

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт
Факультет приборостроения,
информационных технологий и
электроники

Кафедра
«Информационная безопасность
систем и технологий»

Специальность 100503 –

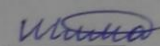
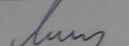
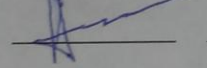
Специализация -

Информационная безопасность
автоматизированных систем
Защищенные автоматизированные
системы управления

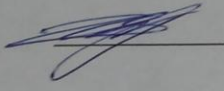
ДИПЛОМНЫЙ ПРОЕКТ

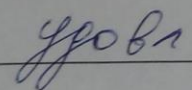
на тему

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ДЛЯ ИЗУЧЕНИЯ ИНФРАСТРУКТУРЫ
ОТКРЫТЫХ КЛЮЧЕЙ

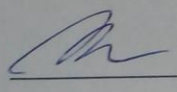
Студент		Шимаханов Антон Витальевич
Руководитель		Липилин О.В.
Нормоконтролёр		Фатеев А.Г.
Рецензент		Акмаев А.Ж.

Работа допущена к защите (протокол заседания кафедры от 20.06.2016 № 14)

Заведующий кафедрой  С.Л. Зефирова

Работа защищена с отметкой  (протокол заседания ГЭК
от 23.06.16 № 9)

Секретарь ГЭК

 А.П. Иванов

Пенза 2016

Проверка ВКР
Зданов
 С.Л. Зефирова

РЕФЕРАТ

Отчет содержит 63 с., 24 рис., 11 источников.

ИНФРАСТРУКТУРА ОТКРЫТЫХ КЛЮЧЕЙ, ЭЛЕКТРОННАЯ ПОДПИСЬ, СЕРТИФИКАТ, СПИСКИ АНУЛЛИРОВАННЫХ СЕРТИФИКАТОВ, ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Объектом исследований является инфраструктура открытых ключей.

Цель дипломного проекта - разработка лабораторного практикума для изучения инфраструктуры открытых ключей.

В процессе работы был проведен обзор инфраструктуры открытых ключей и проведен анализ возможных решений по реализации лабораторного практикума для изучения инфраструктуры открытых ключей, разработана архитектура лабораторного стенда для изучения архитектуры открытых ключей и лабораторный практикум, в котором рассматриваются вопросы генерации ключей, создания и использования сертификатов и списков аннулированных сертификатов.

В результате был разработан лабораторный стенд для изучения инфраструктуры открытых ключей, был разработан лабораторный практикум, включающий рекомендации по выполнению и варианты заданий.

СОДЕРЖАНИЕ

Введение.....	6
1 Обзор инфраструктуры открытых ключей	7
2 Анализ возможных решений по реализации инфраструктуры открытых ключей	16
3 Разработка архитектуры лабораторного стенда для изучения инфраструктуры открытых ключей	31
4 Разработка лабораторного стенда для проведения лабораторного практикума	33
5 Разработка лабораторного практикума для изучения инфраструктуры открытых ключей	36
5.1 Изучаемые вопросы	36
5.2 Создание ключей.....	36
5.3 Создание сертификатов	37
5.4 Установка сертификатов в операционную систему	37
5.5 Просмотр свойств установленных сертификатов.....	37
5.6 Использование сертификатов для формирования электронной подписи документов.....	38
5.7 Проверка электронной подписи документа	38
5.8 Проверка электронной подписи документа при истекшем сроке действия сертификата	39
5.9 Формирование списка отозванных сертификатов.....	39
6 Разработка рекомендаций для проведения лабораторного практикума	41
6.1 Создание ключей.....	41
6.2 Создание сертификатов	42
6.3 Установка сертификатов в операционную систему	43
6.4 Просмотр свойств установленных сертификатов.....	47
6.5 Использование сертификатов для формирования электронной подписи документов.....	49
6.6 Проверка электронной подписи документа	52

6.7 Проверка электронной подписи документа при истекшем сроке действия сертификата	54
6.8 Формирование списка отозванных сертификатов.....	55
7 Разработка вариантов заданий для проведения лабораторного практикума.....	57
Заключение	61
Список использованных источников	63

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

PKI - инфраструктура открытых ключей;

СА - центр сертификации;

УЦ - удостоверяющий центр;

ОС - операционная система;

САС - список аннулированных сертификатов;

ППС - политика применения сертификатов;

РЦ - регистрационный центр.

ВВЕДЕНИЕ

Инфраструктура безопасности для распространения открытых ключей, управления электронными сертификатами и ключами пользователей называется РКІ - инфраструктура открытых ключей (PKI). Термин "PKI" происходит от названия технологии криптографии с открытым ключом, который обладает уникальными свойствами и является основой для реализации функций безопасности в распределенных системах. Инфраструктура открытых ключей реализуется не ради неё самой, но и для поддержки безопасности других приложений. Есть, конечно, и другие механизмы безопасности, которые не используют шифрование с открытым ключом, и они менее сложные, чем PKI. Однако PKI не только предлагает наиболее полное решение, но снимает многие проблемы, присущие более традиционные механизмы безопасности.

Студентам, обучающимся по специальности 10.03.05 «Информационная безопасность автоматизированных систем» необходимо получить знания и выработать практические навыки по применению архитектуры открытых ключей, поэтому актуальной является проблема разработки лабораторного практикума для изучения архитектуры открытых ключей. Таким образом, тема выпускной квалификационной работы является актуальной.

1 Обзор инфраструктуры открытых ключей

Инфраструктура открытых ключей - это комплект средств, распределенных служб и компонентов, в совокупности используемых для поддержки задач шифрования на основе открытого и закрытого ключа. Основой инфраструктуры открытых ключей является использование криптографической системы с открытым ключом и несколько главных принципов:

- закрытый ключ (private key) известен только его владельцу;
- удостоверяющий центр формирует электронный документ - сертификат открытого ключа, таким образом, подтверждающий тот факт, что закрытый (секретный) ключ известен только владельцу этого сертификата, открытый ключ (public key) свободно передается в сертификате;
- никто не доверяет друг другу, но доверяет органу по сертификации;
- удостоверяющий центр подтверждает или опровергает принадлежность открытого ключа заданному лицу, который владеет соответствующим закрытым ключом[1].

Объекты PKI [1].

Инфраструктура открытых ключей реализована в модели клиент-сервер, то есть проверка любой информации от инфраструктуры может происходить только по инициативе клиента.

Основные компоненты PKI [2].

Удостоверяющий центр - это единый центр генерации цифровых сертификатов. У пользователей имеется база публичных ключей, разных СА, и они проверяют ими приходящие сертификаты. Основная предпосылка криптографии с открытыми ключами заключается в том, что два незнакомых друг с другом субъекта должны иметь возможность безопасно связываться между собой. Например, если пользователь желает отправить секретное сообщение другому пользователю, с которым он не знаком, то для шифрования сообщения он должен иметь возможность связать каким - либо образом с другим пользователем и его открытым ключом. Связывания открытых ключей

и их владельцев занимается организация доверенных центров. Этим центрам доверяют выполнение функций связывания ключей и идентификационных данных (идентичности) пользователей.

Такие доверенные центры в терминологии PKI называются удостоверяющими центрами (УЦ). Они производят связывание пары ключей с идентичностью, утверждая цифровой подписью данные, которые включают в себя некоторое представление идентичности и соответствующего открытого ключа. Данную структуру данных называют сертификатом. Сертификат представляет собой зарегистрированное удостоверение, которое хранится в цифровом виде и принимается сообществом пользователей PKI законным и надежным. При создании электронного сертификата используется электронная цифровая подпись УЦ - в этом смысле удостоверяющий центр подобен нотариальной конторе, из-за того что подтверждает подлинность сторон, участвующих в обмене электронными сообщениями или документами.

Хотя удостоверяющий центр не всегда входит в состав PKI (в небольших инфраструктурах, или тех, которые находятся в закрытых средах, где пользователи могут сами выполнять функции управления сертификатами), он является очень важным компонентом многих крупномасштабных PKI. Прямое использование открытых ключей требует прибегнуть к дополнительной защите и идентификации для устанавливаемой связи с секретным ключом. Без такой дополнительной защиты злоумышленник может предоставлять себя как за отправителя подписанных данных, так и за получателя зашифрованных данных, изменив значение открытого ключа или нарушив его идентификацию. Все это приводит к необходимости проверки подлинности, то есть проверки открытого ключа.

Удостоверяющий центр объединяет людей, процессы, программные и аппаратные средства, включенные в безопасное объединение пользователей и их открытых ключей. Удостоверяющий центр известен субъектам инфраструктуры открытых ключей по двум атрибутам: названию и открытому ключу. УЦ включает свое имя в каждый выпущенный им сертификат и в

список аннулированных сертификатов (САС) и подписывает их при помощи собственного секретного ключа. Пользователи могут легко идентифицировать сертификаты по имени УЦ и убедиться в их подлинности, используя его открытый ключ, основные компоненты PKI представлены на рисунке 1.

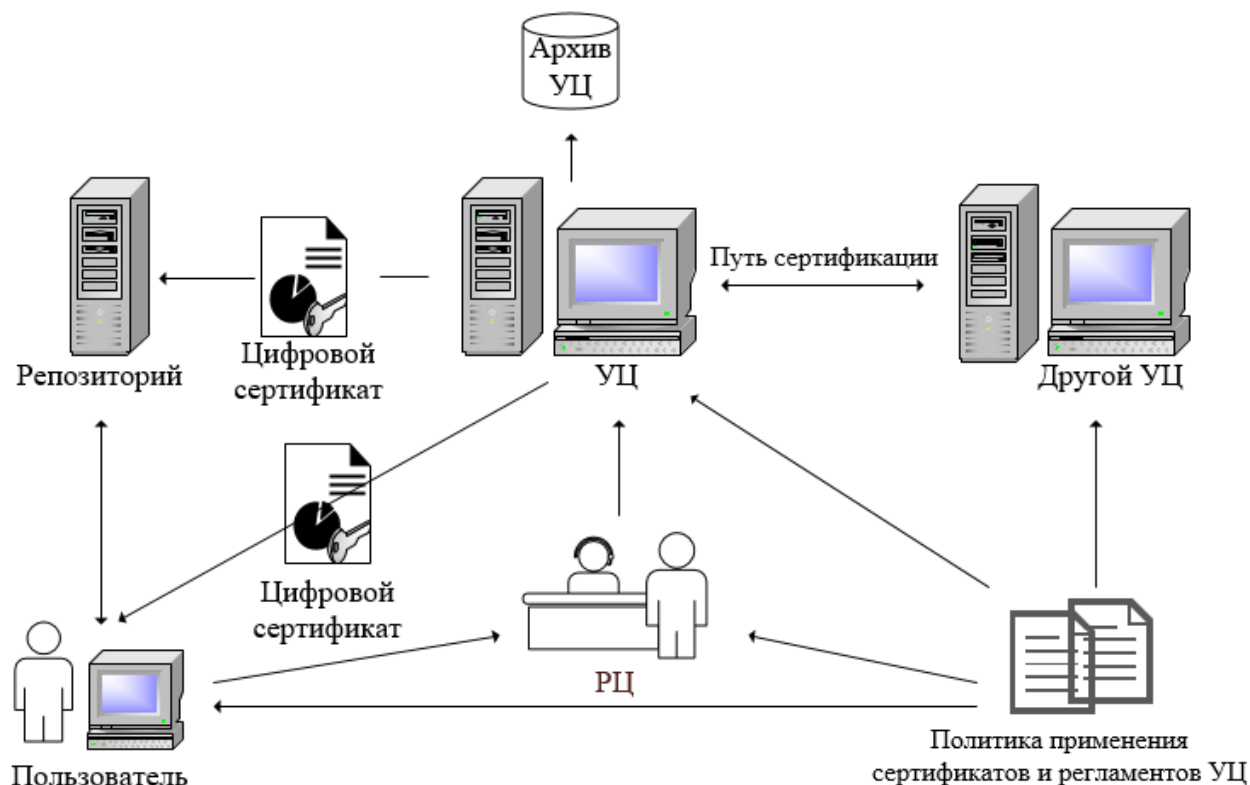


Рисунок 1 - Основные компонент PKI

Удостоверяющий центр - главный управляющий компонент PKI, выполняющий следующие основные функции:

- формирует собственный секретный ключ; если является головным удостоверяющим центром, то издает и подписывает свой сертификат, называемый самоизданным или самоподписанным;
- создает и подписывает сертификаты открытых ключей подчиненных удостоверяющих центров и конечных субъектов РК;
- ведет базу всех изданных сертификатов и формирует списки САС с регулярностью, установленной удостоверяющим центром;
- публикует информацию о статусе сертификатов и списков САС.

При необходимости УЦ может передавать некоторые функции другим компонентам PKI. Для подтверждения, что лицо, указанное в сертификате, владеет секретным ключом, который соответствует этому открытому ключу, выпуская сертификат открытого ключа. Включая в сертификат дополнительную информацию, удостоверяющий центр подтверждает ее принадлежность этому субъекту. Дополнительная информация может быть контактной или содержать сведения о типах приложений, которые могут работать с данным сертификатом. Когда субъектом сертификата является другой удостоверяющий центр, издатель подтверждает надежность выпущенных этим центром сертификатов.

Действия УЦ ограничены политикой применения сертификатов (ППС), которая определяет назначение и содержание сертификатов. Удостоверяющий центр выполняет защиту своего секретного ключа и открыто публикует свою политику, чтобы пользователи могли иметь возможность для ознакомления с назначением и правилами использования сертификатов. Ознакомившись с политикой использования сертификатов и решив, что доверяют удостоверяющему центру и его деловым операциям, пользователи могут полагаться на сертификаты, выпущенные этим центром. Таким образом, в PKI удостоверяющие центры выступают как доверенная третья сторона.

Регистрационный центр (РЦ) - необязательный компонент системы, предназначенный для регистрации пользователей. Для этих целей РЦ обычно использует web-интерфейс. Удостоверяющий центр доверяет регистрационному центру проверку информации о субъекте. Регистрационный центр, убедившись в правильность информации, подписывает своим ключом и передаёт удостоверяющему центру, который, проверив ключ регистрационного центра, выдает сертификат. Один регистрационный центр может работать с несколькими удостоверяющими центрами и регистрационными центрами. Иногда, удостоверяющий центр выполняет функции регистрационного центра.

Репозиторий - это хранилище, содержащее сертификаты и списки отозванных сертификатов (COC) и служащее для распространения этих

объектов среди пользователей. В Федеральном Законе РФ №63 «Об электронной подписи» он называется реестр сертификатов ключей подписей.

Архив сертификатов - это хранилище всех изданных когда-либо сертификатов (включая сертификаты с закончившимся сроком действия). Архив используется для проверки подлинности электронной подписи, которой заверялись документы.

Центр запросов - необязательный компонент системы, где конечные пользователи могут запросить или отозвать сертификат.

Конечные пользователи - пользователи, приложения или системы, являющиеся владельцами сертификата и использующие инфраструктуру.

Основные задачи[1].

Главными задачами систем информативной защищенности, которые решает инфраструктура управления открытыми ключами:

- обеспечение конфиденциальности данных;
- обеспечение целостности данных;
- обеспечение аутентификации пользователей и ресурсов, к которым обращаются пользователи;
- обеспечение способности подтверждения совершенных пользователями действий с данными.

Архитектуры PKI[2].

Элементарная PKI (одиночный УЦ). В этом варианте все без исключения пользователи верят одному УЦ и переписываются между собою. В данной архитектуре, в случае если правонарушитель выдаст себя за УЦ, следует попросту вновь изготовить все без исключения выписанные сертификаты прежде и продолжить нормальную работу.

Иерархическая PKI. Иерархическая состав - это более часто встречающаяся структура PKI. В этом случае в базе всей структуры стоит единственный основной УЦ, которому все без исключения верят и ему подчиняются нижестоящие УЦ. Помимо данного ведущего УЦ в текстуре

существуют еще не один УЦ, который подчиняется вышестоящему, которому в свою очередь приписаны какие-либо пользователи или нижестоящие УЦ.

В основном акцентируют пять типов архитектур РКИ.

Простая РКИ (одионочный УЦ). В этом варианте все без исключения пользователи верят одному УЦ и переписываются между собою. В данной архитектуре, в случае если правонарушитель выдаст себя за УЦ, следует попросту вновь изготовить все без исключения выписанные сертификаты прежде и продлить нормальную работу.

Иерархическая РКИ. Иерархическая состав - это более часто встречающаяся структура РКИ. В этом случае, в базе всей структуры нужен один основной УЦ, которому все без исключения верят и ему подчиняются нижестоящие УЦ. Помимо данного ведущего УЦ в структуре существуют еще никак не единственный УЦ, какой подчиняется вышестоящему, который в свою очередь приписаны какие-либо пользователи или нижестоящие УЦ.

Сетевая РКИ. Сетевая структура РКИ создается равно как линия доверия, множественные подтверждающие центры которые предоставляют РКИ-сервисы и связаны равными взаимоотношениями. Однако в этом случае отсутствует одного головного УЦ, которому все без исключения верят. В данной архитектуре все без исключения УЦ доверяют рядом стоящим УЦ, а любой пользователь верит только лишь этому УЦ, у которого выписывал сертификат. Подтверждающие центры выпускают сертификаты друг для друга; пара сертификатов описывает двухсторонние взаимоотношения доверия. В эту архитектуру РКИ свободно добавляется новейший УЦ, для этого ему необходимо обмениваться сертификатами, согласно последней мере, с одним УЦ, который уже вступает в сеть. В этой архитектуре более непростое построение цепочки сертификации.

