УДК 004.032.26

Использование библиотеки OpenSSL для создания своего собственного сертификата X.509

Ульянов Егор Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
Целью данной статьи является, применение библиотеки OpenSSL, в создание своего личного сертификата формата X.509. Для этого использовались стандартные методы библиотеки. Итогами исследования стал созданный личный сертификат.
Ключевые слова: электронная цифровая подпись, ЭЦП, OpenSSL, личный сертификат

Using the OpenSSL library to create your own X.509 certificate

Ulianov Egor Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to use the OpenSSL library to create your own personal certificate in format X.509. Standard library methods were used for this. The results of the study were a personal certificate created.
Keywords: electronic digital signature, EDS, OpenSSL, personal certificate

1 Введение

1.1 Актуальность исследования

В криптографии стандарт X.509 - это стандарт международного союза электросвязи (ITU-T), определяющий формат сертификатов открытых ключей. Сертификаты X.509 используются во многих интернет-протоколах, включая TLS / SSL, который является основой для HTTPS, безопасного протокола для просмотра веб-страниц, также используются в автономных приложениях, таких как электронные подписи.

Сертификат X.509 привязывает идентификатор к открытому ключу с помощью цифровой подписи. Сертификат содержит идентификационные данные (имя хоста, или организации, или отдельного лица) и открытый ключ (RSA, DSA, ECDSA, ed25519 и т.д.) и либо подписывается центром сертификации, либо подписывается самостоятельно. Когда сертификат подписан доверенным центром сертификации или проверен другими способами, тот, кто владеет этим сертификатом, может использовать

открытый ключ, который он содержит, для установления безопасной связи с другой стороной или проверки документов, подписанных цифровой подписью соответствующим закрытым ключом.

Промышленные предприятия используют технологию цифровой подписи для оптимизации процессов и улучшения целостности документов. К отраслям, использующим цифровые подписи, относятся следующие:

Правительство. Издательство правительства публикует электронные версии бюджетов, государственных и частных законов и законопроектов с цифровыми подписями. Цифровые подписи используются правительствами во всем мире по целому ряду причин, включая обработку налоговых деклараций, проверку транзакций между бизнесом и правительством (B2G), ратификацию законов и управление контрактами. Большинство государственных учреждений должны придерживаться строгих законов, правил и стандартов при использовании цифровых подписей. Многие правительства и корпорации также используют смарт-карты для идентификации своих граждан и сотрудников. Это физические карты, снабженные цифровой подписью, которые могут использоваться для предоставления владельцу карты доступа к системам учреждения или физическим зданиям.

Здравоохранение. Цифровые подписи используются в сфере здравоохранения для повышения эффективности лечения и административных процессов, для усиления безопасности данных, для электронного назначения лекарств и госпитализации.

Производство. Компании-производители используют цифровые подписи для ускорения процессов, включая разработку продукта, обеспечение качества (QA), усовершенствования производства, маркетинг и продажи. Использование цифровых подписей в производстве регулируется Международной организацией по стандартизации (ISO).

Финансовые услуги. Финансовый сектор использует цифровые подписи для контрактов, безбумажного банковского обслуживания, обработки кредитов, страховой документации, ипотеки и многого другого. Этот жестко регулируемый сектор использует цифровые подписи с пристальным вниманием к правилам и рекомендациям.

Криптовалюты. Цифровые подписи также используются в биткойнах и других криптовалютах для аутентификации блокчейна. Они также используются для управления данными транзакций, связанных с криптовалютой, и как способ для пользователей показать право собственности на валюту или свое участие в транзакции.

1.2 Обзор исследований

Исследованиями в данной теме занимались следующие авторы. К. Н. Онуфриев, Д. М. Колодный, В. А. Прилипской, В. Д. Береговой, Д. С. Толстов, П. А. Яшников, С. А. Аладинский, В. С. Повод, Е. А. Смирнов, Д. Н. Анищенко в своей работе описали разработку программы, предназначенную для автоматизации учета и хранения сертификатов ключей проверки

электронной подписи участников электронного взаимодействия [1]. К.Н.Агулова в своей статье определила возможности языка программирования python считывания или проверки данных из сертификата электронной подписи [2]. Такие авторы как Д. Расяева, Е.Тұрсынбек, І.Жолдыбай, А.Шингишева описали электронные цифровые подписи и сертификаты, их преимущества и возможности [3].

