**0Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Отделение - ОИТ

Направление – Информатика и вычислительная техника

**Отчёт по лабораторной работе № 3**

**«Электронная подпись»**

по дисциплине: Защита информации

Выполнил:

студент гр. 8В7Б Мальцев М.Ю.

Проверил:

Доцент ОИТ Ботыгин И.А.

Томск 2020 г.

Оглавление

[Оглавление 2](#_Toc59382253)

[Результат работы 3](#_Toc59382254)

[1. Теоретический материал 3](#_Toc59382255)

[1.1 Введение 3](#_Toc59382256)

[1.2 Основные положения 4](#_Toc59382257)

[1.3 Атаки на электронную цифровую подпись 6](#_Toc59382258)

[1.4 Правовое регулирование электронной цифровой подписи в России 7](#_Toc59382259)

[1.5 Средства работы с электронной цифровой подписью 9](#_Toc59382260)

[1.6 Заключение 11](#_Toc59382261)

[2. Тест 12](#_Toc59382262)

[3. Пример работы электронной подписи на собственной программе 13](#_Toc59382263)

[Вывод 17](#_Toc59382264)

Результат работы

1. Теоретический материал
   1. Введение

Электронная цифровая подпись – реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа электронной цифровой подписи и позволяющей идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажений информации в электронном документе. Электронная цифровая подпись в электронном документе равнозначна собственноручной подписи в документе на бумажном носителе при одновременном соблюдении следующих условий:

сертификат ключа подписи, относящийся к этой электронной цифровой подписи, не утратил силу (действует) на момент проверки или на момент подписания электронного документа при наличии доказательств, определяющих момент подписания;

подтверждена подлинностью электронной цифровой подписи в электронном документе;

электронная цифровая подпись используется в соответствии со сведениями, указанными в сертификате ключа подписи.

При этом электронной документ с электронной цифровой подписью имеет юридическое значение при осуществлении отношений, указанных в сертификате ключа подписи.

В скором будущем заключение договора будет возможно в электронной форме, который будет иметь такую же юридическую силу, как и письменный документ. Для этого он должен иметь механизм электронной цифровой подписи, подтверждаемый сертификатом. Владелец сертификата ключа подписи владеет закрытым ключом электронной цифровой подписи, что позволяет ему с помощью средств электронной цифровой подписи создавать свою электронную цифровую подпись в электронных документах (подписывать электронные документы). Для того, чтобы электронный документ могли открыть и другие пользователи, разработана система открытого ключа электронной подписи.

Для того, чтобы иметь возможность скреплять электронный документ механизмом электронной цифровой подписи, необходимо обратиться в удостоверяющий центр за получением сертификата ключа подписи. Сертификат ключа подписи должен быть внесен удостоверяющим центром в реестр сертификатов ключей подписей не позднее даты начала действия сертификата ключа подписи. Первый в России такой удостоверяющий центр запущен в сентябре 2002 г. российским НИИ развития общих сетей (РосНИИРОС). Удостоверяющий центр по закону должен подтверждать подлинность открытого ключа электронной цифровой подписи.

* 1. Основные положения

Общая суть электронной подписи заключается в следующем. С помощью криптографической хэш-функции вычисляется относительно короткая строка символов фиксированной длины (хэш). Затем этот хэш шифруется закрытым ключом владельца - результатом является подпись документа. Подпись прикладывается к документу, таким образом получается подписанный документ. Лицо, желающее установить подлинность документа, расшифровывает подпись открытым ключом владельца, а также вычисляет хэш документа. Документ считается подлинным, если вычисленный по документу хэш совпадает с расшифрованным из подписи, в противном случае документ является подделанным.

При ведении деловой переписки, при заключении контрактов подпись ответственного лица является непременным атрибутом документа, преследующим несколько целей:

гарантирование истинности письма путем сличения подписи с имеющимся образцом;

гарантирование авторства документа (с юридической точки зрения).

Выполнение данных требований основывается на следующих свойствах подписи:

подпись аутентична, то есть с ее помощью получателю документа можно доказать, что она принадлежит подписывающему;

подпись неподделываема; то есть служит доказательством, что только тот человек, чей автограф стоит на документе, мог подписать данный документ, и никто иной;

подпись непереносима, то есть является частью документа и поэтому перенести ее на другой документ невозможно;

документ с подписью является неизменяемым;

подпись неоспорима;

любое лицо, владеющее образцом подписи, может удостоверится, что документ подписан владельцем подписи.

