



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

**Институт
информационных
технологий**

**Кафедра
Информационных систем**

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ № 6
НА ТЕМУ:
« Освоение программных средств для работы с сертификатами
открытых ключей »

ПО ДИСЦИПЛИНЕ
« Защита информации »

СТУДЕНТА 4 КУРСА бакалавриата ГРУППЫ ИДБ-20-02

ЕРДОГАН ДЕНИЗ ЕРДАЛОВИЧ

Направление: 09.03.01 Информатика и вычислительная техника
Профиль подготовки: Информатика и вычислительная техника

Отчет сдан « _____ » _____ 2023 г.

Оценка _____

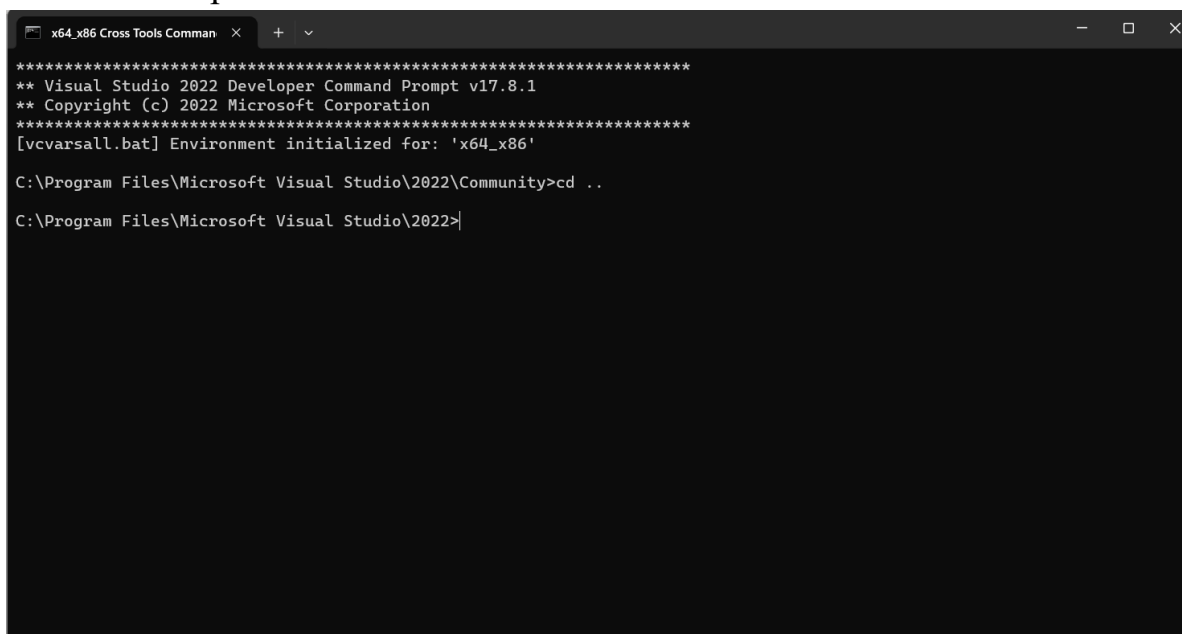
Преподаватель _____ Симонов М.Ф. _____.

МОСКВА 2023

1. Начать сеанс работы;

Для вызова утилит командной строки использовать командную строку Windows (Пуск | Программы | Microsoft Visual Studio 2005 | Visual Studio Tools| Visual Studio 2005 Command Prompt):

Найдём в поиске ОС Windows консоль “Developer Command Prompt for VS 2022” и откроем её:



```
*****  
** Visual Studio 2022 Developer Command Prompt v17.8.1  
** Copyright (c) 2022 Microsoft Corporation  
*****  
[vcvarsall.bat] Environment initialized for: 'x64_x86'  
  
C:\Program Files\Microsoft Visual Studio\2022\Community>cd ..  
  
C:\Program Files\Microsoft Visual Studio\2022>
```

Рисунок № 1 – Developer Command Prompt for VS 2022 (Windows 11)

Для завершения работы в режиме командной строки использовать команду exit;

2. Скопировать в свою индивидуальную папку на рабочей станции документ Microsoft Word «СИСТЕМНЫЕ ПРОГРАММЫ ДЛЯ РАБОТЫ С СЕРТИФИКАТАМИ» из указанного преподавателем места:

Скопируем необходимый файл на рабочую станцию:

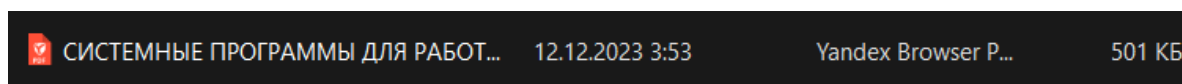


Рисунок № 2 – скопированный файл на рабочей станции (Windows 11)

3. Открыть скопированный в п. 2 документ и ознакомиться с его разделом 1 «Создание сертификатов»:

Ознакомимся с требуемым по заданию разделом:

СИСТЕМНЫЕ ПРОГРАММЫ ДЛЯ РАБОТЫ С СЕРТИФИКАТАМИ

1. Создание сертификатов

Для создания сертификата требуется иметь секретный ключ его издателя. В состав операционной системы Windows входит сертификат издателя по умолчанию (например, Root Agency) и связанный с ним секретный ключ.

Самоподписанные (self-signed) сертификаты могут использоваться как в тестовых целях, так и в качестве доверенных корневых сертификатов. Подпись под таким сертификатом вычисляется с помощью секретного ключа, связанного с открытым ключом из создаваемого сертификата. Доверенные корневые сертификаты могут затем использоваться для удостоверения новых сертификатов.

Для создания самоподписанных сертификатов, а также сертификатов, удостоверенных с их помощью или с помощью имеющихся сертификатов издателей, может использоваться системная программа MakeCert, которая представляет собой утилиту командной строки. Формат строки вызова этой системной программы следующий:

MakeCert [*базовые опции* | *расширенные опции*] [*имя выходного файла*]

Имя выходного файла может быть опущено, если создаваемый сертификат не должен записываться в файл.

Опции, которые могут быть указаны при вызове утилиты MakeCert, разделяются на три группы:

- базовые опции, управляющие созданием и хранением созданного сертификата;
- расширенные опции, применимые к свойствам создаваемого сертификата, *сертификатам издателей* и связанным с ними секретным ключам и их хранению.

Большинство опций программы MakeCert доступны при использовании обозревателя Internet Explorer версии 4.0 и выше. В табл. 1 приведено описание базовых опций программы MakeCert.

...

Рисунок № 3 – раздел для ознакомления (Windows 11)

4. С помощью утилиты командной строки MakeCert выполнить следующее:

MakeCert – инструмент для создания сертификатов, которые предназначены исключительно для тестирования разрабатываемого приложения. Этот инструмент создает пару ключей (открытый и закрытый) для цифровой подписи и помещает её в файл сертификата.

4.1. Создать закрытый ключ ЭЦП и сертификат, подписанный удостоверяющим центром по умолчанию, поместив их в файлы с расширениями соответственно *rvk* и *cer* (имена владельцев сертификатов должны совпадать с фамилиями и инициалами студентов):

Самозаверяющий сертификат – это сертификат, подписанный приложением, которое создало его, т.е. *MakeCert*.

Создадим закрытый ключ ЭЦП и сертификат, подписанный удостоверяющим центром по умолчанию, поместив их в файлы с расширениями *.pvk и *.cer, соответственно. Для этого введем команду – “makecert -n "CN=ErdoganDE" -sv C:\pro\deniz1.pvk C:\pro\ErdoganDE1.cer” (мы создаем сертификаты в отдельной папке на рабочей машине):

- -n (subjectName) – задает имя субъекта. Согласно правилам, к имени субъекта добавляется префикс "CN = " для "Common Name";
- -sv (privateKeyFile) – указывает файл, содержащий контейнер закрытого ключа. То есть закрытый ключ будет храниться не в сертификате, а в файле.

Появится окно, где необходимо задать пароль для закрытого ключа и подтвердить его:

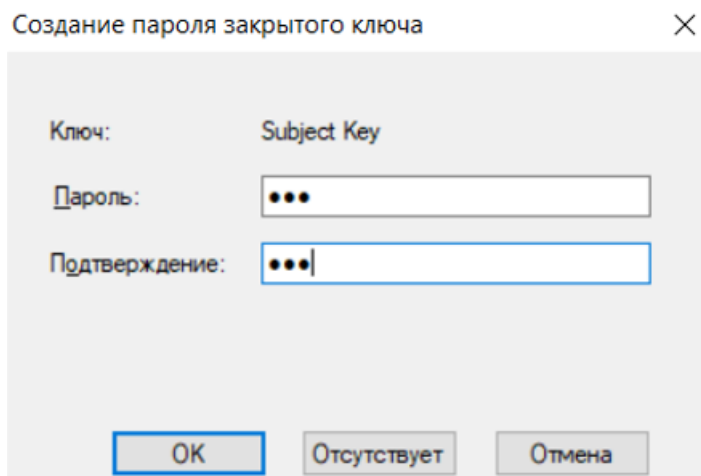


Рисунок № 4 – создание пароля для закрытого ключа (Windows 11)

Если мы ничего не введём, то появится следующая форма:

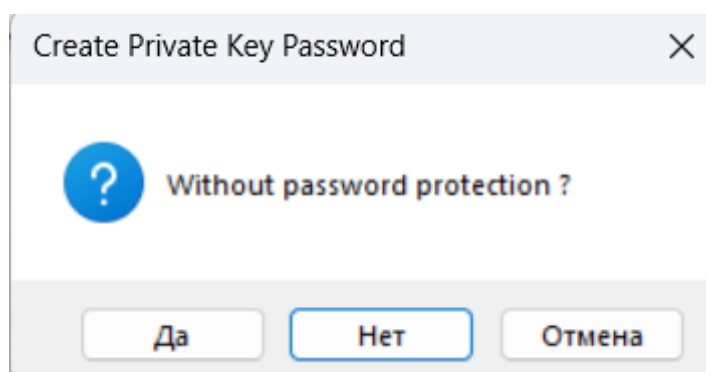


Рисунок № 5 – уточнение при отсутствии пароля (Windows 11)

Если мы не подтвердим/неправильно подтвердим пароль, то появится следующая форма:

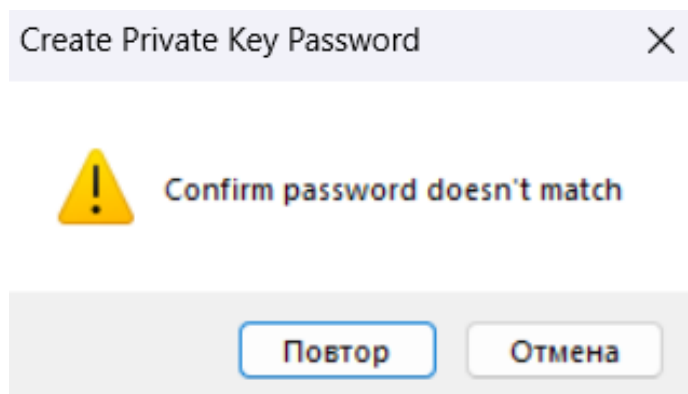


Рисунок № 6 – неправильное подтверждение пароля (Windows 11)

После нужно ввести созданный пароль закрытого ключа:

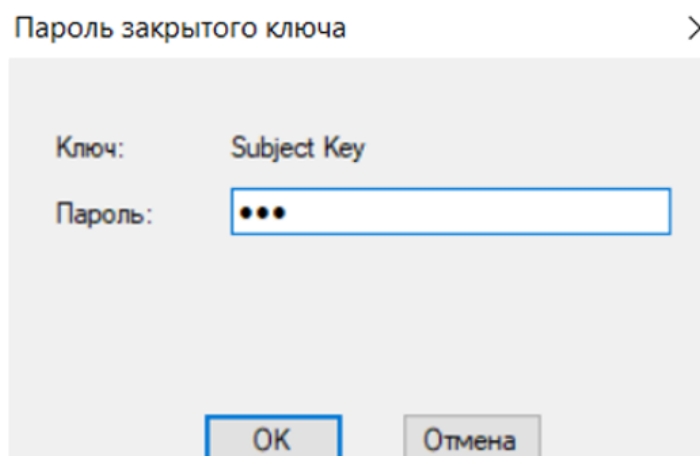


Рисунок № 7 – ввод созданного пароля закрытого ключа (Windows 11)

Если мы не введём пароль/неправильно введём пароль, то появится сообщение об ошибке в консоли:



Рисунок № 8 – ошибка при вводе созданного пароля (Windows 11)

Иначе получим следующее сообщение:

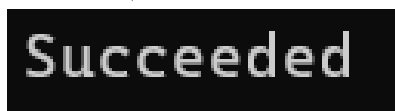


Рисунок № 9 – успех при вводе созданного пароля (Windows 11)

После заходим в нашу папку и видим созданные файлы:

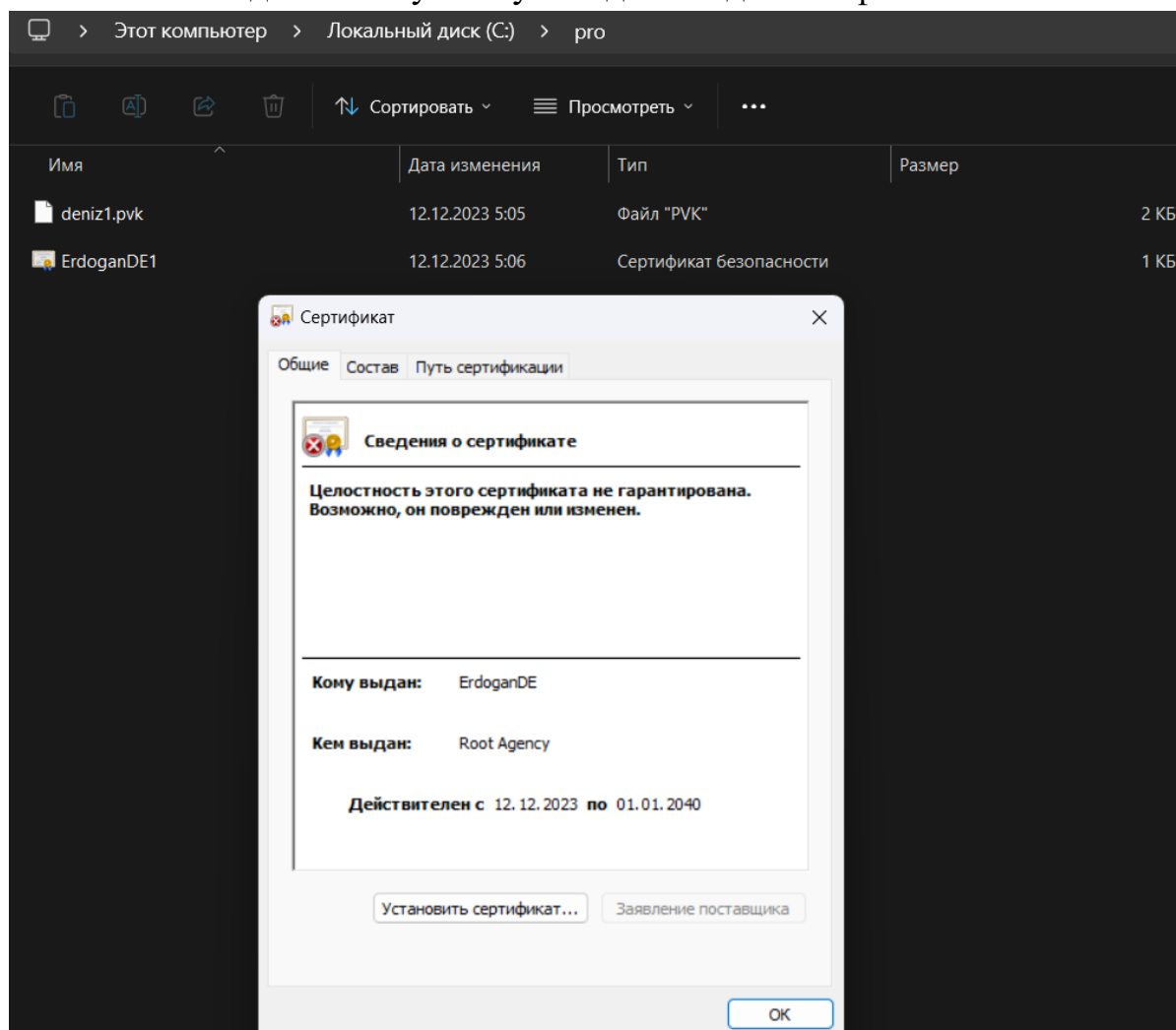


Рисунок № 10 – созданные сертификат и закрытый ключ (Windows 11)

4.2. Повторить п. 4.1, но поместить закрытый ключ и сертификат в хранилище сертификатов *My*:

Повторим предыдущий пункт, но поместим закрытый ключ и сертификат в хранилище сертификатов *My*. Для этого введем команду - «*makecert -n "CN=ErdoganDE" -sv C:\pro\deniz2.pvk -ss My C:\pro\ErdoganDE2.cer*» в консоль. После установим пароль, введём его ещё раз.