1.3 Цель исследования

Цель исследования – применение библиотеки OpenSSL, в создание своего личного сертификата формата X.509.

2. Материалы и методы

Для создания личного сертификата будем использовать полноценную криптографическую библиотеку с открытым исходным кодом – OpenSSL [4], так как данный инструмент обладает поддержкой почти всех низкоуровневых алгоритмов хеширования, шифрования и электронной подписи, а также реализует большинство популярных криптографических стандартов, в том числе: позволяет создавать ключи RSA, DH, DSA, сертификаты X.509, подписывать их, формировать CSR и CRT, шифровать данные и тестировать SSL/TLS соединения.

3 Результаты и дискуссия

Начнём создание сертификата. После того как был скачан и установлен дистрибутив библиотеки с официального сайта (рис. 1)., необходимо в переменные среды «Windows» добавить путь к папке с файлами библиотеки (рис.2).

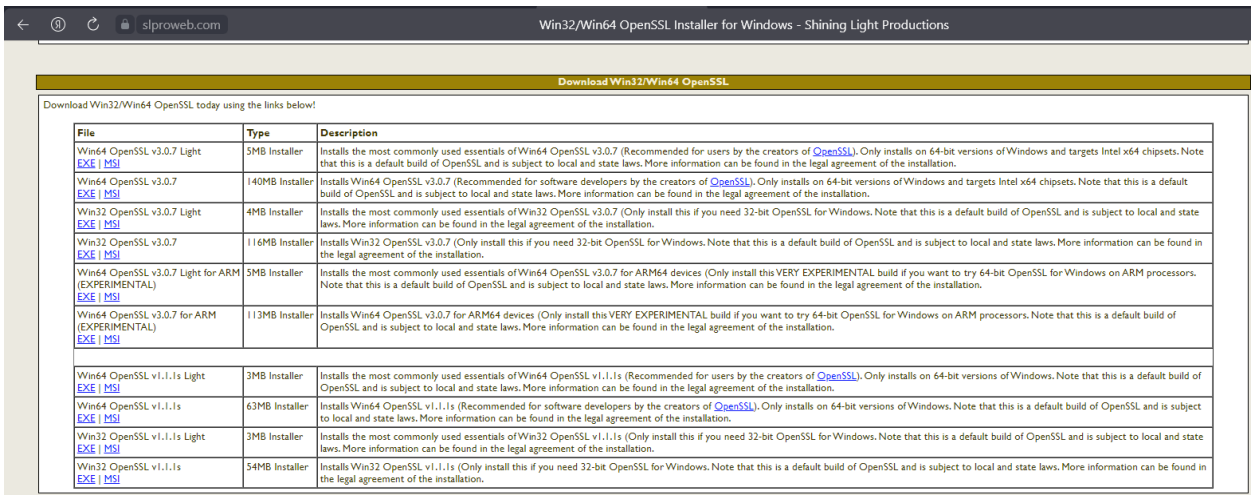

Рисунок 1. Официальный сайт OpenSSL

