

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

ЗНАКОМСТВО С КРИПТОСИСТЕМАМИ

Отчет по лабораторной работе №4

По дисциплине

«Информационная безопасность»

Студент гр. 431-3

_____ Е.П. Бекиш
(подпись)

(дата)

Руководитель:

Ассистент кафедры АСУ

_____ Я.В. Яблонский
(подпись)

Томск 2024

Оглавление

1	Цель работы	3
2	Задание на лабораторную работу	4
3	Алгоритм действий	5
3.1	Создание самоподписанного сертификата	6
3.2	Распространение открытого ключа	12
3.2.1	Экспортирование сертификата	12
3.2.2	Импортирование сертификата	15
3.3	Создание электронной подписи и шифрование сообщения	17
3.3.1	Создание электронной подписи.....	18
3.3.2	Шифрование сообщения	22
3.4	Расшифрование и проверка подписи	26
4	Полученные в результате работы файлы.....	29
5	Вывод.....	31

1 Цель работы

Освоить процесс создания ключей, распространения открытых и сохранения в тайне закрытых ключей, а также шифрования, расшифрования, создания и подтверждения электронных подписей в криптосистемах.

2 Задание на лабораторную работу

Задание по варианту №1: установите программу КриптоАРМ и последовательно выполните следующие шаги:

- Изучите «Руководство пользователю»;
- Ознакомьтесь с работой программы. Внимательно изучите процессы создания ключей, распространения открытых и сохранения в тайне закрытых ключей, схему разделения и сборки ключей.
- Создайте свою собственную пару ключей (открытый и закрытый).
- Распространите свой открытый ключ.
- Получивший открытый ключ пользователь должен проделать следующие действия: подготовить документ (файл), который необходимо переслать другому пользователю, подписать файл цифровой подписью, зашифровать подписанный файл с помощью открытого ключа другого пользователя, передать зашифрованный файл пользователю, чей ключ использовался при шифровании, получатель должен расшифровать файл и проверить достоверность ЭЦП;

3 Алгоритм действий

После изучения «Руководства пользователю» первым делом была проверена информация «О программе». Как видно из рисунка 3.1, версия скачанной криптосистемы является «КриптоАРМ Стандарт Плюс 5». Данная версия является расширенной, в отличие от стандартной версии, так как поддерживает работу с токенами и старт-картами с криптографией «на борту» и не извлекаемыми ключами. В то время как стандартная версия включает в себя функции подписи и шифрования электронных данных, а также поддерживает российские ГОСТ алгоритмы подписи и шифрования. Функция проверки корректности электронной подписи при работе с криптопровайдером «КриптоПро CSP» включена во все версии КриптоАРМ по-умолчанию.

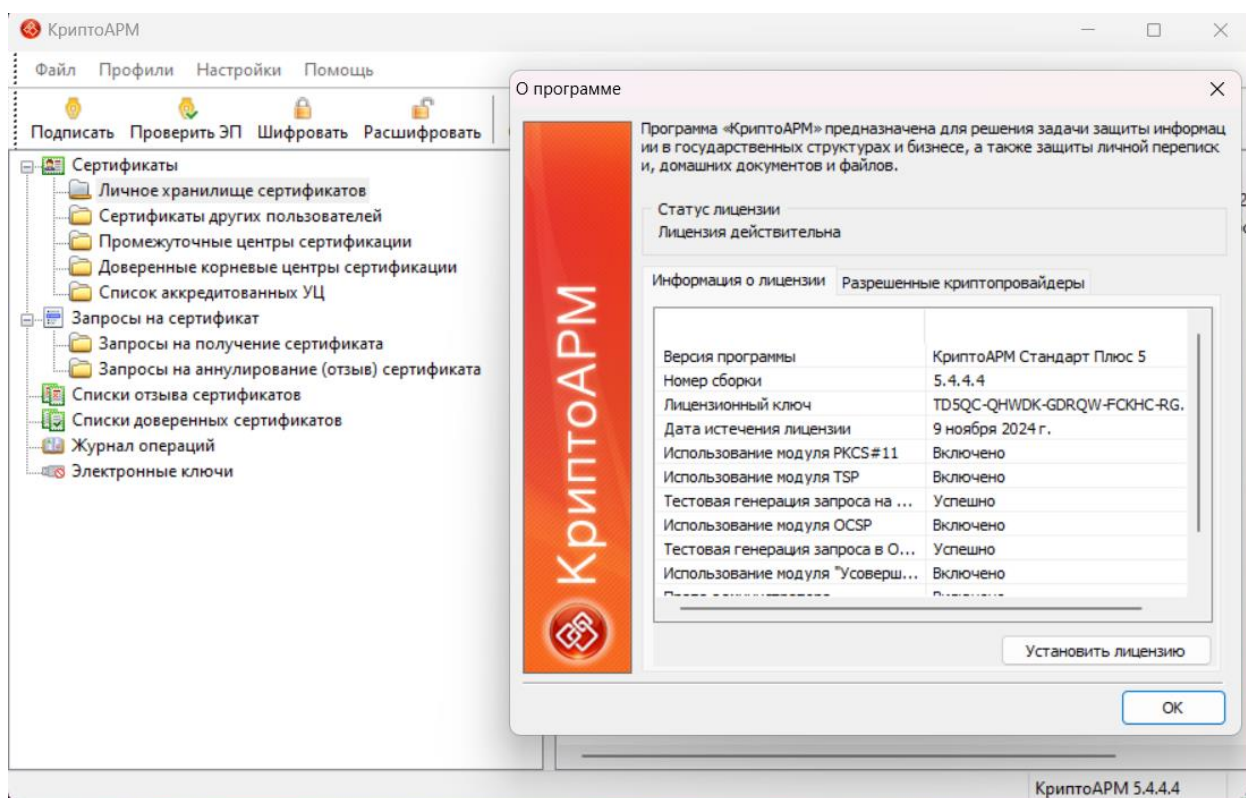


Рисунок 3.1 — Общие сведения о программе

3.1 Создание самоподписанного сертификата

Для того, чтобы создать открытый и закрытый ключи, в криптосистемах используют понятие сертификата.

Самоподписанный сертификат – сертификат, изданный самим пользователем, без обращения к доверенной стороне (Удостоверяющему центру). Самоподписанный сертификат является одновременно личным и корневым (устанавливается в Личное хранилище сертификатов и «Доверенные корневые центры сертификации»). Самоподписанные сертификаты используются для обмена зашифрованными или подписанными документами между людьми, доверяющими друг другу, например, друзьями, коллегами. Обменявшись такими сертификатами между собой, они могут пересылать друг другу подписанные и зашифрованные электронные данные, не беспокоясь при этом, что информация может быть перехвачена, искажена и использована против их интересов.

Таким образом, для создания самоподписанного сертификата были выполнены следующие действия:

- В дереве элементов главного окна был выбран раздел Сертификаты. Далее правой клавишей мыши было вызвано контекстное меню и выбран пункт Создать > Самоподписанный сертификат, как это показано на рисунке 3.2. Далее открывается Мастер создания самоподписанного сертификата.
- На первом шаге мастера создания самоподписанного сертификата требуется ознакомиться с порядком и требованиями создания самоподписанного сертификата.
- На следующем шаге из выпадающего списка был выбран шаблон «Сертификат КЭП физического лица».
- Далее требовалось указать идентификационную информацию о владельце будущего сертификата в соответствии с шаблоном, выбранным на предыдущем шаге, как это показано на рисунке 3.3. При этом стоит обратить внимание на следующие правила: поля, отмеченные знаком «*» являются

обязательными для заполнения, ИНН указывается без пробелов и знаков «-».

- После указания данных в разделе «Основная информация» открывается раздел «Параметры ключа», заполнение которого указано на рисунке 3.4. В процессе создания сертификата «с нуля» был отмечен выбор создания нового ключевого набора, что подразумевает генерацию новой ключевой пары и создания сертификата на его основе. Пометка экспортируемости ключей в данном случае является необязательной, так как возможность экспорта сертификата по умолчанию возможна с открытым ключом, тогда как активация функции позволит экспортировать сертификат ещё и с закрытым ключом, обеспечивая тем самым архивацию сертификата. Таким образом открытый ключ выглядит следующим образом: 48fa6690-a709-4b11-858e-a0fbf03fb027.

- После прохождения вышеупомянутого раздела требовалось подтвердить несколько действий: запрос системы на установку пароля на носитель и подтверждение его, а также запрос системы на установление самоподписанного сертификата в хранилище Доверенных корневых центров сертификации.

Общие сведения о сертификате представлены на рисунке 3.5.