- *-n* (subjectName) – задает имя субъекта. Согласно правилам, к имени субъекта добавляется префикс "CN = " для "Common Name";

Параметры `-sr currentuser -ss My` указывают, что сертификат нужно поместить в хранилище текущего пользователя в раздел *My*.

Все окна будут индентичны пункту выше.

По итогу сертификат успешно создан (чтобы открыть хранилище сертификатов, нужно вбить в командной строке поиска «*Управление сертификатами пользователя*», при этом *My* — это раздел личных сертификатов):

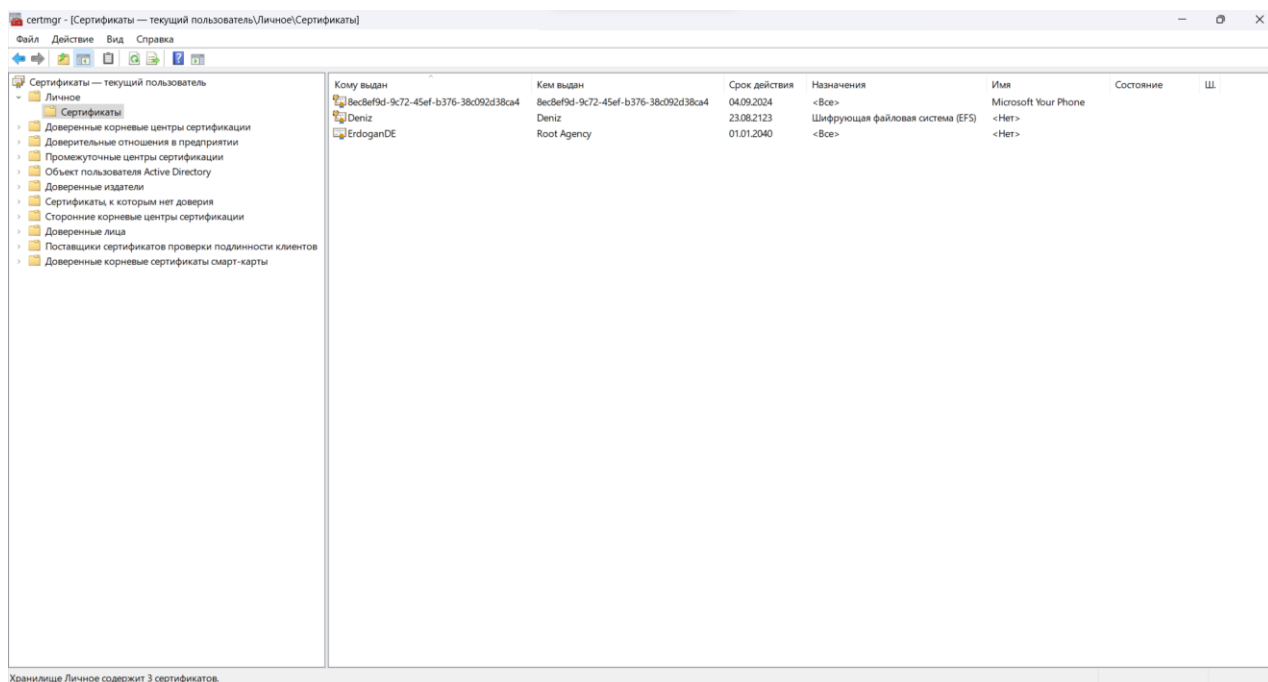


Рисунок № 11 – созданный сертификат в хранилище *My* (Windows 11)

4.3. С помощью созданных в п. 4.2 закрытого ключа и сертификата создать и удостоверить новый сертификат, поместив его в хранилище *TrustedPeople*:

Создадим и удостоверим новый сертификат, поместив его в хранилище *TrustedPeople*. Для того, чтобы это выполнить, введем в консоль команду – “`makecert -is my -ic C:\pro\ErdoganDE2.cer -ss TrustedPeople C:\pro\ErdoganDE3.cer`”.

Нам надо будет ввести пароль закрытого ключа, который мы создавали в предыдущем пункте:

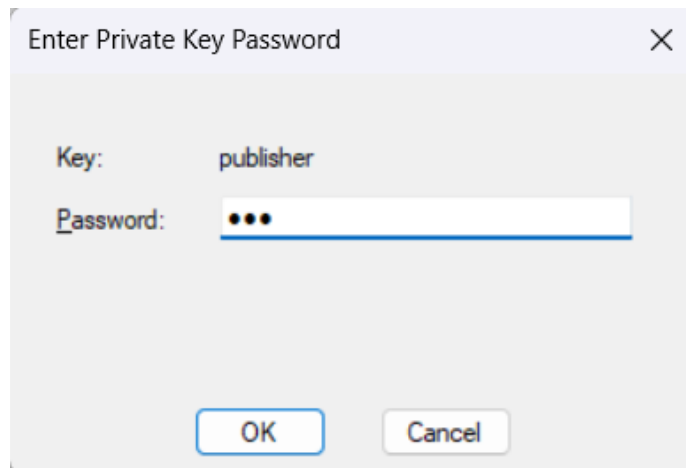


Рисунок № 12 – создание нового сертификата в хранилище TrustedPeople (Windows 11)

Все окна при выполнении данного пункта аналогичны предыдущим.

Рассмотрим новый созданный сертификат в соответствующем хранилище:

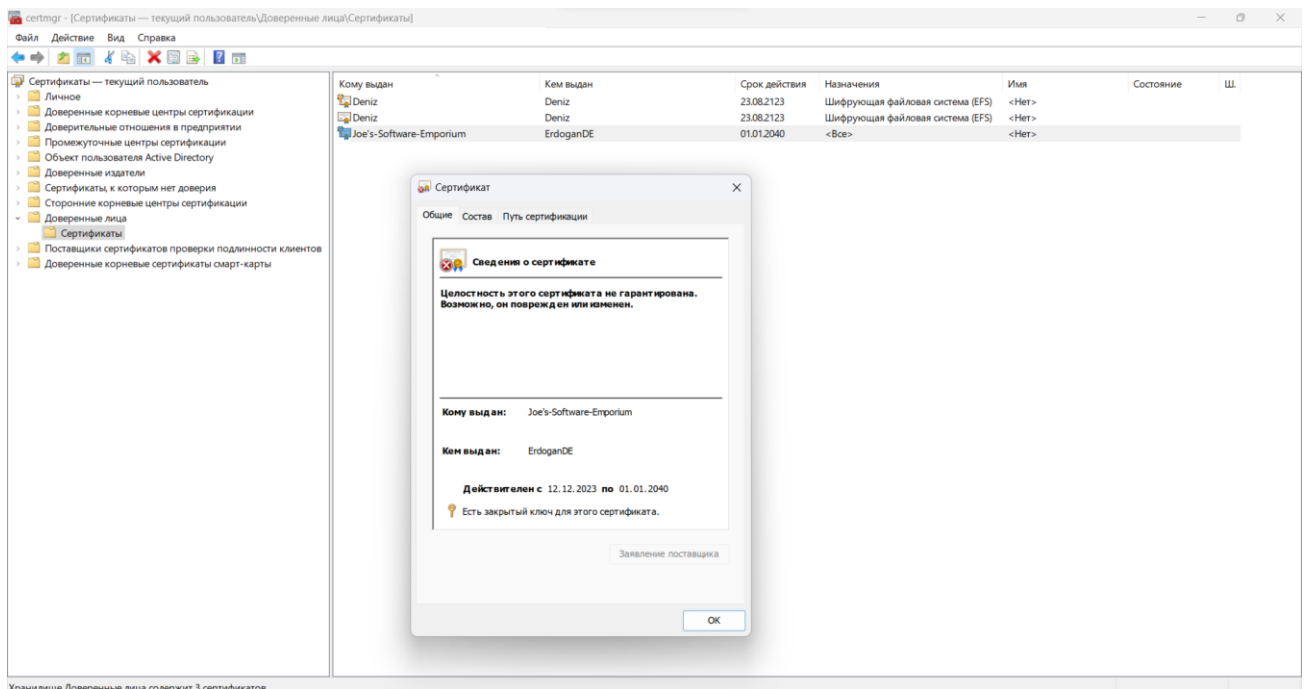


Рисунок № 13 – удостоверенный сертификат в хранилище TrustedPeople (Windows 11)

4.4. Создать закрытый ключ и самоподписанный сертификат, поместив их в хранилище CA:

Создадим закрытый ключ и самоподписанный сертификат, поместив их в хранилище *CA*. Для этого воспользуемся следующей командой – “*makecert –sv C:\pro\deniz4.pvk -r -ss CA C:\pro\ErdoganDE4.cer*”.

Снова задаём и подтверждаем пароль закрытого ключа.

В результате проделанных действий у нас появится новый сертификат вида:

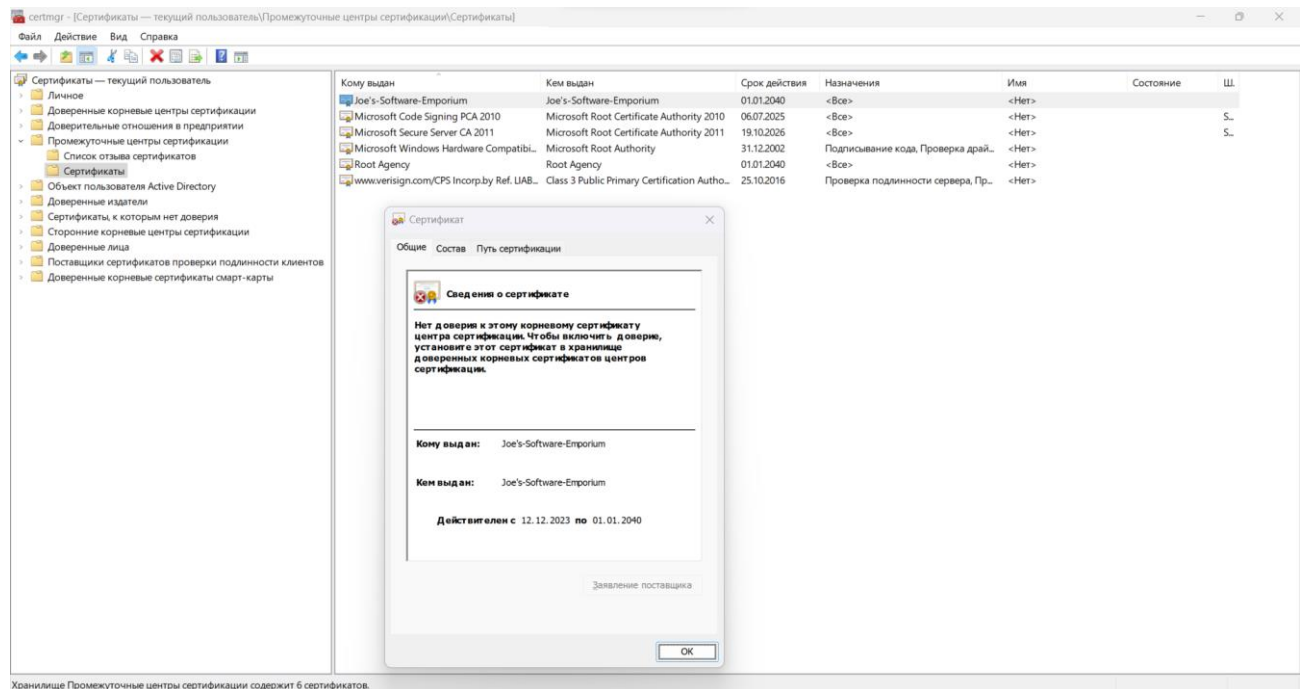


Рисунок № 14 – самоподписанный сертификат (Windows 11)

4.5. Включить в отчет о лабораторной работе:

4.5.1. Сведения о назначении и основных функциях утилиты MakeCert:

Средство *MakeCert* создает сертификат “X.509”, подписанный корневым ключом теста или другим указанным ключом, который привязывает Ваше имя к открытой части пары ключей. Сертификат сохраняется в файле, хранилище системных сертификатов или обоих местах.

Средство устанавливается в папку “*bin*” в пути установки пакета *sdk Microsoft Windows*.

Средство *MakeCert* использует следующий синтаксис команды:

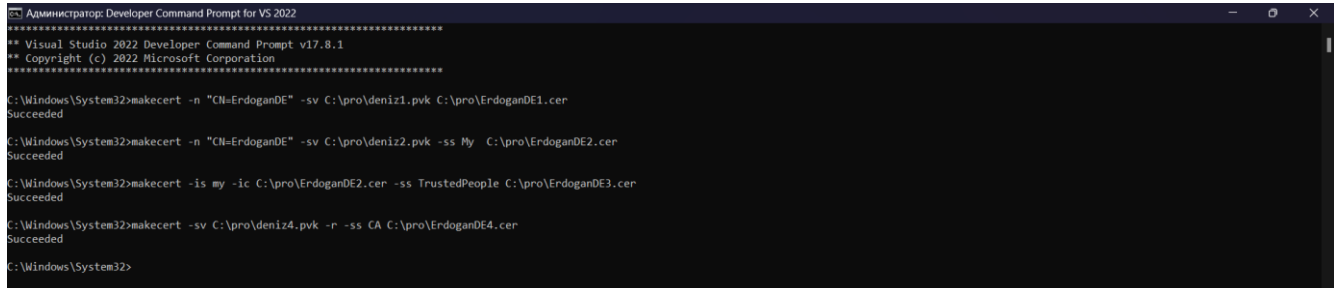
- *MakeCert [BasicOptionsExtendedOptions/] OutputFile:*
 - *OutputFile* — это имя файла, в который будет записан сертификат. Выходной файл можно опустить, если сертификат не записывается в файл.

MakeCert включает базовые и расширенные параметры. Основные параметры используются при создании сертификатов чаще всего.

Дополнительные параметры обеспечивают более гибкое использование программы.