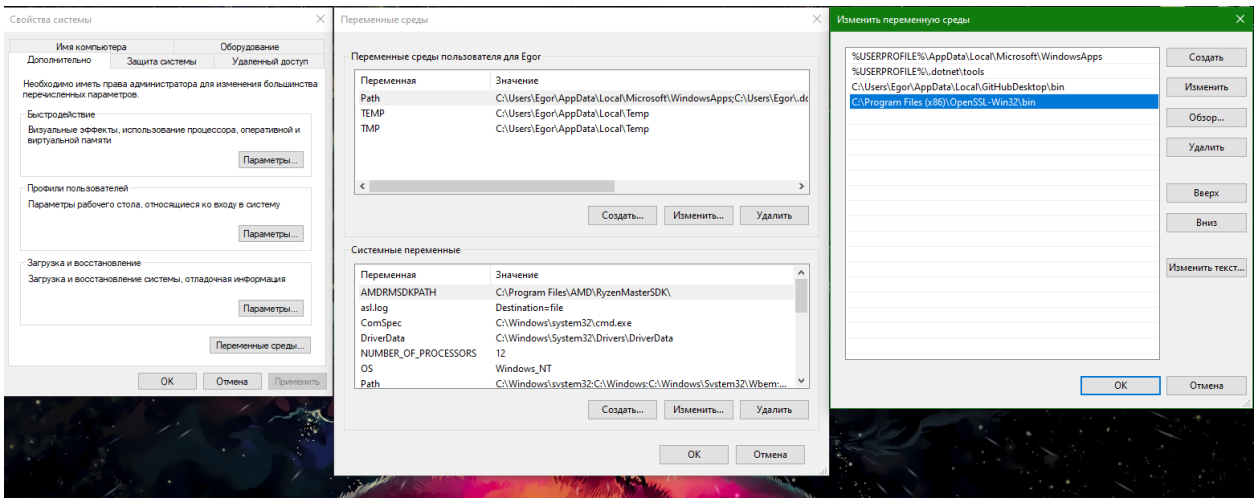

Рисунок 2. Добавление пути к библиотеке

Далее переходим в командную строку и проверяем работу OpenSSL, для этого вводим команду openssl (рис.3).

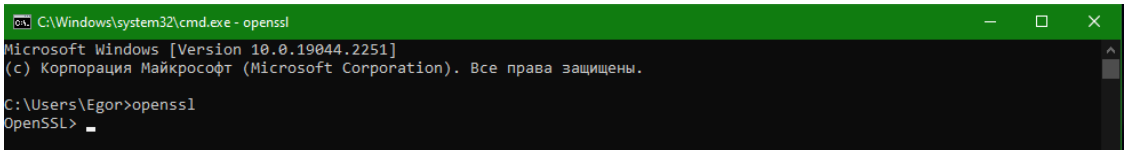

Рисунок 3. Проверка работы командной строки с библиотекой

Возвращаемся обратно в директорию пользователя и прописываем команду “openssl req -x509 -days 365 -newkey rsa:2048 -keyout my-key.pem -out my-cert.pem”, где: команда openssl req вызывает библиотеку, days – срок действия сертификата, опция newkey указывает, что нужно создать новую пару ключей, а в параметрах сообщаем тип rsa и сложность 2048 байт, keyout - указывает домен, для которого генерируеться ключ, out - указывает имя файла, в котором будет сохранен pem. Пишем пароль и подтверждаем, далее отвечаем на вопросы о сертификате: страна, область, город, организация, организационная единица, сайт компании, электронная почта компании (рис.4).