Развитие современных средств безбумажного документооборота, средств электронных платежей немыслимо без развития средств доказательства подлинности и целостности документа. Таким средством является электронно-цифровая подпись (ЭЦП), которая сохранила основные свойства обычной подписи.

Существует несколько методов построения ЭЦП, а именно:

шифрование электронного документа (ЭД) на основе симметричных алгоритмов. Данная схема предусматривает наличие в системе третьего лица – арбитра, пользующегося доверием обеих сторон. Авторизацией документа в данной схеме является сам факт шифрования ЭД секретным ключом и передачи его арбитру.

Использование ассиметричных алгоритмов шифрования. Фактом подписания документа является шифрование его на секретном ключе отправителя.

Развитием предыдущей идеи стала наиболее распространенная схема ЭЦП – шифрование окончательного результата обработки ЭД хеш-функцией при помощи ассиметричного алгоритма.

Появление этих разновидностей обусловлено разнообразием задач, решаемых с помощью электронных технологий передачи и обработки электронных документов.

При генерации ЭЦП используются параметры трех групп:

общие параметры

секретный ключ

открытый ключ

Отечественным стандартом на процедуры выработки и проверки ЭЦП является ГОСТ Р 34.10-94.

* 1. Атаки на электронную цифровую подпись

Стойкость большинства схем ЭЦП зависит от стойкости ассиметричных алгоритмов шифрования и хэш-функций.

Существует следующая классификация атак на схемы ЭЦП:

атака с известным открытым ключом.

Атака с известными подписанными сообщениями – противник, кроме открытого ключа имеет и набор подписанных сообщений.

Простая атака с выбором подписанных сообщений – противник имеет возможность выбирать сообщения, при этом открытый ключ он получает после выбора сообщения.

Направленная атака с выбором сообщения

Адаптивная атака с выбором сообщения.

Каждая атака преследует определенную цель, которые можно разделить на несколько классов:

полное раскрытие. Противник находит секретный ключ пользователя.

Универсальная подделка. Противник находит алгоритм, функционально аналогичный алгоритму генерации ЭЦП.

Селективная подделка. Подделка подписи под выбранным сообщением.

Экзистенциальная подделка. Подделка подписи хотя бы для одного случайно выбранного сообщения.

На практике применение ЭЦП позволяет выявить или предотвратить следующие действия нарушителя:

отказ одного из участников авторства документа.

Модификация принятого электронного документа.

Подделка документа.

Навязывание сообщений в процессе передачи – противник перехватывает обмен сообщениями и модифицирует их.

Так же существуют нарушения, от которых невозможно оградить систему обмена сообщениями – это повтор передачи сообщения и фальсификация времени отправления сообщения. Противодействие данным нарушениям может основываться на использовании временных вставок и строгом учете входящих сообщений.

* 1. Правовое регулирование электронной цифровой подписи в России

В развитых странах мира, в том числе и в Российской Федерации, электронная цифровая подпись широко используется в хозяйственном обороте. Банк России и другие банки Российской Федерации эффективно используют ЭЦП для осуществления своих операций путем пересылки банковских электронных документов по корпоративным и общедоступным телекоммуникационным сетям.

Для преодоления всех существующих в данной области отношений препятствий необходимо создание унифицированных правил, при помощи которых страны могут в национальном законодательстве решить основные проблемы, связанные с юридической значимостью записей в памяти ЭВМ, письменной формой электронных данных (в том числе и документов), подписью под такими данными, оригиналом и копиями электронных данных, а также признанием в качестве судебных доказательств электронных данных, заверенных электронной подписью.

10 января 2002 года был принят Федеральный Закон «Об электронной цифровой подписи», вступивший в силу с 22 января текущего года, который закладывает основы решения проблемы обеспечения правовых условий для использования электронной цифровой подписи в процессах обмена электронными документами, при соблюдении которых электронная цифровая подпись признается юридически равнозначной собственноручной подписи человека в документе на бумажном носителе.

Федеральный Закон «Об электронной цифровой подписи» определяет условия использования ЭЦП в электронных документах органами государственной власти и государственными организациями, а также юридическими и физическими лицами, при соблюдении которых:

средства создания подписи признаются надежными;

сама ЭЦП признается достоверной, а ее подделка или фальсификация подписанных данных могут быть точно установлены;

предоставляются юридические гарантии безопасности передачи информации по открытым телекоммуникационным каналам;

соблюдаются правовые нормы, содержащие требования к письменной форме документа;

сохраняются все традиционные процессуальные функции подписи, в том числе удостоверение полномочий подписавшей стороны, установление подписавшего лица и содержания сообщения, а также роль подписи в качестве судебного доказательства;

обеспечивается охрана персональной информации.