4.5.2. Протокол работы в режиме командной строки, полученный при выполнении п.п. 4.1-4.4 (с помощью системного меню окна командной строки и буфера обмена):

Рассмотрим протокол работы вышепроделанных пунктов:



```
Администратор: Developer Command Prompt for VS 2022
*****
** Visual Studio 2022 Developer Command Prompt v17.8.1
** Copyright (c) 2022 Microsoft Corporation
*****

C:\Windows\System32>makecert -n "CN=ErdoganDE" -sv C:\pro\deniz1.pvk C:\pro\ErdoganDE1.cer
Succeeded

C:\Windows\System32>makecert -n "CN=ErdoganDE" -sv C:\pro\deniz2.pvk -ss My C:\pro\ErdoganDE2.cer
Succeeded

C:\Windows\System32>makecert -is my -ic C:\pro\ErdoganDE2.cer -ss TrustedPeople C:\pro\ErdoganDE3.cer
Succeeded

C:\Windows\System32>makecert -sv C:\pro\deniz4.pvk -r -ss CA C:\pro\ErdoganDE4.cer
Succeeded

C:\Windows\System32>
```

Рисунок № 15 – протокол работы в режиме консоли (Windows 11)

5. С помощью утилиты из состава пакета Microsoft Office SelfCert (в версии Microsoft Office 2003 и старше вызов этой программы возможен через меню Пуск | Программы | Microsoft Office | Средства Microsoft Office | Цифровой сертификат для проектов VBA) выполнить следующее:

5.1. Создать самоподписанный сертификат для субъекта с именем, совпадающим с фамилией и инициалами студента:

Создадим самоподписанный сертификат с помощью программы *selfcert.exe* из пакета *Microsoft Office*. Для этого перейдём по следующему пути *Пуск -> Microsoft Office -> Средство создания цифровых сертификатов для проектов VBA*.

Зададим имя сертификата:

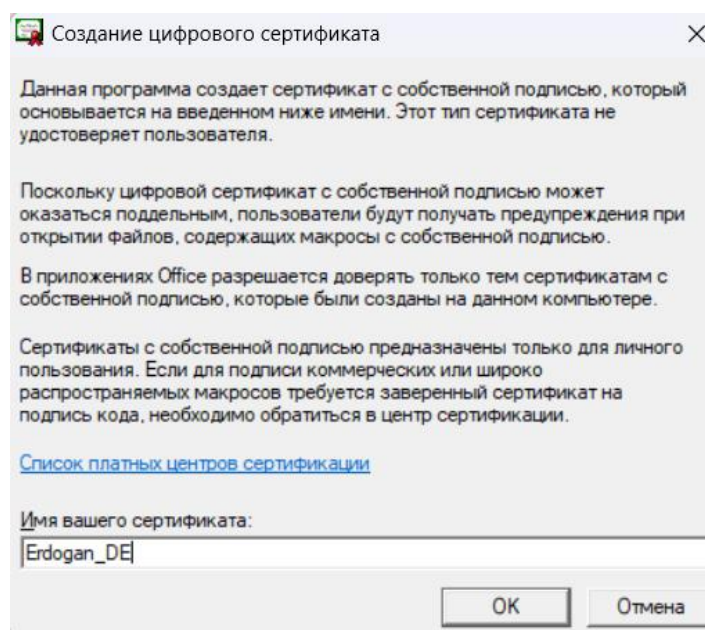


Рисунок № 16 – окно создания цифрового сертификата (Windows 11)

Если не ввести название сертификата, то появится следующее сообщение:

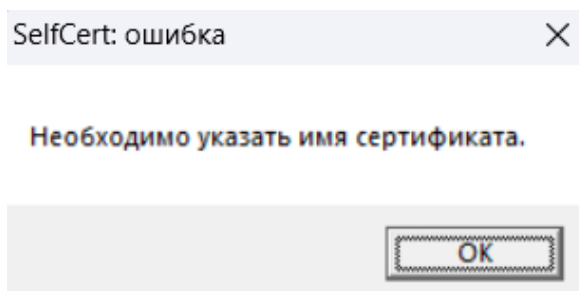


Рисунок № 17 – отсутствие названия сертификата (Windows 11)

После нажатия «OK», появится следующее окно об успешном создании сертификата:

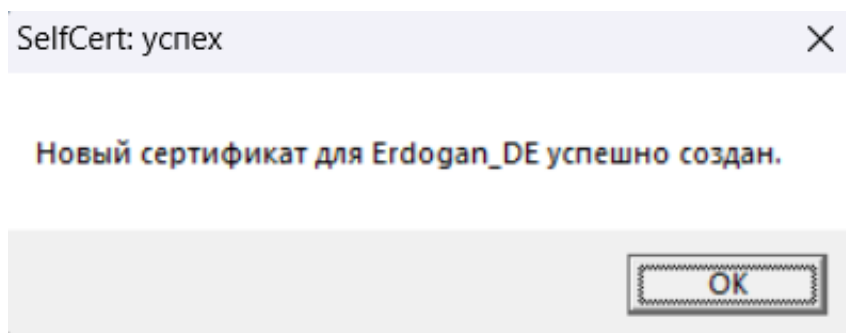


Рисунок № 18 – сообщение о создании нового цифрового сертификата (Windows XP)

Рассмотрим созданный нами сертификат:

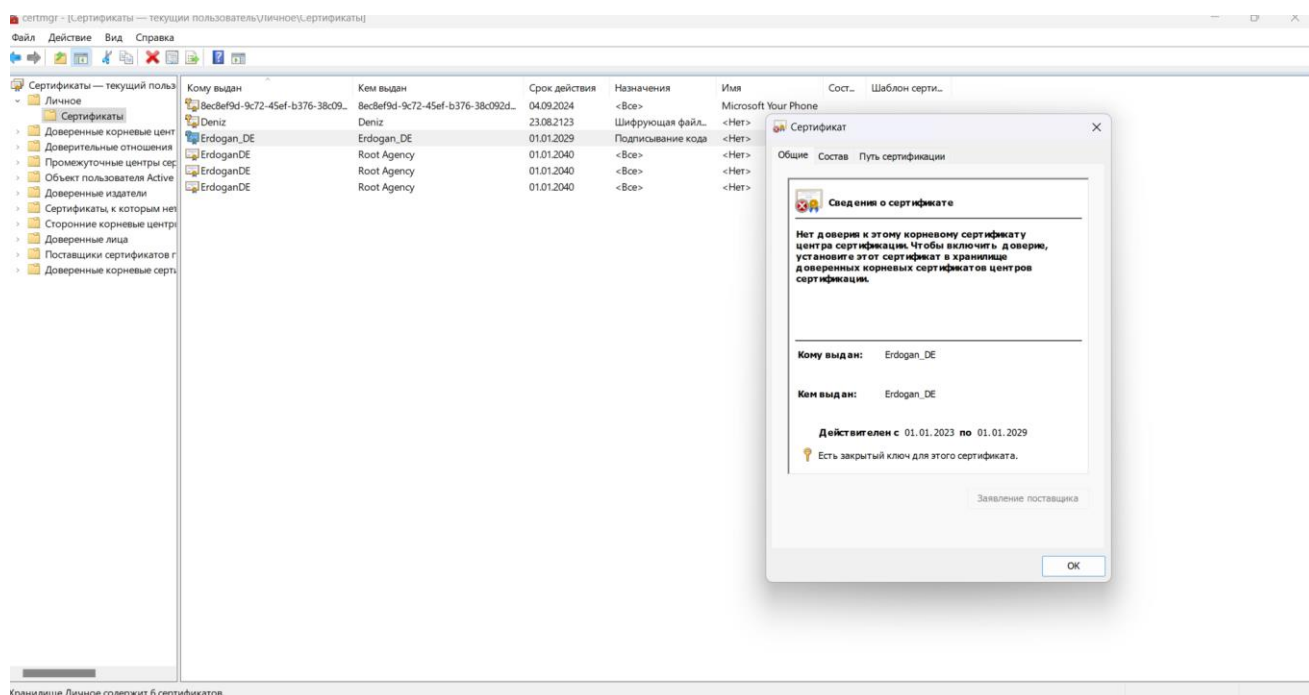


Рисунок № 19 – самоподписанный сертификат (Windows 11)

5.2. Включить в отчет о лабораторной работе:

5.2.1. Сведения о хранилище сертификатов, в которое помещается самоподписанный сертификат:

Самоподписанный сертификат помещается в логическое хранилище *Му* (Личные). В этом хранилище располагаются следующие типы сертификатов:

- Сертификаты, связанные с закрытыми ключами, к которым имеется доступ;
- Сертификаты, которые были выданы вам либо компьютеру или службе, для которых вы выполняете управление сертификатами.

5.2.2. Копии экранных форм, полученных при выполнении п. 5.1;

6. Ознакомиться с разделом 2 «Создание списка доверенных сертификатов» скопированного в п. 2 документа:

Ознакомимся со вторым разделом скопированного файла:

2. Создание списка доверенных сертификатов

Утилита MakeCTL предназначена для создания списков доверенных сертификатов (CTL). Созданный список кодируется и сохраняется в хранилище сертификатов или файле.

Входом утилиты MakeCTL является массив хранилищ сертификатов. Вычисляются хеш-значения всех сертификатов в этих хранилищах, которые и включаются в CTL.

Хранилища сертификатов могут быть заданы следующими способами:

- сохраненным ранее файлом хранилища;
- файлом в формате PKCS #7;
- файлом с закодированным сертификатом;
- именем системного хранилища.

Формат командной строки при вызове утилиты MakeCTL:

```
MakeCTL [/u subjectUsageID] [/s [/r registryLocation]]  
хранилище сертификатов 1 [/s [/r registryLocation]]  
хранилище сертификатов 2 ... [/s [/r registryLocation]]  
хранилище сертификатов N имя выходного файла.stl
```

Здесь *subjectUsageID* – идентификатор объекта для назначения создаваемого CTL (по умолчанию этот список состоит из сертификатов корневых удостоверяющих центров, предназначенных для подписания кода, что задается константой `szOID_TRUSTED_CODESIGNING_CA_LIST`, определенной в файле `Wintrust.h` как 1.3.6.1.4.1.311.2.2.1), *registryLocation* – указатель на размещение в реестре системного хранилища сертификатов (по умолчанию `currentUser`, т.е. используется раздел `HKEY_CURRENT_USER`, но возможно и задание `localMachine` для указания на раздел `HKEY_LOCAL_MACHINE`).

Опция `/s` указывает на то, что используется системное хранилище сертификатов. Может быть дополнительно указана опция `/?` для получения

...

Рисунок № 20 – второй раздел скопированного документа (Windows 11)

7. С помощью мастера списка доверия сертификатов, автоматически активизируемого при вызове утилиты командной строки MakeCTL без параметров, выполнить следующее:

7.1. Создать файл со списком доверенных сертификатов, созданных при выполнении п.п. 4-5 и предназначенных для подписывания кода:

Создадим файл со списком доверенных сертификатов, созданных при выполнении п.п. 4-5 и предназначенных для подписывания кода.

Для этого запустим утилиту *MakeCTL* от имени администратора:

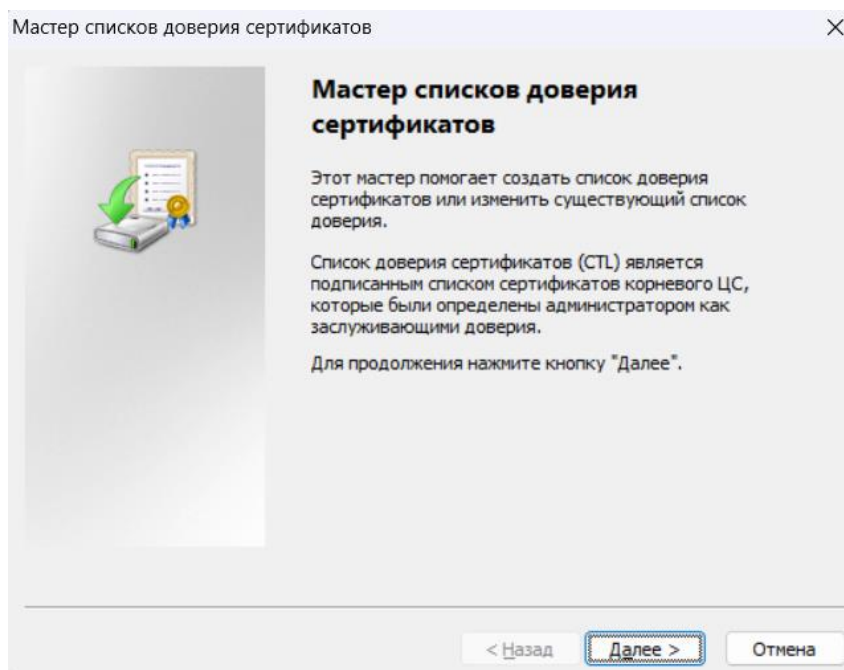


Рисунок № 21 – мастер списков доверия сертификатов (Windows 11)

В следующем окне выбираем назначение «Подписывание кода». Также по желанию можно указать префикс, идентифицирующий этот список, и задать срок действия:

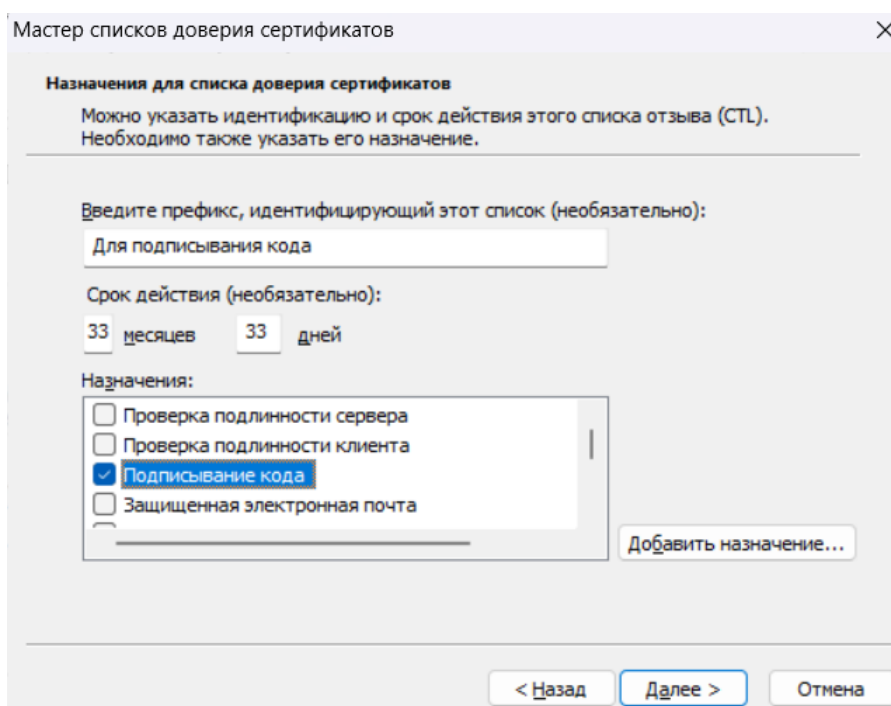


Рисунок № 22 – установка параметров сертификата (Windows 11)

Если мы не выберем назначение списка, то появится следующие сообщение:

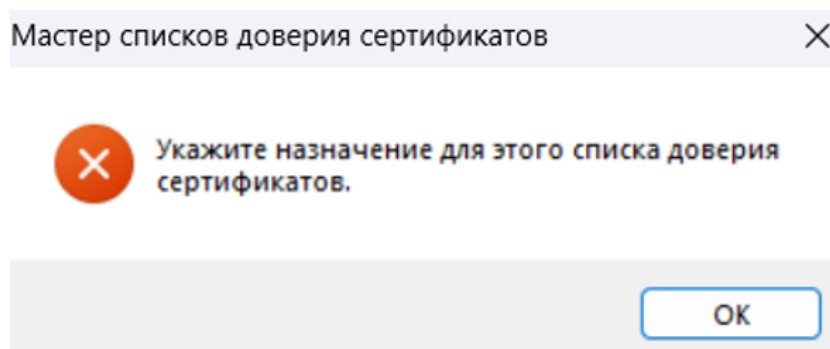


Рисунок № 23 – отсутствие назначение списка (Windows 11)

Выбираем созданные ранее самоподписанные сертификаты:

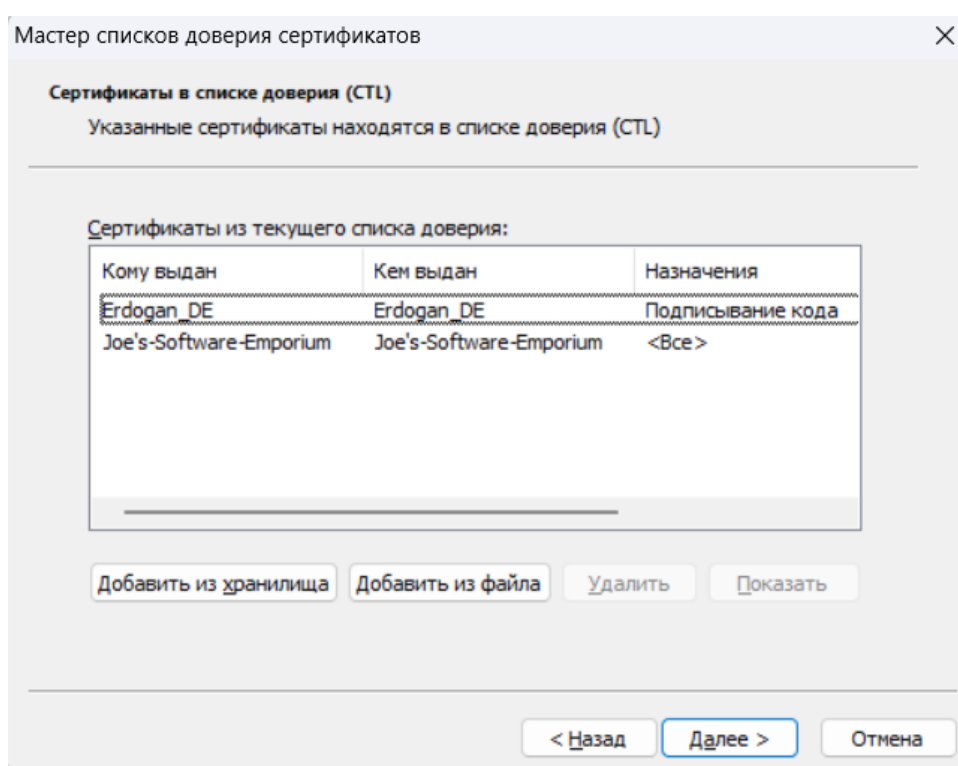


Рисунок № 24 – выбор сапомисных сертификатов (Windows 11)

Если мы не выберем сертификат, то появится следующие сообщение:

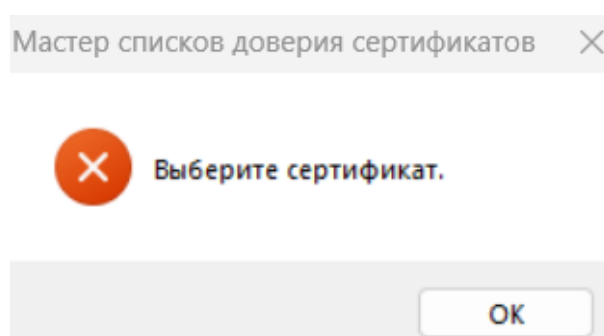


Рисунок № 25 – отсутствие выбора сертификата (Windows 11)

Выбираем хранилище, куда сохранится список доверенных сертификатов:

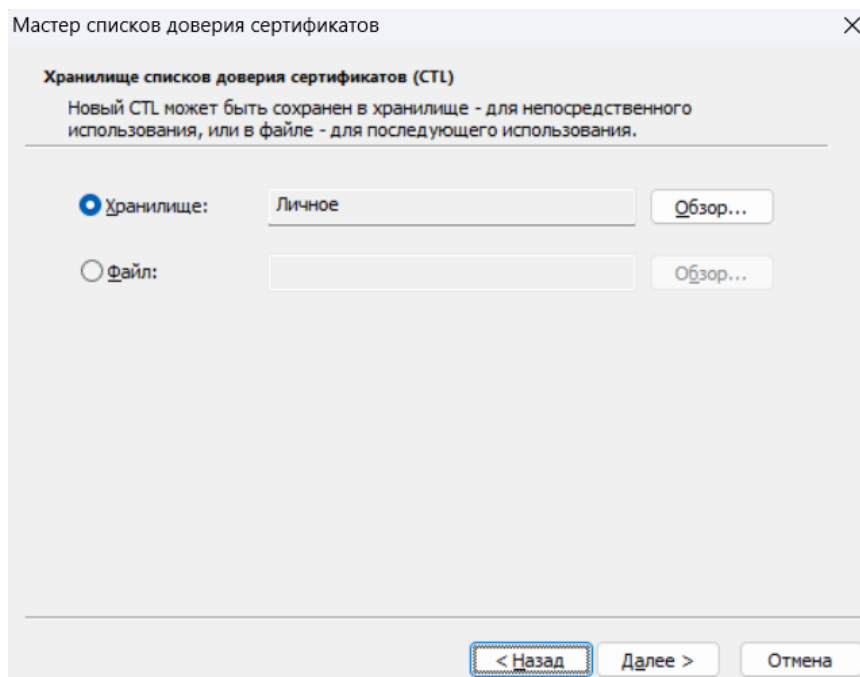


Рисунок № 26 – выбор хранилища для сертификатов (Windows 11)

Задаём имя и описание для нового списка:

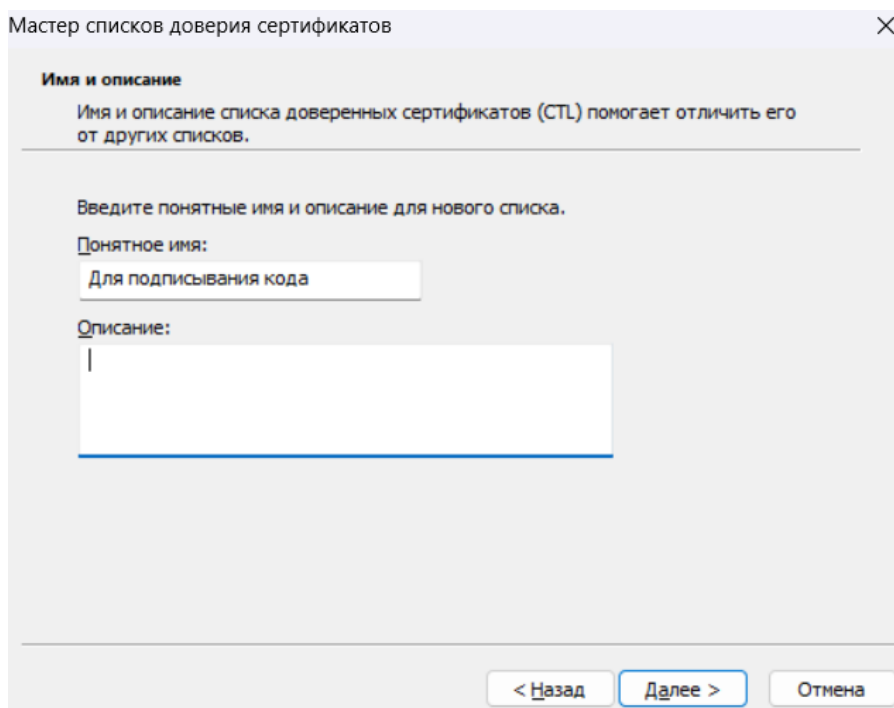


Рисунок № 27 – установка имени и описания (Windows 11)

Подтверждаем выбранные ранее настройки:

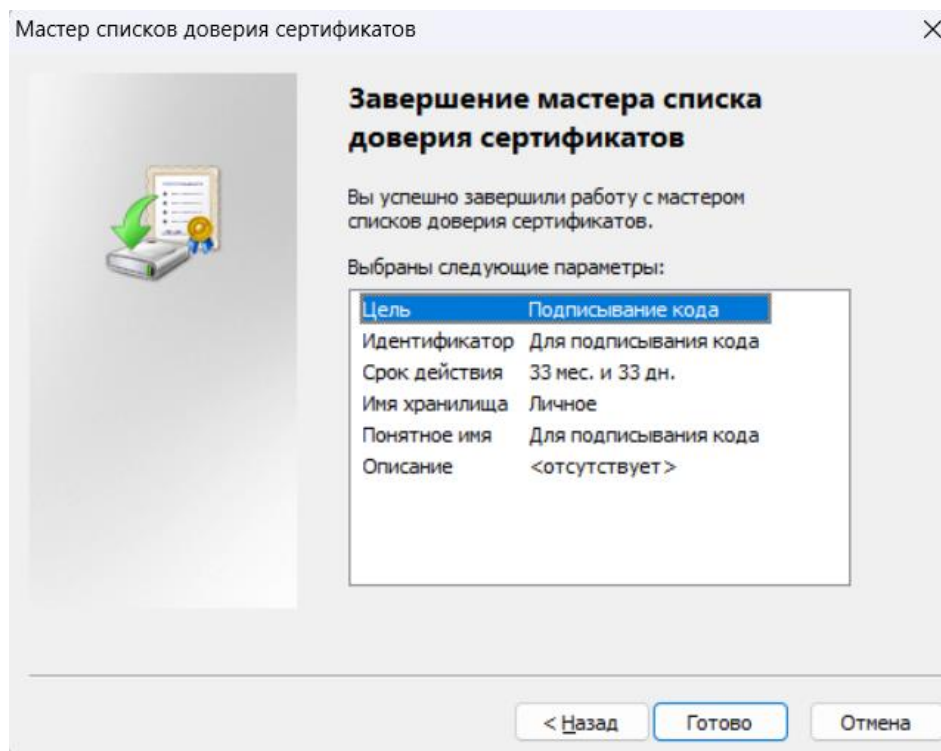


Рисунок № 28 – подтверждение настроек (Windows 11)

Получаем сообщение об успешном завершении:

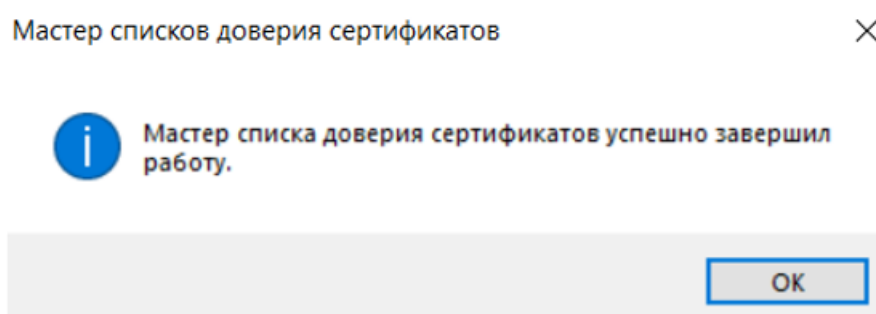


Рисунок № 29 – сообщение об завершении процесса (Windows 11)

Как видно, список доверия сертификатов успешно создан:

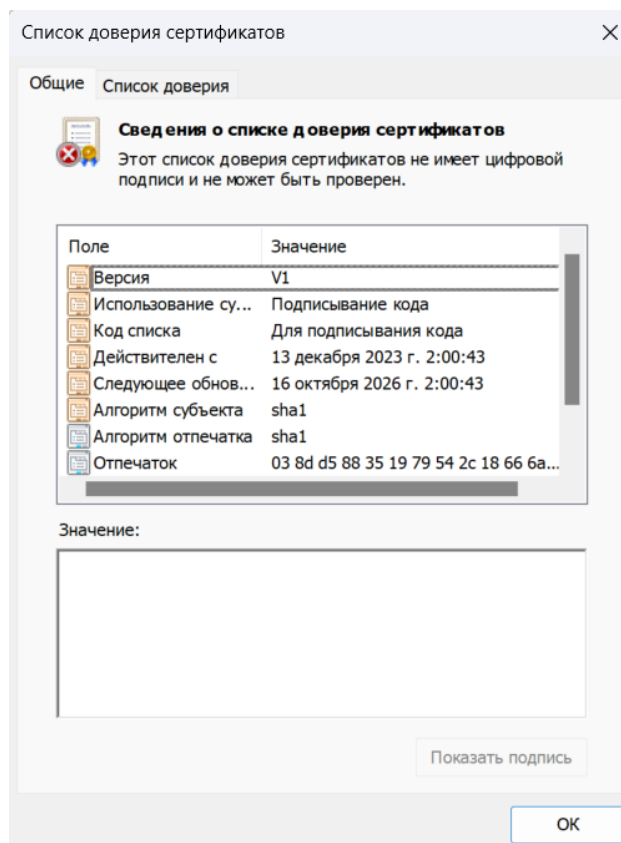


Рисунок № 30 – список доверенных сертификатов – подписывание кода (Windows 11)

Рассмотрим подробнее список сертификатов:

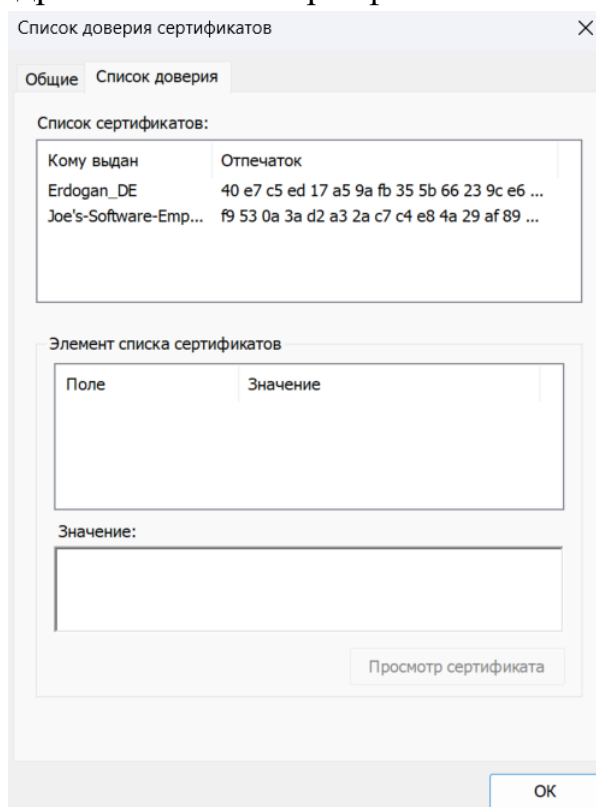


Рисунок № 31 – информация о списке доверенных сертификатов (Windows 11)

7.2. Создать другой файл со списком доверенных сертификатов, созданных при выполнении п.п. 4-5 и предназначенных для шифрования файлов:

Создадим другой файл со списком доверенных сертификатов, созданных при выполнении п.п. 4-5 и предназначенных для шифрования файлов. Для этого повторим проделанный ранее шаги в предыдущем пункте, но с небольшими поправками в установке назначения:

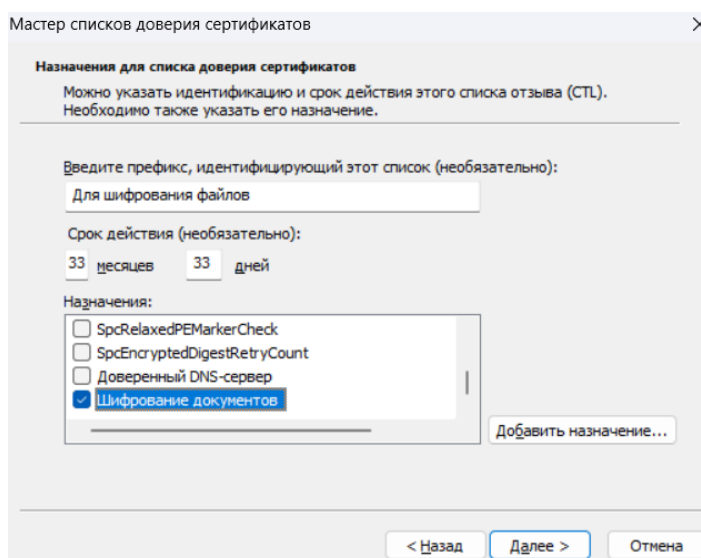


Рисунок № 32 – установление назначения списка сертификатов 0

Рассмотрим сам список сертификатов:

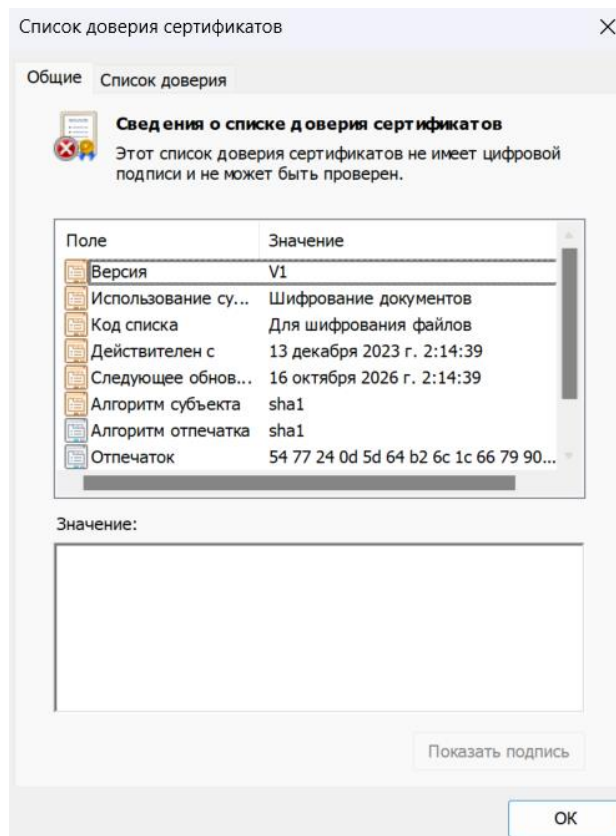


Рисунок № 33 – цифровая подпись для списка сертификатов - шифрование файлов (Windows 11)

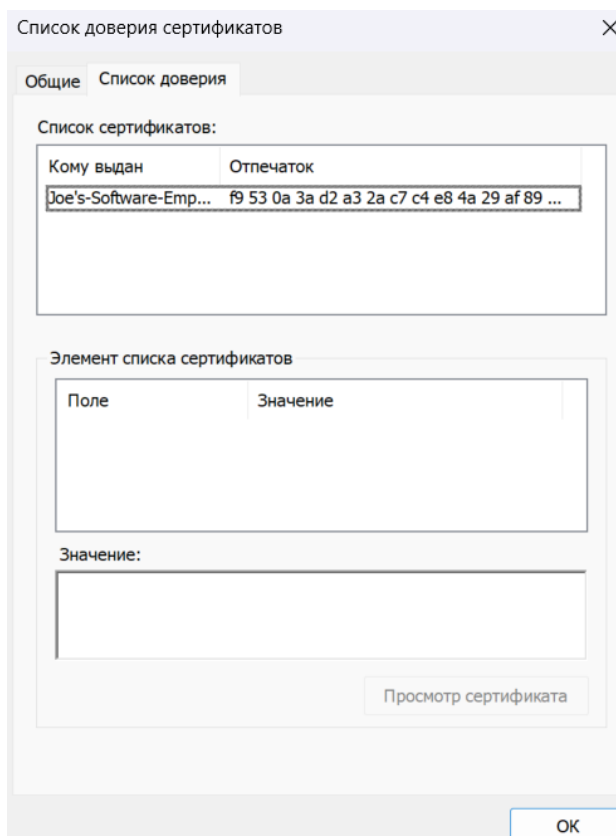


Рисунок № 34 – подробная информация о списке сертификатов (Windows 11)

7.3. Включить в отчет о лабораторной работе:

7.3.1. Сведения о назначении, способах получения и хранения списков отозванных сертификатов:

Списки отозванных сертификатов применяются для того, чтобы установить, был ли сертификат пользователя или удостоверяющего центра отозван в связи с компрометацией ключей. Важное свойство *СОО* — он содержит информацию только о сертификатах, срок действия которых не истёк.

Через утилиту *CertMgr* можно отображать информацию, копировать, удалять, экспортировать и импортировать сертификаты *CTL* и *CRL* из хранилища.

Если программа MakeCTL не установлена, то скопировать ее из папки с описаниями лабораторных работ;

7.3.2. Включить в электронную версию отчета копии экранных форм, полученных при выполнении п. 7;

8. Ознакомиться с разделом 3 «Вычисление и проверка электронной цифровой подписи» скопированного в п. 2 документа:

Ознакомимся с необходимой главой:

3. Вычисление и проверка электронной цифровой подписи

Утилита SignTool с командой sign предназначена для вычисления электронной цифровой подписи под файлом. Если файл уже содержит ЭЦП, то подпись будет вычислена заново. Формат командной строки при вызове утилиты SignTool с командой sign:

SignTool sign [опции] имя файла

Утилита SignTool с командой sign поддерживает три группы опций:

- опции, влияющие на выбор сертификата (табл. 3);
- опции, относящиеся к секретному ключу (табл. 4);
- опции, относящиеся к создаваемой ЭЦП (табл. 5);
- другие опции (табл. 6).

Табл. 3

Опция	Описание опции
/a	Выбирается лучший из подходящих сертификатов (иначе ожидается, что существует один подходящий сертификат)
/c имя	Имя шаблона сертификата
/f имя	Имя файла с сертификатом (для PFX-файла, защищенного паролем требуется опция /p, а если файл не содержит личный ключ, то могут использоваться опции /csp и /k)
/i имя	Имя или часть имени издателя сертификата подписи
/j имя	Имя файла с DLL, возвращающей массив атрибутов подписи
/jp параметр	Параметр (только один) для передачи в определенную предыдущей опцией DLL
/n имя	Имя или часть имени владельца сертификата подписи
/p строка	Пароль для PFX-файла с личным ключом
/г имя	Имя владельца корневого сертификата, удостоверяющего сертификат подписи
/s имя	Хранилище сертификатов, содержащее сертификат и секретный ключ создателя ЭЦП (по умолчанию My)
/sm	Для поиска сертификата подписи используется хранилище в разделе реестра HKEY_LOCAL_MACHINE (иначе в HKEY_CURRENT_USER)
/sha1 отпечаток	Хеш-значение сертификата создателя ЭЦП
/u OID или строка	Расширенное назначение ключа ЭЦП (по умолчанию “Code Signing” (1.3.6.1.5.5.7.3.3), т.е. подписание кода)
/uw	Назначение ключа ЭЦП – “Windows System Component Verification” (1.3.6.1.4.1.311.10.3.6), т.е. проверка компонент Windows)

Табл. 4

Рисунок № 35 – раздел для ознакомления (Windows 11)

SignTool — это средство командной строки, используемое для цифровой подписи пакета приложения или пакета приложений с помощью сертификата.

9. Включить в отчет о лабораторной работе:

9.1 Сведения о назначении и способах получения ЭЦП:

Электронная цифровая подпись – это закодированная информация о лице, как физическом, так и юридическом, которая необходима для его

идентификации при подаче документов в электронном виде. Также она позволяет защитить документ от редактирования сторонними лицами.

Её можно сделать с помощью *SignTool*.

9.2 Ответ на вопрос, какие возможности утилиты *SignTool sign* не поддерживаются мастером создания электронной цифровой подписи:

- Выбор лучшего из подходящих сертификатов;
- Выбор хранилища сертификатов;
- Установка назначения ключа ЭЦП;
- Поиск сертификата подписи в разделе реестра.

9.3 Ответ на вопрос, под файлами каких типов может быть вычислена ЭЦП с помощью утилиты *SignTool signwizard*:

- Файлы программ: *exe, dll, ocx*;
- САВ-файлы: *cab*;
- Файлы списков доверия сертификатов: *stl*;
- Файлы каталогов: *cat*.

10. Включить в отчет о лабораторной работе:

10.1 Сведения о способах проверки ЭЦП и получения ее параметров:

Для проверки истинности подписанного ЭЦП файла может использоваться утилита *SignTool* с командой *verify*. Синтаксис командной строки при вызове этой утилиты следующий:

SignTool verify [опции] имя подписанного файла

Для утилиты *SignTool* с командой *verify* можно указать до четырех опций:

- */q* (при успешном завершении не генерируется никаких сообщений, а при ошибке – минимальное количество сообщений);
- */v* (отображение полной информации об истинности подписанного файла);
- */r* имя (имя владельца корневого сертификата, удостоверяющего сертификат подписи);
- */tw* (генерируется предупреждение, если подписанный файл не имеет отметки времени).

Утилита *SignTool* с командой *verify* определяет тип проверяемого подписанного файла автоматически. Если подпись корректна, то в командной строке выводится соответствующее сообщение, содержащее имя подписанного файла и результат его проверки, например:

SignTool verify my.stl

Successfully verified: my.stl

Если проверка подписи завершилась неудачно, то выводится сообщение о причинах ошибки, например:

SignTool Error: A certificate chain processed, but terminated in a root certificate which is not trusted by the trust provider.

SignTool Error: File not valid: my.stl.

10.2 Ответ на вопрос, как происходит добавление издателя сертификата к списку доверенных сертификатов издателей и на что это оказывает влияние (при ответе на этот вопрос могут потребоваться сведения, полученные при выполнении п.п. 12 и 13):

Путём импорта файла сертификата издателя в хранилище сертификатов “Доверенные издатели” (с помощью мастера импорта сертификата).

Сертификатами, которые являются доверенными издателями нельзя подписывать макросы.

11. Ознакомиться с разделом 4 «Управление сертификатами» скопированного в п. 2 документа:

Ознакомимся с необходимым разделом:

4. Управление сертификатами

Утилита CertMgr обеспечивает поддержку управления сертификатами, списками доверенных сертификатов (CTL) и списками отозванных сертификатов (CRL). К основным функциям этой системной программы относятся:

- отображение информации из сертификатов, CTL и CRL;
- копирование сертификатов, CTL и CRL из одного хранилища сертификатов в другое;

- удаление сертификатов, CTL и CRL из хранилища;
- экспорт (сохранение) закодированных сертификатов, CTL и CRL из хранилища в файл;
- импорт (загрузка) закодированных сертификатов, CTL и CRL из файла в хранилище сертификатов.

Формат командной строки при вызове утилиты CertMgr следующий:

CertMgr [/add | /del | /put][*опции*] [/s [/r *раздел реестра*]]

[*входное имя*] [/s [/r *раздел реестра*]] [*выходное имя*]

В табл. 7 приведено описание флагов операций, выполняемых утилитой CertMgr.

Табл. 7

Флаг операции	Описание
не задан	Отображение сертификатов, CTL и CRL
/add	Копирование сертификатов, CTL и CRL в хранилище сертификатов
/del	Удаление сертификатов, CTL и CRL из хранилища
/put	Экспорт сертификатов, CTL и CRL из хранилища в файл

Если флаг операции не задан, то отображаются все сертификаты, CTL и CRL из файла с сохраненным хранилищем или самого хранилища сертификатов, чье имя задается в качестве входного имени (выходное имя в этом случае не используется).

Если задан флаг операции /add, то входное имя – это имя хранилища сертификатов, содержащее сертификаты, CTL и CRL, которые будут добавлены в хранилище, чье имя задано как выходное имя. В качестве выходного имени может быть задано имя файла с сохраненным хранилищем. Если задана опция /r, то хранилище сохраняется в файле формата PKCS #7 (опция /r не может применяться, если выходное имя указывает на системное хранилище сертификатов).

Если задан флаг операции /del, то входное имя определяет имя хранилища сертификатов, CTL и CRL, а выходное имя – имя хранилища, в которое будут помещены копии элементов входного хранилища, оставшихся в нем после удаления. Если выходное имя не задано, то модифицируется входное хранилище. В качестве выходного имени может быть указано имя файла с сохраненным хранилищем или (при задании опции /r) в формате PKCS #7. Опция /r не применяется, если выходное имя указывает на системное хранилище сертификатов.

Если задан флаг операции /put, то входное имя – это имя хранилища сертификатов, закодированные элементы которого записываются в файл в формате X.509, задаваемый выходным именем (при указании опции /r выходной файл имеет формат PKCS #7).

В табл. 8 приведено описание опций, которые могут быть указаны при вызове утилиты CertMgr.

...