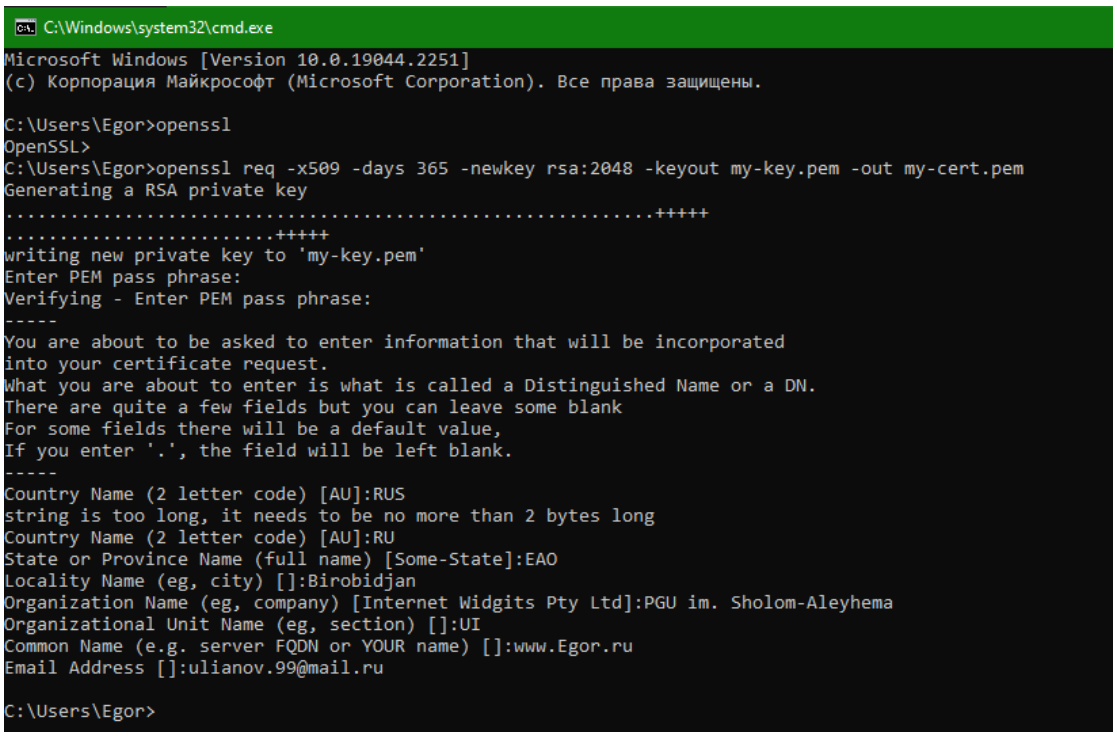

Рисунок 4. Создание сертификата X.509

Таким образом в директории указанной в командной строке появляются два файла, один из файлов является ключом, второй сертификатом (рис. 5).