В Законе устанавливаются права и обязанности обладателя электронной цифровой подписи.

В соответствии с законом владельцем сертификата ключа подписи (обладателем электронной цифровой подписи) является физическое лицо, на имя которого удостоверяющим центром выдан сертификат ключа подписи и которое владеет соответствующим закрытым ключом электронной цифровой подписи, позволяющим с помощью средств электронной цифровой подписи создавать свою электронную цифровую подпись электронных документах (подписывать электронные документы).

Владелец сертификата ключа подписи обязан:

Хранить в тайне закрытый ключ электронной цифровой подписи;

Не использовать для электронной цифровой подписи открытые и закрытые ключи электронной цифровой подписи, если ему известно, что эти ключи используются или использовались ранее;

Немедленно требовать приостановления действия сертификата ключа подписи при наличии оснований полагать, что тайна закрытого ключа электронной цифровой подписи нарушена.

Согласно ст. 6 данного Закона сертификат ключа подписи должен содержать следующие сведения:

Уникальный регистрационный номер сертификата ключа подписи, даты начала и окончания срока действия сертификата ключа подписи, находящегося в реестре удостоверяющего центра;

Фамилия, имя, отчество владельца сертификата ключа подписи или псевдоним владельца;

Открытый ключ электронной цифровой подписи;

Наименование и место нахождения удостоверяющего центра, выдавшего сертификат ключа подписи;

Сведения об отношениях, при осуществлении которых электронный документ с электронной цифровой подписью будет иметь юридическое значение.

* 1. Средства работы с электронной цифровой подписью

**PGP**

Наиболее известный — это пакет PGP (Pretty Good Privacy) – (www.pgpi.org), без сомнений являющийся на сегодня самым распространенным программным продуктом, позволяющим использовать современные надежные криптографические алгоритмы для защиты информации в персональных компьютерах.

К основным преимуществам данного пакета, выделяющим его среди других аналогичных продуктов следует отнести следующие:

Открытость. Исходный код всех версий программ PGP доступен в открытом виде. Любой эксперт может убедиться в том, что в программе эффективно реализованы криптоалгоритмы. Так как сам способ реализации известных алгоритмов был доступен специалистам, то открытость повлекла за собой и другое преимущество - эффективность программного кода.

Стойкость. Для реализации основных функций использованы лучшие (по крайней мере на начало 90-х) из известных алгоритмов, при этом допуская использование достаточно большой длины ключа для надежной защиты данных

Бесплатность. Готовые базовые продукты PGP (равно как и исходные тексты программ) доступны в Интернете в частности на официальном сайте PGP Inc. ( www.pgpi.org ).

Поддержка как централизованной (через серверы ключей) так и децентрализованной (через «сеть доверия») модели распределения открытых ключей.

Удобство программного интерфейса. PGP изначально создавалась как продукт для широкого круга пользователей, поэтому освоение основных приемов работы отнимает всего несколько часов.

**GNU Privacy Guard (GnuPG)**

GnuPG (www.gnupg.org ) - полная и свободно распространяемая замена для пакета PGP. Этот пакет не использует патентованный алгоритм IDEA, и поэтому может быть использован без каких-нибудь ограничений. GnuPG соответсвует стандарту RFC2440 (OpenPGP).

**Криптон**

Пакет программ Криптон (www.ancud.ru) предназначен для использования электронной цифровой подписи (ЭЦП) электронных документов.

В стандартной поставке для хранения файлов открытых ключей используются дискеты. Помимо дискет, пакет Криптон дает возможность использования всех типов ключевых носителей (смарт-карт, электронных таблеток Touch Memory и др.).

* 1. Заключение

Цифровая подпись обеспечивает:

Удостоверение источника документа. В зависимости от деталей определения «документа» могут быть подписаны такие поля как автор, внесённые изменения, метка времени и т. д.

Защиту от изменений документа. При любом случайном или преднамеренном изменении документа (или подписи) изменится хэш, следовательно подпись станет недействительной.

Невозможность отказа от авторства. Так как создать корректную подпись можно лишь зная закрытый ключ, а он известен только владельцу, то владелец не может отказаться от своей подписи под документом.

Возможны следующие угрозы цифровой подписи:

Злоумышленник может попытаться подделать подпись для выбранного им документа.

Злоумышленник может попытаться подобрать документ к данной подписи, чтобы подпись к нему подходила.

При использовании надёжной хэш-функции, вычислительно сложно создать поддельный документ с таким же хэшем, как у подлинного. Однако, эти угрозы могут реализоваться из-за слабостей конкретных алгоритмов хэширования, подписи, или ошибок в их реализациях.