Сетевые РКИ владеют огромной гибкостью, так как обладают множественными пунктами доверия. Компрометация одного УЦ не отображается в сетевой РКИ в целом. Удостоверяющие центры, которые изготовили сертификаты с целью скомпрометированного УЦ, попросту

аннулируют их, этим самым удаляя с инфраструктуры ненадежный УЦ. Вследствие чего, не нарушается деятельность пользователей, связанных с иными удостоверяющими центрами, - они по-прежнему имеют все шансы полагаться в надежные пункты доверия, и защищено соединяться с другими пользователями своей PKI. Компрометация сетевой PKI приводит к тому, что сворачивается деятельность одного УЦ совместно с его сообществом пользователей, либо, в случае если не внушает доверие несколько удостоверяющих центров, к тому, что PKI распадается на несколько меньших инфраструктур. Возобновление после компрометации сетевой PKI совершается легче, нежели иерархической, в первую очередь в целом, потому что компрометация касается меньше количества пользователей.

Создать путь сертификации в сети довольно трудно, так как данная процедура никак не детерминирована и существуют многочисленные виды развития цепи сертификатов. Одни из них приводят к построению верного пути, прочие - заводят в безвыходное положение. Согласно данному обстоятельству валидация пути сертификации зачастую производится одновременно с его построением, частью данного процесса считается устранение ложных ветвей. С целью построения правильного пути применяется ряд добавочных полей сертификатов.

Кросс-сертифицированные корпоративные PKI. Этот тип архитектуры, возможно, анализировать равно как совместный тип иерархической и сетевой архитектур. Имеется несколько фирм, у каждой из которых организована какая-то собственная PKI, однако они желают общаться между собой, вследствие чего же появляется их общая межфирменная PKI. В архитектуре кросс-сертифицированной корпоративной PKI наиболее сложная система цепочки сертификации.

Архитектура мостового УЦ. Структура мостового УЦ разрабатывалась с целью того, чтобы убрать недостатки сложного процесса сертификации в кросс-сертифицированной корпоративной PKI. В этом случае все без исключения компании вероятнее какой-то одной или двум фирмам, доверяют

одному установленному мостовому УЦ, который считается почти их головным УЦ, однако он не является главным пунктом доверия, а выступает в роли посредника между другими УЦ.

Формат сертификатов открытых ключей X.509[3].

Формат сертификата открытого ключа определен в рекомендациях Международного Союза по телекоммуникациям (X.509) и документе RFC 3820, организации инженерной поддержки Internet Engineering Task Force (IETF). IETF является открытым сообществом исследователей, разработчиков сетевых протоколов, операторов и производителей, занимающихся проблемами развития сетей Интернет и предоставлением непрерывного функционирования существующей инфраструктуры. Несмотря на то, что документ RFC 3820 адресован Интернет - сообществу, параметры сертификата открытого ключа даёт эластичный механизм передачи различной информации и используется в корпоративных PKI.

Сертификат открытого ключа подписи или шифрования подразумевает собой высокоструктурированную двоичную запись в виде абстрактной синтаксической нотации ASN.1. Сертификат содержит элементы данных, сопровождающиеся цифровой подписью издателя сертификата. В сертификате существует десять основных полей: шесть обязательных и четыре опциональных. Значительная доля информации, которую необходимо указать в сертификате, не является обязательной, а сущность обязательных полей сертификата может колебаться. К обязательным полям относятся:

- серийный номер сертификата;
- идентификатор алгоритма подписи;
- имя издателя;
- период действия;
- открытый ключ субъекта;
- имя субъекта сертификата.

Под субъектом сертификата подразумевается область, которая осуществляет контроль секретного ключа, подходящий данному открытому

ключу. Присутствие необязательных полей свойственно, для сертификатов версий два и три, к необязательным полям сертификата принадлежит номер версии, два уникальных идентификатора и дополнения. Структура сертификата представлена на рисунке 2.

Поле "Версия" задает синтаксис сертификата, по умолчанию подразумевается первая версия сертификата. Если в поле версии указывается два, то сертификат включает только лишь уникальные идентификаторы, а если три, то в сертификат включаются и уникальные идентификаторы и дополнения, что свойственно для всех современных сертификатов. Сертификаты первой версии не включают неповторимые идентификаторы или дополнения.

Версия	Версия v1	Версия v2	Версия v3
Серийный номер			
Идентификатор алгоритма подписи			
Имя издателя			
Период действия (не ранее / не позднее)			
Имя субъекта			
Информация об открытом ключе субъекта			
Уникальный идентификатор издателя	Все версии		
Уникальный идентификатор субъекта			
Дополнения			
Подпись	Все версии		

Рисунок 2 – Структура сертификата

2 Анализ возможных решений по реализации инфраструктуры открытых ключей

Реализация PKI в Windows выполняется следующими элементами.

CryptoAPI[4].

Функции CryptoAPI гарантируют прикладным программам доступ к криптографическим способностям Windows. Но они считаются лишь «передаточным звеном» в сложной цепи обработки информации. Главную службу осуществляют тайные с глаза разработчика программного обеспечения функции, вступающие в специальные программные (либо программно-аппаратные) модули – провайдеры шифровальных услуг (CSP - Cryptographic Service Providers), либо криптопровайдеры. Программная часть криптопровайдера представляет собой dll-файл, подписанный Microsoft; периодически Windows проверяет цифровую подпись, что исключает возможность подмены криптопровайдера.

В зависимости от версии операторной системы структура определенных криптопровайдеров может значительно меняться. Но в каждом ПК с Windows возможно отыскать Microsoft Base Cryptographic Provider, принадлежащий к ранее известному нам типу PROV_RSA_FULL. Непосредственно с данным провайдером согласно умолчанию станут взаимодействовать все без исключения программы.

Использование шифровальных способностей Windows напоминает работу программы с графическим механизмом. Криптопровайдер аналогичен графическому драйверу: он способен гарантировать связь программного обеспечения с оборудованием (устройство чтения смарт-карт, аппаратные датчики случайных чисел и пр.). С целью вывода данных на графическое устройство приложение никак не обязано напрямую обращаться к драйверу - взамен этого необходимо получить у системы контекст устройства, с помощью которого и исполняются все без исключения действия. Это дает возможность прикладному разработчику программного обеспечения использовать графическое устройство, ничего не зная о его аппаратной реализации.

Конкретно, таким образом, с целью применения шифровальных функций приложение обращается к криптопровайдеру не напрямую, а посредством CryptoAPI. При этом сначала следует сделать запрос у системы контекст криптопровайдера.

Первым делом, выясним, какие же криптопровайдеры определены в системе. Для этого потребуются четыре функции CryptoAPI:

- CryptEnumProviders - возвращает название и вид i-го согласно порядку криптопровайдера в системе (перечисление наступает с нулевой отметки);

- CryptAcquireContext- осуществляет присоединение к криптопровайдеру с заданным типом и именованием и возвращает его дескриптор (контекст). При подключении мы будем передавать функции флаг CRYPT_VERIFYCONTEXT, предназначенный для извлечения контекста в отсутствии подключения к контейнеру ключей;

- CryptGetProvParam - возвращает значение подтвержденного параметра провайдера, к примеру, версии (второй параметр при призыве функции - PP_VERSION), типа осуществления (программный, аппаратный, смешанный - PP_IMPTYPE), поддерживаемых алгоритмов (PP_ENUMALGS). Перечень поддерживаемых алгоритмов при поддержке, данной функции способен быть приобретен соответствующим способом: при одном вызове функции возвращается сведения об одном алгоритме; при первом вызове функции необходимо передать значение флага CRYPT_FIRST, а при последующих флаг обязательно равен 0;

- CryptReleaseContext - освобождает дескриптор криптопровайдера.

После освоения информации о структуре CryptoAPI, возможно пользоваться ею в практических целях. Наверное, наиболее прогнозируемым воздействием криптографической подсистемы считается шифрование файлов - таким образом, для того чтобы только пользователь, знающий конкретный пароль, имел возможность получить к ним доступ.

С целью кодирования данных в CryptoAPI используются симметричные алгоритмы. Симметричность означает, то что с целью шифрования и расшифровки сведений применяется один и тот же ключ, знаменитый как шифрующей, таким образом и расшифровывающей стороне. При этом плохо выбранный ключ кодирования способен предоставить противнику возможность взломать шифр. По этой причине одной из функций шифровальной подсистемы должна являться генерация «хороших» ключей или случайным способом, или в основе некоторой информации, предоставляемой пользователем, к примеру, пароля.

В случае формирования ключа в основе пароля должно осуществляться последующее обязательное требование: при неоднократном повторении операции генерации ключа на одном и том же пароле обязаны получаться идентичные ключи. Ключ кодирования обладает, как правило, строго конкретную длину, характеризуемую применяемым алгоритмом, а протяженность пароля может являться свободной. В том числе и подсознательно понятно, то, что для однозначной генерации ключей необходимо привести различные пароли к некоторой общей форме. Это достигается с помощью хеширования.

Хешированием называется преобразование строки свободной длины в битовую последовательность фиксированной длины с предоставлением последующих условий:

- согласно хеш - значению нельзя возобновить начальное сообщение;
- практически нельзя отыскать два разных текста, предоставляющих похожие хеш - значимости (подобные условия называют коллизиями).

При соблюдении приведенных обстоятельств хеш-значение предназначается компактным цифровым отпечатком (дайджестом) сообщения. Имеется большое число алгоритмов хеширования. CryptoAPI поддерживает, к примеру, алгоритмы MD5 (MD - Message Digest) и SHA (Secure Hash Algorithm).

Таким образом, для того чтобы сформировать ключ шифрования в основании пароля, необходимо сначала получить хеш данного пароля. Для

этого необходимо сформировать с помощью CryptoAPI хеш-объект, воспользовавшись функцией CryptCreateHash (провайдер, ID_алгоритма, источник, флаги, хеш), которой необходимо передать дескриптор криптопровайдера (приобретенный с помощью CryptAcquireContext) и идентификатор алгоритма хеширования (другие характеристики могут являться нулями). В следствии получили дескриптор хеш-объекта. Данный объект, возможно, представить себе как черный ящик, который берет на себя всевозможные данные и «перемалывает» их, сохраняя изнутри себя только хеш-значение. Подать данные во вход хеш-объекта дает возможность функция CryptHashData (дескриптор, сведения, размер данных, флаги).

Непосредственно формирование ключа выполняет функция CryptDeriveKey (провайдер, ID_алгоритма, хеш-объект, флаги, ключ), что берет на себя хеш-объект в качестве начальных данных и строит оптимальный ключ для алгоритма кодирования, установленного собственным ID. Итогом станет дескриптор ключа, который возможно применять с целью кодирования.

Следует сосредоточить внимание, то, что при работе с CryptoAPI все время имеем дело не с самими объектами или их адресами, а с дескрипторами - целыми числами, характеризующими состояние предмета во внутренних таблицах криптопровайдера. Сами таблицы размещаются в оберегаемой сфере памяти, таким образом, то, что программы - «шпионы» никак не могут приобрести к ним доступ.

Методы шифрования, поддерживаемые CryptoAPI, возможно поделить на блочные и поточные:

- первоначальные обрабатывают данные сравнительно большими по объему блоками (к примеру, 64, 128 битов либо более);
- вторые - побитно (теоретически, в практике же - побайтно).

В случае если объем данных, доступных шифрованию, никак не кратный объему блока, в таком случае завершающий, неполный блок данных, станет, дополнен важным количеством случайных битов, вследствие чего объем зашифрованных данных способен несколько повыситься. Безусловно, при

применении поточных шифров объем данных при шифровании остается неизменным.

Кодирование выполняется функцией `CryptEncrypt` (ключ, хеш, финал, флаги, данные, `размер_данных`, `размер_буфера`):

- через параметр `ключ` передается дескриптору ключа кодирования;
- параметр `хеш` применяется, в случае если в то же время необходимо определить хеш-значение шифруемого текста;
- параметр `финал` равный `true`, в случае если зашифровываемый блок текста завершающий либо единственный (шифрование возможно реализовывать компонентами, вызывая функцию `CryptEncrypt` несколько раз);
- значение флага обязано быть нулевым;
- параметр `данные` предполагает собою адрес буфера, в котором, при вызове функции, располагается первоначальный текст, а, по окончании, деятельности функции, зашифрованный;
- следующий параметр, в соответствии с этим, представляет размер входных либо выходящих сведений;
- последний параметр задает размер буфера.

С целью расшифровки сведений применяется функция `CryptDecrypt` (`ключ`, `хеш`, `финал`, `флаги`, `данные`, `размер_данных`), выделяющаяся с шифрующей функции только лишь тем, что размер буфера указывать не необходимо: так как размер данных при расшифровке может только снизиться, отведенного под них буфера безусловно станет достаточным. Текст никак не уместится в буфере, возникнет ошибка.

Описание `CryptoAPI`.

Криптографические функции поделены на два уровня `CryptoAPI 2.0` и `CryptoAPI 1.0`.

Уровень `CryptoAPI 2.0` осуществляет «крупные» шифровальные функции. Как правило, непосредственно данные функции вызываются из конечных приложений. Комплект этих функций призван удовлетворить все без исключения требования в области криптографии со стороны разработчика

программного обеспечения. Он сильно стимулирует процесс осуществления функций конечных приложений сопряженных с криптографией и защищенной передачей данных.

CryptoAPI 1.0 - это так именуемый уровень шифровальных примитивов. Это определенный комплект функций реализующих основные шифровальные операции - шифрование/расшифрование данных, хеширование данных, генерация и контроль электронной цифровой подписи (ЭЦП). Данные функции в свою очередь вызываются с функций уровня CryptoAPI 2.0.

Функции данного уровня реализуются в виде криптопровайдера (Cryptographic Service Provider - CSP). С точки зрения операционной системы криптопровайдер представляется как библиотека шифровальных функций с стандартизованным интерфейсом. Интерфейс криптопровайдера стандартизирован с целью абстрагирования конечного приложения с определенной реализации шифровальных функций. Приложение и функции CryptoAPI 2.0 стимулируют не сами реализации шифровальных функций, а переходные, «абстрактные» функции CryptoAPI 1.0, которые экспортируются системной библиотекой advapi32.dll.

На рисунке 3 изображена схема вызова функций CryptoAPI 1.0, и последующий вызов криптопровайдера.

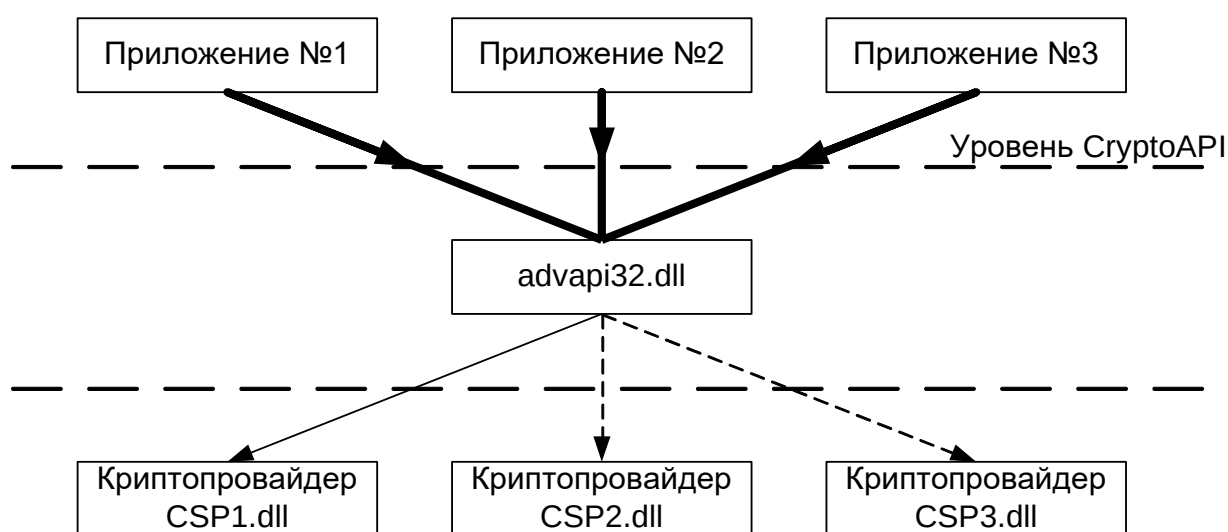


Рисунок 3 - Схема вызова функций CryptoAPI 1.0

CryptoSPI (cryptographic service provider interface) регламентирует состав характеристик вызова функций криптопровайдера, и кроме того устанавливает определенный обычный набор флагов и значений характеристик вызова. Кроме того регламентируется набор возвращаемых ошибок

Криптопровайдер Crypton CSP[5].

Криптопровайдер Crypton CSP специализирован для компании допуска к российским шифровальным алгоритмам, реализованным в Crypton CSP и в аппаратных и программных шифраторах серии КРИПТОН фирмы "АНКАД", с поддержкой стандартных интерфейсов CryptoAPI, CAPICOM, Certificate Enrollment Control.

Crypton CSP выполнен в соответствии с шифровальным интерфейсом Microsoft - Cryptographic Service Provider (CSP). Вследствие чего, многочисленные приложения Microsoft и сторонних создателей, использующие отмеченные интерфейсы, имеют все шансы работать с российскими криптоалгоритмами.

Crypton CSP призван найти решение этих проблемы, так как он дает возможность применять в стандартных продуктах Microsoft (Office, Outlook и других) стойкие российские криптоалгоритмы ГОСТ с целью кодирования и подписи электронной информации, и отступить от интегрированных в Microsoft Office не внушающих доверие шифровальных алгоритмов с небольшой длиной ключа. Помимо этого, Crypton CSP дает возможность развернуть полноценную инфраструктуру открытых ключей (PKI - Public Key Infrastructure), кроме того основанную на проверенных российских криптоалгоритмах. Криптопровайдер может свободно встраивать в любые прочие защищенные приложения для Windows.

Криптографические преобразования выполняются в Crypton CSP на уровне ядра операционной системы (при применении Crypton Emulator) либо на аппаратном уровне (при применении шифраторов серии КРИПТОН). Такого рода подход почти ликвидирует вероятность постороннего вмешательства в процесс шифрования.

Криптопровайдер Crypton CSP предъявляет следующие программно-аппаратные требования:

- ОС Windows 2000/XP/2003; для установки сервера сертификатов - серверная ОС Windows 2000/2003;
- присутствие устройства криптографической охраны сведений серии КРИПТОН либо его программного драйвера-эмулятора для Windows - Crypton Emulator, обладающих сертификаты ФСБ;
- наличие Crypton API для Windows.

