Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт Космических и Информационных технологий
институт
Кафедра Информатики
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А.С.Кузнецов
подпись инициалы, фамилия
« 08 » 06 2017г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 «Программная инженерия»
код – наименование направления

Кроссплатформенная программная система хранения параметров учетных записей
тема

Руководитель: _____ 08.06.17 Доцент, к. т. н. А.С.Кузнецов
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник: _____ 08.06.17 С.В.Прохоров
подпись, дата инициалы, фамилия

Нормоконтроль: _____ 08.06.17 Доцент, к. т. н. О.А.Антамошкин
подпись, дата должность, ученая степень инициалы, фамилия

Красноярск 2017

РЕФЕРАТ

Бакалаврская работа 41 стр., 18 рисунков, 16 источников.

Объект исследования – защита персональных данных.

Цель работы – разработать кроссплатформенную программную систему хранения параметров учетных записей.

Метод исследования – практический эксперимент.

Результат – изучена предметная область по данной теме, проанализированы действующие менеджеры паролей и их алгоритмы шифрования, также выбран алгоритм шифрования и реализована программа с его использованием.

Оглавление

Введение 4
1. Анализ и обзор программных систем хранения учетных записей 5
1.1 Особенности предметной области и решаемых задач 5
1.2 Анализ существующих менеджеров паролей 7
1.2.1 KeePass 7
1.2.2 1Password 9
1.2.3 Dashlane 11
1.3 Выделение существенных функций программных систем для хранения паролей 13
2. Используемые технологии 14
2.1 Выбор средства разработки. Язык программирования 14
2.2 Среда разработки 18
2.3 Метод шифрования 20
2.4 Метод хранения информации 28
2.4 Архитектура системы 29
2.5 Техническое задание ..... **Ошибка! Закладка не определена.**
3. Реализация и описание программы 33
3.1 Разработка программы 33
3.2 Руководство пользователя 34
Заключение 39
Список использованных источников 40

Введение

В нашем современном обществе главным ресурсом является информация. Информацией владеют и используют абсолютно все. Но каждый человек сам решает, какую информацию ему нужно получить, какой поделиться, а какую лучше оставить при себе и если придется защитить ее.

На сегодняшний день большинство информации имеет цифровой формат. И с помощью выхода в интернет и персонального компьютера или смартфона можно получить практически любую услугу. Теперь не нужно посещать лично большинство организаций и учреждений, потому что можно воспользоваться онлайн-доступом.

Сейчас люди используют множество веб-сервисов, включая социальные сети, электронную почту, операции в банке, государственные порталы предоставления услуг и т.д. Естественно, сервисы нуждаются в аутентификации всех пользователей, и самым распространенным способом является пара логин-пароль. Этот способ является простым и удобным для пользователя.

Отсюда вытекает проблема, что нужно запомнить большое количество комбинаций логина и пароля. Плюс ко всему, пароли должны быть устойчивыми к взлому: длинные, состоящие из разных регистров, с использованием цифр и различных знаков, меняться хотя бы раз в полгода. Однако, такие пароли сложны для запоминания, и со временем пользователь начинает использовать простые пароли или небезопасные методы хранения (текстовый файл, стикер на рабочем столе, браузер с автоматическим хранением логина и пароля).

К счастью, на выручку приходит наиболее удобный и безопасный инструмент – менеджер паролей.

Менеджер паролей представляет собой приложение, которое хранит пароли и другую информацию в зашифрованном виде.

Целью работы является разработка кроссплатформенной программной системы хранения параметров учетных записей на платформе Java.

1. Анализ и обзор программных систем хранения учетных записей

1.1 Особенности предметной области и решаемых задач

Менеджер паролей — программное обеспечение (далее – ПО), которое оказывает помощь пользователю при работе с паролями и PIN-кодами. У такого ПО, как правило, присутствует местная локальная база данных или файлы, в которых хранятся зашифрованные сведения пароля. Большинство систем хранения паролей также работают как заполнитель формы, то есть они заполняют поле логина и данные пароля автоматически. В большинстве случаев они представлены как расширение браузера.

Менеджеры паролей делятся на три основных категории:

- Десктоп — хранят пароли к ПО, установленному на жестком диске или на SSD-накопителе компьютера.
- Сетевые — менеджеры паролей онлайн, где пароли сохранены на веб-сайтах провайдеров.
- Портативные — хранят пароли к программному обеспечению на мобильных устройствах, таких как планшет, смартфон или к портативным приложениям на USB-накопителе [1].