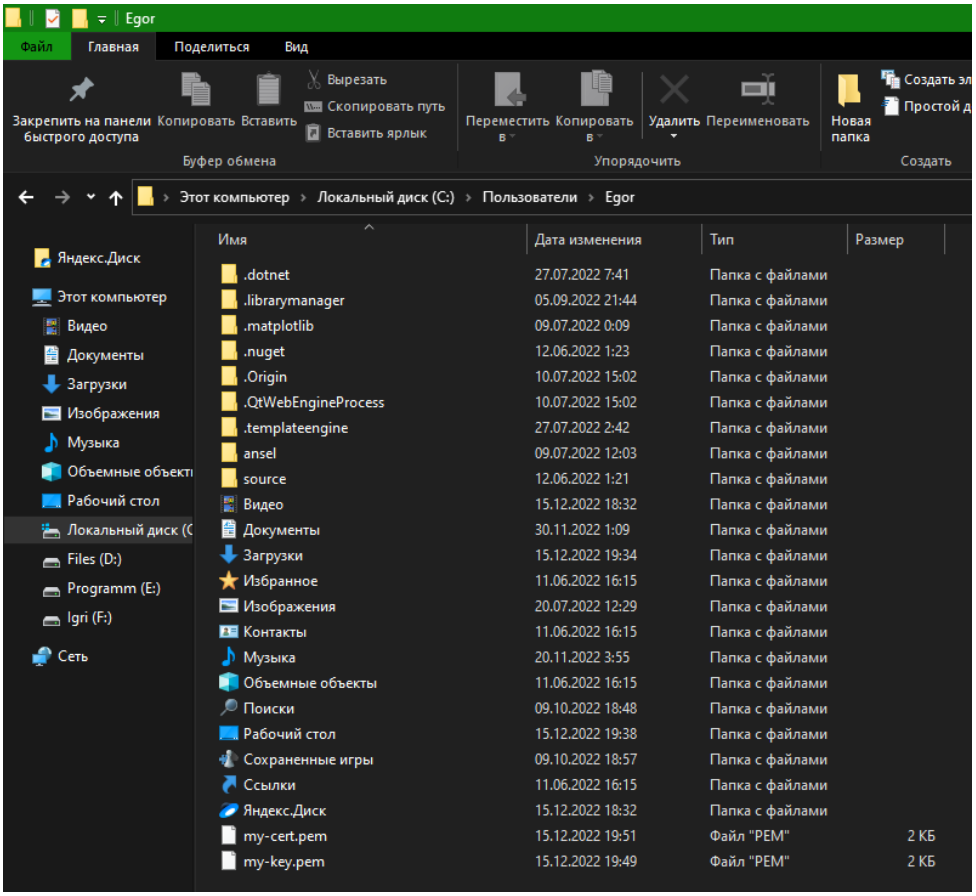

Рисунок 5. Результат работы команды

Ключ и сертификат можно объединить, для этого используем данную команду “openssl pkcs12 -export -in my-cert.pem -inkey my-key.pem -out Egor-test-sert.pfx”, где: используем стандарт PKCS #12, который определяет формат архивного файла для хранения множества объектов криптографии в виде одного файла, export – экспорт двух ранее созданных файлов. В конце пишем выходной файл с расширением (рис.6).

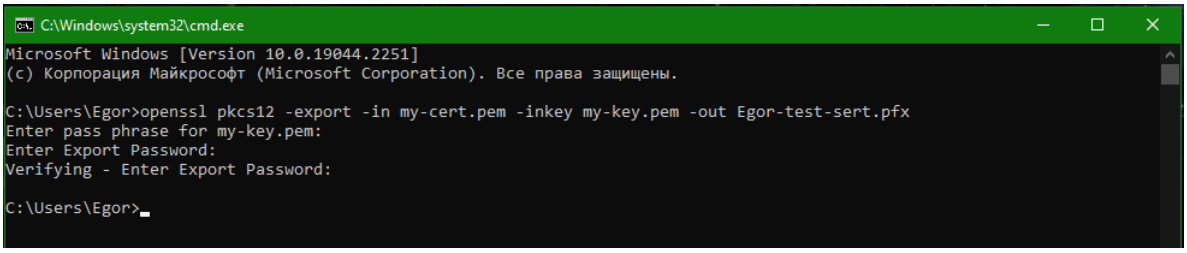

Рисунок 6. Объединение сертификата с ключом

В директории указанной в командной строке появиться файл с названием, указанным выше, и расширением pfx. Pfx - это двоичный формат, который часто используется для хранения всех элементов цепочки доверия, таких как сертификат сервера, любые промежуточные сертификаты и закрытый ключ, в одном зашифрованном файле