Ключевые носители - дискеты, электронные брелоки Touch Memory, сменные носители с интерфейсом USB, смарт-карты.

Криптопровайдер Crypton CSP может применяться для:

- формирования ключей шифрования и ключей электронной цифровой подписи - как на программном, так и на аппаратном уровне, с помощью сертифицированного ФСБ датчика случайных чисел (ДСЧ);
- формирования и проверки электронной цифровой подписи (ЭЦП) в соответствии с российскими стандартами ГОСТ Р 34.10-2012[6] с длиной закрытого ключа - 256 бит и открытого ключа - 1024 бита;
- шифрования и имитозащиты в соответствии с ГОСТ 28147-89[7] для обеспечения конфиденциальности и контролирования целостности информации, с длиной симметричных ключей - 256 бит.

Использование криптопровайдера дает возможность использовать российские алгоритмы охраны информации в разных продуктах, поддерживающих интерфейс Microsoft.

В их число входит:

- удостоверяющий центр Microsoft Certification Authority;
- почтовый клиент Microsoft Outlook Express;
- диспетчер контактов Microsoft Outlook;
- приложения Microsoft Office (MS Word, MS Excel, MS Power Point);
- приложение контроля целостности ПО Microsoft Authenticode;
- почтовый клиент The Bat! (разработчик - компания "RITLABS");

– система защиты конфиденциальной информации на ПК Secret Disk NG (разработчик - компания "ALADDIN Software Security R.D.").

Сертификация ключей ЭЦП обеспечивается посредством Microsoft Certification Authority.

ViPNet Криптопровайдер[8].

ViPNet Криптопровайдер создан в соответствии с криптографическим интерфейсом компании Microsoft - Cryptographic Service Provider (CSP). ViPNet Криптопровайдер вступает в структура СКЗИ «Домен-КС2» и СКЗИ «Домен-КМ». При поставке в составе СКЗИ «Домен-КМ» существует несколько отличительных черт и ограничений в применении криптопровайдера, которые отображены в данном документе и в документации в СКЗИ «Домен-КМ».

ViPNet Криптопровайдер выполнен в виде динамической библиотеки ITCCSP и предназначен для использования:

- на IBM PC компьютерах с процессорами не ниже Intel Pentium II (или их аналогов) под управлением операционных систем Windows 98SE, Windows NT, Windows 2000, Windows 2003 Server, Windows XP;

- на процессорах не ниже i386 под управлением операционных систем RedHat Linux (7.2, 7.3, 9.0), RedHat Enterprise Linux 3, Fedora Core 3, Mandrake Linux (9.0, 9.1, 10.1), Debian Woody 3.0, Linux SuSe (9.0, 9.1, 9.2, 10.0, SuSE Linux Enterprise Server 9), Linux Slackware 10.2 (только ядро 2.4.31).

Криптопровайдер обеспечивает:

- создание ЭЦП по алгоритмам ГОСТ Р 34.10-2012[6];
- вычисление и проверку ЭЦП по алгоритмам ГОСТ Р 34.10-2012[6];
- хеширование данных в соответствии с алгоритмом ГОСТ Р 34.11-2012[9];
- шифрование и имитозащиты данных в соответствии с алгоритмом ГОСТ 28147-89[7];
- поддержку различных устройств для хранения ключей (eToken, iKey, Smard Card и т.п.);

– хранение сертификатов открытых ключей непосредственно в ключевом контейнере.

ViPNet Криптопровайдер поддерживает параметры и идентификацию алгоритмов и параметров, обеспечивающие совместимую работу с криптопровайдерами иных производителей ("КриптоПРО", "СигналКом").

В состав криптопровайдера входят вспомогательные модули, обеспечивающие вызов криптографических функций провайдера через интерфейс Microsoft CryptoAPI 2.0 при работе под управлением ОС , Windows 2000, Windows 2003 Server, Windows XP. Это позволяет обеспечить вызов криптографических функций из стандартных приложений Microsoft (Outlook, Outlook Express и др.).

Криптопровайдер входит в состав программного обеспечения ViPNet Cryptoservice (ViPNet Client). Модули ПО ViPNet Cryptoservice обеспечивают поддержку ключевой структуры ViPNet, предоставляют интерфейс взаимодействия с Ключевым и Удостоверяющим Центром ViPNet.

Криптопровайдер Signal-COM CSP[2].

Фирма «Сигнал-КОМОК» приступила к продвижению на рынок собственной новейшей разработки - криптопровайдера Signal-COM CSP, реализующего российские алгоритмы криптопреобразований, в том числе алгоритм формирования/проверки ЭЦП, основанный на эллиптических кривых (ГОСТ Р 34.10-2001).

Криптопровайдер Signal-COM CSP выполнен в соответствии с технологией Cryptographic Service Provider (CSP) и защищен цифровой подписью фирмы Microsoft, что дает возможность использовать во многих её приложениях (Microsoft Certification Authority, Microsoft Outlook Express, Microsoft Outlook, Microsoft Authenticode и др.) российские алгоритмы кодирования и ЭЦП.

С помощью открытых интерфейсов Microsoft CryptoAPI 2.0 и CAPICOM 1.0 Signal-COM CSP способен свободно интегрироваться в ранее имеющиеся

либо формируемые защищенные приложения самостоятельных разработчиков, функционирующие под управлением Windows 9x/Me/NT/2000/XP.

Криптопровайдер Signal-COM CSP обеспечивает возможность организации инфраструктуры открытых ключей (Public Key Infrastructure) в соответствии с международными рекомендациями. В качестве криптоядра, отвечающего за осуществление абсолютно всех криптографических действий, в Signal-COM CSP применяется способ криптографической защиты данных (СКЗИ) производства «Сигнал-КОМ», сертифицированное ФАПСИ на соответствие «Требованиям к средствам шифровальной защиты секретной данных» согласно уровням KC1 и KC2.

GNU Privacy Guard[10].

GNU Privacy Guard (GnuPG либо GPG) - независимая альтернатива комплекту криптографического программного обеспечения PGP, выданная под лицензией GNU General Public License. Считается частью проекта GNU, получила гранты с германского правительства. GnuPG целиком совместим со стандартом IETF OpenPGP.

Текущие версии GnuPG имеют все шансы взаимодействовать с PGP и иными OpenPGP-совместимыми системами в порядке совместимости. GnuPG дает возможность зашифровывать и подписывать сведения в целях безопасного сохранения и передачи данных.

GnuPG - программное обеспечение, зачастую включаемое в свободные операционные системы, такие как GNU/Linux, FreeBSD, OpenBSD и NetBSD.

Хотя главным интерфейсом считается инструктивная строка, существуют разные дополнительные интерфейсы, к примеру, GnuPG включен в KMail и Evolution (графические клиенты электронной почты). Система плагинов целиком совместим со стандартом OpenPGP.

С поддержкой расширения Enigmail GnuPG функционирует в почтовом клиенте Mozilla Thunderbird с целью кодирования и аутентификации сообщений. Кроме того поддержка GnuPG существует в почтовых клиентах Mutt, The Bat! и Gnus.

Функцию GnuPG в шифровании и подписи электронных писем, возможно, применять с почтовыми клиентами, функционирующими по протоколам POP3, IMAP4, SMTP, при помощи программы GPGrelay.

Также, применяя дополнительные плагины GnuPG либо SecureIM, возможно достигнуть шифрования сообщений с помощью GnuPG в клиенте сетей мгновенных сообщений Miranda IM.

GnuPG поддерживается Jabber-клиентами Psi и Gajim.

Вплоть до 7 июня 2010 года проводилась создание FireGPG - дополнения для Mozilla Firefox, что позволило дополнить в данный интернет-браузер поддержку GnuPG. Кроме других функций оно позволило интегрировать GnuPG в Gmail. Из-за того, что web интерфейс Gmail регулярно изменялся, автор плагина прекратил вначале поддерживать шифрование Gmail почты, а потом и всего плагина. Данным плагином FireGPG воспользовалось приблизительно тридцать тысяч человек.

Другой плагин для браузеров WebPG существует в версиях для Mozilla Firefox (SeaMonkey, Thunderbird) и Google Chrome (Chromium). Поддерживает функции: шифрование, расшифровывание, цифровая подпись, управление ключами. Имеется экспериментальная интеграция с Google почтой.

Иной действующий плагин позволяющий применять криптозащиту (шифрование и ЭЦП) в сообщении на странице в Сети интернет для браузера Google Chrome это GPG4Browsers. Данный плагин сначала выпущен германской фирмой Recurity Labs, однако потом отдан в отдельный проект OpenPGP.js.

Использование плагинов в браузерах дает возможность гарантировать принадлежность этого либо другого сообщения, тому либо другому человеку (посредством ЭЦП) либо дает возможность прочитать сообщение, находящееся в публичном доступе кому-то одному (для кого зашифровано это сообщение). В том числе допустимо применение данных плагинов в социальных сетях либо электронной почте.

Особенности GNU Privacy Guard:

- полная альтернатива PGP;
- не использует патентованные алгоритмы;
- распространяется под GNU General Public License;
- полная реализация OpenPGP (RFC2440);
- расшифрование и аутентификация сообщений, созданных с помощью PGP 5, 6 и 7;
- поддержка электронной подписи с помощью алгоритмов ElGamal, DSA, RSA и хеш-функций MD5, SHA-1, SHA-2, RIPE-MD-160 и TIGER;
- работа с асимметричным шифрованием ElGamal и RSA (длина ключа от 1024 до 4096 бит);
- поддержка блочных алгоритмов симметричного шифрования AES, CAST5, 3DES, Twofish, Blowfish, Camellia, а также IDEA с помощью плагина;
- лёгкая реализация новых алгоритмов с помощью дополнительных модулей;
- поддержка просроченных ключей и подписей;
- интегрированная поддержка НКР-серверов ключей.

Программно-аппаратный комплекс ШИПКА[11].

Программно-аппаратный комплекс ШИПКА представляет собой USB-устройство, главным назначением которого являются:

- безопасное хранение криптографических ключей и секретных ключей ключевой пары в памяти устройства;
- аппаратное осуществление криптографических операций.

Как ранее рассказывалось выше, криптопровайдеры реализуются в виде динамически загружаемых библиотек, при этом они обладают цифровой подписью и при работе с криптопровайдером в системных библиотеках проверяется целостность его DLL. Однако контроль целостности DLL криптопровайдера выполняется программно, то что предоставляет вероятность воздействовать программным посредством в процесс исполнения проверки, а, следовательно, и в развитие алгоритмов. Таким образом как в комплексе

ШИПКА криптографические алгоритмы выполнены аппаратно, в таком случае, нельзя привнести изменения в его код, в таком случае имеется оказать влияние в реализацию какого-либо алгоритма.

Выявив отличия криптопровайдера, реализованного на основе программно-аппаратного комплекса ШИПКА и программно реализованного криптопровайдера "Microsoft Base Cryptographic Provider", который является одним из основных криптопровайдеров Windows. Эти криптопровайдеры имеют одинаковый тип - PROV_RSA_FULL, то есть реализуют одинаковый набор алгоритмов вычисления значения хеш-функции, генерации криптографических ключей и ключевых пар, шифрования, вычисления подписи. Таким образом, оба криптопровайдера, имея на входе каждой функции одинаковые параметры и значения входных переменных, на выходе дают так же одинаковые значения. Однако, в связи с различным способом реализации этих криптопровайдеров, криптопровайдер для комплекса ШИПКА имеет следующие особенности реализации по сравнению с криптопровайдером "Microsoft Base Cryptographic Provider".

Все выше представленные криптопровайдеры платные, или для их нормального функционирования требуется доступ к сети интернет. В лабораторном практикуме не будет использоваться сеть интернет, поэтому будут использоваться бесплатные программные продукты, позволяющие выполнять основные функции УЦ:

- формирование собственного секретного ключа; если является головным удостоверяющий центр, то издает и подписывает свой сертификат, называемый самоизданным или самоподписанным;
- выпускать (то есть создает и подписывает) сертификаты открытых ключей подчиненных удостоверяющих центров и конечных субъектов PKI;
- поддерживать реестр сертификатов (базу всех изданных сертификатов) и формировать списки САС с регулярностью, определенной регламентом удостоверяющего центра;
- публиковать информацию о статусе сертификатов и списков САС.

Для выполнения лабораторного практикума выбран бесплатный программный продукт OpenSSL.

OpenSSL - это система охраны и сертификации сведений, наименование SSL переводится, равно как система безопасных сокетов (secure socket layer). OpenSSL применяется почти абсолютно всеми сетевыми серверами с целью охраны передаваемой информацией. Имеется программное API SSL(SSLEAY), позволяющее формировать безопасные сокет с шифрацией передаваемых сведений в своих разработках.

OpenSSL поддерживает весьма большое количество разных стандартов сертификации, шифрования, хеширования, в таком случае использование этой команды достаточно сложно. Внутри OpenSSL имеются отдельные элементы, отвечающие за то или иное действие, с целью извлечения списка доступных компонентов, возможно, вызвать openssl с параметрами list-standart-commands. Возможно кроме того получить перечень доступных алгоритмов хеширования (list-message-digest-commands) и алгоритмов кодирования (list-cipher-commands).

3 Разработка архитектуры лабораторного стенда для изучения инфраструктуры открытых ключей

Поскольку PKI подразумевает использование в архитектуре клиент-сервер, то лабораторный стенд должен включать в себя как минимум две рабочие станции.

В лабораторном стенде для изучения инфраструктуры открытых ключей используются две виртуальные машины с установленными ОС:

- centOS, выступает в качестве сервера;
- windows 7, используется в качестве клиента.

Для проведения лабораторного практикума необходимо, чтобы в лабораторном стенде были реализованы компоненты для выполнения основных действий.

Должен быть УЦ, установленный на отдельной виртуальной машине, который позволяет:

- изготавливать сертификаты электронных цифровых подписей;
- создавать ключи электронных цифровых подписей по обращению участников информационных систем;
- приостанавливать и возобновлять действия сертификатов электронных цифровых подписей, а также аннулирование их;
- осуществлять подтверждение подлинности электронной цифровой подписи в электронном документе в отношении выданных им сертификатов ключей подписей.

Должно быть клиентское приложение, которое позволяет:

- устанавливать сертификаты;
- подписывать документы электронной цифровой подписью;
- проводить проверку подлинности подписи.

Диаграмма вариантов использования приведена на рисунке 4.

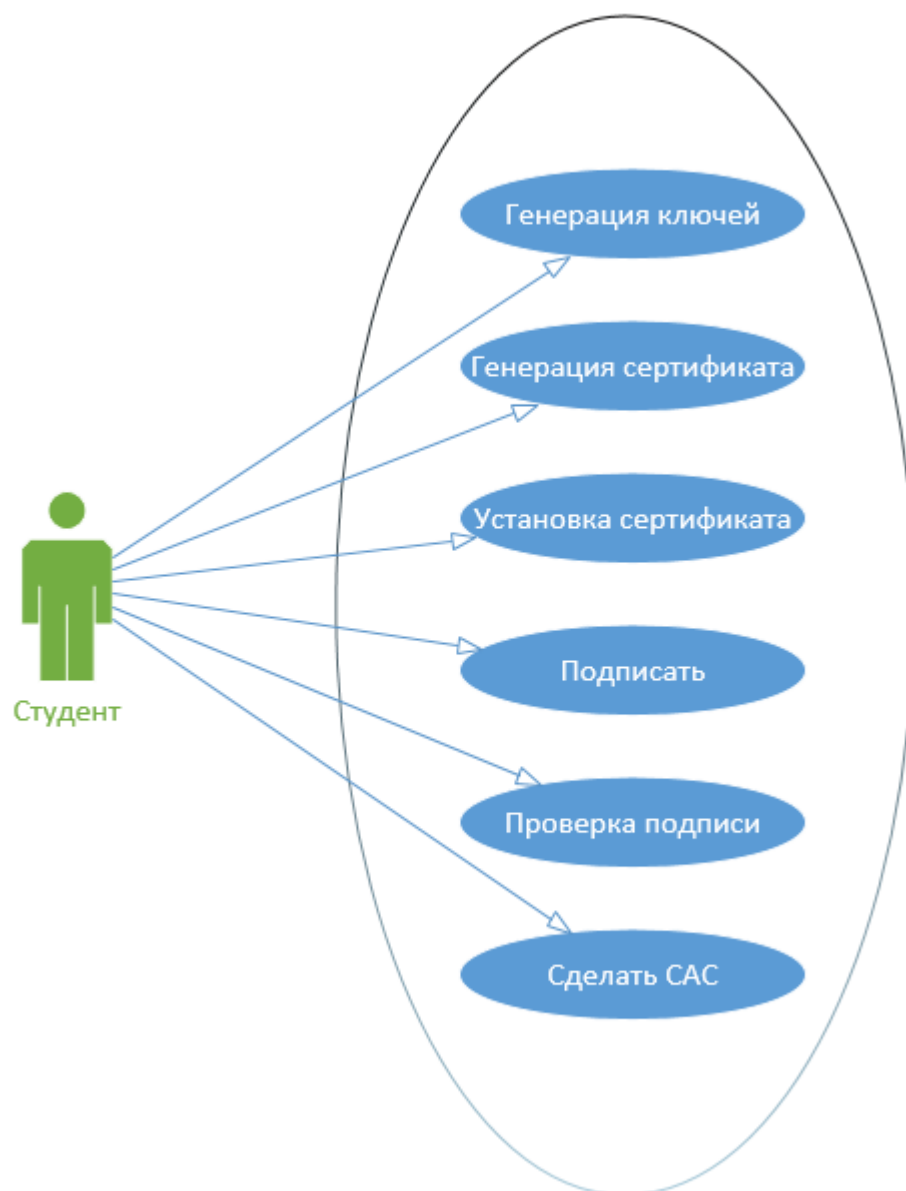


Рисунок 4 – Диаграмма вариантов использования

4 Разработка лабораторного стенда для проведения лабораторного практикума

В роли УЦ будет использоваться программный продукт OpenSSL, который позволяет выполнять следующие задачи:

- создавать и управлять ключами RSA и DSA - команды `rsa`, `dsa`, `dsaparam`;
- создавать сертификаты формата `x509`, запросы на сертификацию, восстановление - команды `x509`, `req`, `verify`, `ca`, `crl`, `pkcs12`, `pkcs7`;
- зашифровывать данные с помощью симметрического или асимметрического шифрования - команды `enc`, `rsautl`;
- высчитывать хеши различных типов - команда `dgst`;
- работа с S/MIME - команда `s/mime`;
- проверка работы серверов и клиентов `ssl` - команды `s_client`, `s_server`.