Системы хранения паролей могут также использоваться как защита от фишинга (вид интернет-мошенничества основанного на принципах социальной инженерии, которым движет возможность украсть конфиденциальные данные пользователей — логины и пароли). В отличие от пользователей, менеджер паролей способен использовать автоматизированный скрипт логина не восприимчиво к визуальным подражаниям, которые имеют сходства с веб-сайтами. С этим внедренным преимуществом использование систем хранения паролей благоприятно, даже если у пользователя есть всего несколько паролей, которые он держит в голове [2].

Менеджеры паролей, как правило, применяют выбранный пользователем главный пароль, или же секретное слово, чтобы создать ключ, применяемый для зашифровки хранимых паролей. Этот основной пароль следует делать достаточно сложным, чтобы удержать атаки злоумышленников.

Если главный пароль будет украден, то, следовательно, будут рассекречены все хранимые в БД программы пароли. Это демонстрирует обратную связь между удобством использования и безопасностью: единственный пароль может быть более практичен, но если он будет взломан, то под угрозой окажутся все хранимые пароли.

Основной пароль также может быть обнаружен и атакован при применении криптоанализа. Такая угроза может быть снижена путём использования виртуальной клавиатуры.

Некоторые системы хранения паролей имеют у себя генератор паролей. Созданные пароли могут быть отгадываемыми, если менеджер пароля не использует криптографически безопасный генератор случайных чисел[1].

1.2 Анализ существующих менеджеров паролей

Сейчас на рынке представлен большой выбор различных менеджеров паролей. Проведем анализ трех наиболее популярных программ: KeePass, 1Password, Dashlane. Для анализа выделим данные параметры, которые позволят нам оценить функциональность этих менеджеров паролей наиболее полно:

- Цена;
- Удобство пользовательского интерфейса (где 1 – перегруженный интерфейс и имеет неприятный вид, 5 – перегрузка отсутствует и имеет приятный внешний вид);
- Кроссплатформенность;
- Генератор паролей;
- Перенос базы данных;
- Поиск по записям;
- Автоматический ввод данных в браузер;
- Смена мастер-пароля;
- Портативность.

Представленный анализ существующих приложений поможет нам пополнить и окончательно сформировать перечень функций, необходимых нам для разработки своего менеджера паролей.

1.2.1 KeePass

KeePass Password Safe — свободная и многофункциональная программа. Официальная реализация есть только под Windows, но исходники открыты под GPLv2, поэтому есть огромное количество реализаций, в том числе под Linux и под OS X. Также есть приложения и для мобильных ОС (рисунок 1). База данных паролей шифруется с помощь симметричного AES-256, а мастер-пароль хэшируется с SHA-256. В качестве синхронизации обычно используют либо

старую добрую «флешку», либо один из облачных сервисов, например Dropbox. Некоторые мобильные клиенты, кстати, умеют работать автоматически с хранилищем в Dropbox [3].

Для KeePass есть большое количество плагинов и дополнительных инструментов: утилиты для импорта/экспорта паролей из БД, браузерные плагины, позволяющие автоматически заполнять формы логина, и дополнительные средства бэкапа и синхронизации.

Данный менеджер имеет свои минусы: Сейчас используется две версии (1 и 2) базы данных, несовместимых друг с другом. При этом есть клиенты, поддерживающие только одну из версий. Несмотря на то, что основное приложение бесплатно, существуют и платные клиенты, например под iOS.

Как это часто бывает с подобными проектами, интерфейс у некоторых клиентов оставляет желать лучшего [2].