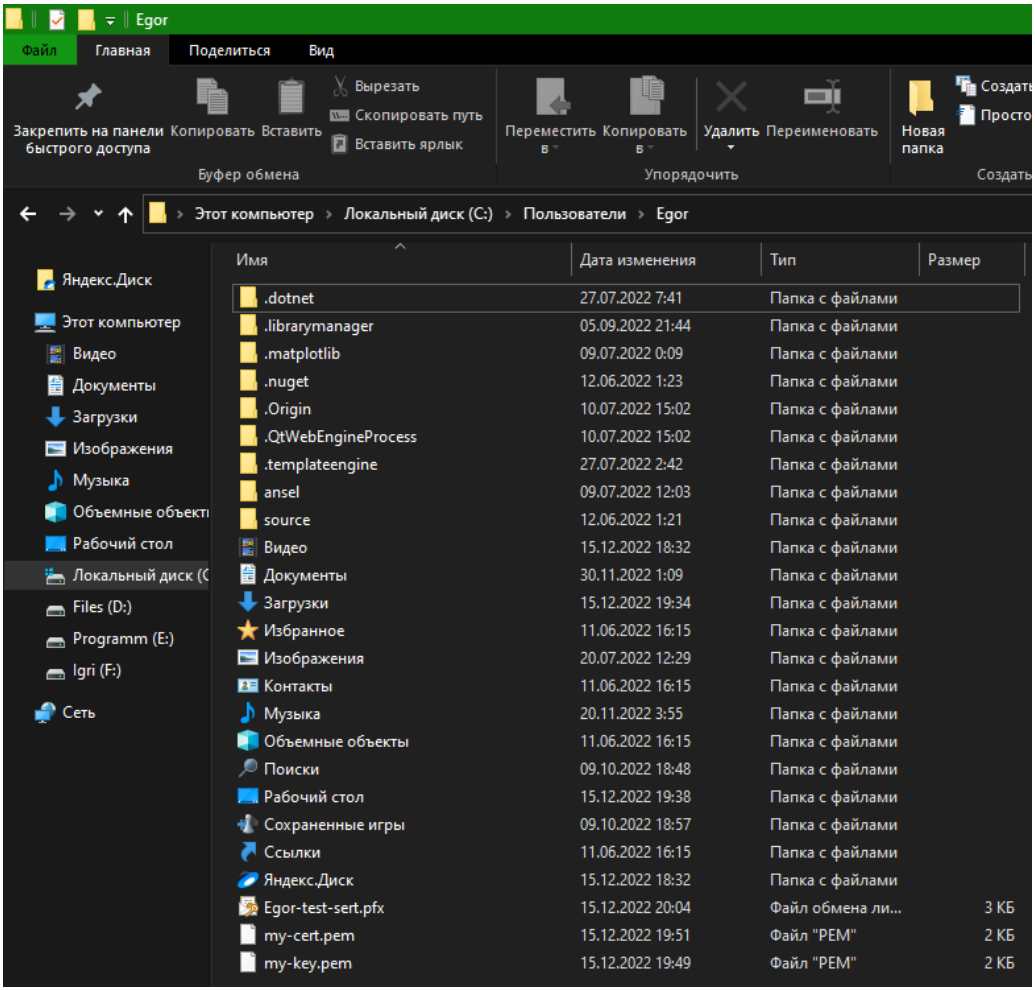

Рисунок 7. Форматирование оставшихся категориальных данных

Далее для тестирования работы сертификата, извлечём из него открытый ключ командой “openssl pkcs12 -in Egor-test-sert.pfx -clcerts -nokeys -out Egor-test-cert-public.pem” (рис.8).

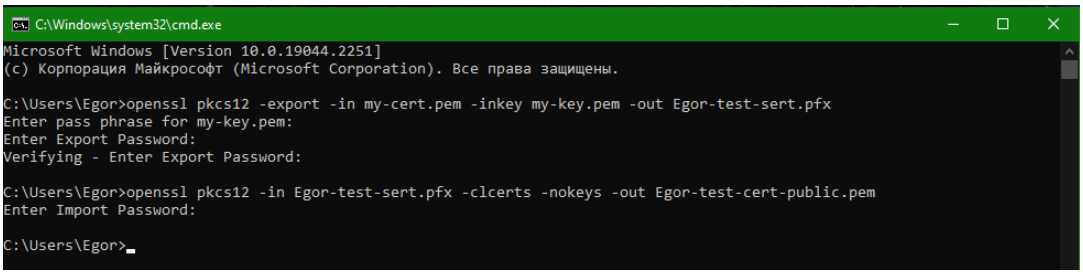

Рисунок 8. Извлечение открытого ключа

Работа командой строки завершилась и в директории с сертификатом появился файл, содержащий открытый ключ (рис.9).

