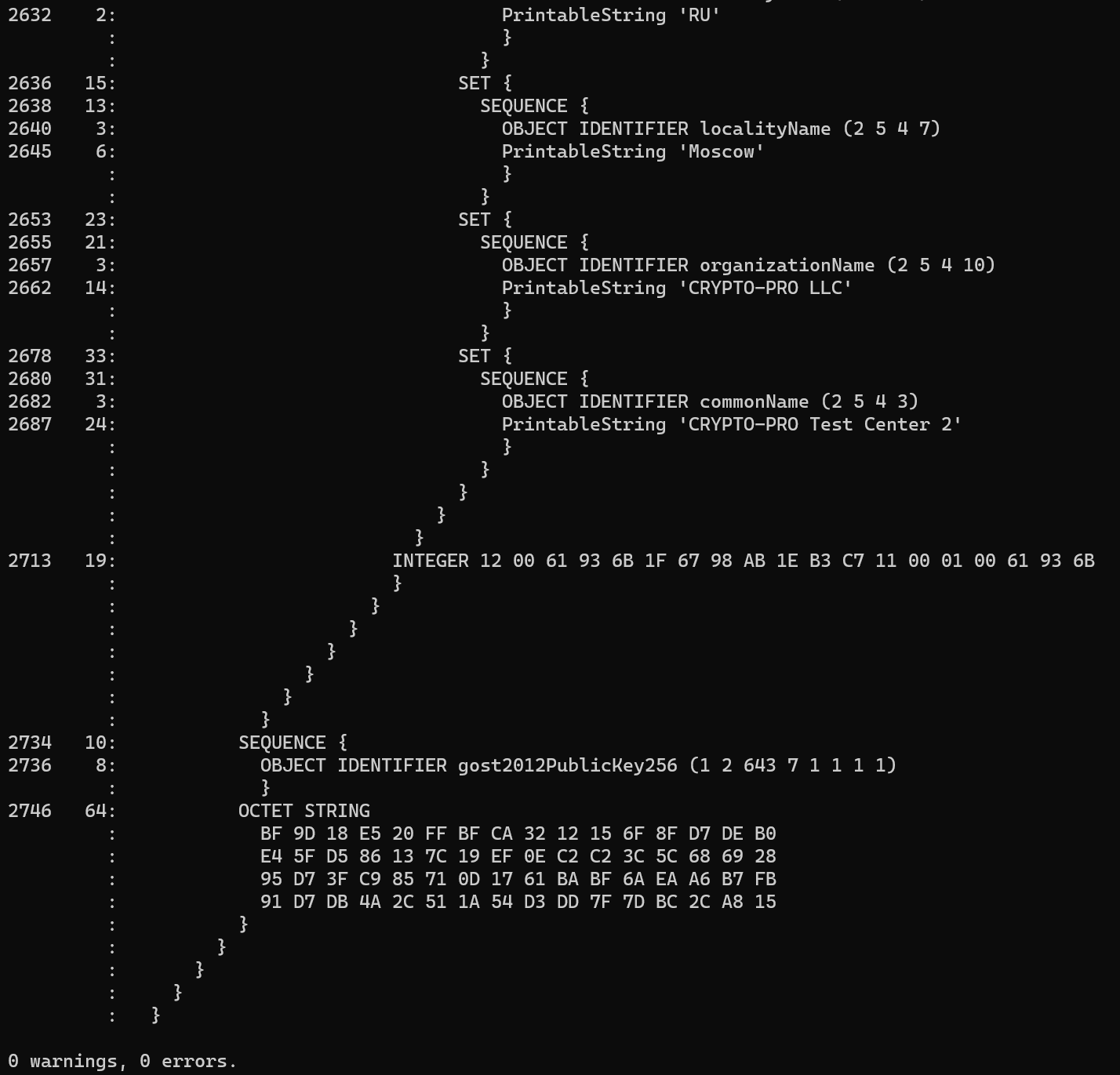
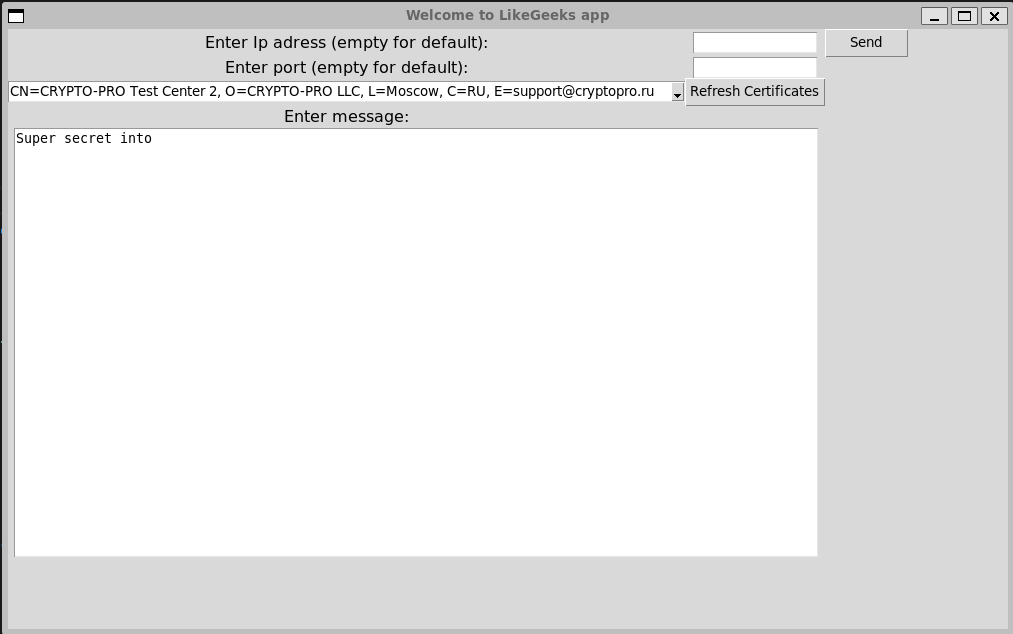
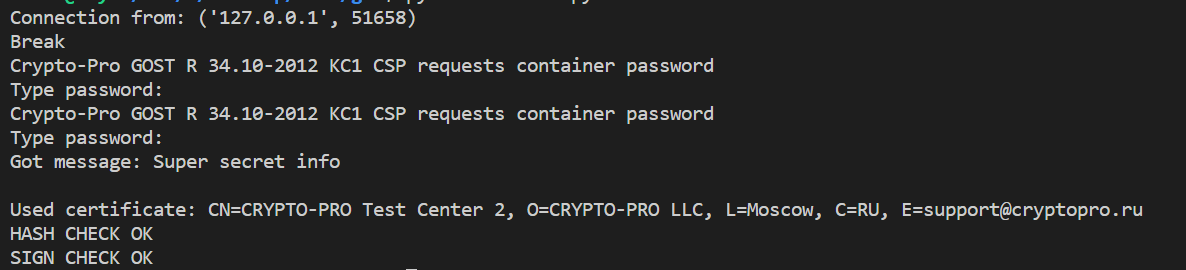