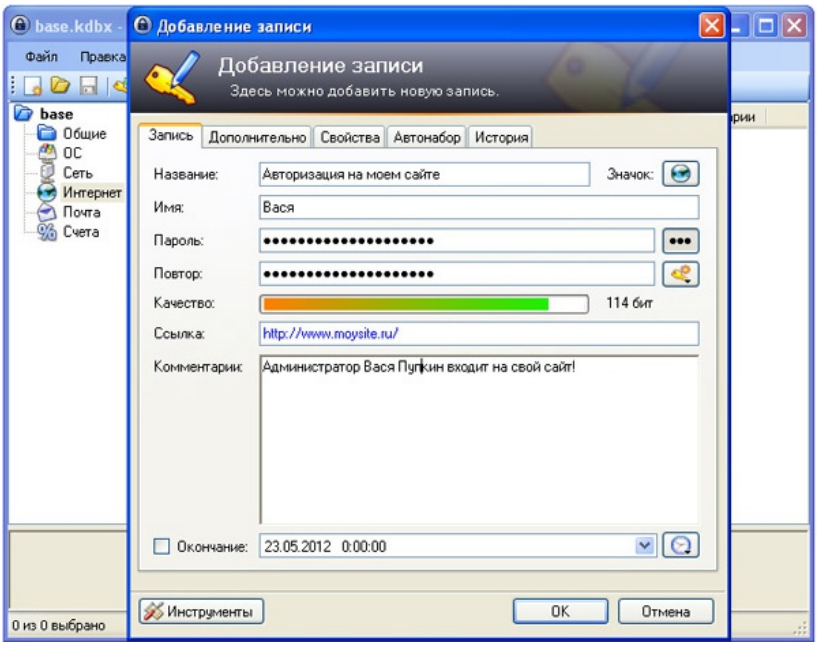

Рисунок 1 - Создание учетной записи интернета в KeePass Password Safe

Анализ:

- Цена – Официальная версия для Windows бесплатна, но так же имеются платные версии приложений;
- Удобство пользовательского интерфейса – 2;
- Кроссплатформенность – присутствует;

- Генератор паролей – присутствует;
- Перенос базы данных – присутствует;
- Поиск по записям – присутствует;
- Автоматический ввод данных в браузер – присутствует;
- Смена мастер-пароля – присутствует;
- Портативность – присутствует.

Помимо этого, KeePass имеет такие функции как:

- Создание записи;
- Дублирование записи;
- Сортировка записей;
- Копирование данных записи и удаление из буфера обмена, скопированного через какое-то определенное время;
- Настройка базы и программы;
- Хранение дат;
- Помимо мастер-пароля поддержка ключевого файла.

1.2.2 1Password

1Password — популярное на Mac OS X решение для хранения паролей, лицензий на ПО и другой персональной информации от компании AgileBits. Также есть версия под Windows, и к тому же предлагается нативный клиент под iOS (рисунок 2).

Все версии 1Password обладают встроенной функцией синхронизации базы с помощью сервиса Dropbox. Эта функциональность не обязательна, база по умолчанию хранится локально. База данных зашифрована AES-128 [4].

1Password for Mac интегрируется с Safari, Firefox, Chrome и Camino «из коробки». Версия для Windows интегрируется с Firefox, Chrome и IE. Также обе версии 1Password предлагают удобный интерфейс для использования хранимой информации в любых других приложениях.

Кроме интеграции с разными платформами, 1Password предоставляет еще один оригинальный способ доступа к своей базе. Хранилище паролей (agile keychain) представляет из себя набор файлов, один из которых — HTML-файл с полноценным интерфейсом для работы с базой, который может быть открыт любым браузером на практически любом устройстве[3].

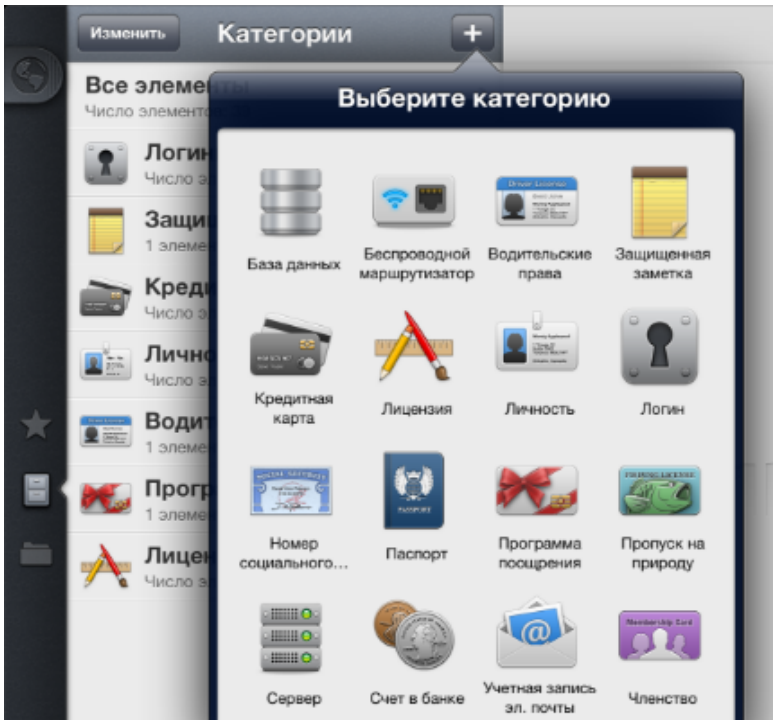

Рисунок 2 – Выбор категории в 1Password

Анализ:

- Цена – первый месяц использования бесплатно, далее 1 месяц – 5$ или навсегда 65$;
- Удобство пользовательского интерфейса – 5;
- Кроссплатформенность – присутствует частично;
- Генератор паролей – присутствует;
- Перенос базы данных – присутствует;
- Поиск по записям – присутствует;
- Автоматический ввод данных в браузер – присутствует;
- Смена мастер-пароля – присутствует;
- Портативность – присутствует.