В качестве клиентского приложения используется программный продукт OpenOffice, который позволяет использовать следующие возможности:

- устанавливать сертификаты;
- подписывать документы электронной цифровой подписью;
- проводить проверку подлинности подписи.

Для работы с сертификатами используется стандартное средство ОС Windows 7 – диспетчер сертификатов "`certmgr.exe`".

Диспетчер сертификатов (`Certmgr.exe`) предназначен для управления сертификатами, списками доверия сертификатов (CTL) и списками отзыва сертификатов (CRL). Диспетчер сертификатов позволяет выполнять следующие основные функции:

- отображать сведения о сертификатах, CTL и CRL на консоли;
- добавлять сертификаты, CTL и CRL в хранилище сертификатов;
- удалять сертификаты, CTL и CRL из хранилища сертификатов;
- сохранять в файл сертификата X.509, CTL или CRL из хранилища сертификатов.

Программа Certmgr.exe работает с двумя типами хранилищ сертификатов, системным и StoreFile. Указывать тип хранилища необязательно, поскольку программа Cedrtmgr.exe может автоматически определить тип хранилища и выполнить соответствующие действия.

При запуске программы Certmgr.exe без параметров выполняется оснастка "certmgr.msc" с графическим интерфейсом пользователя, облегчающим управление сертификатами, что также можно сделать из командной строки. В графическом интерфейсе пользователя имеется мастер импорта, копирующий сертификаты, CTL и CRL с диска в хранилище сертификатов. Графический интерфейс программы "Certmgr.exe" представлен на рисунке 5.

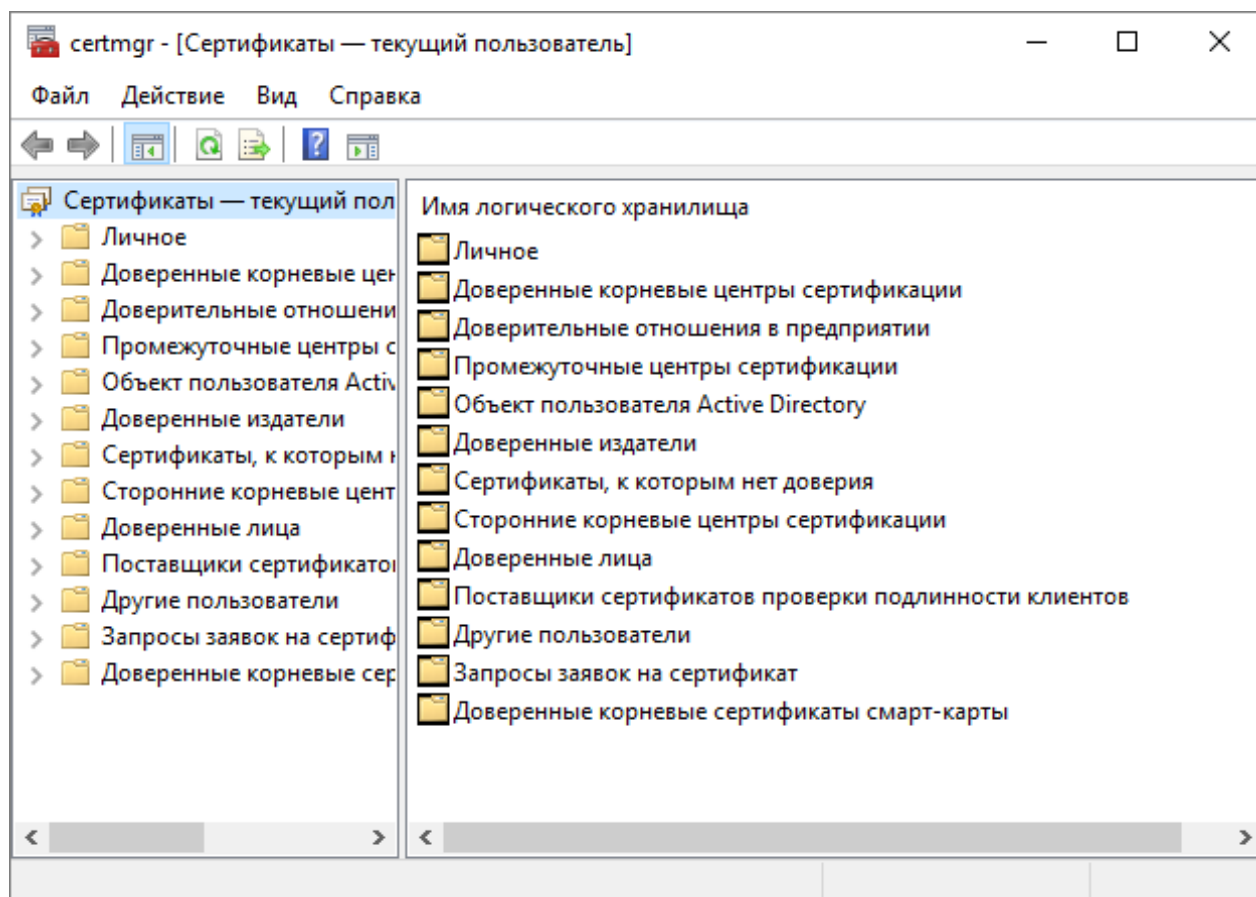


Рисунок 5 - Графический интерфейс программы Certmgr.exe

Диаграмма компонентов разработанного лабораторного стенда приведена на рисунке 6.

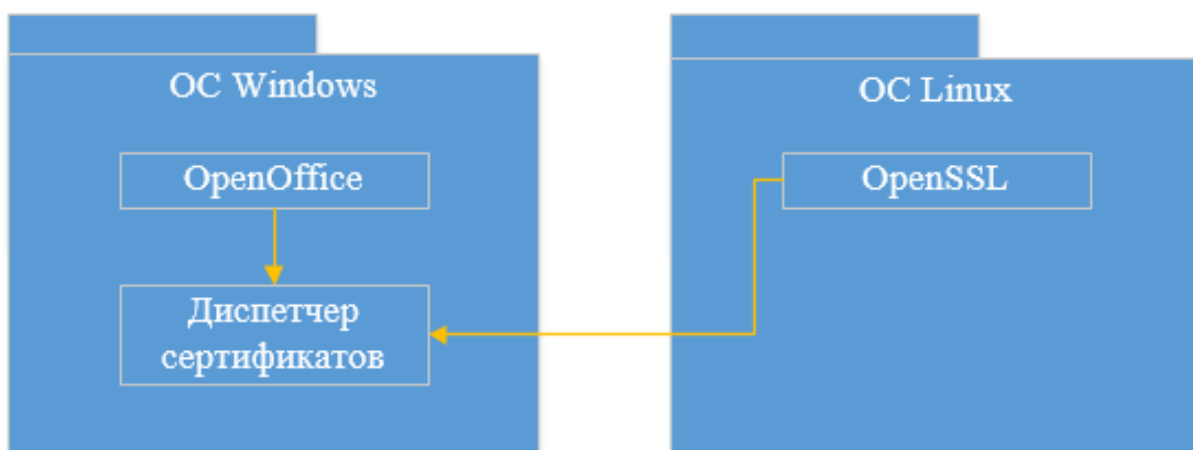


Рисунок 6 – Диаграмма компонентов лабораторного стенда для проведения лабораторного практикума

5 Разработка лабораторного практикума для изучения инфраструктуры открытых ключей

5.1 Изучаемые вопросы

В лабораторном практикуме по изучению инфраструктуры открытых ключей должны быть рассмотрены следующие вопросы:

- создание ключей;
- создание сертификатов;
- установка сертификатов в операционную систему;
- просмотр свойств установленных сертификатов;
- использование сертификатов для формирования электронной подписи документов;
- проверка электронной подписи документа;
- проверка электронной подписи документа при истекшем сроке действия сертификата;
- формирование списка отозванных сертификатов;
- проверка электронной подписи документа с помощью отозванного сертификата.

5.2 Создание ключей

Для выполнения лабораторного практикума необходимо сгенерировать две пары ключей по алгоритму RSA.

Первая пара - открытый и секретный ключи удостоверяющего центра. При генерации секретного ключа удостоверяющего центра необходимо дважды ввести проверочную фразу для генерации ключа. Далее необходимо сгенерировать секретный ключ сервера и файл запроса на получение сертификата сервера от центра сертификации, на основании секретного ключа удостоверяющего центра формируем открытый ключ, потребуется заполнить поля «Country Name», «State or Province Name», «Locality Name», «Organization Name», «Organizational Unit Name», «Email Address», «Common Name» .

Вторая пара - открытый и секретный ключи пользователя. Далее необходимо сгенерировать секретный ключ сервера и файл запроса на

получение сертификата сервера от центра сертификации, потребуется заполнить поля «Country Name», «State or Province Name», «Locality Name», «Organization Name», «Organizational Unit Name», «Email Address», «Common Name».

5.3 Создание сертификатов

Для выполнения лабораторного практикума необходимо сгенерировать два сертификата.

Первый сертификат проверки электронной подписи УЦ - самоподписанный сертификат на секретный ключ УЦ. При этом в строке «Enter pass phrase for ca.key:» необходимо указать парольную фразу, использованную при генерировании приватного ключа центра авторизации.

Второй сертификат пользователя - проверка электронной подписи пользователя, подписанный секретным ключом УЦ. Далее необходимо соединить сгенерированные ранее закрытый и открытый ключ сервера в один ключевой файл.

Значения соответствующих полей приведены в варианте задания.

В ходе выполнения лабораторного практикума, при генерации ключей и сертификатов необходимо установить срок действия сертификатов и ключей равным одному году (days 365).

В данном лабораторном практикуме для генерации ключей и сертификатов используется библиотека OpenSSL.

5.4 Установка сертификатов в операционную систему

Для выполнения лабораторного практикума необходимо скопировать сертификат проверки электронной подписи пользователя и его секретный ключ на виртуальную машину Windows 7 и сертификат удостоверяющего центра.

Для того чтобы установить сертификат в ОС Windows 7 потребуется воспользоваться диспетчером сертификатов «certmgr.exe».

5.5 Просмотр свойств установленных сертификатов

Для просмотра свойств установленных сертификатов используется диспетчер сертификатов. Сведения, которые нужно просмотреть:

- серийный номер сертификата;
- идентификатор алгоритма подписи;
- имя издателя;
- период действия;
- открытый ключ субъекта;
- имя субъекта сертификата.

5.6 Использование сертификатов для формирования электронной подписи документов

Для формирования электронной подписи документа в данном лабораторном практикуме требуется:

- создать электронный документ;
- ввести текст;
- сохранить документ;
- подписать документ секретным ключом пользователя.

В данном лабораторном практикуме для формирования электронной подписи используется OpenOffice. Текст, для заполнения документа, приведен в варианте задания.

5.7 Проверка электронной подписи документа

На данном этапе лабораторного практикума необходимо проверить правильность электронной подписи документа, выполненной на предыдущем этапе, определить результат проверки. Проверить подтверждение подписи документа либо не прохождение подписи документа.

В данном лабораторном практикуме для проверки электронной подписи используется бесплатное программное обеспечение OpenOffice.

Внести изменения в текст документа и выполнить проверку электронной подписи, определить результат проверки. Результату проверки предоставить в виде отчета о проделанной работе.

5.8 Проверка электронной подписи документа при истекшем сроке действия сертификата

Для проверки электронной подписи документа при истекшем сроке действия сертификата потребуется созданный ранее подписанный документ.

Для проверки электронной подписи документа при истекшем сроке действия сертификата потребуется изменить системное время виртуальной машины с установленной ОС Windows7 на 365 дней (один год) вперед. Так как при указании действия ключа и сертификата было указано время действия равным одному году. Из-за смены времени на виртуальной машине, сертификат будет не действителен про времени действия.

После изменения времени, проверить состояние подписи документа. Результаты проведения операций по проверки электронной подписи документа при истекшем сроке действия включить в отчет о проделанной работе.

По окончании поверки электронной подписи документа при истекшем времени требуется вернуть время виртуальной машины Windows 7 обратно.

5.9 Формирование списка отозванных сертификатов

Для формирования списка аннулированных (отозванных) сертификатов потребуется:

- сгенерировать список отзыва;
- посмотреть список отзыва;
- отозвать ранее выпущенный сертификат.

Установить список аннулированных (отозванных) сертификатов на виртуальную машину с установленной ОС Windows 7.

Издаваемые УЦ списки аннулированных (отозванных) сертификатов должны соответствовать рекомендациям. Все поля и дополнения, включаемые в список аннулированных (отозванных) сертификатов должны быть заполнены.

Выполнить проверку электронной подписи документа, определить результат проверки.

Результаты проверки включить в отчет о проделанной работе.

Порядок проведения все описанных выше действий представлен в виде диаграммы на рисунке 7.

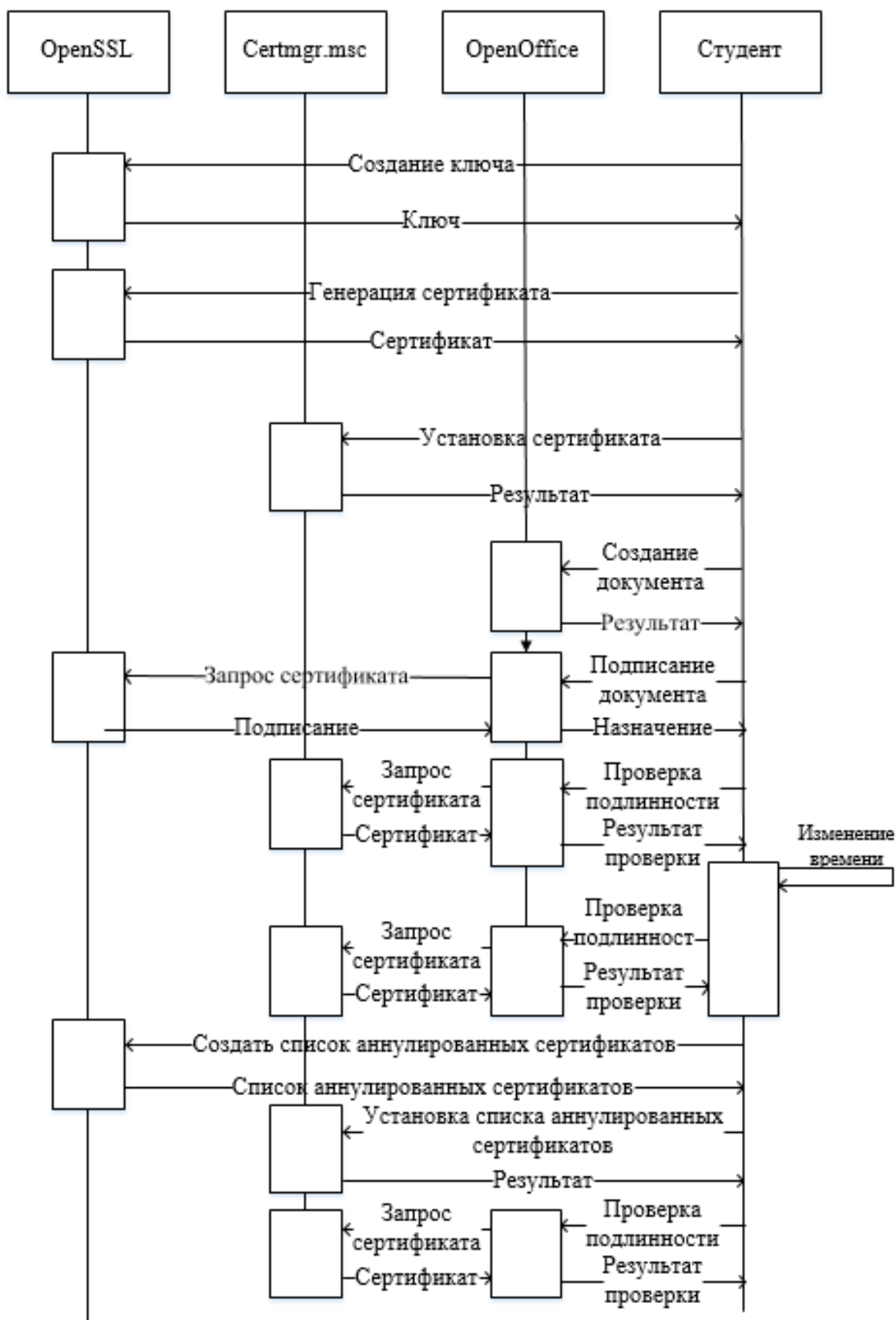


Рисунок 7 – Диаграмма последовательности действий

6 Разработка рекомендаций для проведения лабораторного практикума

6.1 Создание ключей

Перед началом генерирования ключей и сертификатов необходимо создать папку «cert», где они будут храниться. Это можно выполнить с помощью следующих действий:

```
[root@localhost ~]# mkdir ./cert
```

```
[root@localhost ~]# cd ./cert
```

```
[root@localhost cert]#
```

Далее с помощью библиотеки «openssl» необходимо сгенерировать закрытый ключ центра сертификации (CA – Certification Authority):

```
[root@localhost cert]# openssl genrsa -des3 -out ca.key 4096
```

При этом дважды необходимо ввести проверочную фразу для генерируемого ключа.