Тем не менее, возможны ещё такие угрозы системам цифровой подписи:

Злоумышленник, укравший закрытый ключ, может подписать любой документ от имени владельца ключа.

Злоумышленник может обманом заставить владельца подписать какой-либо документ, например используя протокол слепой подписи.

Злоумышленник может подменить открытый ключ владельца (см. управление ключами) на свой собственный, выдавая себя за него.

1. Тест

Какие функции выполняет электронная цифровая подпись?

* помогает гарантировать, что поставивший подпись — тот, кем он является в действительности;
* помогает гарантировать, что содержимое документа не менялось и не подделывалось после ввода цифровой подписи;
* помогает доказать любой из сторон авторство подписанного содержимого;
* **все функции, перечисленные выше.**

Какой ключ должен обязательно присутствовать в документе с электронной цифровой подписью?

* закрытый ключ;
* **открытый ключ;**
* оба.

Какие действия можно выполнять с документом, подписанным цифровой подписью?

* только редактирование;
* **только чтение;**
* чтение и редактирование.

Какая информация хранится в электронной цифровой подписи?

* имя файла закрытого ключа подписи;
* только информация о лице, сформировавшем подпись;
* **дата формирования подписи, информация о лице, сформировавшем подпись и имя файла открытого ключа подписи.**

Что представляет собой строка электронной цифровой подписи?

* **графический объект;**
* строка MS Office Word;
* надпись.

Какую роль выполняет электронная цифровая подпись?

* роль дополнительной информации о передаваемых данных;
* это данные о времени передачи информации;
* **роль обычной подписи в электронных документах;**
* роль обратного адреса отправителя.

Может ли документ одновременно быть зашифрованным и подписанным ЭЦП:

* **Да**
* Нет

При отправке по электронной почте подписанного ЭЦП документа будет отправлен:

* **сам файл и подпись файла;**
* только файл;
* только подпись.

При подписании документа ЭЦП используется:

* Открытый ключ
* **Секретный ключ**
* Сертификат ключа

Первый ФЗ «Об электронной цифровой подписи» в РФ был принят:

* В 2000 г.
* **В 2002 г.**
* В 2006 г.

1. Пример работы электронной подписи на собственной программе

В данном разделе представлен пример создания электронной подписи при помощи языка Kotlin для операционной системы Android.

Для реализации функции электронной подписи были использованы классы Signature и KeyPair из библиотеки java.security. Для демонстрации примера работы был создан класс SignatureExample, имеющий метод подписи строки, метод проверки подписи и метод обновления ключей.

При создании экземпляра класса инициализируются переменные пары ключей, подписи, а также два байтовых массива, хранящих подпись и подписанную строку соответственно.

**class** **SignatureExample**(

signatureAlgorithm: String = "SHA256WithDSA",

keyPairAlgorithm: String = "DSA"

) {

// Создание пары ключей

**private** **var** keyPair: KeyPair = KeyPairGenerator.getInstance(keyPairAlgorithm).generateKeyPair()

**private** **val** signature = Signature.getInstance(signatureAlgorithm)

// Эл. подпись в виде массива байтов

**private** **var** digitalSignature: ByteArray = byteArrayOf()

**private** **var** data: ByteArray = byteArrayOf()

}

В классе определен метод подписи строки, передающийся в параметр, и возвращающий строковое представление подписи.

// Метод подписывает строку и возвращает подпись

// в виде массива байтов

**fun** **sign**(input: String): ByteArray {

**val** secureRandom = SecureRandom()

signature.initSign(keyPair.**private**, secureRandom)

data = input.toByteArray(charset("UTF-8"))

signature.update(data)

// Возвращаем подпись

digitalSignature = signature.sign()

**return** digitalSignature

}

После создания подписи, ее можно проверить методом checkSign():

// Проверка подписи

**fun** **checkSign**(): Boolean {

signature.initVerify(keyPair.**public**)

signature.update(data)

**return** signature.verify(digitalSignature)

}

Для проверки корректной работы метода checkSign(), был создан метод, создающий новую пару ключей.