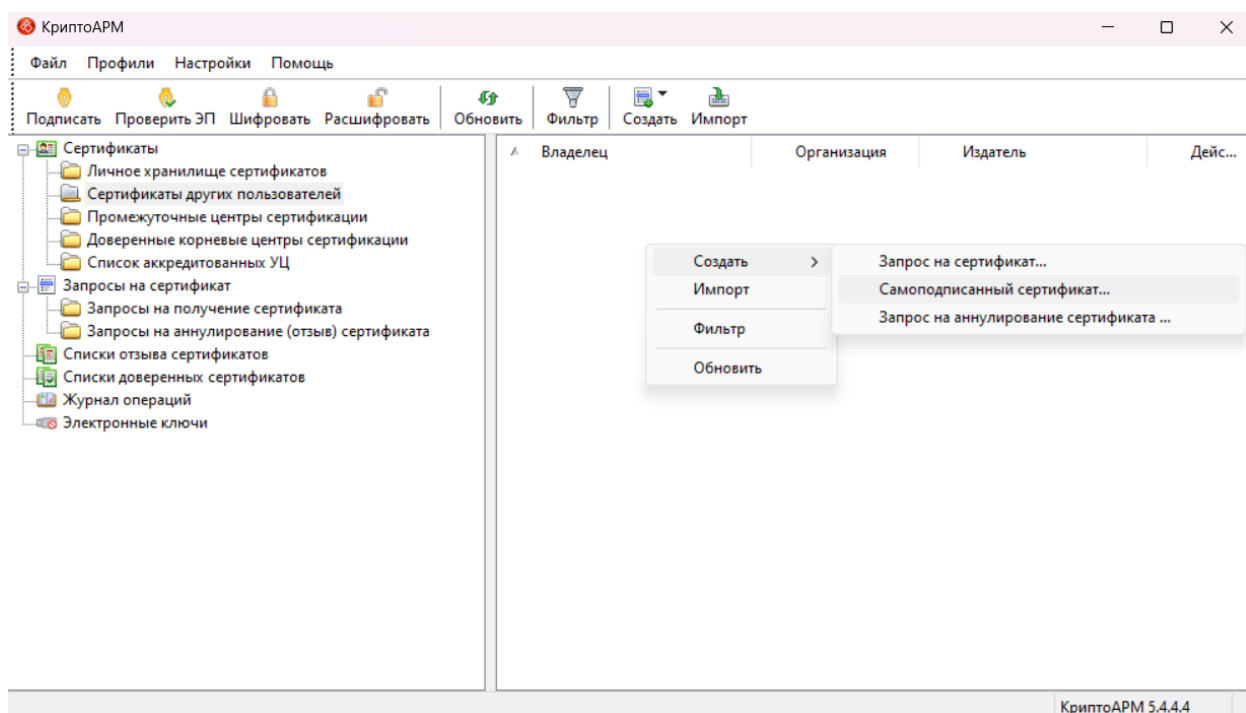


Рисунок 3.2 — Вызов контекстного меню для создания самоподписанного сертификата

КриптоАРМ :: Создание запроса

Основная информация

Указанные на этом шаге параметры будут храниться в поле "Subject" созданного сертификата

Идентификационная информация

Идентификатор (CN)*:	Егор Бекиш
Организация:	ТУСУР
Город:	Томск
Область:	Томская область
Страна:	Российская Федерация (RU) ▾
E-mail:	egorbeckish@mail.ru
ИНН:	244603142927

< Назад Далее > Отмена

Рисунок 3.3 — Внесение идентификационной информации о владельце

КриптоАРМ :: Создание запроса

Параметры ключа

На этом шаге вам следует указать параметры ключа, связанного с сертификатом

Используемый криптопровайдер:
Microsoft Base Cryptographic Provider v1.0

☒ Создать новый ключевой набор
☐ Использовать существующий ключевой набор

Имя ключевого набора:
48fa6690-a709-4b11-858e-a0fbf03fb027 Выбрать...

Назначение ключа

☐ Создание ЭП Длина ключа: 1024
☐ Шифрование
☒ Шифрование и создание ЭП Дополнительно...

☒ Пометить ключи как экспортируемые

< Назад Далее > Отмена

Рисунок 3.4 — Указание параметров ключа

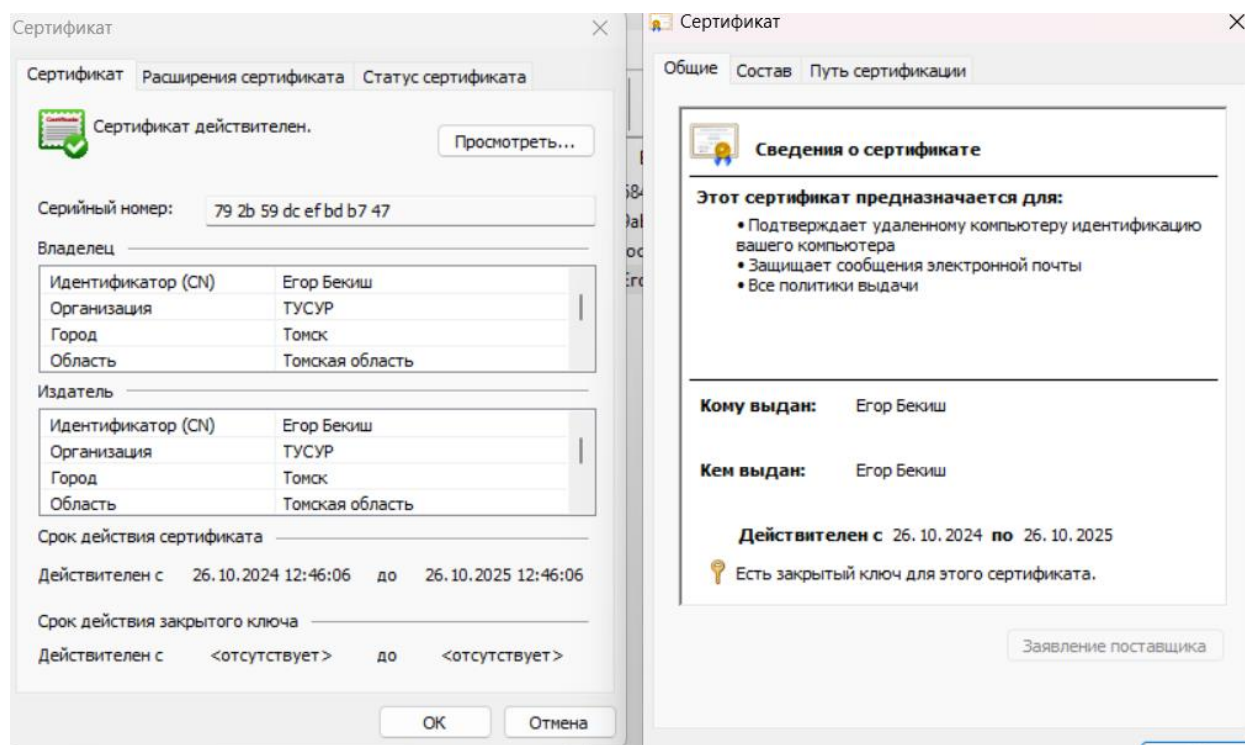


Рисунок 3.5 — Общие сведения о сертификате

3.2 Распространение открытого ключа

3.2.1 Экспортирование сертификата

Для того, чтобы смоделировать ситуацию передачи зашифрованного текста с невозможностью расшифровать данные посторонними лицами, необходимо экспортировать сертификат без закрытого ключа. Экспорт сертификатов в криптосистеме КриптоАРМ осуществляется «Мастером экспорта сертификатов», как это показано на рисунке 3.6. При этом для наших целей необходимо указать пункт «Нет, не экспортировать закрытый ключ».