Рисунок № 36 – раздел для ознакомления (Windows 11)

12. С помощью утилиты командной строки CertMgr выполнить следующее:

12.1. Добавить списки доверенных сертификатов, созданных при выполнении п. 7 и подписанных при выполнении п.9, в системное хранилище Trust:

Добавим списки доверенных сертификатов, созданных при выполнении п. 7 в системное хранилище *Trust*. Для этого в консоли введем команду «certmgr -add -all -s MyErdoganDE1.cer -s trust» (-add – копирует файлы; -all –

выбирает все файлы; -s – указывает хранилище, где хранятся файлы 1 и куда их перемещать 2; MyErdoganDE1.cer и *trust* – названия хранилищ).

```
Microsoft Windows [Version 10.0.22631.2792]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Deniz>certmgr -add -all -s MyErdoganDE1.cer -s trust

C:\Users\Deniz>
```

Рисунок № 37 – консоль с введенной командой (Windows 11)

Списки доверенных сертификатов переместились в системное хранилище *Trust* (Доверительные отношения в предприятии):

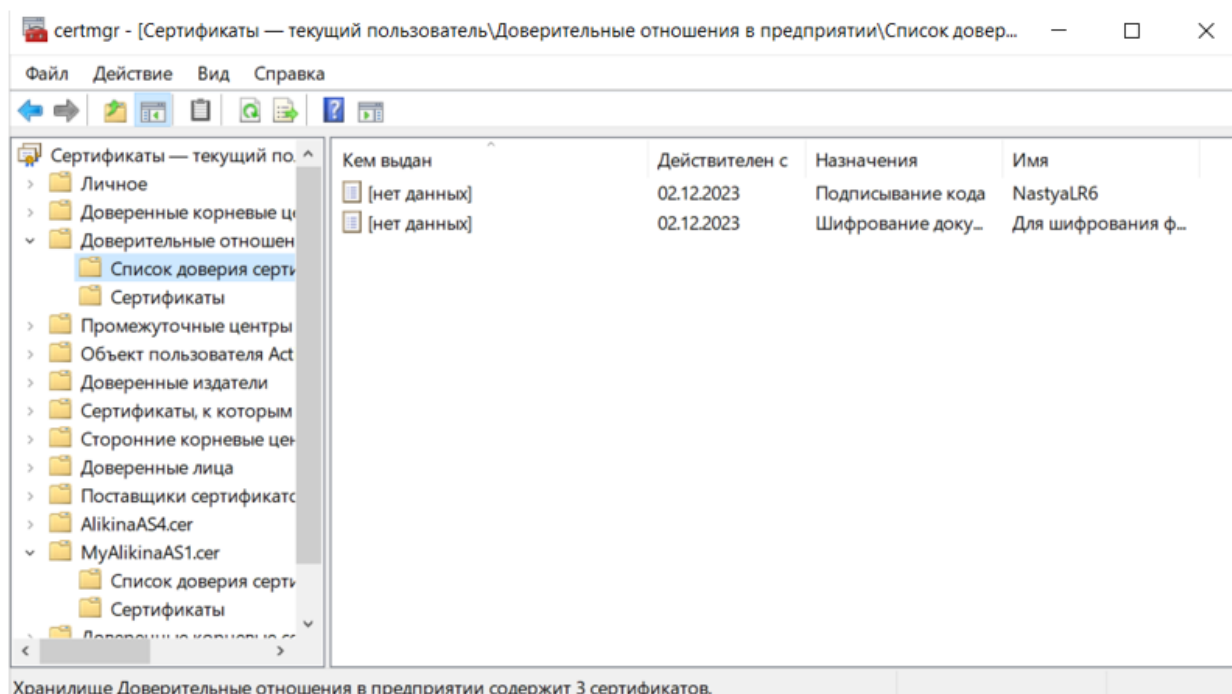


Рисунок № 38 – хранилище Trust (Windows 11)

12.2. Включить в отчет о выполнении лабораторной работы:

12.2.1. Сведения о назначении и основных функциях утилиты CertMgr:

Утилита *CertMgr* обеспечивает поддержку управления сертификатами, списками доверенных сертификатов (CTL) и списками отозванных сертификатов (CRL).

К основным функциям этой системной программы относятся:

- отображение информации из сертификатов, CTL и CRL;
- копирование сертификатов, CTL и CRL из одного хранилища;

- сертификатов в другое;
- удаление сертификатов, CTL и CRL из хранилища;
- экспорт (сохранение) закодированных сертификатов, CTL и CRL из хранилища в файл;
- импорт (загрузка) закодированных сертификатов, CTL и CRL из файла;
- в хранилище сертификатов.

Формат командной строки при вызове утилиты *CertMgr* следующий:

CertMgr [/add | /del | /put][опции] [/s [/r раздел реестра]] [входное имя] [/s [/r раздел реестра]] [выходное имя].

12.2.2. Протокол работы в режиме командной строки, полученный при выполнении п. 12.1-12.2 (с помощью системного меню окна командной строки и буфера обмена);

13. С помощью менеджера управления сертификатами, автоматически активизируемого при вызове утилиты *CertMgr* без параметров, выполнить следующее:

13.1. Освоить способы отбора сертификатов с требуемым назначением:

Запустим менеджер управления сертификатов через командную строку, вызвав утилиту *CertMgr* без параметров:

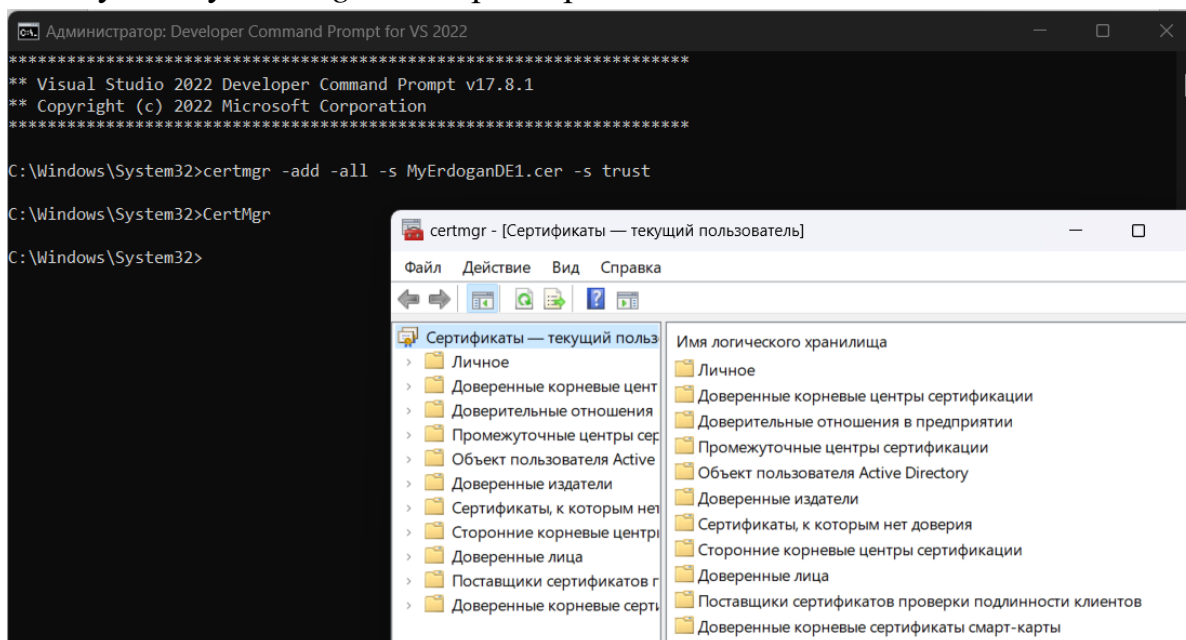


Рисунок № 39 – менеджер управления сертификатов (Windows 11)

Для отбора сертификатов по назначению, нужно выбрать необходимое назначение в выпадающем списке:

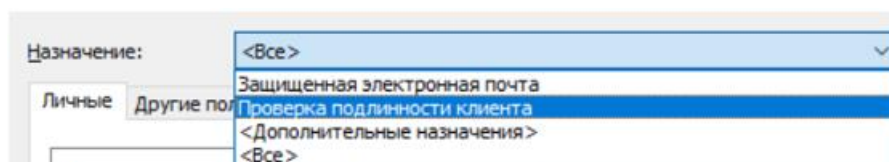


Рисунок № 40 – назначение сертификата (Windows 11)

13.2. Освоить способы просмотра характеристик сертификатов:

Для просмотра характеристик сертификата можно дважды кликнуть *ЛКМ* по нужному сертификату или выбрать сертификат и нажать “*Подробнее*”:

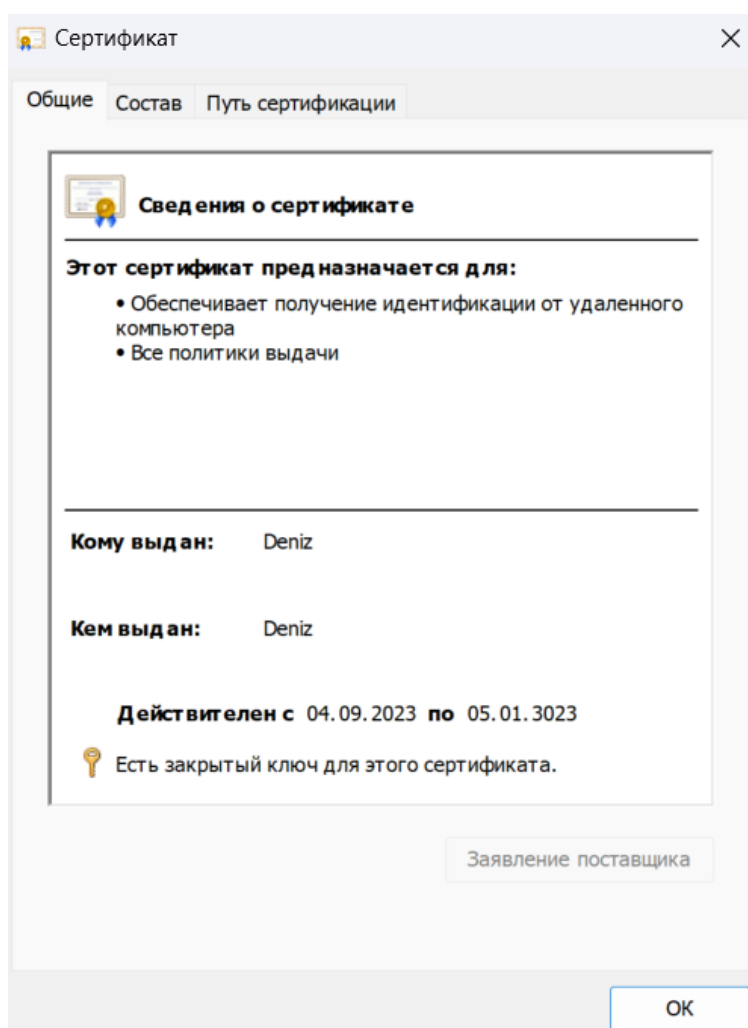


Рисунок № 41 – характеристики сертификата (Windows 11)

13.3. На примере файлов с сертификатами, созданными при выполнении п. 4, освоить работу с мастером импорта сертификатов: Запустим мастер импорта сертификатов, нажав кнопку “*Импорт*”:

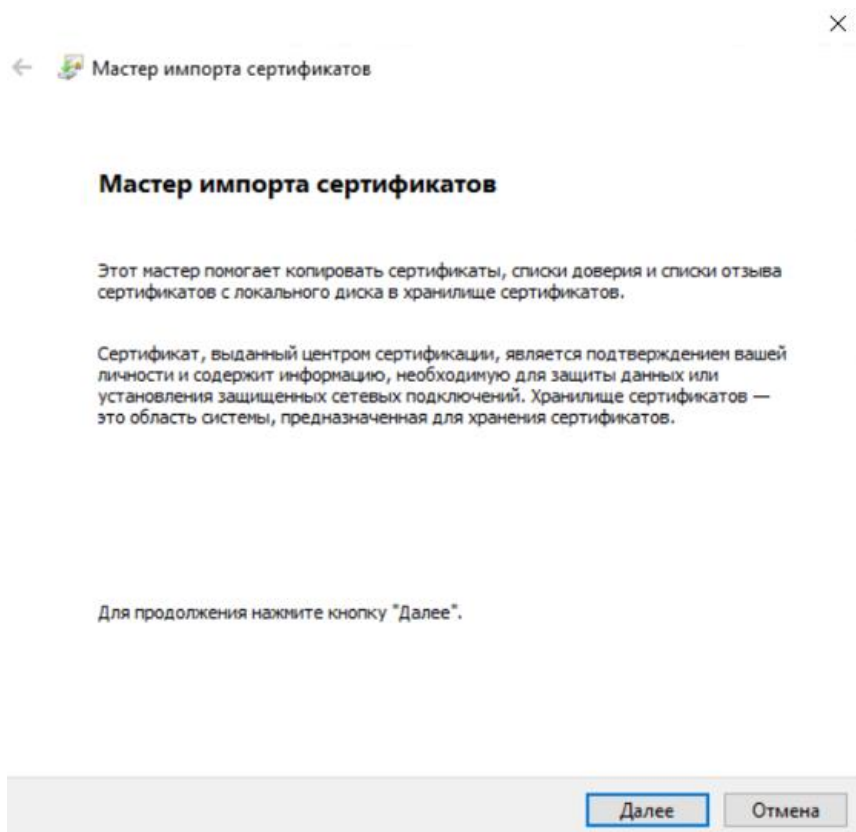


Рисунок № 42 – мастер импортов сертификатов (Windows 11)

Выберим сертификат для импорта:

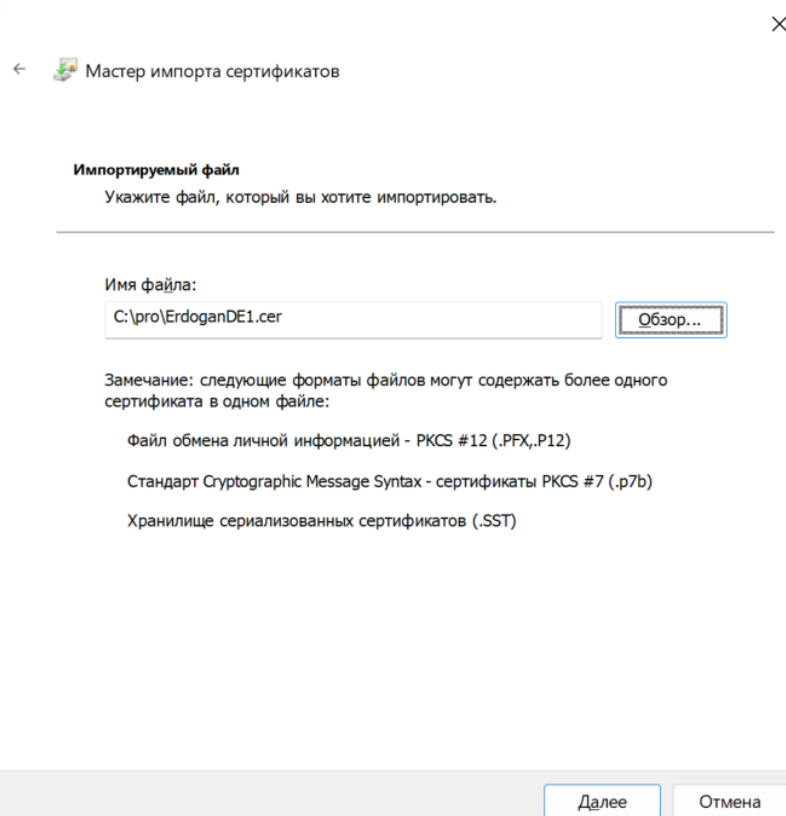


Рисунок № 43 – выбор сертификата для импорта (Windows 11)

Выберим место, куда будем импортировать сертификат:

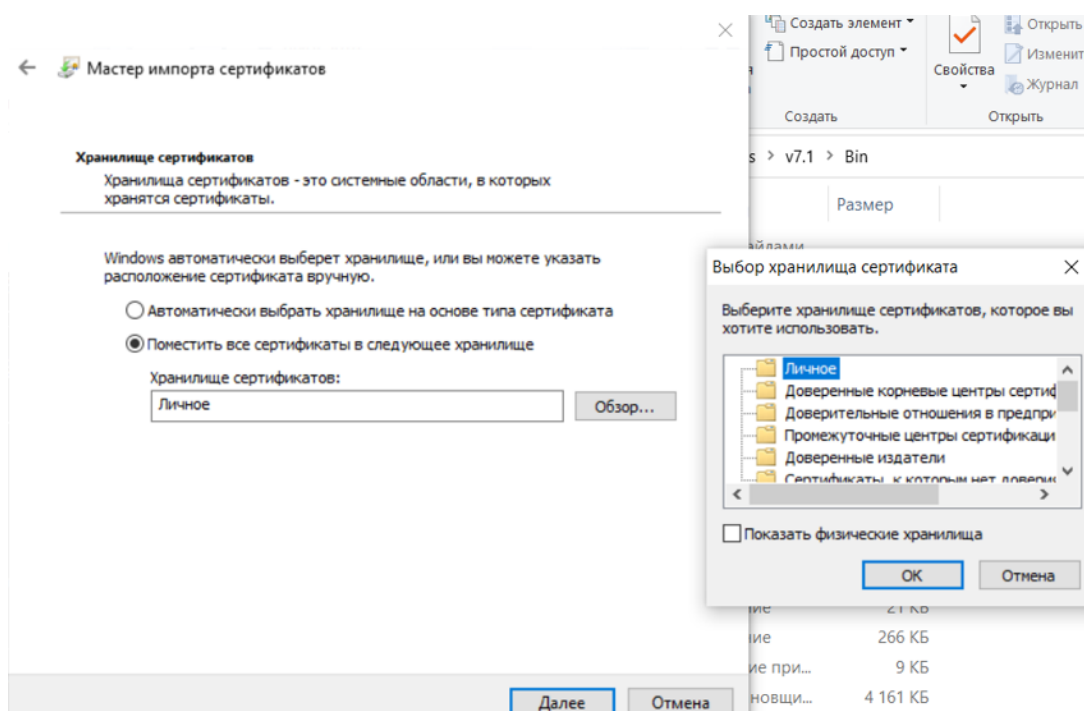


Рисунок № 44 – выбор хранилища (Windows 11)