Также, у 1Password имеются такие функции как:

- Создание записей – быстрые кнопки для создания разных типов;
- Навигационная колонка – все записи, избранное, категории, папки;
- Избранное;
- Модуль для браузеров;
- Поделиться доступом: группы, электронная почта;
- Удаление из буфера обмена, скопированного, через какое-то определенное время.

1.2.3 Dashlane

Dashlane поддерживается Windows, Mac, iOS, Android (рисунок 3). Присутствует функция автоматизированной замены паролей на популярных сайтах и сделать это можно в пару кликов. Также реализован импорт паролей, сохраненных в браузерах Chrome, Firefox и Internet Explorer. Кроме того Dashlane позволяет импортировать пароли из LastPass, RoboForm, Everywhere 7 и нескольких других конкурентных решений[5].

Пароли в базе данных хранятся шифрованными AES-256. Есть возможность синхронизации через собственное облако. Одна из самых интересных фич — возможность использовать двухфакторную аутентификацию через Google Authenticator, что позволяет повысить защищенность данных. Также есть всякие приятные мелочи, типа дашборда безопасности, в котором выводится сводная информация по паролям, или довольно строгие требования к мастер-паролю (разный регистр, цифры, минимум восемь символов). Есть также возможность веб-доступа к паролям[2].

В стандартной поставке также присутствуют браузерные плагины, работающие привычным способом — позволяющие автоматически заполнять известные формы и сохранять результаты введенных паролей.

Dashlane может привлечь своими инструментами для облегчения покупок в сети. Так, он умеет автоматически заполнять всю необходимую финансовую информацию на торговых площадках, включая номера кредитных карт и адрес доставки, отслеживать прохождение платежа и доставку, начислять бонусы за активные покупки и так далее[6].

Интересна также и ценовая политика компании. Во-первых, все устанавливаемые приложения бесплатны, а оплачиваются услуги пользования сервисом. Во-вторых, есть бесплатный тарифный план (без синхронизации, бэкапов и веб-доступа), а есть премиум, стоящий вполне адекватные 40 долларов в год.

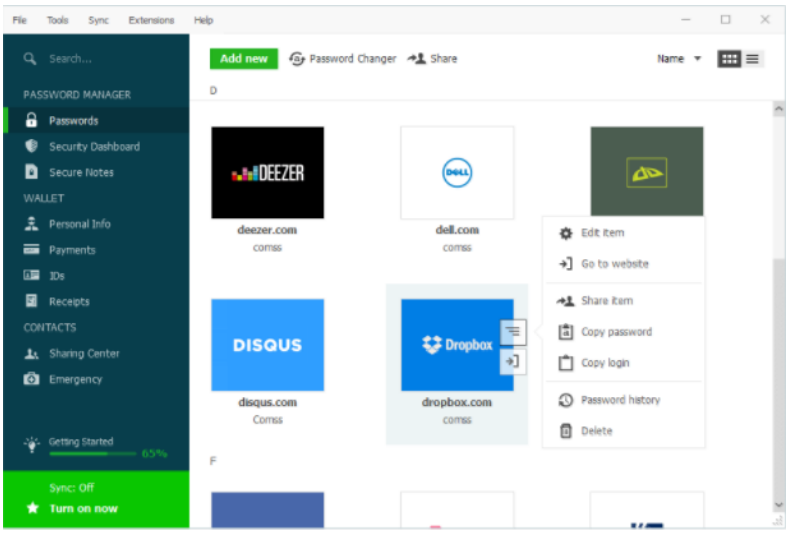

Рисунок 3 – Редактор Dashlane

Анализ:

- Цена – бесплатно (без синхронизации, бэкапов и веб-доступа), 40$ в год за премиум;
- Удобство пользовательского интерфейса – 5;
- Кроссплатформенность – присутствует частично;
- Генератор паролей – присутствует;
- Перенос базы данных – присутствует;
- Поиск по записям – присутствует;
- Автоматический ввод данных в браузер – присутствует;
- Смена мастер-пароля – присутствует;
- Портативность – отсутствует.