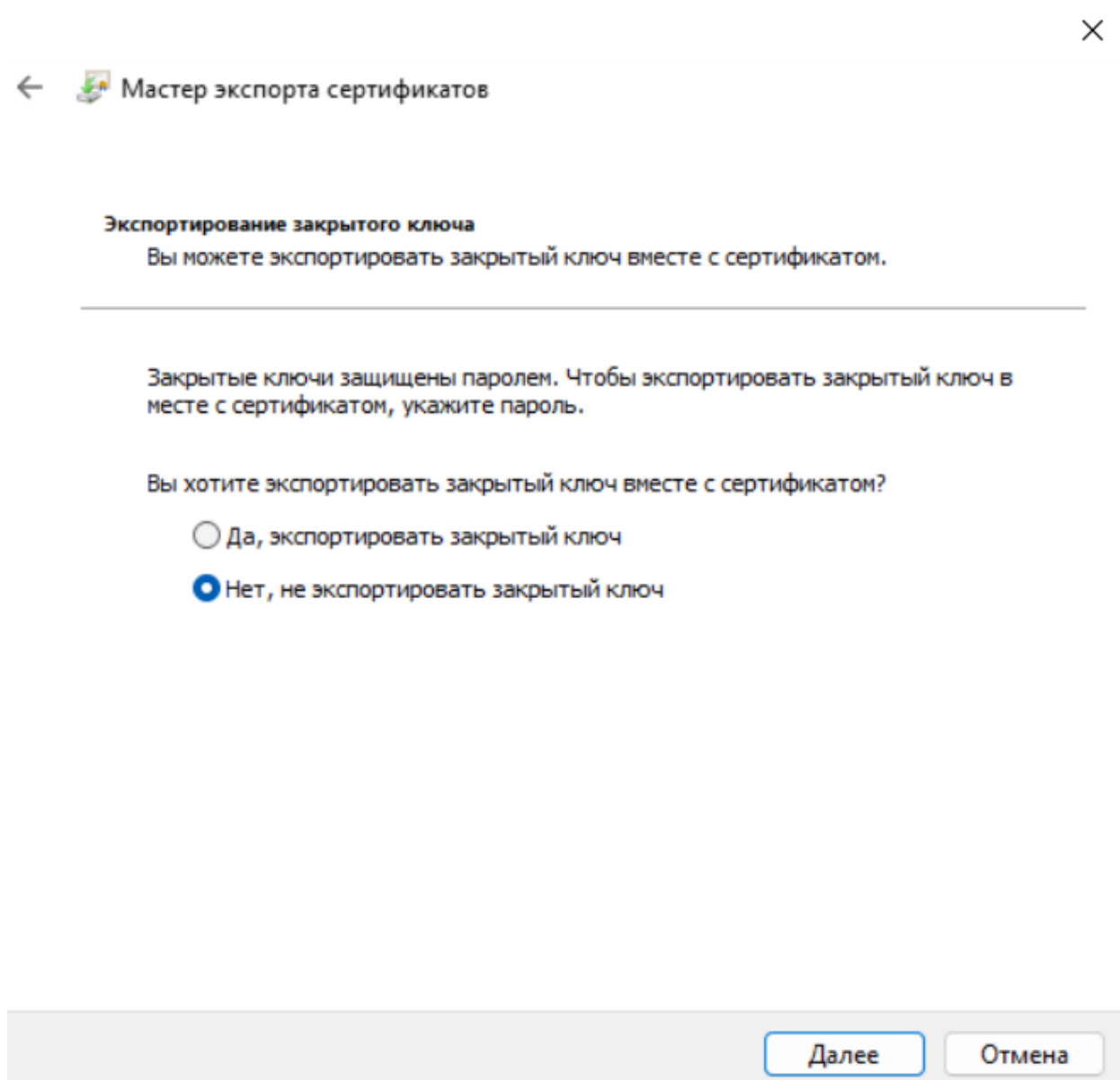


Рисунок 3.6 — Экспорт сертификата без закрытого ключа

Далее открывается раздел с указанием формата экспортируемого файла, как показано на рисунке 3.7. В качестве такого расширения был выбран «Стандарт Cryptographic Message Syntax — сертификаты PKCS #7 (.p7b)». Общая информация об экспортировании сертификата указана на рисунке 3.8.

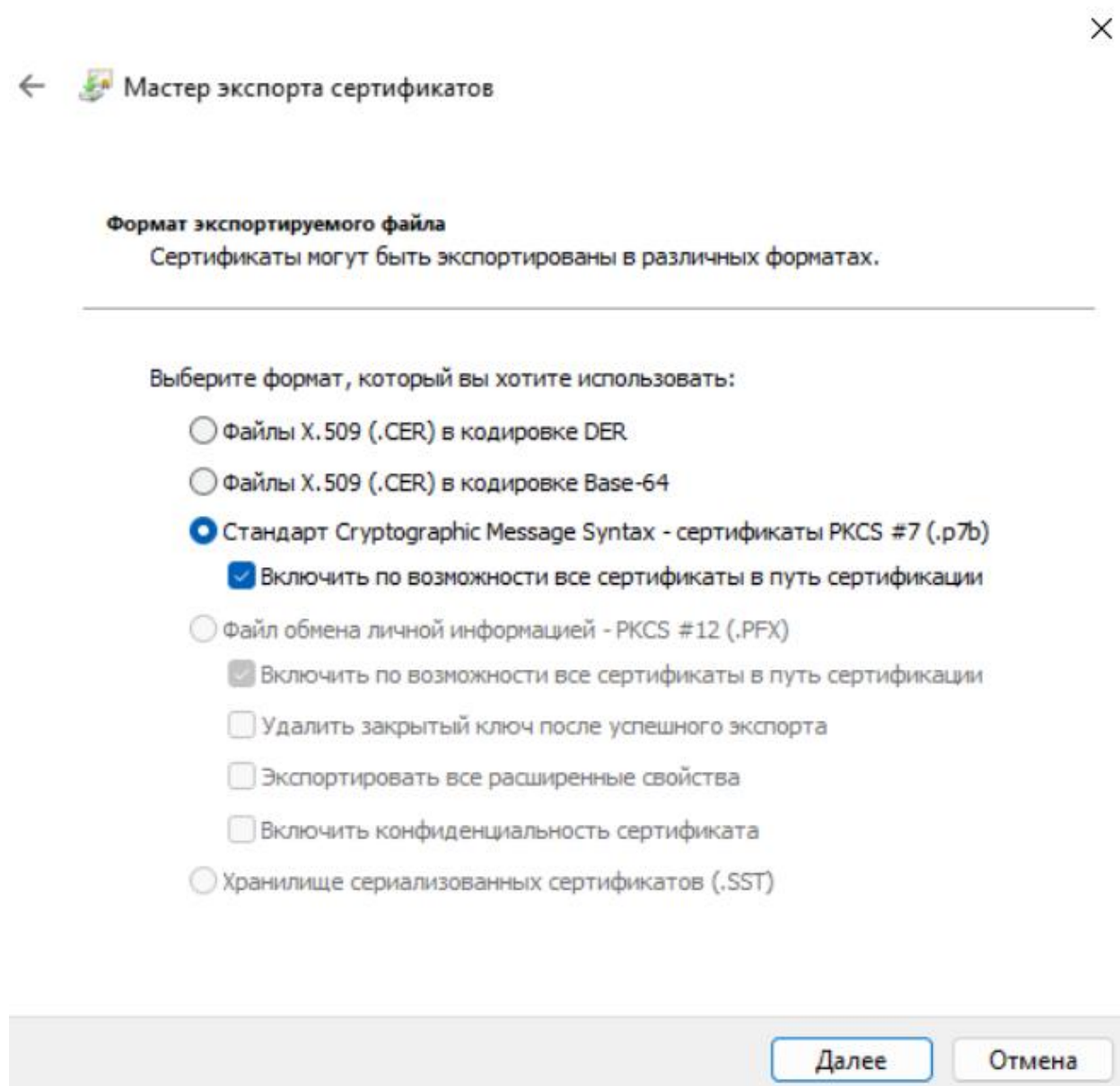


Рисунок 3.7 — Указание формата экспортируемого файла

Завершение работы мастера экспорта сертификатов

Вы успешно завершили работу с мастером экспорта сертификатов.

Были указаны следующие параметры:

Файл	D:\Учеба\ИБ\lab4\SCMS_egorbedish.p7b
Экспорт ключей	Нет
Включить в путь все сертификаты	Да
Формат файлов	Стандарт Cryptographic Message Syntax — c

Готово

Отмена

Рисунок 3.8 — Завершение работы мастера экспорта сертификатов

3.2.2 Импортирование сертификата

Переходя на стадию создания электронной подписи и шифрования данных с помощью открытого ключа получателя, предварительно необходимо импортировать сертификат получателя в список доверенных корневых центров сертификации.

Для этого в меню выбирается функция «Импортировать», после чего вызывается окно «Установка сертификатов, CRL и CTL». Переходя в раздел «Выбор файла сертификата, CRL или CTL», был указан путь до файла сертификата, полученный через flash-носитель от пользователя-получателя, как это показано на рисунке 3.9. Так как данный сертификат не содержится в списке доверенных сертификатов, то его статус на данном этапе будет отображаться как «недействительный».

В приветственном окне не был установлен флаг «Установить личный сертификат», следовательно системой на следующем шаге было предложено указать хранилище, в котором будет храниться импортируемый сертификат. Для данного сертификата было указано хранилище «Доверенные корневые центры сертификации».

В результате проделанных действий в хранилище доверенных сертификатов пользователя-отправителя добавится сертификат получателя, содержащий его открытый ключ.