Далее на основании сгенерированного закрытого ключа центра сертификации «ca.key» формируется сертификат центра сертификации CA (его открытый ключ):

```
[root@localhost cert]# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

При указании «Country Name», «State or Province Name», «Locality Name», «Organization Name», «Organizational Unit Name», «Email Address», «Common Name»

Далее необходимо сгенерировать закрытый ключ сервера:

```
[root@localhost cert]# openssl genrsa -out key.pem 1024
```

и файл запроса на получение сертификата сервера от центра сертификации:

```
[root@localhost cert]# openssl req -new -key key.pem -out req.csr
```

При указании «Country Name», «State or Province Name», «Locality Name», «Organization Name», «Organizational Unit Name», «Email Address», «Common Name»

Значения соответствующих полей приведены в варианте задания.

6.2 Создание сертификатов

После этого необходимо сгенерировать сертификат сервера (его открытый ключ) на основе файла запроса на получение сертификата и подписать его с помощью закрытого ключа центра сертификации и его сертификата:

```
[root@localhost cert]# openssl x509 -req -days 365 -in req.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out cert.crt
```

При этом в строке «Enter pass phrase for ca.key:» необходимо указать парольную фразу, использованную при генерировании приватного ключа центра авторизации.

После указанных операций содержимое папки «/etc/crt» должно быть следующим:

```
[root@localhost cert]# ll
total 20
-rw-r--r--. 1 root root 1952 май 29 16:22 ca.crt
-rw-r--r--. 1 root root 3311 май 29 16:22 ca.key
-rw-r--r--. 1 root root 1298 май 29 16:25 cert.crt
-rw-r--r--. 1 root root 887 май 29 16:24 key.pem
-rw-r--r--. 1 root root 680 май 29 16:25 req.csr
```

Далее необходимо соединить сгенерированные закрытый и открытый ключ сервера в один ключевой файл с помощью следующих команд:

```
[root@localhost cert]# cat ./key.pem > pgusert.pem
[root@localhost cert]# cat ./cert.crt >> pgusert.pem
[root@localhost cert]#
```

После указанных действий будет сформирован сертификат сервера «pgusert.pem», хранящийся на нем по адресу «/etc/ cert». Пример содержимого файла «pgusert.pem» может быть следующим:

```
[root@localhost cert]# cat pgusert.pem
-----BEGIN RSA PRIVATE KEY-----
```

```
MIICXAIBAAKBgQDBJEinU/H+0GM9JMcDFA3zvNHEW0tGY5k00eSA  
mIuS7bEUD339
```

```
...
```

```
BQCnTPRYkcnaoAQyv2VLpoCMwJyvXpat9FeLSG8DG8A=
```

```
-----END RSA PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
```

```
MIIDIDCCAXwCAQEwDQYJKoZIhvcNAQEFBQAwwUjELMAkGA1UEBh  
MCWFgxFtATBgNV
```

```
...
```

```
PnaK7INvL2c=
```

```
-----END CERTIFICATE-----
```

```
[root@localhost cert]#
```

Получен сертификат и секретный ключ в формате PEM (Файл сертификата, закодированный в Base64, к примеру, сертификат PEM X509. Применяется для безопасной идентификации на веб-сайте. Обычно импортируется из веб-сервера Apache на основе Unix и совместим с приложениями OpenSSL), нужно воспользоваться утилитой openssl для преобразования его в PKCS#12. В окне консоли необходимо ввести:

```
openssl pkcs12 -in pgusert.pem -export -out pgusert.p12
```

При этом дважды необходимо ввести проверочную фразу для генерируемого ключа.

6.3 Установка сертификатов в операционную систему

Для выполнения лабораторного практикума необходимо скопировать сертификат проверки электронной подписи пользователя и его секретный ключ на виртуальную машину Windows 7 и сертификат удостоверяющего центра.

Для этого можно воспользоваться функцией «общих папок».

Далее необходимо установить сертификат в систему с помощью менеджера сертификатов, вызвать которой можно с помощью системной команды «certmgr.msc» Для этого необходимо перейти в меню «Пуск – Поиск - certmgr.msc». Вызов менеджера сертификатов представлен на рисунке 8.

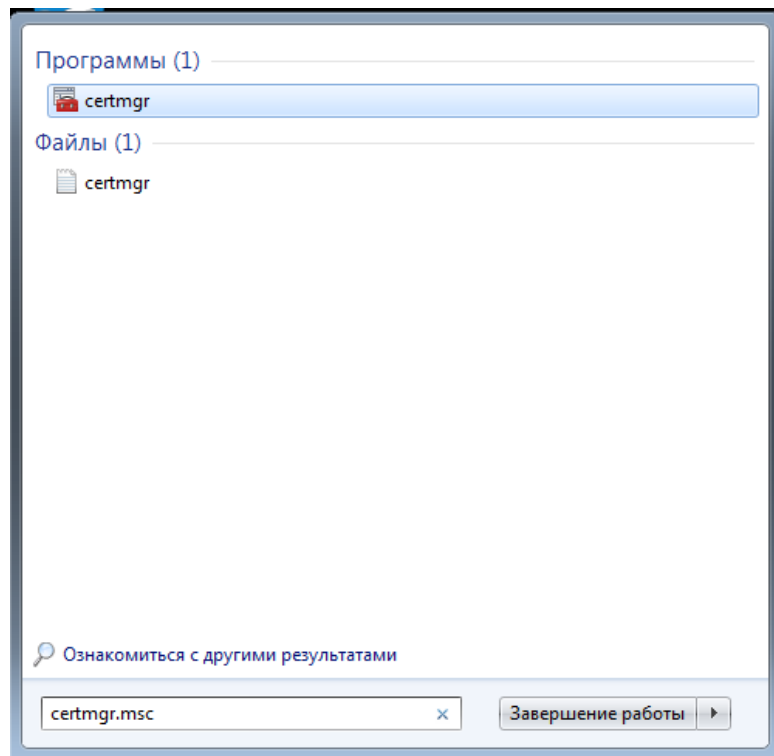


Рисунок 8 - Вызов менеджера сертификатов

Для добавления пользовательского сертификата необходимо выбрать в списке «Доверенные корневые центры сертификации» подписок «Сертификаты», в меню «Действие» выбрать опцию «Все задачи – Импорт...». Указанная операция представлена на рисунке 9.

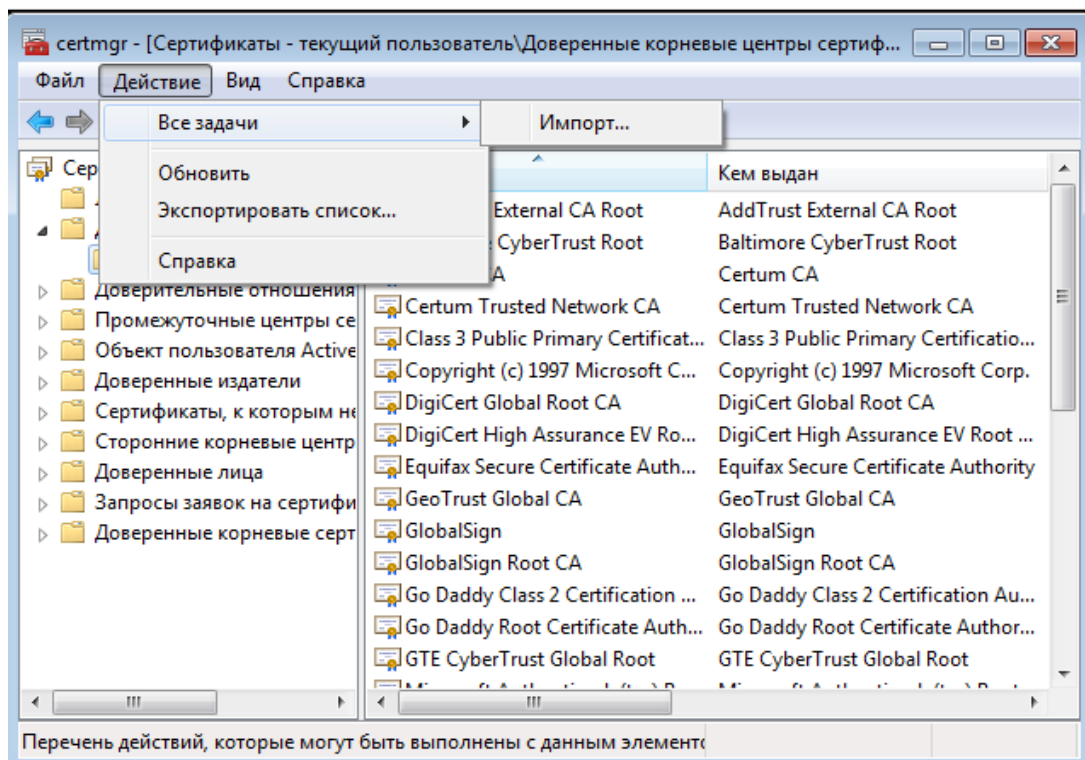


Рисунок 9 - Процедура добавления сертификата

В окне импорта сертификата необходимо указать скопированный с CentOS файл сертификата в соответствии с рисунком 10.

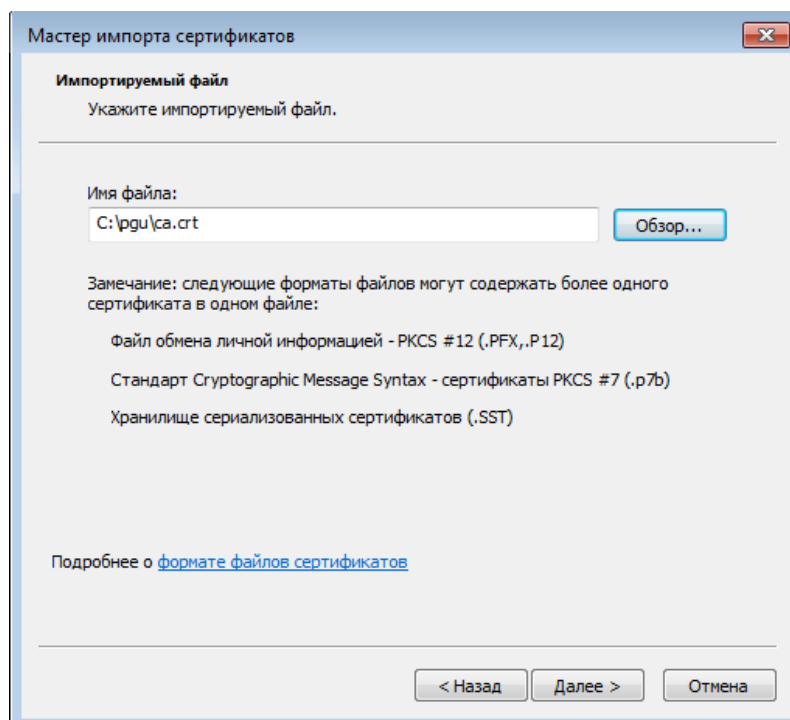


Рисунок 10 - Мастер импорта сертификатов

После нажатий кнопки «Далее» появится уведомление о невозможности проверки принадлежности устанавливаемого сертификата его владельцу, представленное в соответствии с рисунком 11.

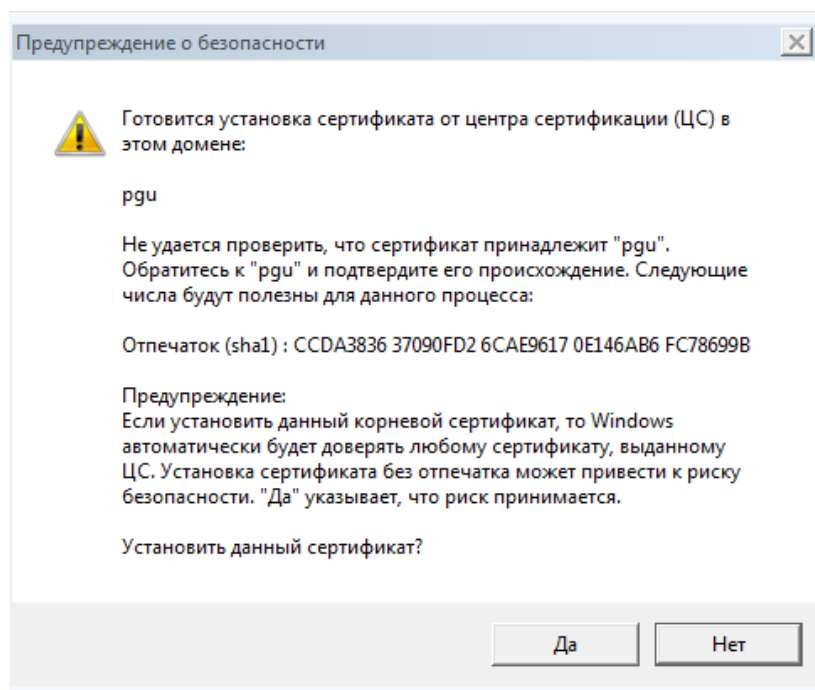


Рисунок 11 - Уведомление о невозможности проверки принадлежности сертификата его владельцу

С уведомлением необходимо согласиться, нажав кнопку «Да», после чего устанавливаемый сертификат появится в списке «Сертификаты». После установки сертификата необходимо перезагрузить виртуальную машину стандартными средствами ОС Windows.

Далее добавляем созданный ранее сертификат сервера скопированный с CentOS в Windows 7.

Запустив файл сертификата с расширением «.p12». Откроется окно мастера импорта сертификатов. При импорте сертификата потребуется ввести проверочную фразу указанную ранее при генерации сертификата. Окно мастера импорта сертификатов представлено на рисунке 12.

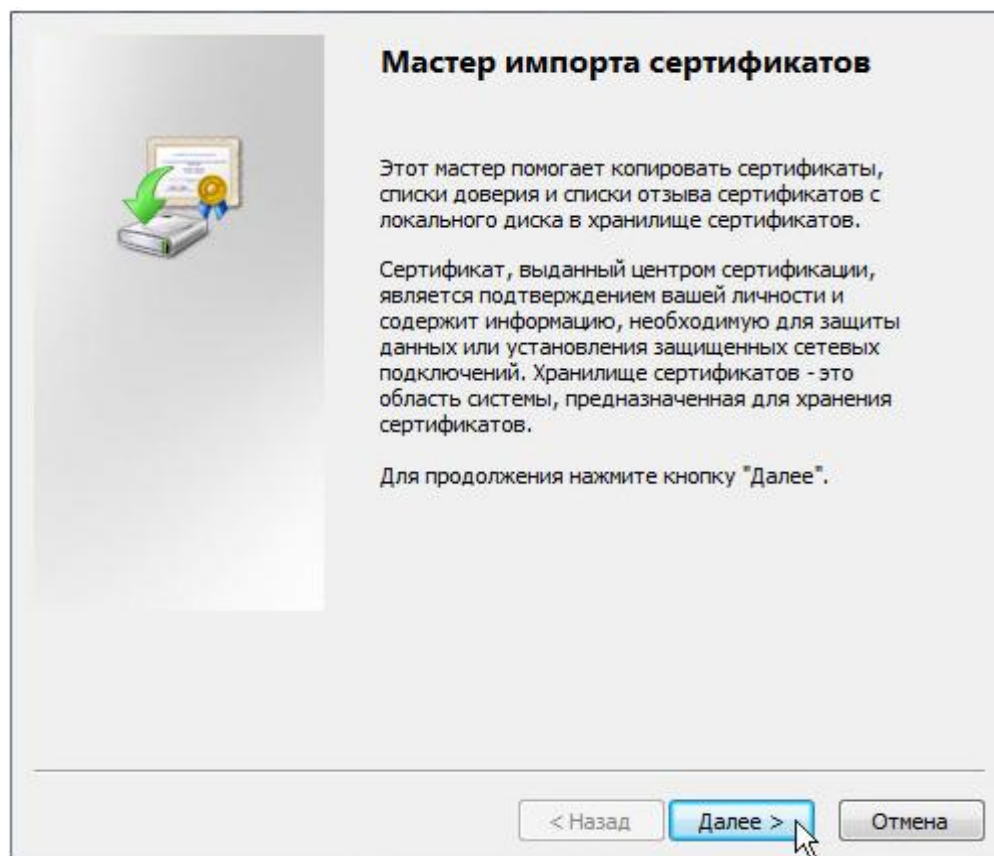


Рисунок 12 - Мастер импорта сертификатов

В последующих шагах установки необходимо нажать на кнопку «Далее», не меняя параметров установки. После завершения установки, на экран будет выведено окно оповещения. Окно оповещения представлено на рисунке 13.

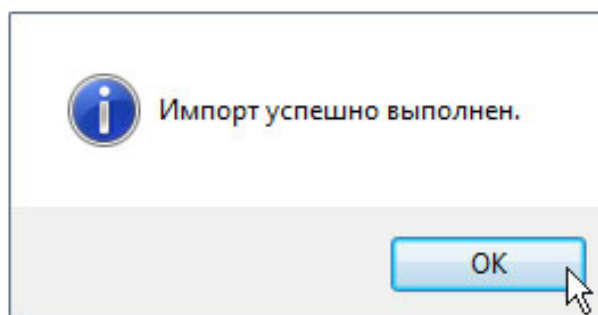


Рисунок 13 - Окно оповещения

Нажмите на кнопку «ОК», окно будет закрыто.

6.4 Просмотр свойств установленных сертификатов

Для просмотра свойств установленных сертификатов используется диспетчер сертификатов. Для этого необходимо перейти в меню «Пуск - Поиск - «certmgr.msc». Будет открыт диспетчер сертификатов. Для просмотра свойств установленного пользовательского сертификата необходимо выбрать в списке «Доверенные корневые центры сертификации» подсписок «Сертификаты», найти установленный ранее сертификат, нажать на него правой кнопкой мыши и выбрать команду «Открыть». Указанная операция представлена на рисунке 14.

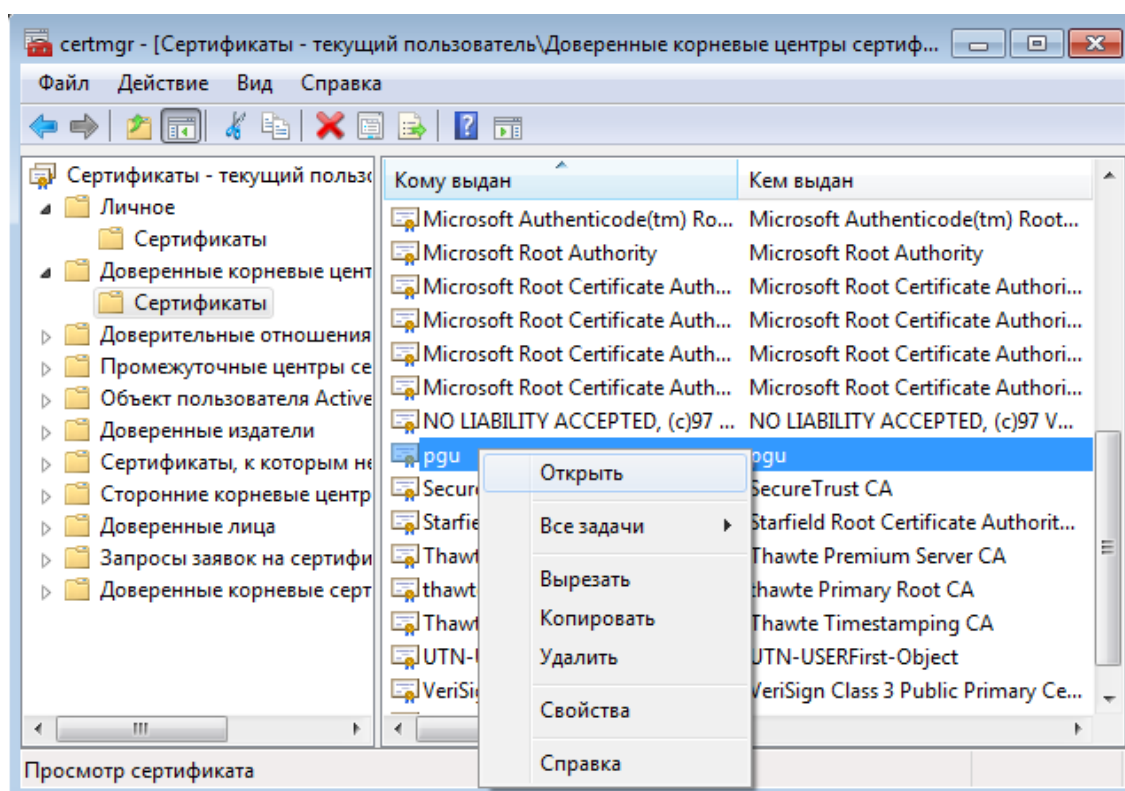


Рисунок 14 - Просмотр свойств установленных сертификатов

В открывшемся сертификате необходимо проверить его свойства:

- серийный номер сертификата;
- идентификатор алгоритма подписи;
- имя издателя;
- период действия;
- открытый ключ субъекта;
- имя субъекта сертификата.

Пример свойств пользовательского сертификата представлен на рисунке

15.

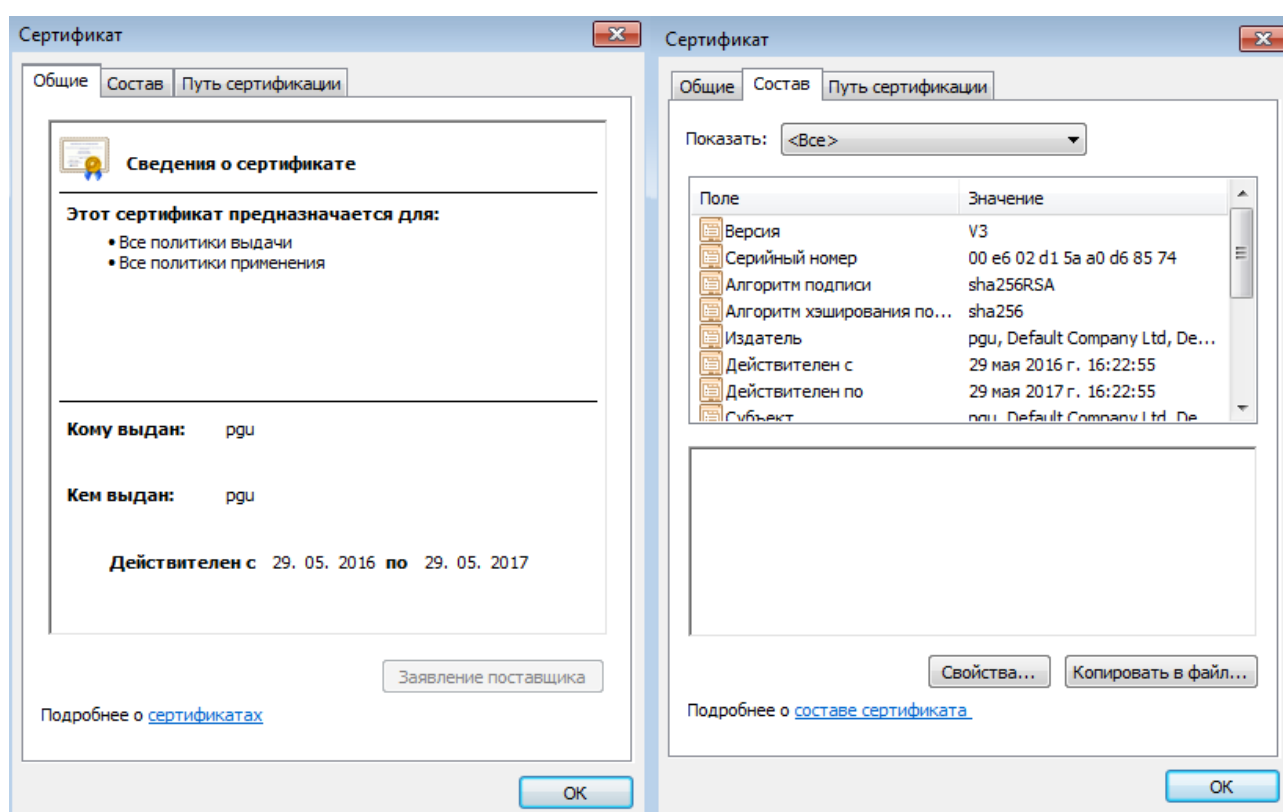


Рисунок 15 - Пример свойств пользовательского сертификата

Для просмотра свойств установленного сертификата сервера необходимо выбрать в списке «Личное» подсписок «Сертификаты», найти установленный ранее сертификат, нажать на него правой кнопкой мыши и выбрать команду «Открыть».

В открывшемся сертификате проверяем его свойства:

- серийный номер сертификата;
- идентификатор алгоритма подписи;

- имя издателя;
- период действия;
- открытый ключ субъекта;
- имя субъекта сертификата.

Пример свойств сертификата сервера представлен на рисунке 16.

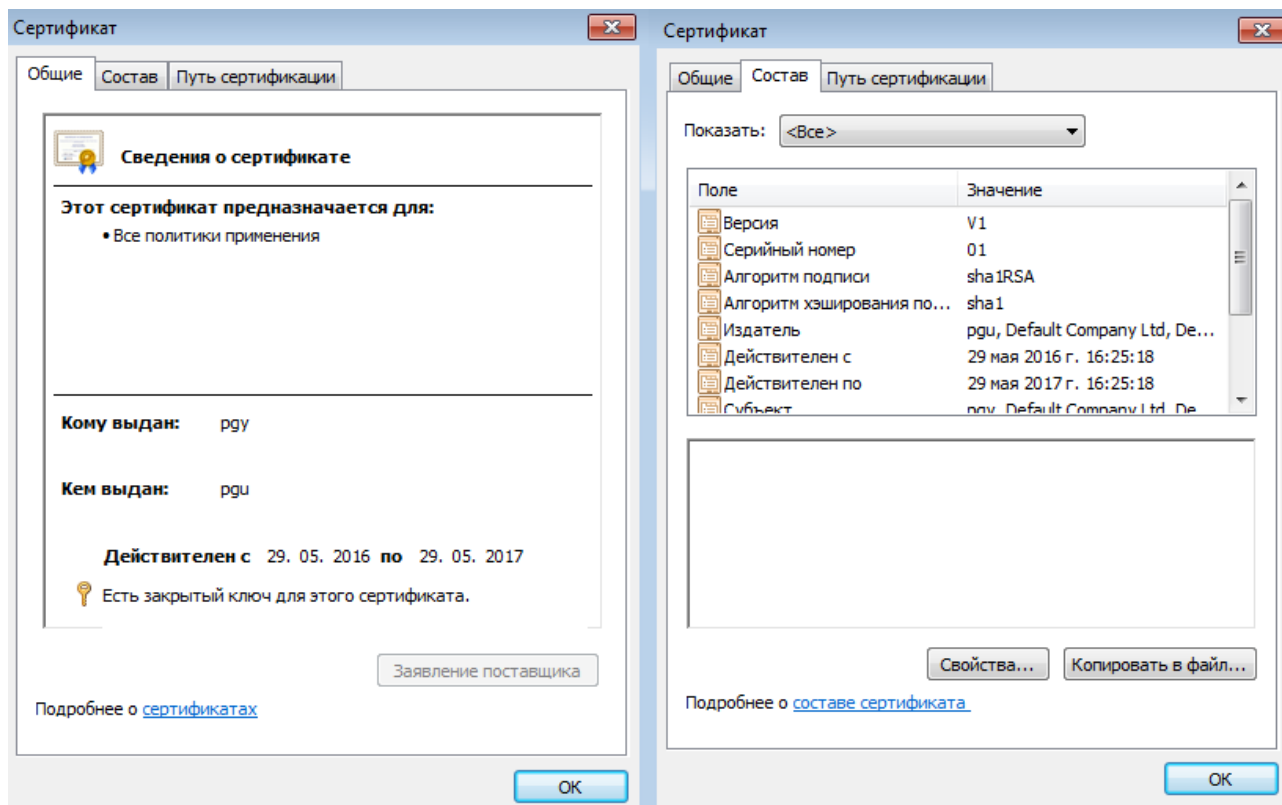


Рисунок 16 - Пример свойств сертификата сервера

6.5 Использование сертификатов для формирования электронной подписи документов