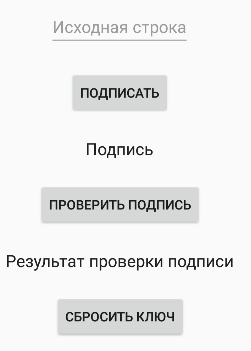
// Метод пересоздания пары ключей

**fun** **refreshKeyPair**() {

keyPair = KeyPairGenerator.getInstance("DSA").generateKeyPair()

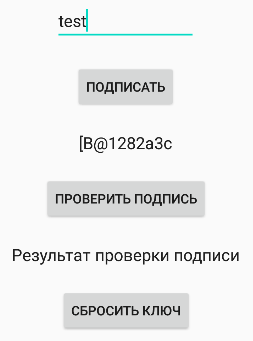
}

Для выполнения приведенного выше примера был создан простой интерфейс в приложении android, представленный на рисунках ниже.



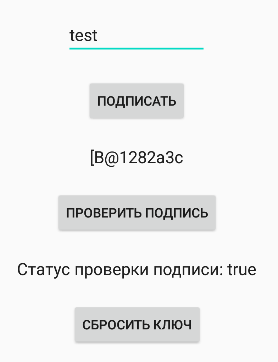
1. – Изначальный интерфейс

На рисунке выше представлены три кнопки, отвечающие за каждый метод класса SignatureExample.



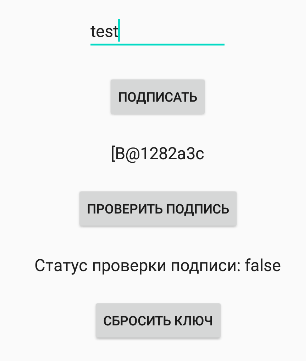
1. – Строкове представление подписи

При нажатии кнопки «Подписать», строка из поля ввода подписывается классом Signature, и ее строковое представление выводится в поле «Подпись».



1. – Проверка подписи

Далее, можно проверить подпись с помощью нажатия соответствующей кнопки. Результат проверки выводится в поле под кнопкой «проверить подпись».



1. – Проверка подписи

Чтобы проверить корректность метода checkSign(), можно создать новую пару ключей с помощью кнопки «Сбросить ключ» и снова нажать «проверить подпись». Как видно на рисунке выше, с новой парой ключей статус проверки вывод значение false, что говорит о том, что строка была подписана другим ключом.

Из приведенных выше рисунков можно сделать вывод, что программа корректно выполняет свою работу.

Вывод

В процессе выполнения лабораторной работы было проведено исследование электронных подписей. Полученные знания были закреплены с помощью реализации примера работы электронной подписи на языке программирования Kotlin с графическим интерфейсом операционной системы Android.

Приложение А. Исходный код приложения

SignatureExample:

**class** **SignatureExample**(

signatureAlgorithm: String = "SHA256WithDSA",

keyPairAlgorithm: String = "DSA"

) {

// Создание пары ключей

**private** **var** keyPair: KeyPair = KeyPairGenerator.getInstance(keyPairAlgorithm).generateKeyPair()

**private** **val** signature = Signature.getInstance(signatureAlgorithm)

// Эл. подпись в виде массива байтов

**private** **var** digitalSignature: ByteArray = byteArrayOf()

**private** **var** data: ByteArray = byteArrayOf()

// Метод пересоздания пары ключей

**fun** **refreshKeyPair**() {

keyPair = KeyPairGenerator.getInstance("DSA").generateKeyPair()

}

// Метод подписывает строку и возвращает подпись

// в виде массива байтов

**fun** **sign**(input: String): ByteArray {

**val** secureRandom = SecureRandom()

signature.initSign(keyPair.**private**, secureRandom)

data = input.toByteArray(charset("UTF-8"))

signature.update(data)

// Возвращаем подпись

digitalSignature = signature.sign()

**return** digitalSignature

}

// Проверка подписи

**fun** **checkSign**(): Boolean {

signature.initVerify(keyPair.**public**)

signature.update(data)

**return** signature.verify(digitalSignature)

}

}

Android View:

**val** signatureExample = SignatureExample()

// Делаем подпись

signButton.setOnClickListener {

**try** {

signResultView.text =

signatureExample.sign(stringInputView.text.toString()).toString()

} **catch** (e: Exception) {

e.printStackTrace()

Toast.makeText(

requireContext(),

"Ошибка при попытке подписать строку",

Toast.LENGTH\_SHORT

).show()

}

}

// Проверяем подпись

checkSignButton.setOnClickListener {

**try** {

**val** result =

"Статус проверки подписи: " + signatureExample.checkSign()

.toString()

checkResultView.text = result

} **catch** (e: Exception) {

e.printStackTrace()

Toast.makeText(

requireContext(),

"Ошибка при попытке проверит ьстроку",

Toast.LENGTH\_SHORT

).show()

}

}

// Делаем новые ключи

refreshKeysButton.setOnClickListener {

signatureExample.refreshKeyPair()

}