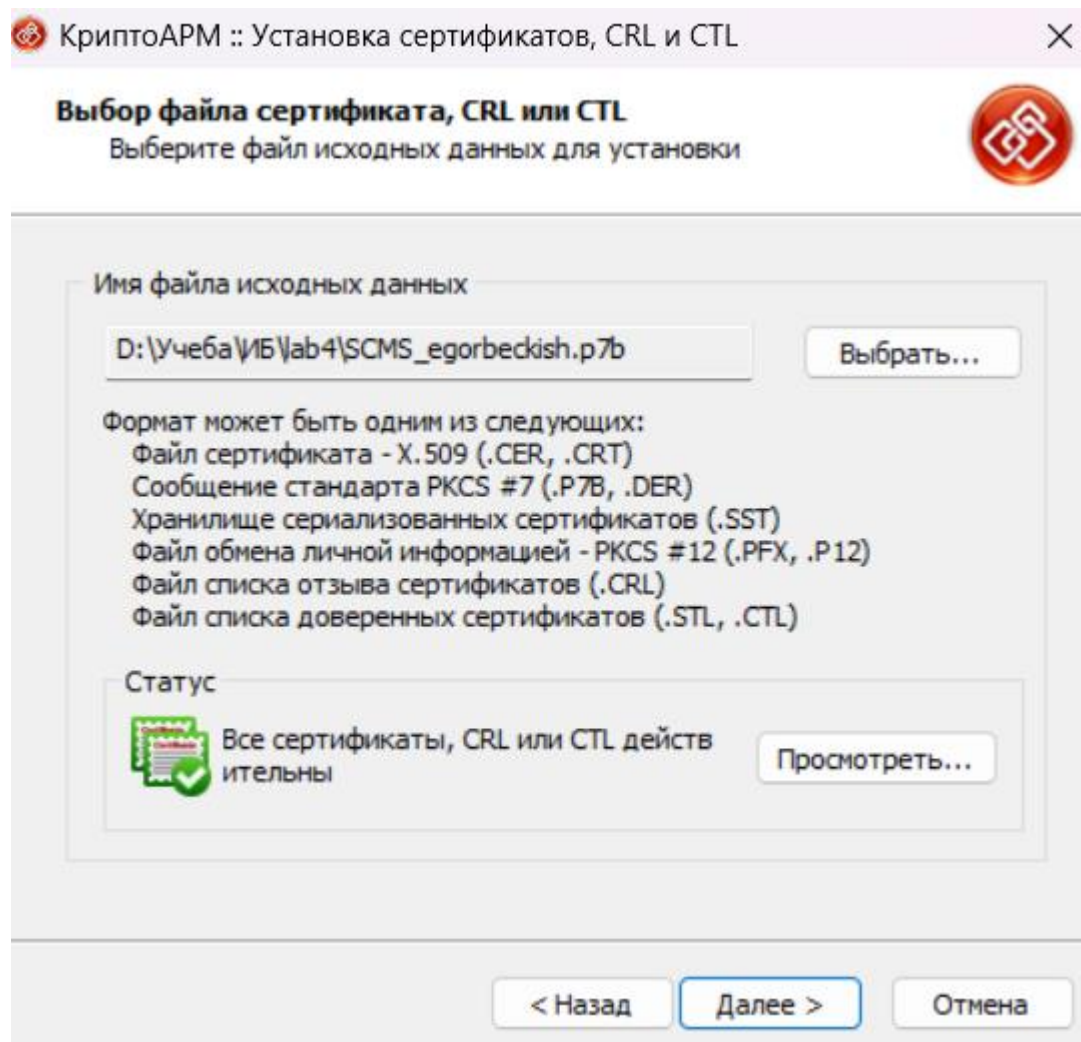


Рисунок 3.9 — Выбор файла сертификата для импортирования

3.3 Создание электронной подписи и шифрование сообщения

Предварительно у пользователя-отправителя должен существовать сертификат. Если этого не было сделано, то осуществляются все те же действия, которые были описаны в пункте 3.1.

В результате проделанных действий на машине отправителя был получен результат, представленный на рисунке 3.5.

Далее, для того чтобы подписать и зашифровать сообщения в одном диалоговом окне, был вызван мастер подписи и шифрования посредством ниспадающего списка из панели меню, как указано на рисунке 3.10.

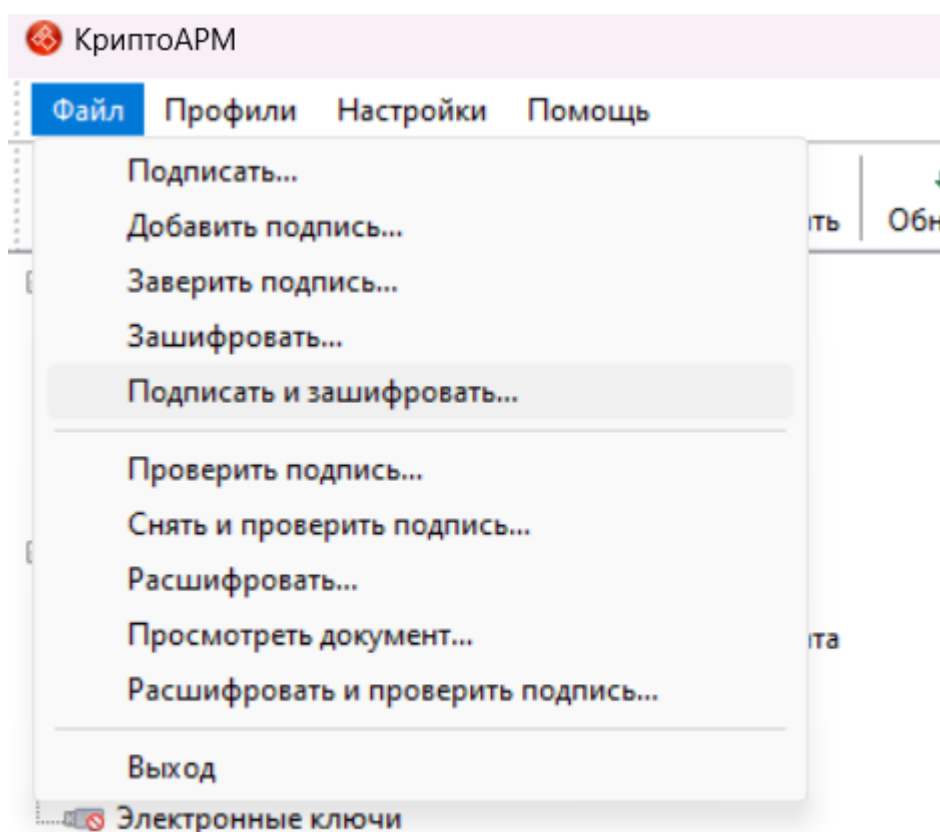


Рисунок 3.10 — Активация мастера подписи и шифрования

3.3.1 Создание электронной подписи

Открыв мастер выполнения операции подписи и шифрования, программа предлагает выбрать файлы и папки, которые необходимо зашифровать и подписать. После добавления документов открывается раздел «Выходной формат», в котором указываются параметры для создания электронной подписи данных. В качестве кодировки и расширения выходного файла был выбран Base64 encoded X.509 с расширением подписанного файла *.sig, как изображено на рисунке 3.11.

В следующем разделе под названием «Параметры подписи» устанавливаются непосредственно параметры подписи. Заполнение этого раздела представлено на рисунке 3.12.

Нажимая кнопку «Далее», мастер предлагает указать сертификат для создания подписи. От имени пользователя-отправителя в качестве личного сертификата для подписи был указан основной сертификат пользователя-отправителя. Хеш-алгоритм, который был выбран на данном этапе, стал SHA-1, как показано на рисунке 3.13.