Рисунок 17 – Шестая часть ASN1 заголовка ЭЦП

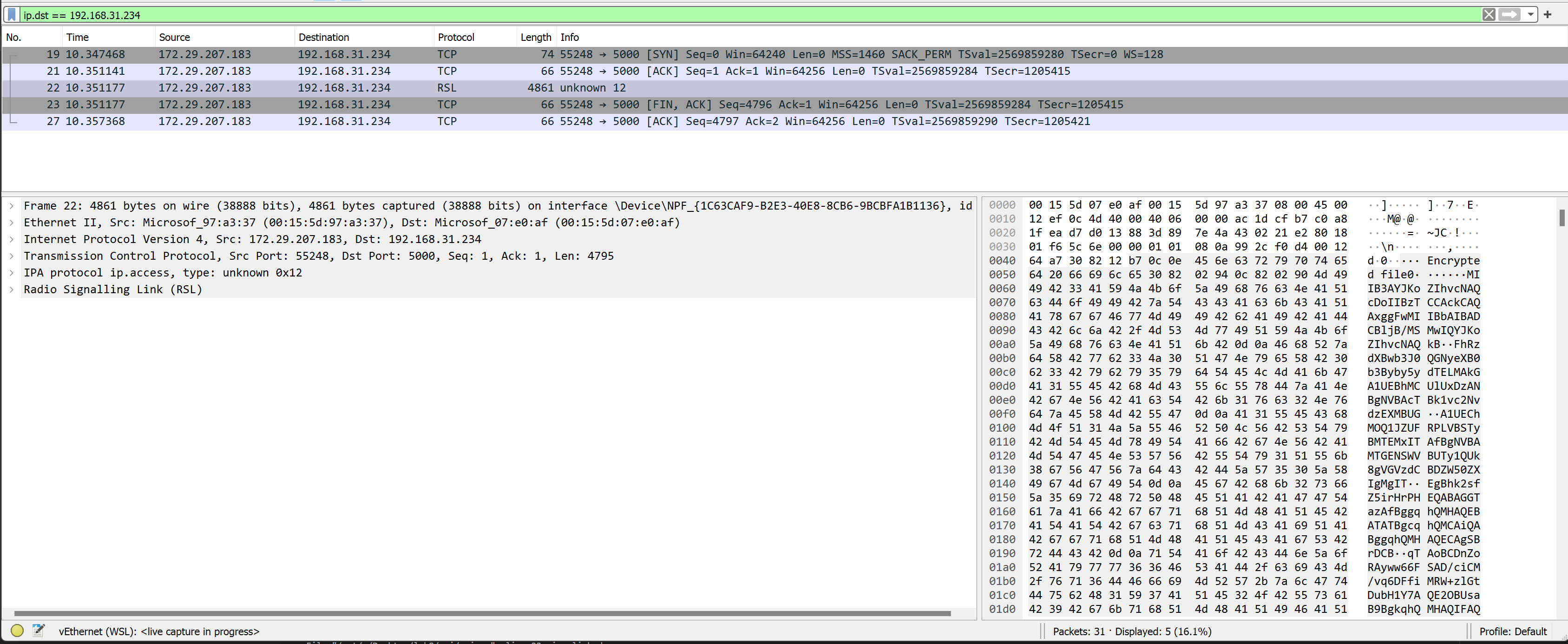
1. **Полученный результат**

****

**Рисунок 18 – GUI клиента**

****

**Рисунок 19 – Полученное сообщение сервером**

****

**Рисунок 20 – Передача сообщения по сети**

1. **Ответы на контрольные вопросы**
2. На чём основывается стойкость шифра ГОСТ 28147-89?

Стойкость шифра ГОСТ 28147-89 основывается на использовании длинных ключей (до 256 бит) и сложности алгоритма шифрования. Кроме того, шифр ГОСТ 28147-89 использует нелинейные преобразования, которые затрудняют атаки на основе линейного криптоанализа.

1. Какие криптопровайдеры поддерживаются стандартной библиотекой CryptoAPI в ОС семейства Windows?

Стандартная библиотека CryptoAPI в ОС семейства Windows поддерживает криптопровайдеры, такие как Microsoft Base Cryptographic Provider, Microsoft Enhanced Cryptographic Provider и Microsoft RSA SChannel Cryptographic Provider, а также криптопровайдеры от других производителей, таких как Gemalto, Safenet и другие.

1. В чём заключаются отличия между алгоритмами ГОСТ Р 34.10-94 и DSS?

Алгоритмы ГОСТ Р 34.10-94 и DSS отличаются в нескольких аспектах. Например, ГОСТ Р 34.10-94 использует умножение точек на эллиптической кривой, а DSS использует возведение в степень. Наконец, форматы ключей, используемые в ГОСТ Р 34.10-94 и DSS, отличаются.

1. Какова возможные разрядности закрытого и открытого ключей в ГОСТ Р 34.10-94?

В ГОСТ Р 34.10-94 используются ключи длиной от 512 до 1024 бит для закрытого и открытого ключей. Размер ключа выбирается в зависимости от желаемого уровня стойкости шифрования.

1. Чем обеспечивается защита ключей в электронных носителях типа eToken, ruToken?

Защита ключей в электронных носителях типа eToken и ruToken обеспечивается путем использования аппаратных средств шифрования, таких как криптопроцессоры и криптоконтроллеры, а также защиты паролем и механизмами аутентификации. Ключи, хранящиеся на этих устройствах, защищены от несанкционированного доступа и могут использоваться только после аутентификации пользователя.

**7) Вывод**

**В ходе выполнения данной лабораторной работы были изучены основные алгоритмы криптографии, используемые в современных криптосистемах. Был реализован клиент-серверное приложение с использованием криптопровайдера КриптоПро CSP, которое позволило защитить передаваемую информацию с помощью алгоритма шифрования данных ГОСТ 28147-89, а также проверяло целостность сообщения при помощью Алгоритма хеширования данных ГОСТ Р 34.11-94 и вычислять ЭЦП с помощью Алгоритмов цифровой подписи ГОСТ Р 34.10-94.**