Для использования сертификатов по формированию электронной подписи документов потребуется запустить программное обеспечение OpenOffice в ОС Windows 7, далее для создания текстового документа необходимо перейти по вкладке «Текстовый документ», и ввести текст, представленный в варианте заданий. Прежде чем подписать документ, необходимо его сохранить. Пример запущенного и сохранённого документа представлен на рисунке 17.

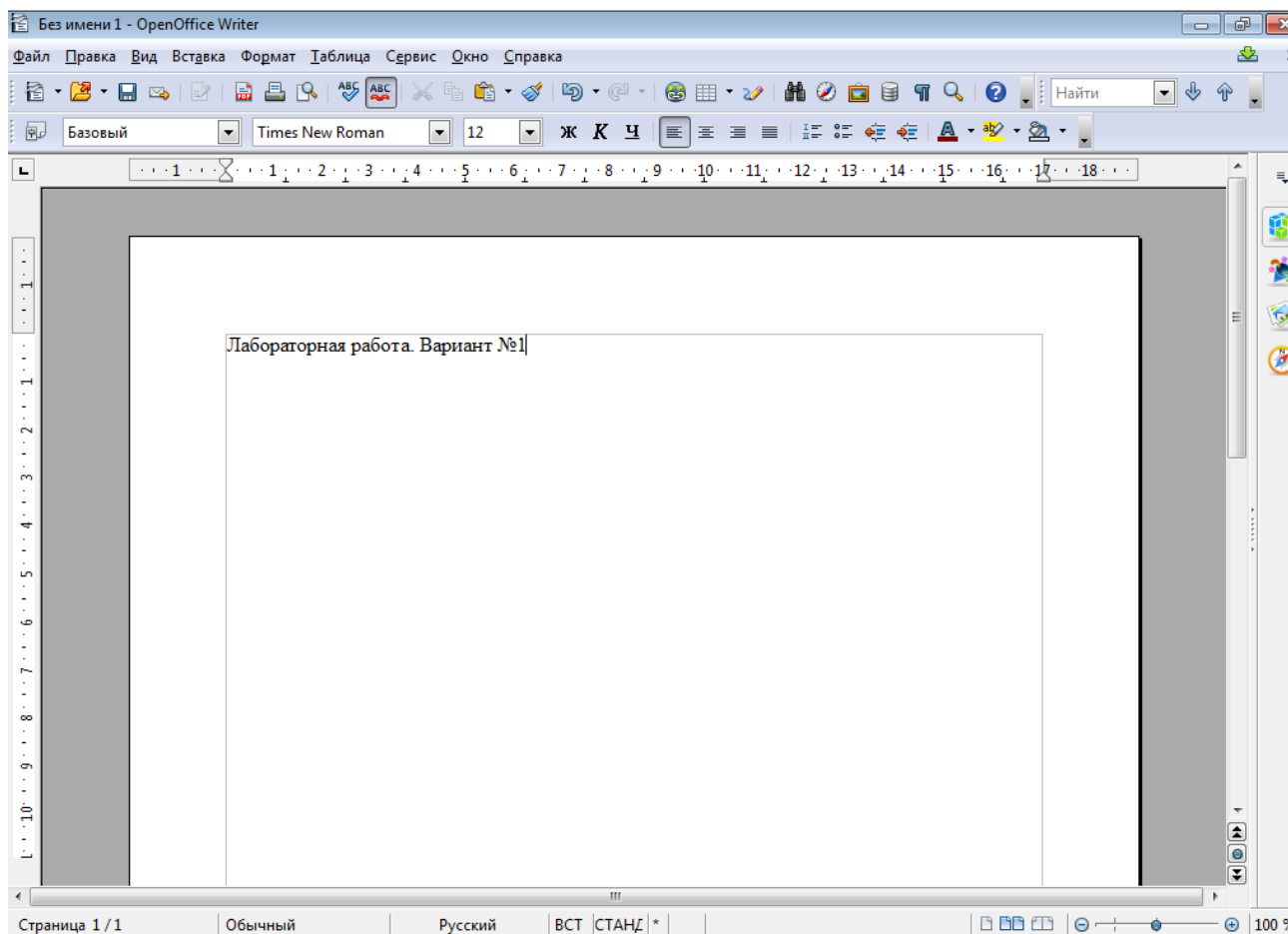


Рисунок 17 - Созданный и сохраненный документ

Для подписи документа нужно открыть вкладку «Файл - Цифровые подписи...» Указанная операция представлена на рисунке 18.

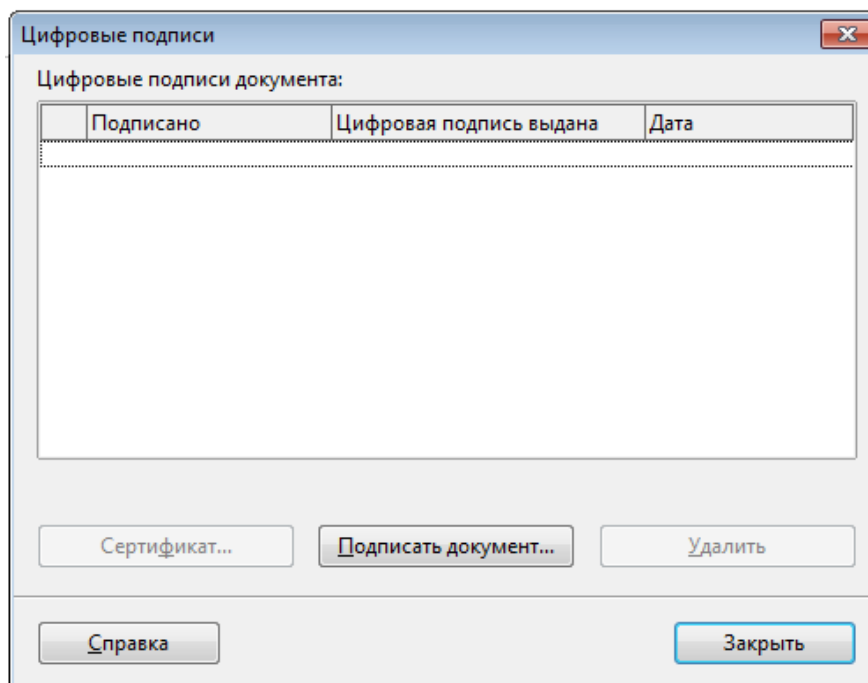


Рисунок 18 - Цифровые подписи

Далее потребуется перейти по вкладке «Подписать документ» необходимо выбирать ранее добавленный сертификат в ОС, подтвердить выбор сертификата, нажав на кнопку “ОК”. Окно выбора сертификата представлено на рисунке 19.

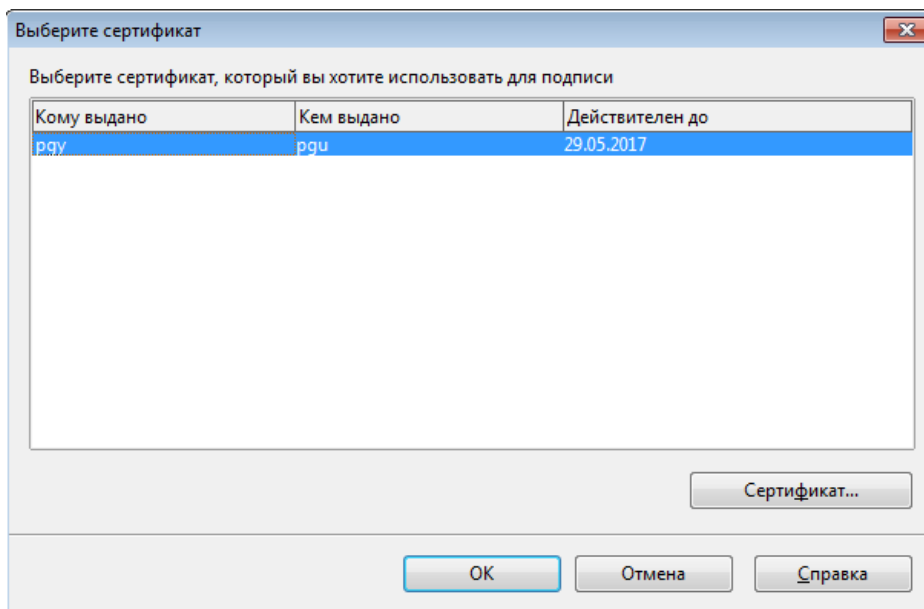


Рисунок 19 - Окно выбора сертификата

Теперь во вкладке «Цифровые подписи» отображается выбранный сертификат, для подписи документа нужно нажать на кнопку «Заккрыть». Окно «Цифровые подписи» с выбранным сертификатом представлен на рисунке 20.

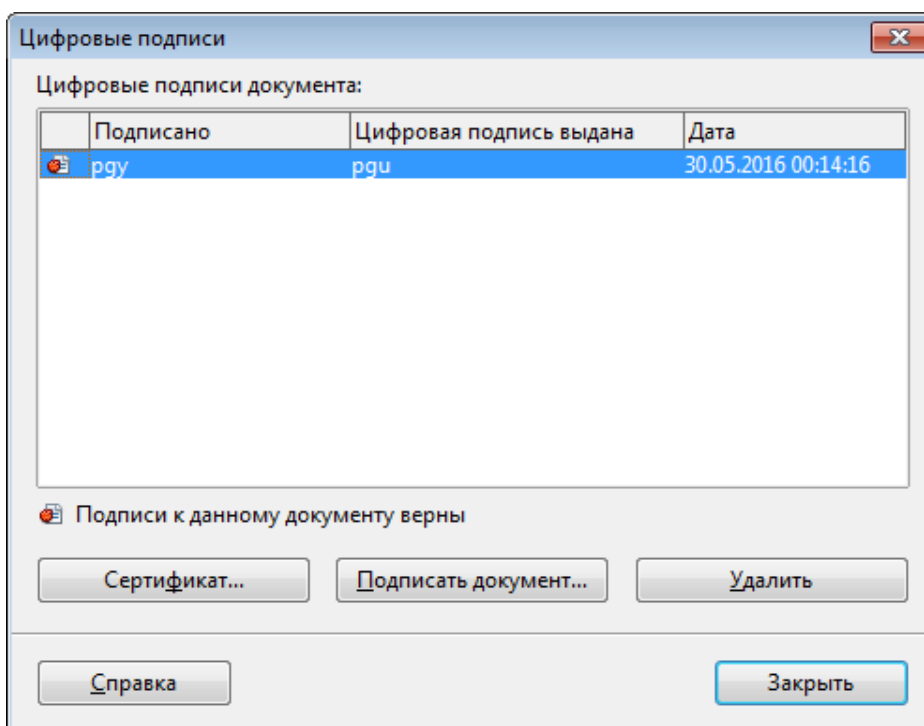


Рисунок 20 - Цифровые подписи

6.6 Проверка электронной подписи документа

При верном выполнении представленных выше операций должно быть выведено уведомление том что «Цифровая подпись: подпись подтверждена». И в строке с название документа будет добавлена запись (Подписано).

Пример успешной подписи документа представлен на рисунке 21.

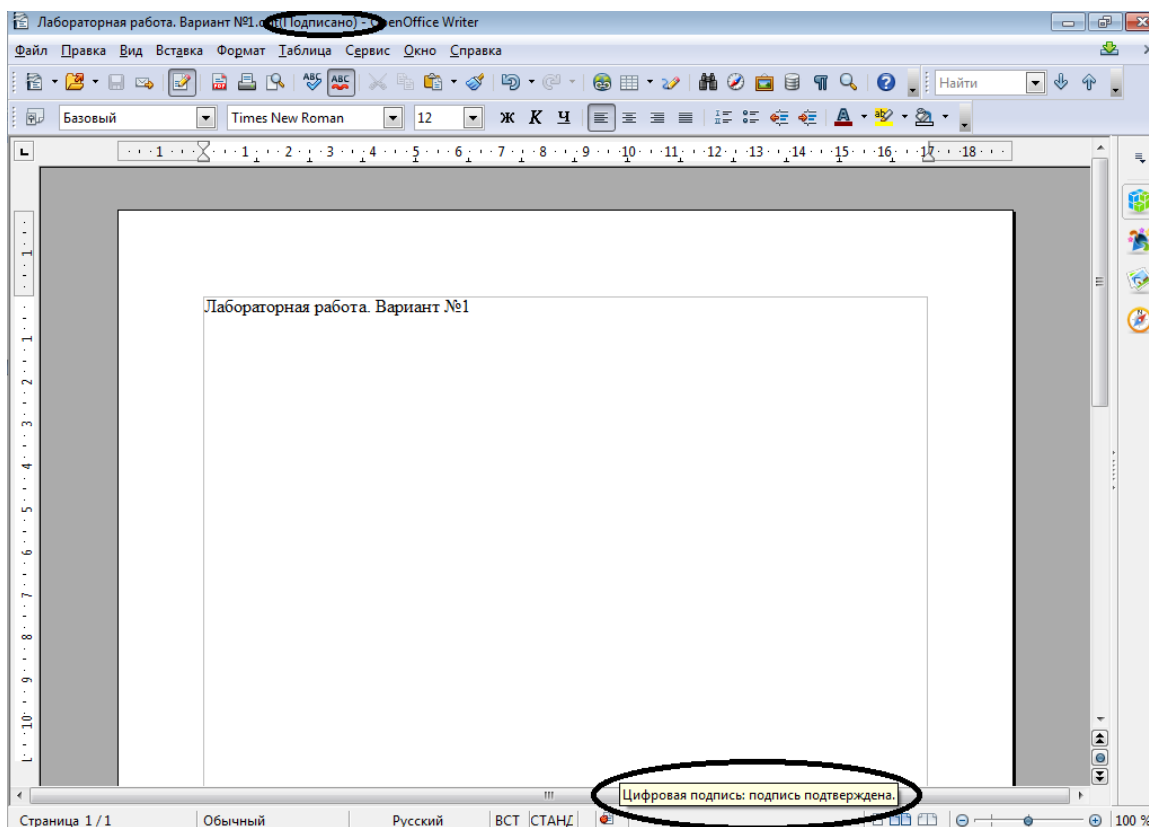


Рисунок.21 - Успешная подпись документа

При не успешном выполнении представленных выше операций должно быть выведено уведомление том что «Цифровая подпись: цифровая подпись подтверждена, но невозможно проверить сертификаты» Данное уведомление указывает на то что, не установлен пользовательский сертификат. Для устранения данной проблемы необходимо проверить правильность выполнения пункта лабораторного практикума по изучению инфраструктуры открытых ключей 6.3 «Установка сертификатов в операционную систему» и пункта лабораторного практикума по изучению инфраструктуры открытых 6.4 «Просмотр свойств установленных сертификатов».

Пример не успешной подписи документа представлен на рисунке 22.

Результаты выполнения данного пункта лабораторного практикума по изучению инфраструктуры открытых ключей в соответствии со своим вариантом включить в отчет о проделанной работе.

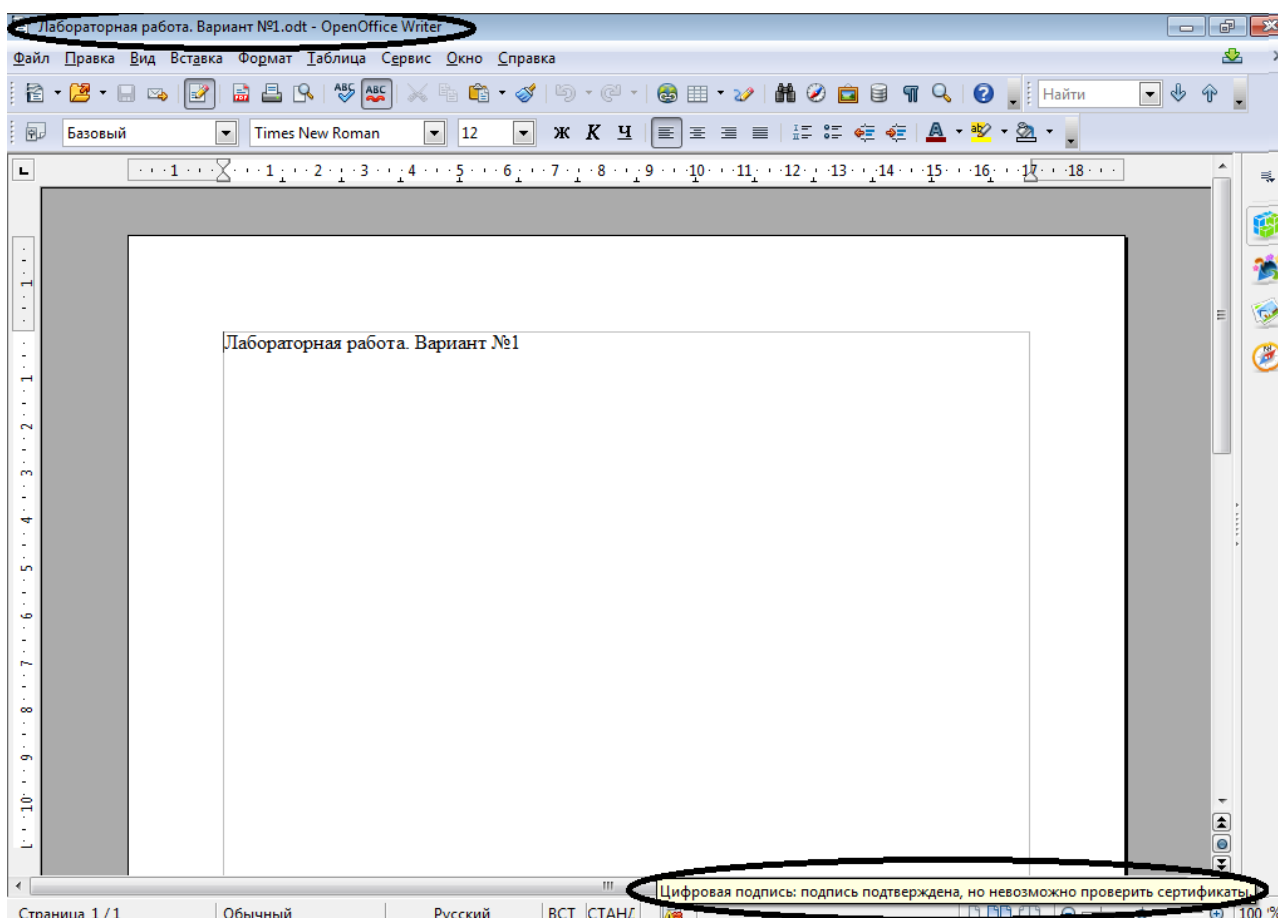


Рисунок 22 - Не успешная подпись документа

Для проверки, что документ подписан, верно, необходимо внести изменения в тест документа, если же документ изменил название «Лабораторная работа. Вариант№1.odt (Подписано)» на «Лабораторная работа. Вариант№1.odt» и перестанет высвечиваться уведомление о цифровой подписи. Значит, выше указанные пункты, по проведению практикума выполнены правильно. Пример выполнения замены текста установленного вариантом задания представлен на рисунке 23.

Результаты выполнения лабораторного практикума по изучению инфраструктуры открытых ключей, в соответствии со своим вариантом необходимо включить в отчет о проделанной работе.

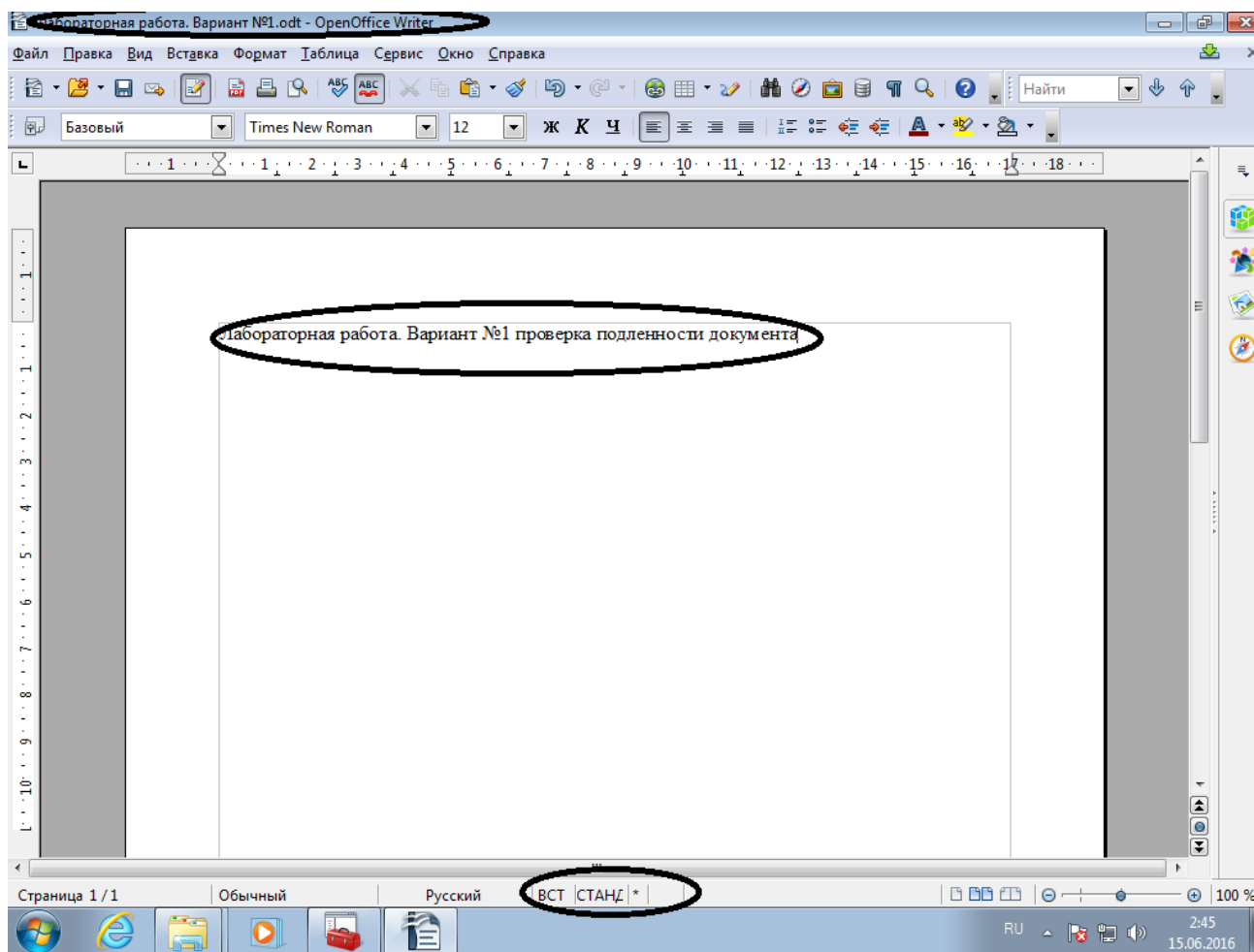


Рисунок 23 - Замена исходящего текста зашифрованного документа

6.7 Проверка электронной подписи документа при истекшем сроке действия сертификата

Для проверки электронной цифровой подписи документа при истекшем сроке действия сертификата необходимо заново создать и подписать документ, выполнив пункты заданий лабораторного практикума по созданию и подписи документа. Далее необходимо заменить системное время виртуальной машины с установленной ОС Windows 7 на один год вперед. Так как при создании ключей и сертификатов был указан период действия, равный одному году.

Для того что сменить время в установленной ОС Windows 7 потребуется открыть «Пуск - Панель управления - Дата и время - Изменить дату и время...» Смена даты и времени представлена на рисунке 24.

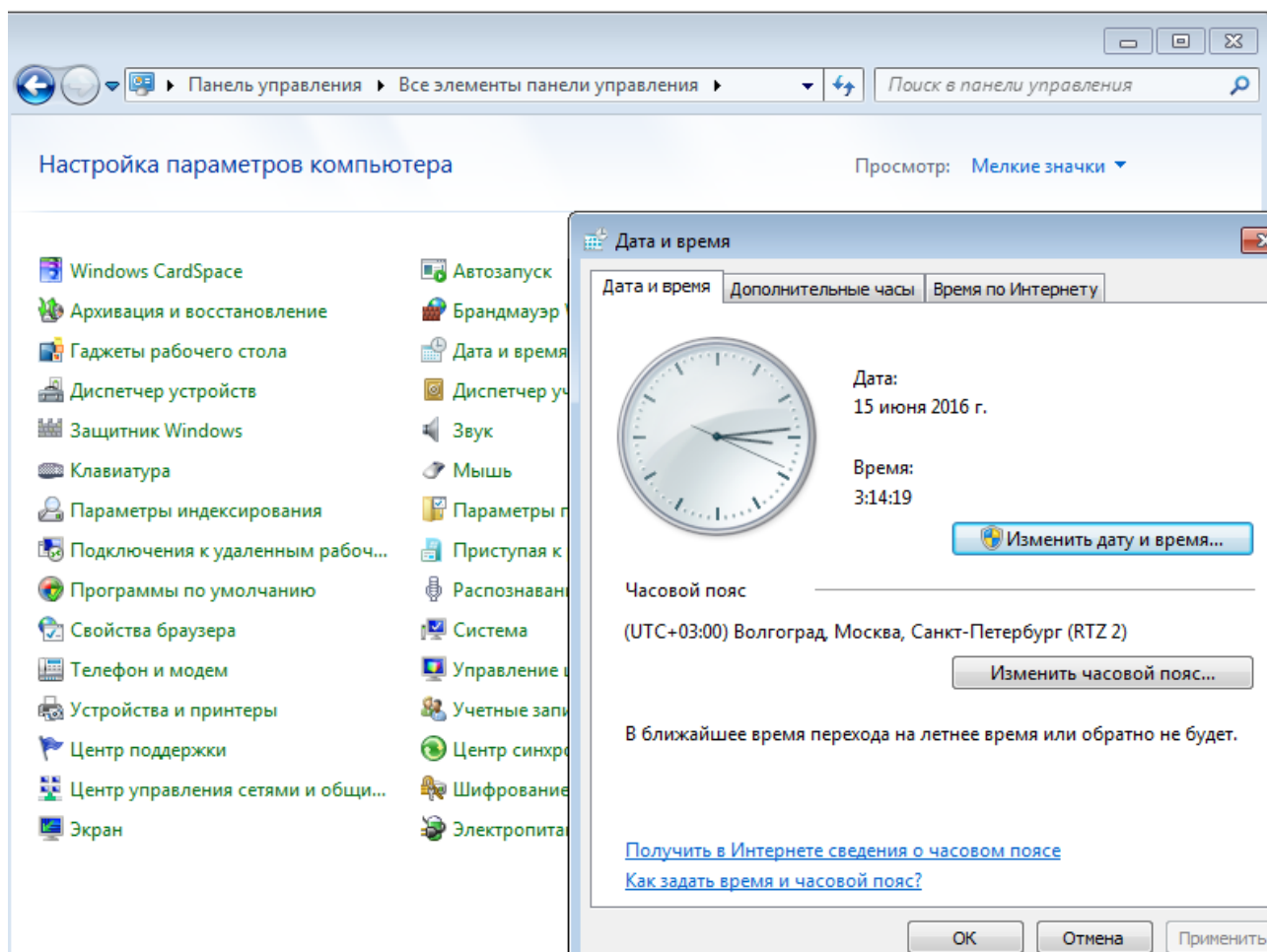


Рисунок 24 Смена даты и времени в ОС Windows 7

После замены даты и времени, при открытии подписанного ранее документа видим, что подпись подтверждена, но невозможно проверить сертификат, так как срок действия сертификата закончился.

Результаты выполнения данного пункта включите в отчет о проделанной работе по лабораторной практике.

6.8 Формирование списка отозванных сертификатов

Для генерирования списка отзыва аннулированных сертификатов следует воспользоваться командой:

```
[root@localhost cert]# openssl ca -gencrl -out crl.pem
```

В процессе выполнения команды нужно будет ввести пароль, установленный на стадии генерации сертификатов.

Команда для просмотра списка отзыва:

```
[root@localhost cert]# openssl crl -in crl.pem -text -noout
```

Команда для отзыва ранее выпущенных сертификатов:

```
[root@localhost cert]# openssl ca -revoke example.pem
```

Данные команды требуется вводить на виртуальной машине с установленной ОС CentOS.

Установить список аннулированных (отозванных) сертификатов на виртуальную машину с установленной ОС Windows 7 и выполнить проверку электронной подписи документа, аналогично пункту 6.6 и определить результат проверки.

Результаты проверки включить в отчет о проделанной работе.

7 Разработка вариантов заданий для проведения лабораторного практикума

Для выполнения лабораторного практикума были разработаны варианты заданий для обучающихся по специальности 10.03.05 «Информационная безопасность автоматизированных систем».

Варианты заданий различаются названием организации, адресом электронной почты, фамилией студента и текстом электронного документа. Варианты заданий представлены ниже.

1 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu1; Organizational Unit Name; Email Address - pgu1@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №1.

2 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu2; Organizational Unit Name; Email Address – pgu2@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №2.

3 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu3; Organizational Unit Name; Email Address – pgu3@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №3.

4 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu4; Organizational Unit Name; Email Address – pgu4@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №4.

5 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu5; Organizational Unit Name; Email Address – pgu5@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №5.

6 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu6; Organizational Unit

Name; Email Address – pgu6@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №6.

7 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu7; Organizational Unit Name; Email Address – pgu7@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №7.

8 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu8; Organizational Unit Name; Email Address – pgu8@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №8.

9 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu9; Organizational Unit Name; Email Address – pgu9@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №9.

10 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu10; Organizational Unit Name; Email Address - pgu10@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №10.

11 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu11; Organizational Unit Name; Email Address - pgu11@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №11.

12 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu12; Organizational Unit Name; Email Address - pgu12@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №12.

13 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu13; Organizational Unit Name; Email Address - pgu13@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №13.

14 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu14; Organizational Unit Name – оставить пустым; Email Address - pgu14@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №14.

15 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu15; Organizational Unit Name – оставить пустым; Email Address - pgu15@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №15.

16 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu16; Organizational Unit Name – оставить пустым; Email Address - pgu16@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №16.

17 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu17; Organizational Unit Name – оставить пустым; Email Address - pgu17@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №17.

18 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu18; Organizational Unit Name – оставить пустым; Email Address - pgu18@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №18.

19 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu19; Organizational Unit Name – оставить пустым; Email Address - pgu19@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №19.

20 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu20; Organizational Unit Name – оставить пустым; Email Address – pgu20@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №20.

21 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu21; Organizational Unit Name – оставить пустым; Email Address – pgu21@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №21.

22 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu22; Organizational Unit Name – оставить пустым; Email Address – pgu22@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №22.

23 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu23; Organizational Unit Name – оставить пустым; Email Address – pgu23@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №23.

24 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu24; Organizational Unit Name – оставить пустым; Email Address – pgu24@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №24.

25 вариант: Country Name – Россия; State or Province Name – оставить пустым; Locality Name - Пенза; Organization Name – pgu25; Organizational Unit Name – оставить пустым; Email Address – pgu25@gmail.com; Common Name - ФИО студента; Текст для созданного документа – Практикум для изучения PKI, Вариант №25.

ЗАКЛЮЧЕНИЕ

В процессе выполнения дипломного проекта были решены следующие задачи:

- произведен обзор инфраструктуры открытых ключей;
- произведен анализ возможных решений по реализации инфраструктуры открытых ключей;
- разработана архитектура лабораторного стенда для изучения инфраструктуры открытых ключей;
- разработан лабораторный стенд для проведения лабораторного практикума;
- разработан лабораторный практикум для изучения инфраструктуры открытых ключей;
- разработаны варианты заданий для проведения лабораторного практикума;
- разработаны рекомендации для проведения лабораторного практикума.

В результате разработан лабораторный практикум для изучения инфраструктуры открытых ключей, включающий лабораторный стенд, рекомендации для проведения практикума и варианты заданий.

В разработанном практикуме рассмотрены следующие вопросы:

- генерация сертификатов удостоверяющего центра;
- генерация сертификатов пользователей;
- аннулирование сертификатов пользователя;
- использование сертификатов пользователя для формирования электронной подписи.

Лабораторный стенд содержит виртуальную машину с установленной операционной системой CentOS и криптографическим пакетом OpenSSL, выполняющим функции удостоверяющего центра. Для формирования и проверки электронной подписи используется Open Office.

Таким образом, цель ВКР достигнута, требования технического задания на выпускную квалификационную работу выполнены в полном объеме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Инфраструктура открытых ключей [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Инфраструктура_открытых_ключей, свободный. – Загл. с экрана.
- 2 Инфраструктуры открытых ключей / Полянская О.Ю., Горбатов В.С. - М.: Интернет-университет информационных технологий - ИНТУИТ.ру, 2007
- 3 Сертификатов открытых ключей X.509 [электронный ресурс]. – Режим доступа: <http://www.inssl.com/x509-open-key-specifications.html>, свободный. – Загл. с экрана.
- 4 Использование Crypto API [электронный ресурс] – Режим доступа: <https://rsdn.ru/article/crypto/usingcryptoapi.xml>, свободный. – Загл. с экрана.
- 5 Криптопровайдер Crypton CSP [электронный ресурс]. – Режим доступа: http://www.itsec.ru/newstext.php?news_id=14394, свободный. – Загл. с экрана.
- 6 ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи».
- 7 ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования».
- 8 ViPNet Криптопровайдер [электронный ресурс]. – Режим доступа: http://www.infotecs.ru/products/catalog.php?ELEMENT_ID=2096, свободный. – Загл. с экрана.
- 9 ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования».
- 10 The GnuPG [электронный ресурс]. – Режим доступа: <https://www.gnupg.org>, свободный. – Загл. с экрана.
- 11 Программно-аппаратные средства защиты информации от НСД [электронный ресурс]. – Режим доступа: <http://www.suritel.ru/cgi-bin/view.pl?cid=1187156006&CatId=55>, свободный. – Загл. с экрана.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

УДК 004.056.55

Политехнический институт
Факультет приборостроения,
информационных технологий и
электроники

Выпускающая кафедра:
Информационная безопасность систем
и технологий
Учебная группа 11ПИ1

УТВЕРЖДАЮ
Зав. кафедрой ИБСТ
к.т.н., доцент С.Л. Зефирова

«27» 04 2016 г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ
РАБОТУ**

студента группы 11ПИ1 Шимаханова Антона Витальевича

Специальность 100503 – Информационная безопасность
автоматизированных систем

Тема дипломного проекта: Лабораторный практикум для изучения
инфраструктуры открытых ключей.

Руководитель ВКР
к.т.н., доцент кафедры ИБСТ
Нормоконтролер
к.т.н., доцент




Липилин О.В.

Фатеев А.Г.

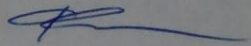
1 Цели и задачи ВКР

Цель работы: разработка лабораторного практикума для изучения инфраструктуры открытых ключей (далее - практикум).

Практикум предназначен для выработки у обучаемых практических навыков по применению инфраструктуры открытых ключей.

2 Тактико-технические требования к выполнению ВКР

Технические требования:

- практикум должен состоять из лабораторного стенда и рекомендаций по проведению практикума;
 - в практикуме должны рассматриваться следующие вопросы:
 - генерация сертификатов удостоверяющего центра;
 - генерация сертификатов пользователей;
 - аннулирование сертификатов пользователя;
 - использование сертификатов пользователя для формирования электронной подписи;
 - для выполнения практикума необходим компьютер с установленной операционной системой Windows 7 и средство виртуализации Virtual Box;
 - лабораторный стенд должен содержать виртуальную машину с установленной операционной системой CentOS и криптографическим пакетом OpenSSL;
 - должна быть разработана архитектура инфраструктуры открытых ключей, включающая:
 - удостоверяющий центр;
 - клиентское приложение, поддерживающее шифрование с открытым ключом;
 - тип применяемой архитектуры – простая, с одним удостоверяющим центром;
- 

- для реализации удостоверяющего центра должно использоваться свободное программное обеспечение Open SSL;

- должен быть разработан порядок проведения лабораторного практикума.

В ВКР должны быть рассмотрены следующие вопросы:

- обзор инфраструктуры открытых ключей;

- анализ возможных решений по реализации инфраструктуры открытых ключей;

- разработка архитектуры лабораторного стенда для изучения инфраструктуры открытых ключей;

- разработка лабораторного стенда для проведения лабораторного практикума;

- разработка лабораторного практикума для изучения инфраструктуры открытых ключей;

- разработка вариантов заданий для лабораторного практикума;

- разработка рекомендаций для проведения лабораторного практикума.

3 Требования к разрабатываемой документации

Отчет по ВКР выполняется в соответствии с требованиями ГОСТ 7.32.

Перечень и содержание графической части ВКР:

- инфраструктура открытых ключей – 1 л., ф. А1 (плакат);

- жизненный цикл ключей – 1 л., ф. А1 (плакат);

- формат сертификатов открытых ключей – 1 л., ф. А1 (плакат);

- диаграмма последовательности действий по выполнению лабораторного практикума – 1 л., ф. А1 (плакат);

- диаграмма компонентов лабораторного стенда – 1 л., ф. А1 (плакат);
- диаграмма вариантов использования – 1 л., ф. А1 (плакат);
- демонстрация выполнения практикума.

Электронная копия отчета и графической части ВКР должна быть представлена на магнитном или оптическом носителе.

4 Сроки выполнения ВКР

Тема утверждена приказом ректора ПГУ № 491а/о от «25» 04 2016 г.

Дата выдачи задания «09» февраля 2016 г.

Время дипломного проектирования с 18.04 2016 г. по 10.07 2016 г.

Задание к исполнению принял «09» февраля 2016 г.

Исполнитель ВКР Шимаханов А.В. Шимаханов А.В.



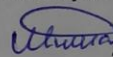
Ректору Пензенского
государственного университета
А. Д. Гулякову
студента 5 курса ФПИТЭ
очной формы обучения
Шимаханова Антона Витальевича

заявление.

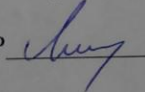
Я, Шимаханов Антон Витальевич, студент 5 курса очной формы обучения специальности 10.05.03 – «Информационная безопасность автоматизированных систем», прошу провести проверку с использованием системы «Антиплагиат» дипломного проекта на тему «Лабораторный практикум для изучения инфраструктуры открытых ключей», выполненной мной самостоятельно, на содержание элементов плагиата.

Все прямые заимствования из печатных и электронных источников, а также из защищенных ранее выпускных квалификационных работ, кандидатских и докторских диссертаций имеют соответствующие ссылки.

Я ознакомлен с действующим Стандартом университета «Выпускная квалификационная работа обучающихся по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры», согласно которому обнаружение плагиата является основанием для недопуска ВКР к защите и отчисления из университета.

 А.В. Шимаханов/
21.06.2016

ВКР представлена на проверку

Руководитель ВКР  /О.В. Липилин/

Протокол

проверки на оригинальность в системе «Антиплагиат.ВУЗ» выпускной квалификационной работы на тему «Лабораторный практикум для изучения инфраструктуры открытых ключей» студента 5 курса А.В. Шимаханова специальности 10.05.03 – «Информационная безопасность автоматизированных систем».

Руководитель ВКР к.т.н., Липилин Олег Владимирович.

Отчет о проверке № 1

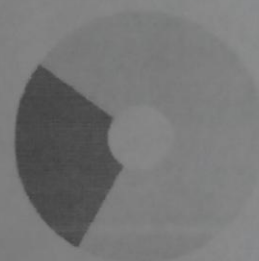
ФИО: Иванов Алексей Петрович
дата загрузки: 17.06.2016 12:51:25
пользователь: ap_ivanov@mail.ru / ID: 1267386
отчет предоставлен сервисом «Антиплагиат»
на сайте <http://www.antiplagiat.ru>

Информация о документе

№ документа: 44
Имя исходного файла: ВКР_Шимаханов_АВ_11ПИ1.doc
Размер текста: 1736 кБ
Тип документа: Не указано
Символов в тексте: 75333
Слов в тексте: 8923
Число предложений: 329

Информация об отчете

Дата: Отчет от 17.06.2016 12:51:25 - Последний готовый отчет
Комментарий: не указано
Оценка оригинальности: 72,6%
Заминствования: 27,4%
Цитирование: 0%



Оригинальность: 72,6%
Заминствования: 27,4%
Цитирование: 0%

08.06.2016 г.

Секретарь ГЭК

О.В. Липилин

А.П.Иванов

ОТЗЫВ
на выпускную квалификационную работу
Шимаханова Антона Витальевича

Тема выпускной квалификационной работы: «Лабораторный практикум для изучения инфраструктуры открытых ключей»

В соответствии с Федеральным государственным образовательным стандартом по специальности 10.05.03 «Информационная безопасность автоматизированных систем» студенты должны уметь эффективно использовать криптографические методы и средства защиты информации в автоматизированных системах. В современных автоматизированных системах множество защитных мер, в которых используются криптографические методы и средства защиты информации основанных на использовании инфраструктуры открытых ключей. В связи с этим, возникает необходимость разработки практикума, в котором студенты получают знания и практические навыки по инфраструктуре открытых ключей.

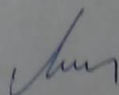
Целью ВКР является разработка лабораторного практикума для изучения инфраструктуры открытых ключей. Практикум предназначен для выработки у обучающихся практических навыков по применению инфраструктуры открытых ключей.

В работе Шимаханов А.В. провел обзор инфраструктуры открытых ключей, провел анализ возможных решений по реализации инфраструктуры открытых ключей, разработал лабораторный стенд для проведения лабораторного практикума, рекомендации по проведению практикума и варианты заданий.

Итогом работы является лабораторный практикум для изучения инфраструктуры открытых ключей. Требования технического задания ВКР полностью выполнены.

ВКР Шимаханова А.В. выполнена на хорошем уровне, соответствует требованиям, предъявляемым к ВКР по специальности 10.05.03 и может быть представлена к защите в ГАК. Выполнение ВКР оценивается в 13 баллов.

Руководитель ВКР



О.В. Липилин

РЕЦЕНЗИЯ

на выпускную квалификационную работу студента

Пензенского государственного университета

Шимаханова Антона Витальевича

Тема дипломного проекта: Практикум для изучения
инфраструктуры открытых ключей

На рецензию представлена выпускная квалификационная работа студента кафедры ИБСТ Шимаханова А.В., содержащая пояснительную записку в объеме 63 стр., 24 рис., 11 источников.

Целью выпускной квалификационной работы является разработка лабораторного практикума для изучения инфраструктуры открытых ключей. В процессе работы был проведен обзор инфраструктуры открытых ключей, анализ возможных решений по реализации инфраструктуры открытых ключей, разработан лабораторный стенд для проведения лабораторного практикума, рекомендации по проведению практикума и варианты заданий.

Материал в пояснительной записке изложен на достаточно квалифицированном уровне. Следует отметить широкий спектр программных продуктов и компонентов, составляющих лабораторный практикум.

Среди достоинств можно отметить актуальность работы, наглядность рекомендаций для проведения лабораторного практикума и возможность его применения в процессе обучения студентов кафедры ИБСТ.

Выпускная работа имеет следующие замечания:

- в лабораторном практикуме не включена проверка электронной подписи удостоверяющим центром;
- в лабораторном практикуме отсутствует возможность выработки сертификата с использованием Российских государственных стандартов в области криптографии.

В целом, работа оставляет хорошее впечатление. Выполнение ВКР оценивается в 7 баллов. Считаю, что выпускная квалификационная работа может быть допущена к защите, а Шимаханов А.В. достоин присвоения квалификации специалиста по информационной безопасности автоматизированных систем.

Научный сотрудник АО «ПНИИ»

А.Ж. Акмаев

Согласен А.Ж. Акмаев
инженер ОК