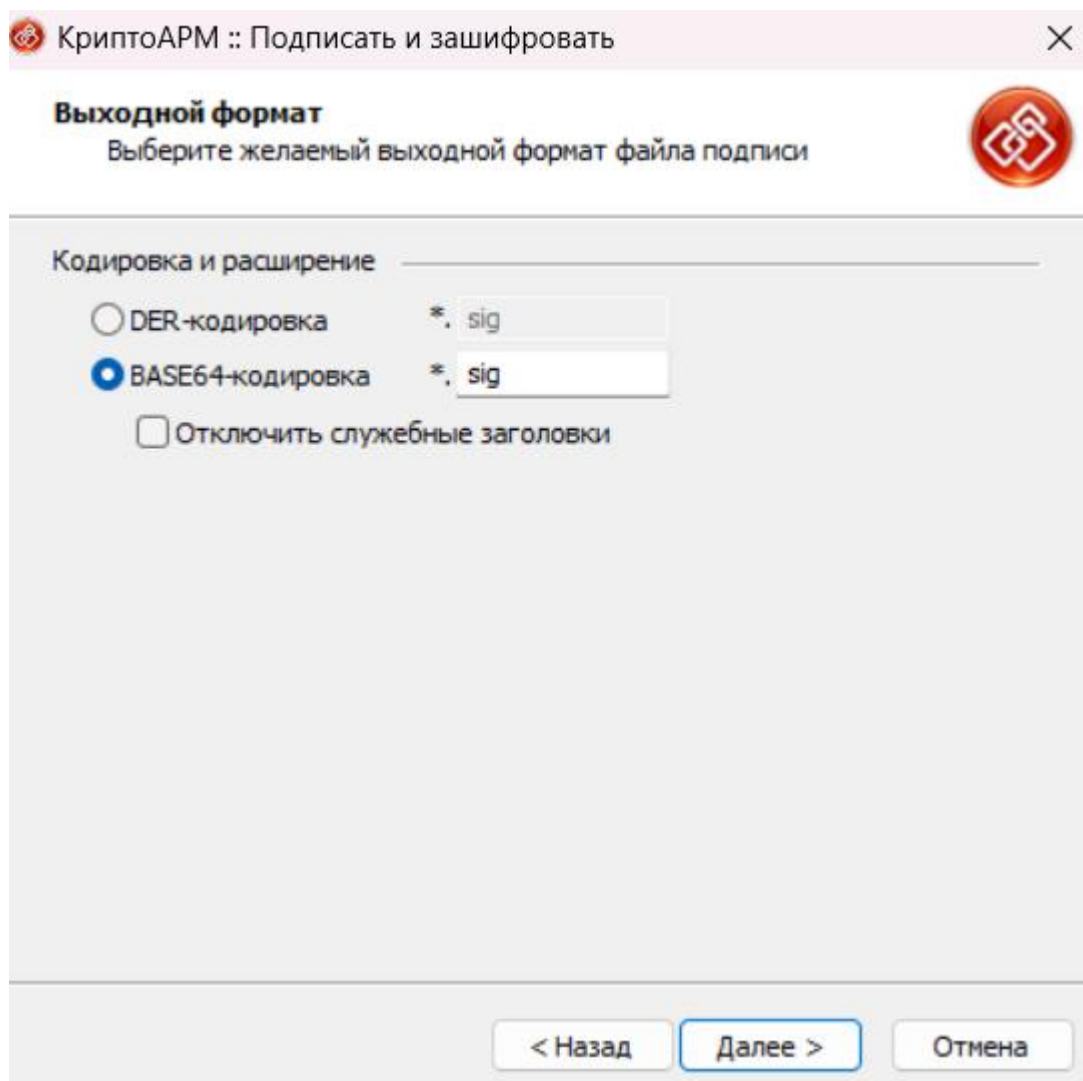


Рисунок 3.11 — Указание кодировки при создании электронной подписи отправителя

КриптоАРМ :: Подписать и зашифровать

Параметры подписи
Установите желаемые параметры подписи

Свойства подписи

Использование подписи: Подписано

Комментарий к подписи: Бекиш Егор Павлович

Идентификатор ресурса: SCMS_egorbeckish.p7b

☒ Поместить имя исходного файла в поле "Идентификатор ресурса"

Включить в подпись: Только сертификат владельца

☐ Сохранить подпись в отдельном файле

☐ Удалить исходный файл после выполнения операции

Уровень безопасного удаления: Выключено

☒ Включить время создания подписи

☐ Включить штамп времени на подписываемые данные

☐ Включить штамп времени на подпись

☐ Включить в подпись доказательства подлинности

< Назад Далее > Отмена

Рисунок 3.12 — Настройка параметров подписи отправителя

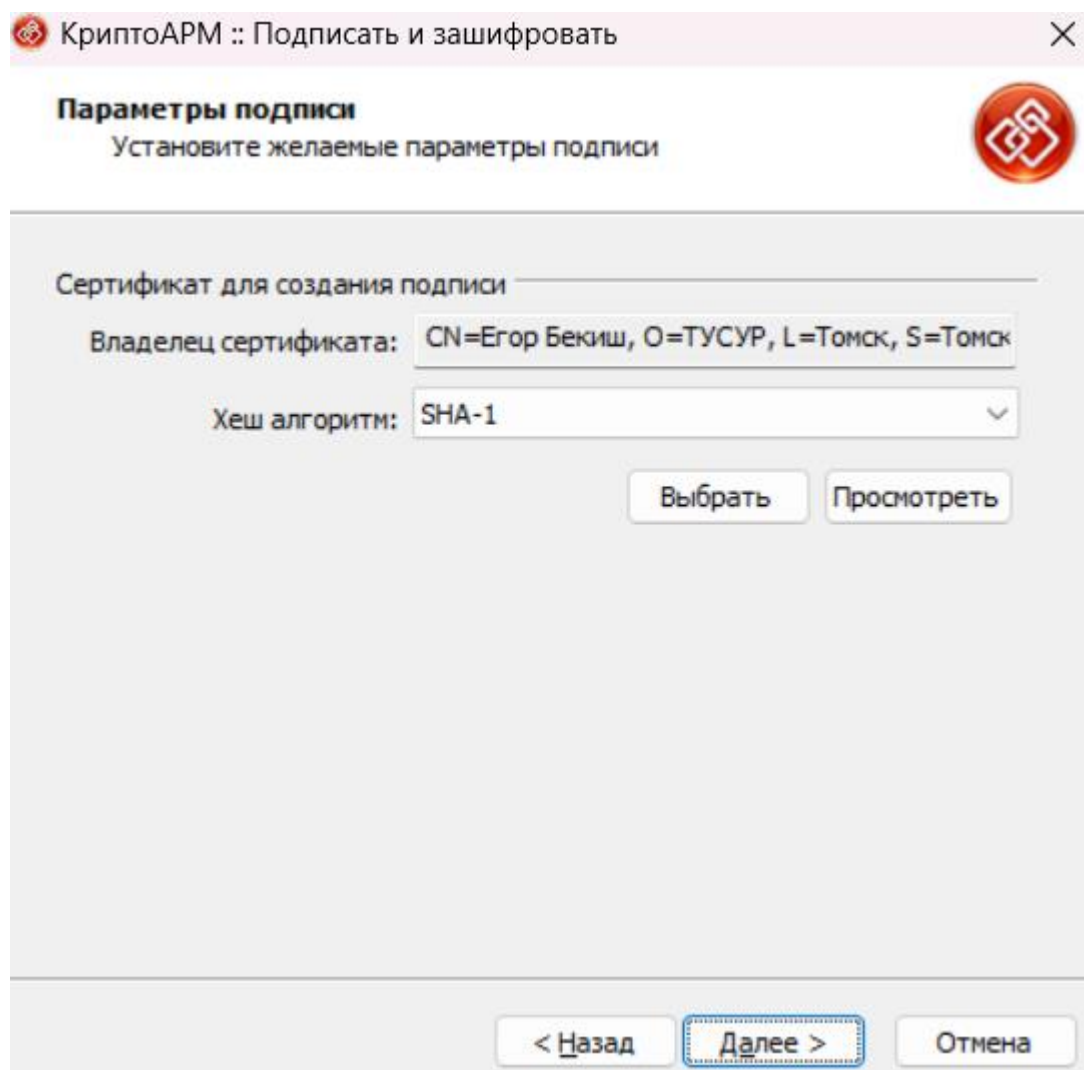


Рисунок 3.13 — Продолжение настройки параметров подписи отправителя

3.3.2 Шифрование сообщения

В том же самом мастере выполнения операции подписи и шифрования после нажатия на кнопку «Далее» мастер переходит к части с настройкой параметров непосредственно шифрования данных.

Подобно созданию подписи, в разделе «Выходной формат файла» на данном этапе указываются настройки выходного формата файла, а именно: кодировка и расширение, флаг архивирования, путь для выходных файлов, настройка сохранения структуры вложенности каталогов и отправка зашифрованного письма по электронной почте. Таким образом указанные для нашего случая настройки изображены на рисунке 3.14.

Далее открывается раздел с настройкой свойств шифрования. В данном разделе необходимо указать режим шифрования для отправителя сообщения. В качестве криптопровайдера был выбран Microsoft Enhanced RSA and AES Cryptoprotider и алгоритм шифрования AES 128, как можно понять из рисунка 3.15.

На следующем шаге мастер предлагает выбрать сертификаты получателей шифруемого файла, используя кнопку «Добавить». Так как мы уже предваритель добавляли сертификат получателя в доверенный список сертификатов, то теперь не доставит труда указать данный сертификат в список получателей шифруемого файла. Результат представлен на рисунке 3.16.

После завершения сбора параметров для выполнения шифрования возникает окно с информацией о статусе операции и об используемых параметрах: сертификат, которым был зашифрован файл и сертификат получателя (-ей). Для продолжения была нажата кнопка «Готово».

Далее возникает окно «Результат выполнения операции» со статусом завершения операции. Для просмотра детальной информации о результатах шифрования и используемых параметрах, необходимо нажать кнопку «Детали». Далее «Менеджер сообщения», в котором можно просмотреть

сертификаты получателей. Детали операции «Подписи и Шифрования» представлены на рисунке 3.17.