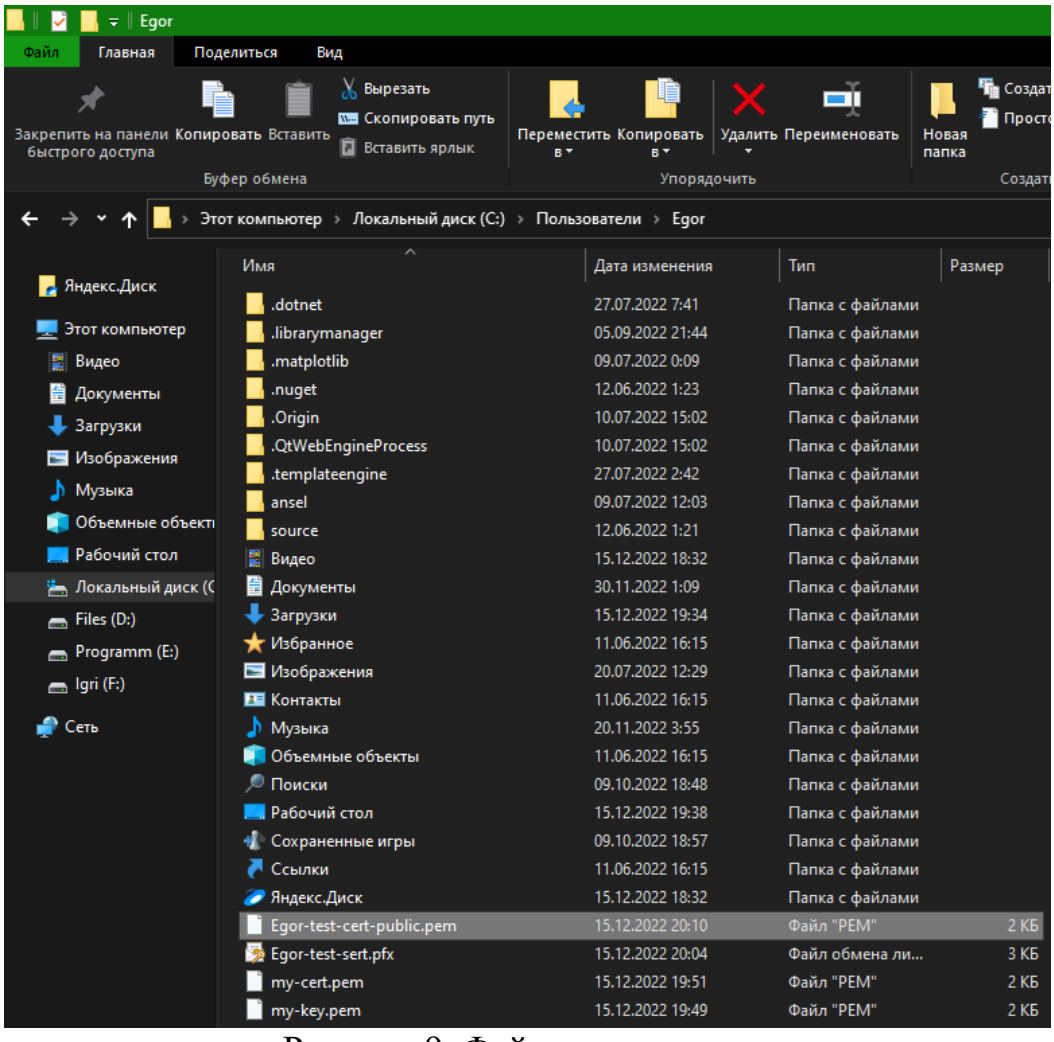

Рисунок 9. Файл с открытым ключом

4 Выводы

В рамках данного исследования описана разработка и небольшой тест собственного электронного сертификата с помощью библиотеки OpenSSL.

В дальнейшем планируется добавление новых функций, улучшение безопасности и применение подобного сертификата на собственном сайте.

Библиографический список

1. Онуфриев К. Н., Колодный Д. М., Прилипской В. А., Береговой В. Д., Толстов Д. С., Яшников П. А., Аладинский С. А., Повод В. С., Смирнов Е. А., Анищенко Д. Н. Программный модуль автоматизации учета и хранения сертификатов ключей проверки электронной подписи участников электронного взаимодействия. 2022.
2. Агулова К.Н. Возможности языка python при обработке сертификата электронной подписи // Материалы и методы инновационных научно-практических исследований и разработок. 2019. С. 31-33.
3. Расяева Д., Тұрсынбек Е., Жолдыбай І., Шингишева А. Электронная цифровая подпись и сертификат// Актуальные научные исследования в современном мире. 2021. С. 118-120.
4. OpenSSl URL: https://www.openssl.org/ (дата обращения: 15.12.2022).