Проверим данные по импорту сертификата:

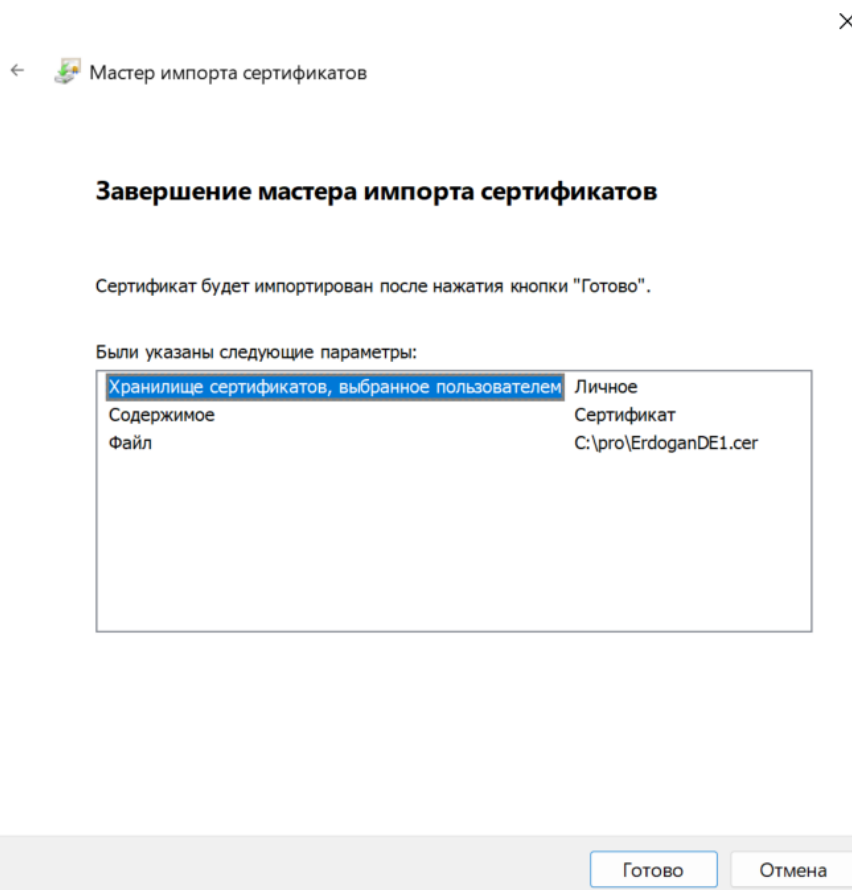


Рисунок № 45 – завершение работы с мастером импорта сертификатов (Windows 11)

После увидим сообщение об успешном импорте:

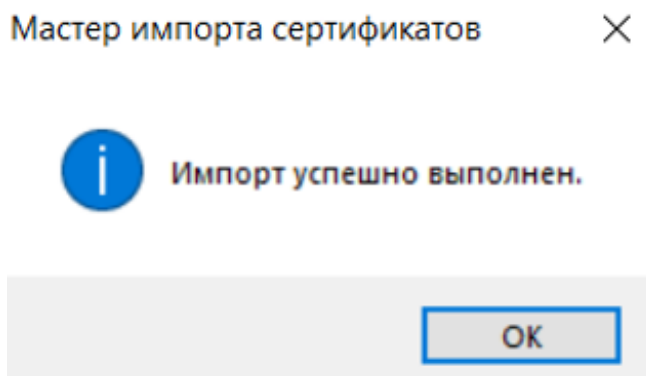


Рисунок № 46 – сообщение об успешном импорте (Windows 11)

Как видим сертификат успешно добавился в хранилище *My*:

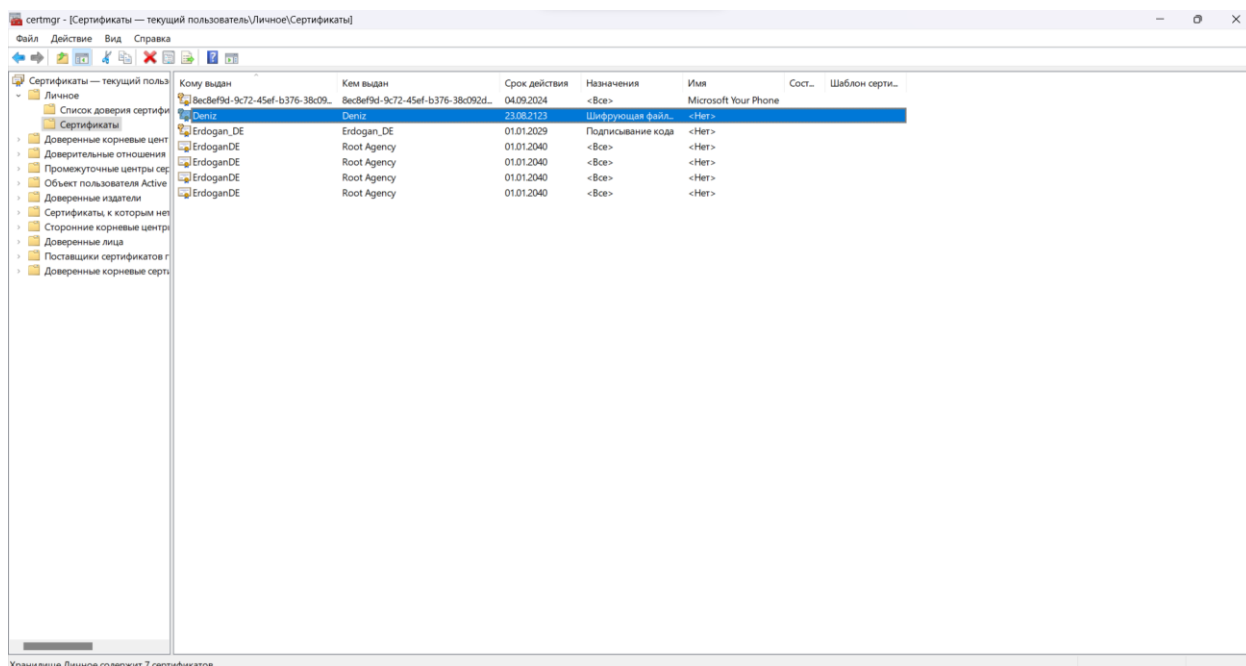


Рисунок № 47 – импортированный сертификат (Windows 11)

13.4. Освоить работу с мастером экспорта сертификатов:

Запустим мастер экспорта сертификатов, выбрав нужный сертификат и нажав кнопку “Экспорт”:



Мастер экспорта сертификатов

Этот мастер помогает копировать сертификаты, списки доверия и списки отзыва сертификатов из хранилища сертификатов на локальный диск.

Сертификат, выданный центром сертификации, является подтверждением вашей личности и содержит информацию, необходимую для защиты данных или установления защищенных сетевых подключений. Хранилище сертификатов — это область системы, предназначенная для хранения сертификатов.

Для продолжения нажмите кнопку "Далее".

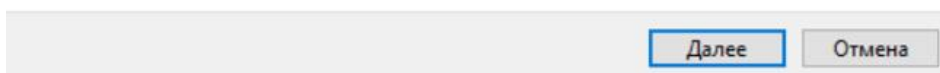


Рисунок № 48 – мастер экспорта сертификатов (Windows 11)

Выберем дополнительные пункты экспорта:

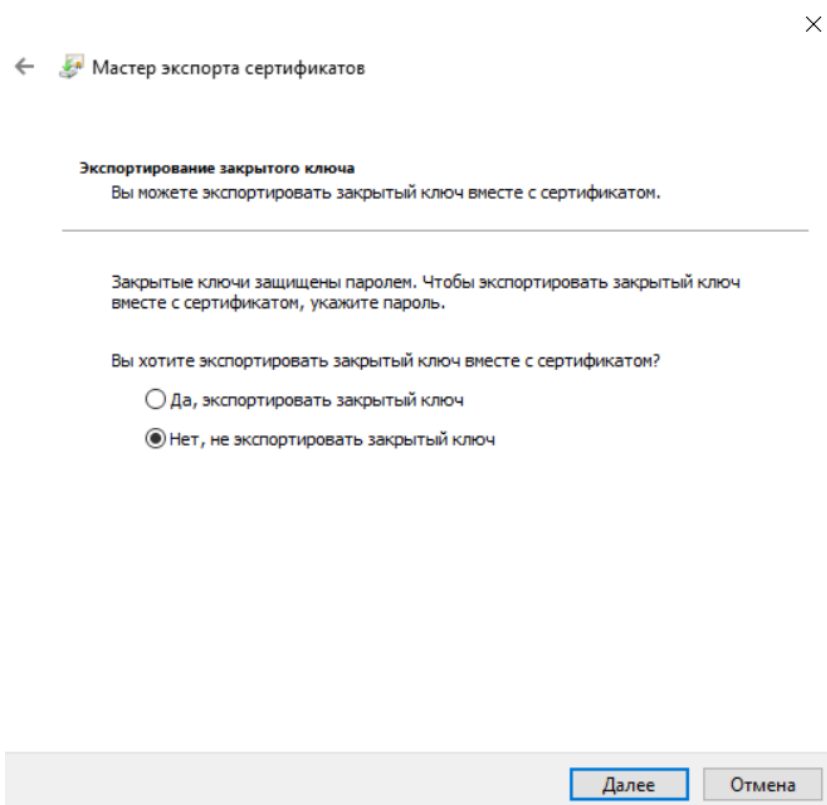


Рисунок № 49 – выбор экспортирования закрытого ключа (Windows 11)

Выберим формат экспортирования:

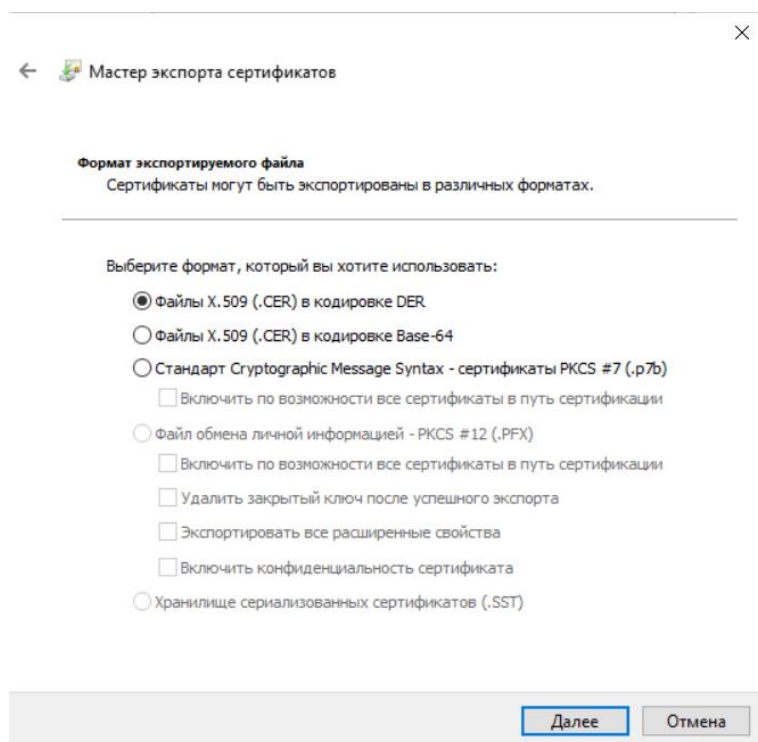


Рисунок № 50 – формат экспортируемого файла (Windows 11)

Введём имя экспортируемого файла:

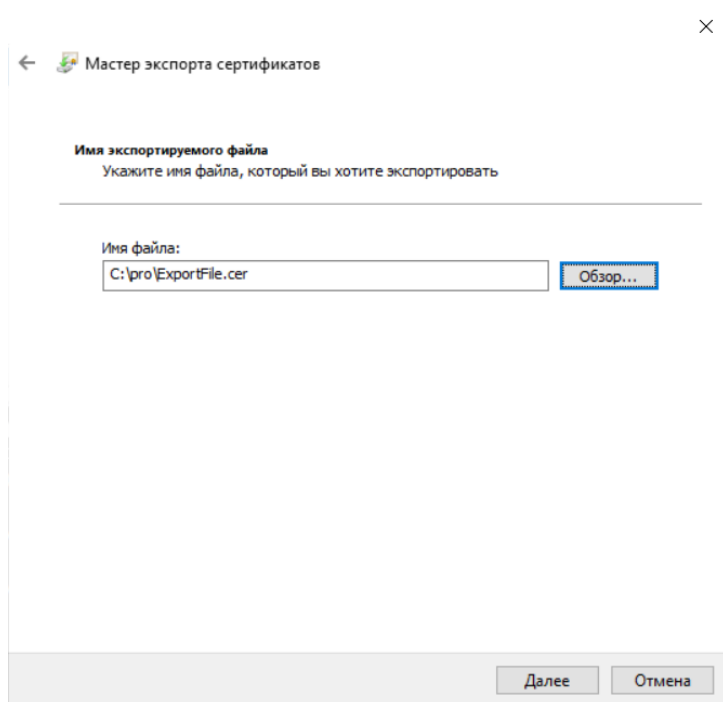


Рисунок № 51 – ввод имени файла экспорта (Windows 11)

После увидим отчёт по экспорту:

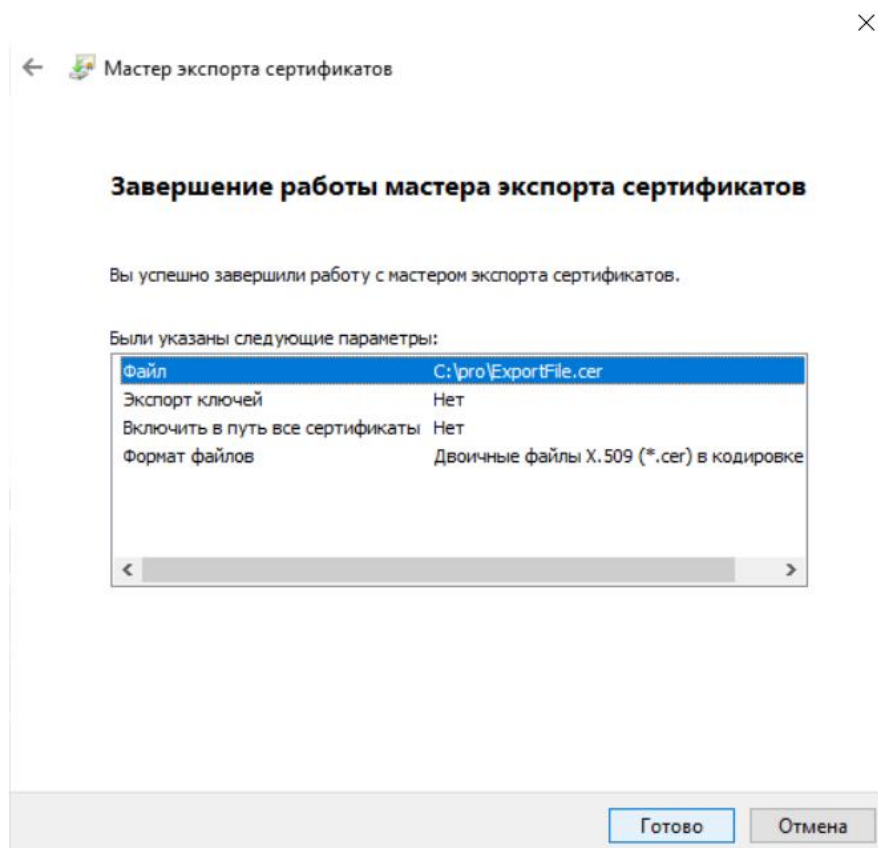


Рисунок № 52 – завершение работы мастера экспорта сертификата (Windows 11)

После мы увидим следующие сообщение:

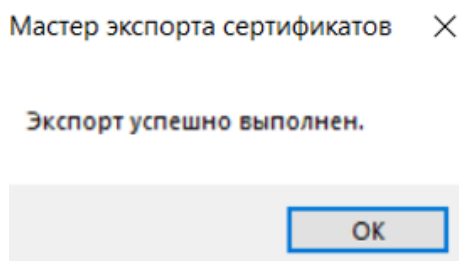


Рисунок № 53 – сообщение об успешном экспорте (Windows 11)

Сертификат с именем «ExportFile» успешно экспортировался в указанный каталог:

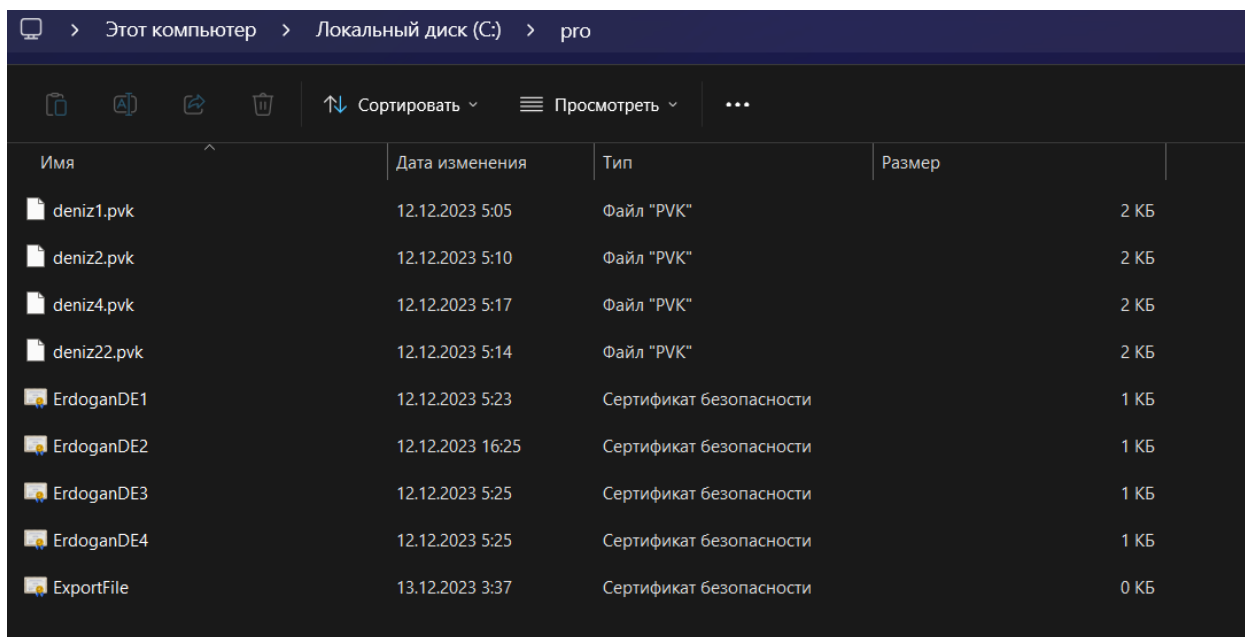


Рисунок № 54 – экспортированный файл (Windows 11)

13.5. Освоить процедуру удаления сертификата, не удаляя их окончательно:

Для удаления сертификата нужно выбрать сертификат и нажать кнопку “Удалить”. Далее появляется окно для подтверждения удаления:

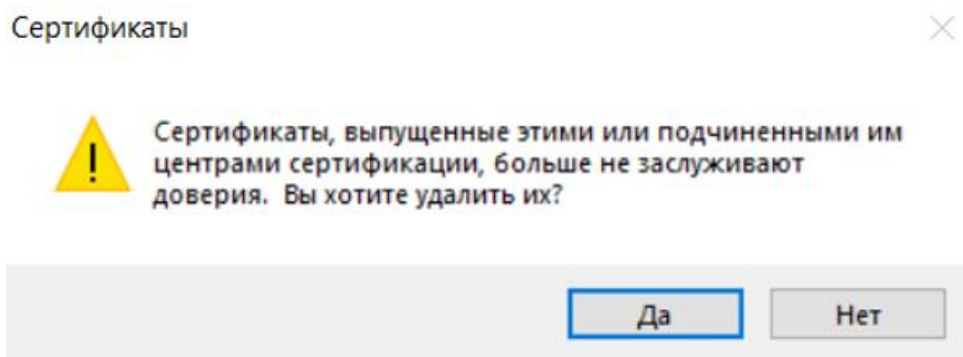


Рисунок № 55 – окно удаления сертификата (Windows 11)

13.6. Включить в отчет о лабораторной работе ответы на вопросы:

13.6.1. Все ли возможности утилиты CertMgr поддерживаются менеджером сертификатов:

Нет. Например, менеджер сертификатов не поддерживает работу со списками доверенных сертификатов и списками отозванных сертификатов;

13.6.2. Как еще может быть начат диалог с менеджером сертификатов:

Двойным нажатием ЛКМ по файлу *CertMgr.exe*.

13.6.3. Как может быть получена информация о составе списка доверенных издателей сертификатов:

Нужно перейти во вкладку “Доверенные издатели” менеджера сертификатов;

13.6.4. Копии экранных форм, полученных при выполнении;

14. Ознакомиться с разделом 5 «Получение сертификата в удостоверяющем центре» скопированного в п. 2 документа:

Ознакомимся с необходимым разделом:

5. Получение сертификата в удостоверяющем центре

В операционных системах Microsoft Windows 2000/XP/2003 для управления сертификатами может быть использована *оснастка* (snap-in) «Сертификаты». Эта системная программа может быть добавлена в *консоль управления Microsoft* (Microsoft Management Console, MMC), которая может быть вызвана с помощью команды Пуск | Выполнить | mmc. Для добавления оснастки «Сертификаты» требуется использовать команду меню Консоль | Добавить / удалить оснастку. В появившемся диалоговом окне нужно нажать кнопку «Добавить» и в списке доступных оснасток выбрать «Сертификаты».

С помощью оснастки «Сертификаты» можно просматривать содержимое хранилищ сертификатов, просматривать, импортировать и экспортировать сертификаты (аналогично менеджеру сертификатов, рассмотренному в предыдущем разделе). Эти операции доступны с помощью команд меню оснастки и с помощью контекстного меню соответствующих объектов (хранилищ и сертификатов). Дополнительно доступна команда «Поиск сертификата» в одном или нескольких хранилищах, позволяющая искать нужные сертификаты по именам издателя или владельца, серийному номеру, хеш-значению и другим критериям.

К другим дополнительным возможностям оснастки «Сертификаты» относятся просмотр списков отозванных сертификатов и запрос сертификата в удостоверяющем центре. Для просмотра списка отозванных сертификатов (CRL) нужно выделить узел «Список отзыва сертификатов» в левой части

Рисунок № 56 – раздел для ознакомления (Windows 11)

15. С помощью оснастки «Сертификаты» выполнить следующее:

15.1. Освоить использование основных функций, доступных с помощью этой оснастки (просмотр хранилищ сертификатов, запрос, просмотр, импорт, экспорт, удаление и поиск сертификатов, просмотр списков отозванных сертификатов):

Нажимаем Win + R и выполняем команду *certmgr.msc* для того, чтобы запустить оснастку *Сертификаты* или забить в командной строке поиска «Управление сертификатами пользователя»:

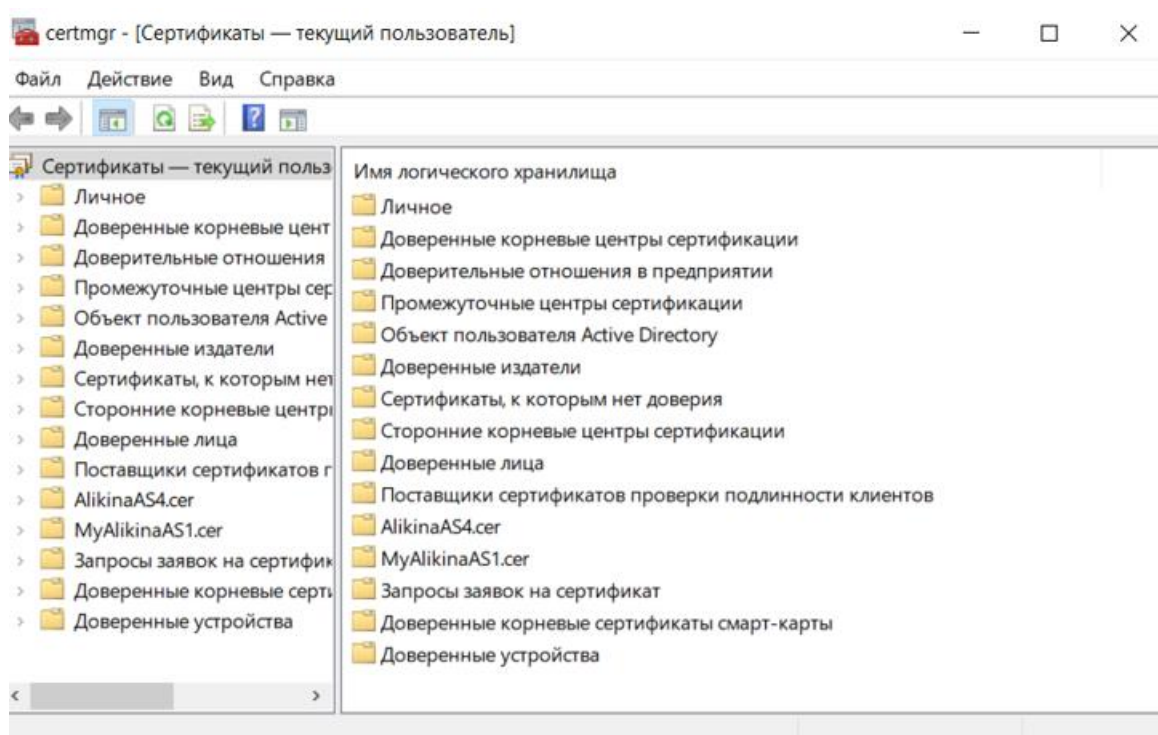


Рисунок № 57 – оснастка сертификаты (Windows 11)

Оснастка «*Сертификаты*» применяется пользователями и администраторами в качестве основного средства просмотра и управления сертификатами для пользователя, компьютера или службы. С помощью оснастки «*Сертификаты*» пользователи могут запрашивать, обновлять, находить, просматривать, перемещать, копировать и удалять сертификаты.

Оснастка «*Сертификаты*» является гибким средством просмотра и управления сертификатами для пользователя, компьютера или службы. С ее помощью можно определять, какие сертификаты хранятся на компьютере, где именно они хранятся, а также их параметры конфигурации.

15.2. Включить в отчет о лабораторной работе ответы на вопросы:

15.2.1. При каких условиях возможен запрос сертификата с помощью оснастки «Сертификаты»:

Сертификат должен находиться в хранилище сертификатов «*Личные*»;

15.2.2. Какие дополнительные возможности имеет оснастка «Сертификаты» по сравнению с менеджером сертификатов (п.п. 12-13):

- Выполнение запроса к личным сертификатам;
- Выполнение запроса на создание нового сертификата;
- Перемещение сертификатов из одного хранилища в другое;
- Экспортировать список сертификатов в текстовый документ;

- Обновление существующих сертификатов;
- Наличие раздела “Справка”.

15.2.3. Копии полученных при выполнении п. 15 экранных форм;

17. Включить в отчет о лабораторной работе ответы на контрольные вопросы:

Назовите основные свойства и типы устройств аутентификации:

Любая система аутентификации представляет собой совокупность элементов, выполняющих ту или иную роль в реализуемом ей сценарии.

К таким элементам относятся:

- Субъект аутентификации – лицо, проходящее процедуру аутентификации;
- Характеристика субъекта (фактор) – отличительная черта, характеризующая субъект;
- Владелец системы аутентификации – лицо, несущее ответственность и контролирующее работу системы;
- Механизм аутентификации – принцип, по которому осуществляется проверка подлинности предоставленного субъектом фактора;
- Механизм предоставления прав – механизм, обеспечивающий авторизацию, то есть предоставление тех или иных прав, приписанных данному субъекту, прошедшему проверку подлинности.

Типы систем аутентификации:

- Парольная аутентификация;
- Биометрическая аутентификация;
- Карточная аутентификация;
- Токенная аутентификация;
- Многофакторная аутентификация;
- Одноразовые пароли.

Как происходила непрямая аутентификация в ОС Windows NT:

В операционных системах *Windows NT* для не прямой аутентификации используется система доменных имен (DNS). При входе в систему пользователь вводит свое имя и пароль. Затем операционная система отправляет запрос к *DNS*-серверу с указанием имени пользователя. *DNS*-сервер проверяет наличие записи в своей базе данных для указанного имени

пользователя и возвращает операционной системе ответ, подтверждающий или опровергающий подлинность введенного имени.

Если ответ положительный, операционная система выполняет аутентификацию пользователя и предоставляет ему доступ к системе.

Из каких шагов состоит стандартный вариант протокола SSL:

SSL (Secure Socket Layer) - это протокол безопасности, который обеспечивает шифрование данных между сервером и клиентом. Стандартный вариант протокола SSL включает в себя следующие шаги:

Шаг 1: Установка соединения. Клиент и сервер устанавливают соединение и обмениваются информацией о протоколе, например, версиями SSL;

Шаг 2: Обмен сертификатами. Сервер отправляет свой сертификат клиенту, а клиент проверяет его на достоверность;

Шаг 3: Аутентификация. Клиент использует сертификат сервера для подтверждения его подлинности;

Шаг 4: Шифрование. Данные, передаваемые между клиентом и сервером, шифруются с использованием ключа, полученного из сертификата сервера;

Шаг 5: Передача данных. Зашифрованные данные передаются по сети;

Шаг 6: Расшифровка. На стороне получателя данные расшифровываются с использованием ключа из сертификата сервера;

Шаг 7: Завершение соединения. Соединение закрывается, и клиент и сервер подтверждают, что данные были переданы без ошибок.