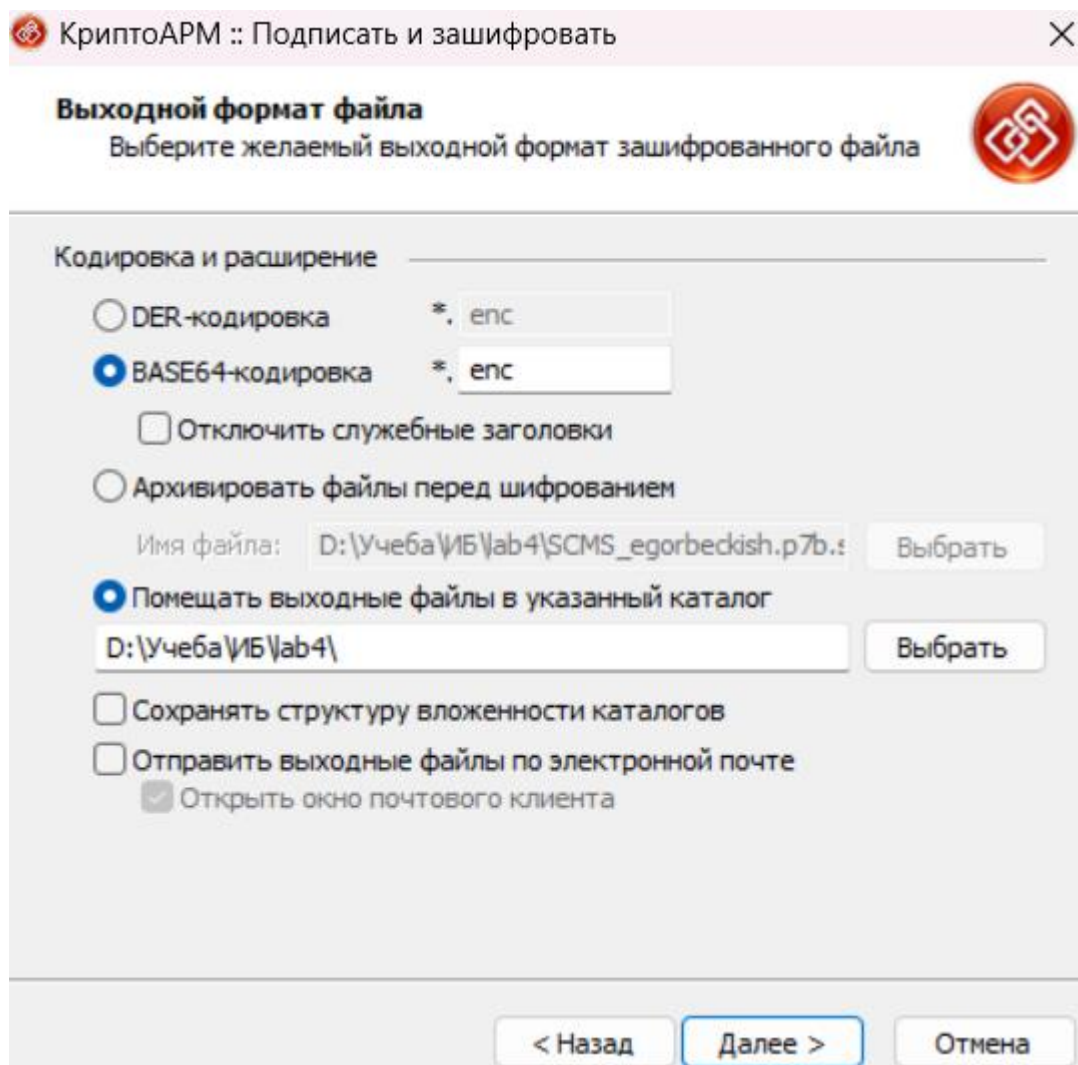


Рисунок 3.14 — Указание выходного формата файла

КриптоАРМ :: Подписать и зашифровать

Свойства шифрования
Выберите необходимые свойства шифрования

Режим шифрования для отправителя сообщения

☒ Использовать криптопровайдер

Тип криптопровайдера: Microsoft Enhanced RSA and AES Cryptogr

Алгоритм шифрования: AES 128

☐ Использовать собственный сертификат

Владелец сертификата:

Алгоритм шифрования:

Выбрать Просмотреть

☐ Включить сертификат в список получателей

< Назад Далее > Отмена

Рисунок 3.15 — Настройка свойств шифрования сообщения

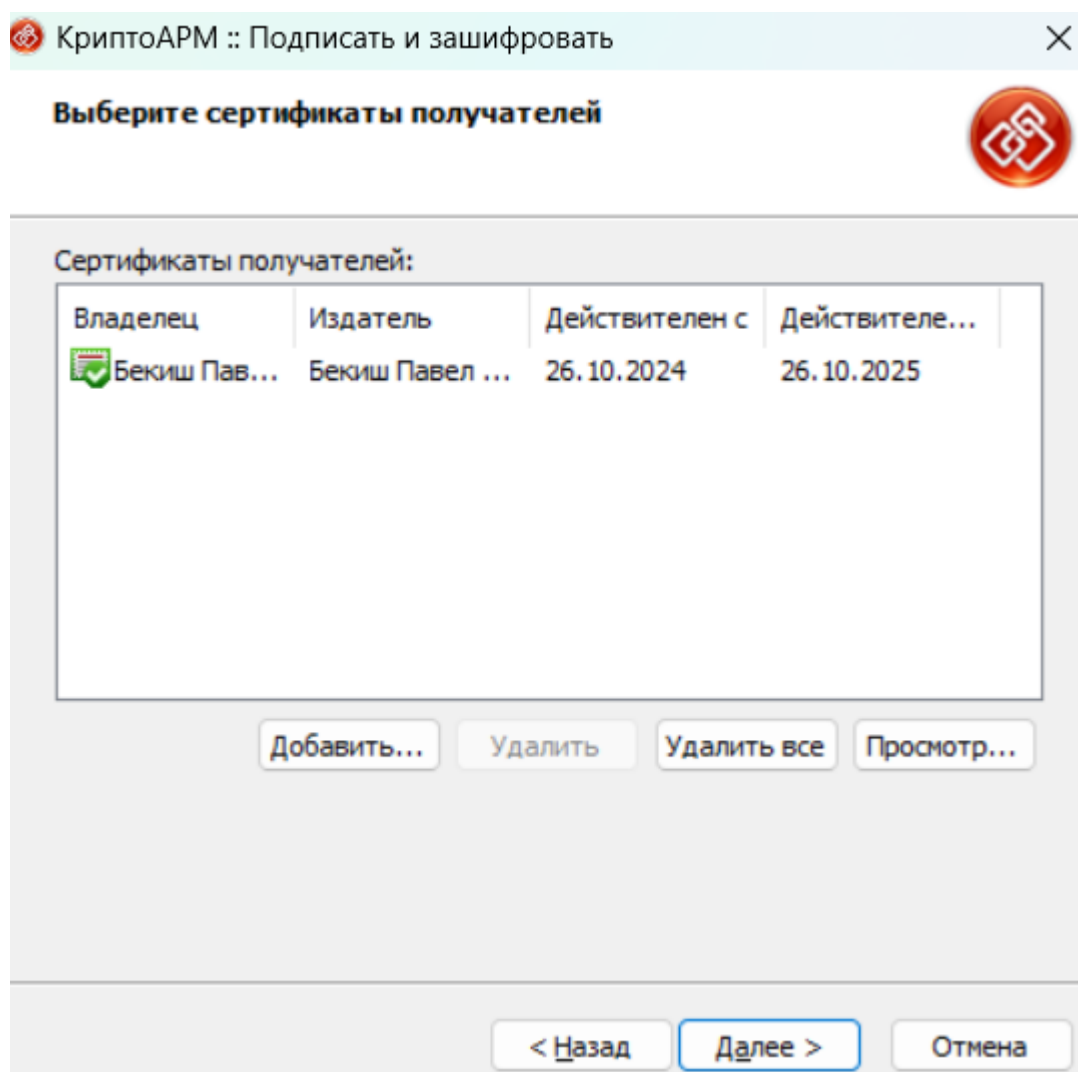


Рисунок 3.16 — Добавление сертификатов получателей

Результат	Дата и время	Вид операции	Пользователь	Имя файла
Успех	26.10.2024 15:45:55	Подпись и шиф...	egorb	Отчет Бекиш Е.П. ЛР2.pdf
Успех	26.10.2024 14:56:41	Обновление TSL	egorb	

Рисунок 3.17 — Результат шифрования

3.4 Расшифрование и проверка подписи

С помощью программы «КриптоАРМ» пользователь-получатель может расшифровать и проверить ЭП отдельного файла или группы файлов, папку с файлами (при этом каждый файл, входящий в указанную папку, будет расшифрован и проверена подпись) или расшифровать архивы.

После открытия мастера операции «Расшифрования и проверка подписи» и выбора настройки по-умолчанию, открывается раздел для выбора файлов с зашифрованными и подписанными данными. Выбрав соответствующие файлы, как показано на рисунке 3.18, мастер переходит в раздел с настройкой сертификатов расшифрования, то есть именно того сертификата, в котором содержится закрытый ключ для расшифрования.

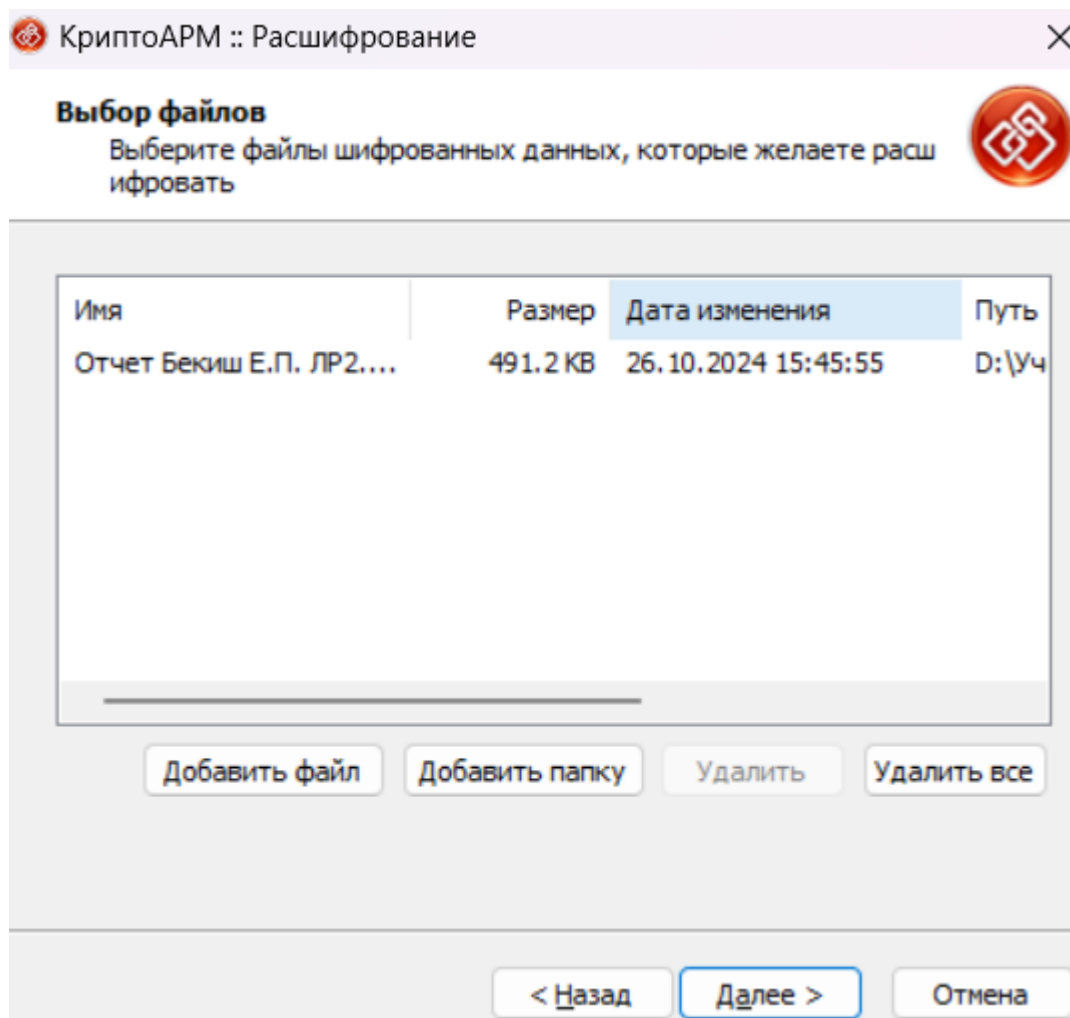


Рисунок 3.18 — Выбора файлов для расшифрования получателем

Далее мастер переходит в обзорный раздел, в котором указаны все данные, как показано на рисунке 3.19.

Нажимая на кнопку «Готово», мастер выполняет расшифрование и отображает результат, как показано на рисунке 3.20.

Переходя в «Файл», можно просмотреть статус подписи сертификатом отправителя, как показано на рисунке 3.21. Как видно из изображения, статус сертификата и подписи «подтверждён». В этом же окне можно сохранить расшифрованный документ.

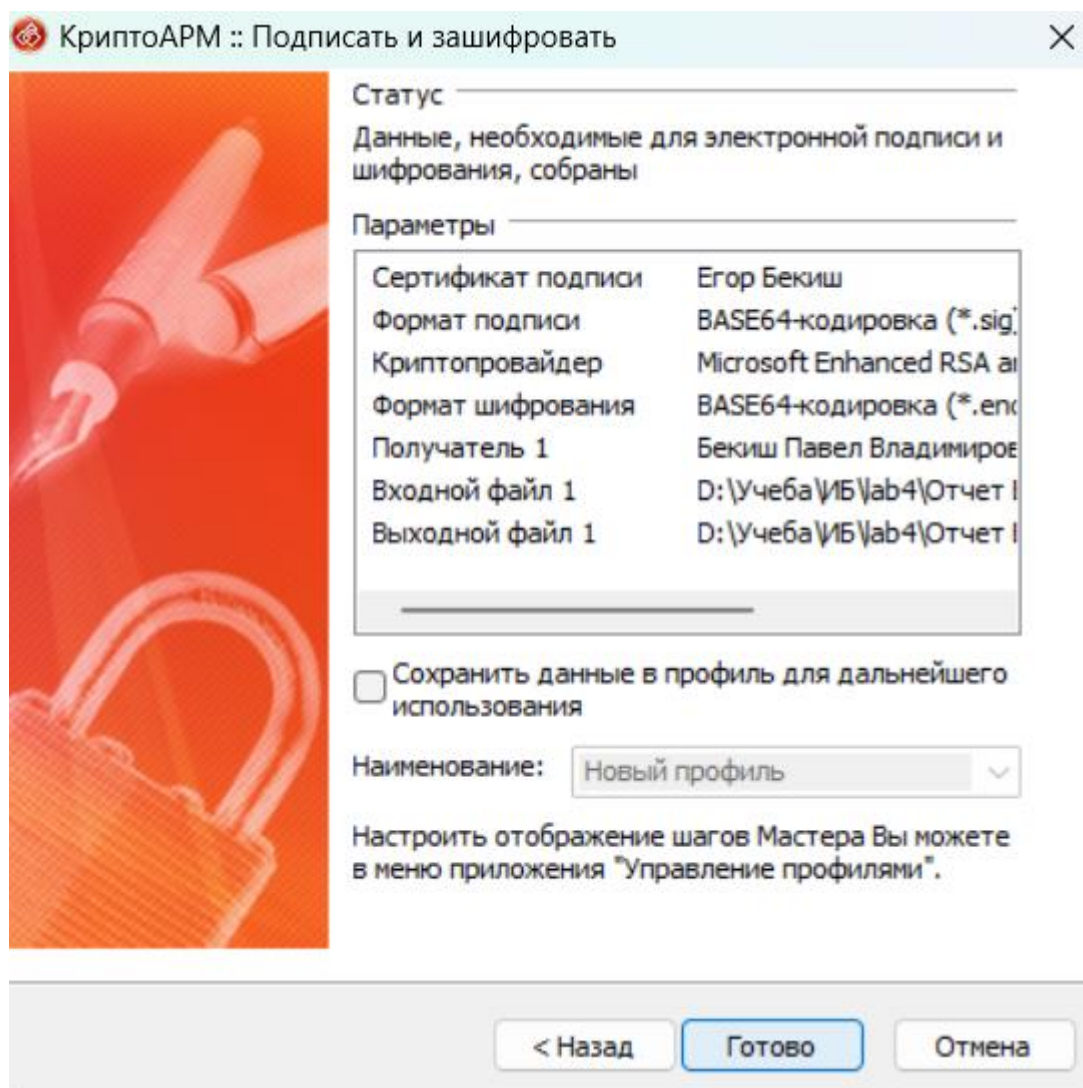


Рисунок 3.19 — Общие сведения настройки расшифрования и проверки подписи


Результ...	Дата и время	Вид опера...	Пользователь	Имя файла
 Успех	26.10.2024 16:19:06	Расшифрование	egorb	Отчет Бекиш Е.П. ЛР2.pd...

Рисунок 3.20 — Результат расшифрования файла

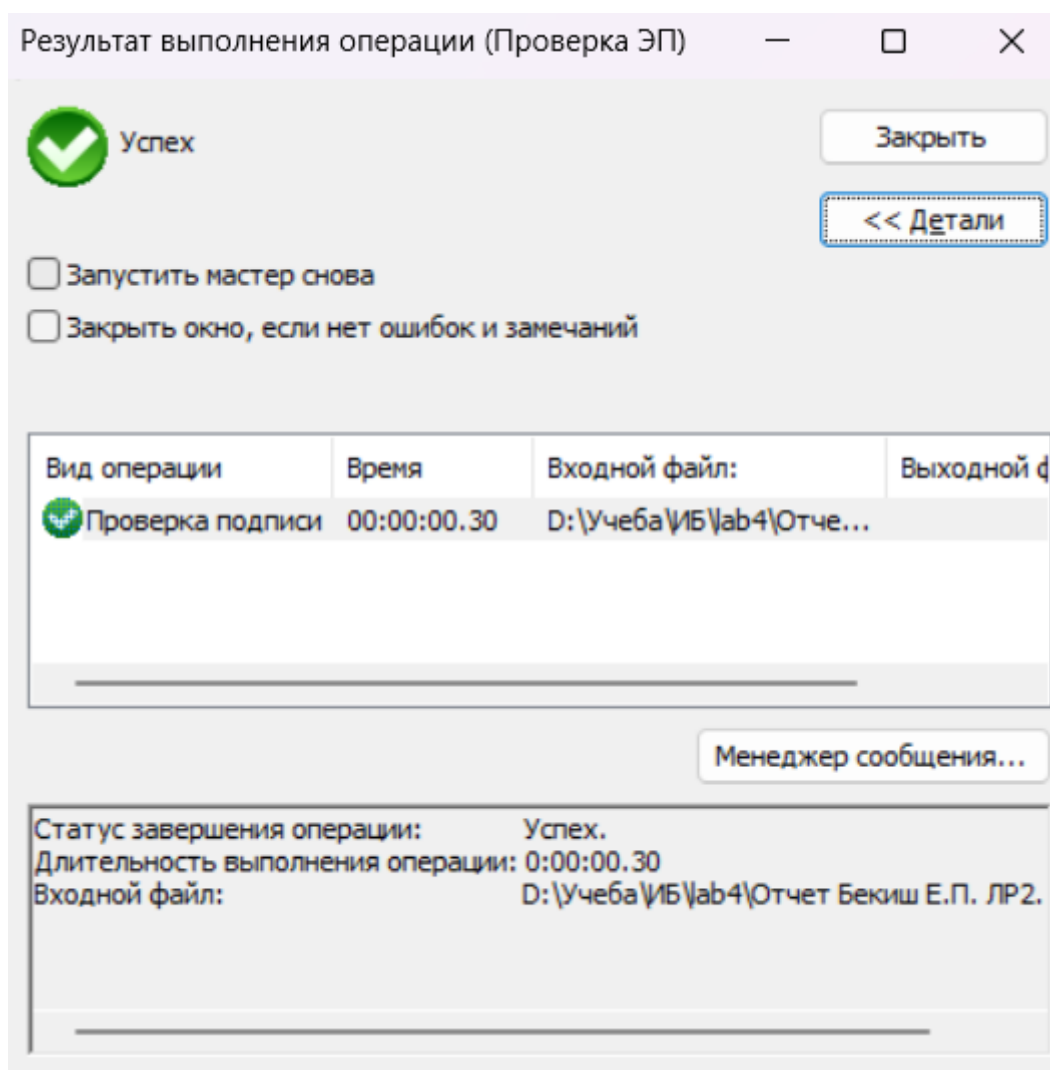
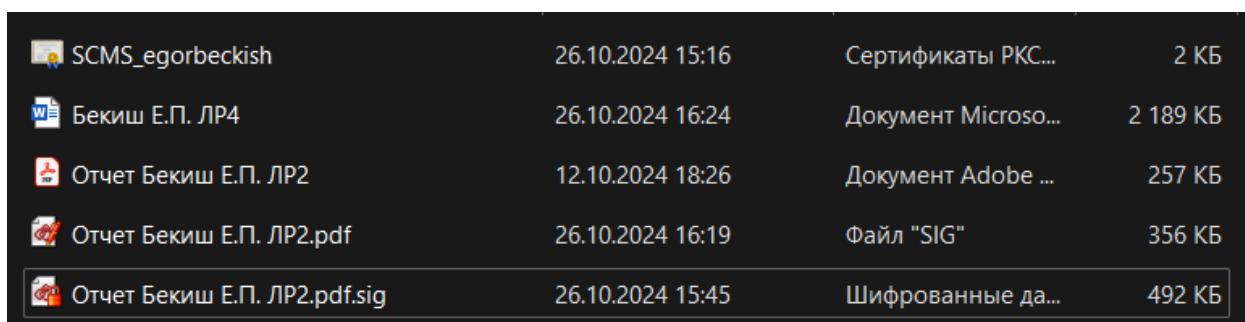


Рисунок 3.21 — Проверка подписи сертификата отправителя

4 Полученные в результате работы файлы

В результате шифрования данным отправителем был получен подписанный и зашифрованный файл: *Отчет Бекиш Е.П. ЛР2.pdf.sig*. Расширение этого файла «.sig» означает, что данный файл был подписан с помощью Base64 encoded X.509, в свою очередь расширение .enc означает, что данный файл зашифрован. На рисунке 4.1 можно увидеть все входные и выходные файлы. Часть содержимого зашифрованного файла можно посмотреть на рисунке 4.2.








	SCMS_egorbeckish	26.10.2024 15:16	Сертификаты PKC...	2 КБ
	Бекиш Е.П. ЛР4	26.10.2024 16:24	Документ Microso...	2 189 КБ
	Отчет Бекиш Е.П. ЛР2	12.10.2024 18:26	Документ Adobe ...	257 КБ
	Отчет Бекиш Е.П. ЛР2.pdf	26.10.2024 16:19	Файл "SIG"	356 КБ
	Отчет Бекиш Е.П. ЛР2.pdf.sig	26.10.2024 15:45	Шифрованные да...	492 КБ

Рисунок 4.1 — Наличие файлов в работе

```

Отчет Бекиш Е.П. ЛР2.pdf.sig.enc X
Отчет Бекиш Е.П. ЛР2.pdf.sig.enc
1  -----BEGIN  CMS-----
2  MIAGCSqGSIb3DQEHA6CAMIACAQAxgGfOMIIBZAIBADCBzDCBvzEaMBgGCCqFAwOB
3  AwEBEgwyNDQ2MDMxNDI5MjcXIjAgBgkqhkiG9w0BCQEWEE2Vnb3JiZWNaXNoQG1h
4  aWwucnUxCzAJBgNVBAYTA1JVMScwJQYDVQQIHH4EIGQ+BDWEQQQ6BDAETwAgBD4E
5  MQQ7BDAEQQRCEWxEzARBgNVBAceCgQiBD4EPARBBDOxEzARBgNVBAoeCgQiBCME
6  IQQjBCAxHTAbBgNVBAMeFAQVBDMEPgRAACAEEQQ1BDoEOARIAgh5K1nc7723RzAN
7  BgkqhkiG9w0BAQcwAASBgLVN0ucJz60cbyj+Hfyo87zn9gErhBxZ0o0tod5/0vOA
8  vWK1SL20SZ8SeSxgMKiuI0ZvMSkgpkQvd5h0XBtdG2PNQ817Ke6ue7pzY+ki+UTX
9  NHPzwSxnuwIUV/KG1bkTU2JIRlpe4IEBHAvC0s9j7xBw/RDz1RC7mst/3VvTO6
10 MIAGCSqGSIb3DQEHA6AdBgIghkgBZQMEAAQIEEIO1FNwIBUHjy385xDdUA+2ggASC
11 BABFvk8USyk6zir5FXLXDtJK1MWSpbMTeYHrM5BmXZyk8uB15M7xAUweAN244BLF
12 vN1rjbXBVPQU8w+em4WCWV8sZYbzrcOf3cMEhJTNHEq7ykTNQjnSB1TZvVjo/ttJ
13 XwD9Cb2mF7vmrFbmHk8WNb3iKH1HMnRJkrnp1wJEyZ6A00UBCj81r6Dwlm8ASUf/
14 3n0gdjej0BZJevYtOnBWZhlybODiRLUR0nye+H1hVxaJ/sBrkW8HZZAwMHMnmCg3
15 sPV2djeFLkxx94e30NzdWz13MMIQC41Ej9qnXFF7dgIAClix1NQTW01G7MI3v6Jj
16 pKGx8vKy0/2oYNt5fBLMjUEBdqXVkJrDZ6NuMOosheVhUoe2YBFKWNVvq9LivhDt
17 0TZ5LmIxV3xEjRQv8yv3Jelra/6VrRsB1og1Tga288quo1FhvCyPaTWwzhEBYPq+
18 DN7T40F68wVsPZ8KebqBy14F4T8apWZREnrF/S5rAHQAj18JzOmAKqsCJqQcvNIK
19 SHLtKuyZ4GVUEqBPODNDPrxvSAp00I6osQVepseWnN4swu3bj4dFVw6ficx3MIwd
20 ynwFxVGAEOamQOpX4h0hhQ4SK9IxxY6dmQgkKkD0EbTiHuDPsvhUcdvktty1Y8R
21 U2PTmg7PFVB0EAg9CzN4Ek9SKw6Kb0K6TgQzLarJ7YoxLCbUse4IQ/8/qK04cgLz
22 1+tk/ct1PyWu5bj9k8BTpuoSmsRt67W8CMPbSX4f4iLDwQnmbQi8LeA09JLft52
23 nM7HifuU0Yep8MQSxOxqn8RarolcvWVCYpbgKPX2E1XAunCfCFow7hWm8BtznMD
24 Er0K6w3voTaDe4jgHnzNT0M0/4XXkSrEWznsMcjjJ9QP1xfg2xIt23WBXiYhBlKk
25 uD/b0wJbWmWZRNLKIym9A2Z1Y5VSNc8ZjexA3Zizzb9uaNJ37ROj3veSSR8svmi
26 fKYH3CF7pGdfK+QP5Y5yfYcfY++ikdIy+nHsX92p221t5WSZSenY8Q4tbbbZH2b
27 M4IG3VHs3g8dcpt7tnMwseghT1kpY1sLSDUiknEUf1sONP64HNoX0DGSIXLHdc6x
28 HiEFD5K73/NUpmvBgI76L32yH4UDFwF//Obju2pu13EOPa+30FQZLEE6736ySHBv
29 JDZ+RZh4AXQEblZkGkrtGPYLYwaJNjVxme5+E2o06doolT7gj3CEmacrPT0c1LvR

```

Рисунок 4.2 — Содержимое зашифрованного файла

5 Вывод

В результате выполнения лабораторной работы я освоил процесс создания ключей, распространения открытых и сохранения в тайне закрытых ключей, а также шифрования, расшифрования, создания и подтверждения электронных подписей в криптосистемах.