Кроме того, у Dashlane имеются такие функции как:

- Создание записей;
- Организация по категориям;
- Автоматизированная замена паролей на популярных сайтах;
- Импорт паролей, сохраненных в браузерах, а также из некоторых менеджеров паролей;
- Двухфакторная аутентификация через Google Authenticator;
- Аудит безопасности - присвоения рейтинга безопасности всем вашим паролям;
- Функция экстренного доступа в случае непредвиденных обстоятельств;
- Сохранение чеков.

1.3 Выделение существенных функций программных систем для хранения паролей

На основе анализа существующих программ был составлен общий список требований к разрабатываемой программной системе хранения учетных записей:

- Кроссплатформенность – одно из самых важных требований, так как это позволяет использовать программу на различных устройствах;
- Удобный пользовательский интерфейс – является важным критерием, так как именно через него пользователей взаимодействует с программой. Очень важно, чтобы интерфейс был интуитивно понятен, а также легкость обучения работе с ним. Трудоемкость решения определенных задач с его помощью;
- Генератор паролей – функция, которая в случайном порядке формирует набор нелогичных символов большей или меньшей степени сложности, зависящей от заданных параметров. Эта функция нужна, чтобы пользователь не сидел и не придумывал свой пароль;
- Бесплатное распространение – менеджер паролей создается с целью упростить жизнь пользователям веб-ресурсов, а не с целью получения материальной выгоды;
- Портативность – эта функция нужна, чтобы я мог воспользоваться нужным веб-ресурсом с любого компьютера и любого места, а не только со своего;
- Ключ-файл – ключ-файл позволяет нам сделать менеджер паролей еще более безопасным, если использовать его в связке с мастер-паролем. Или же использовать его просто вместо мастер-пароля.

2. Используемые технологии

2.1 Выбор средства разработки. Язык программирования

Выше, были выявлены требования и критерии к программной системе хранения паролей, на которые будем ориентироваться при разработке.

При выборе среды реализации всегда сравнивают программные продукты и их возможности. Использование возможностей средств разработки программ позволяет автоматизировать процесс разработки.

Инструментальные средства позволяют:

- создавать интерфейс, используя стандартные компоненты;
- создавать базы данных и оболочки для них же;
- разрабатывать более надежные программы.

Современные средства разработки характеризуются параметрами:

- поддержка объектно-ориентированного программирования (далее ООП);
- поддержка баз данных;
- возможность использования CASE-технологий, как для проектирования разрабатываемой системы, так и для разработки моделей реляционных баз данных;
- использование визуальных компонентов для наглядного проектирования интерфейса.

Свойствами, перечисленными выше обладают такие языки программирования как:

- C++;
- C#;
- Java.

Каждое из этих средств разработки содержит весь спектр современного инструментария, который был перечислен ранее. Главное отличие состоит в области использования рассматриваемых средств [7].

При решении поставленной задачи для меня было оптимальным использовать язык программирования Java, который является языком высокого уровня. Первая причина, по которой выбор был сделан в пользу Java, является тот факт, что он является самым популярным языком программирования последних двух лет, по мнению авторитетных аналитических изданий. По версии Red Monk по данным за январь 2017 года – Java занимает второе место после JavaScript по популярности[8]. В издание Tiobe – Java занимает первое место с большим отрывом от языка C по данным за июнь 2017 года[9]. Но, ни это является его плюсом, а, то из-за чего Java возглавляет этот рейтинг.

Java легок в изучении. Синтаксис языка очень прост и похож на обычный английский[10], в нем минимум символов (например, угловых скобочек), что значительно упрощает понимание и изучение этого языка.

Как упоминалось ранее, Java является объектно-ориентированным языком программирования. Если в процессе разработки применять принципы ООП, то структуру приложения можно организовать по модульному принципу, а само приложение и его код, становится легко читаемым, гибким и простым в расширении[10]. В Java используются основные принципы ООП, такие как абстракции, наследование, полиморфизм и инкапсуляция. Также в язык Java внедрены лучшие техники построения шаблонов. Язык поддерживает основные принципы дизайна классов в ООП в виде проектов с исходным кодом, таких как Spring, а так же позволяет легко проводить манипуляции над проектом при помощи методологии внедрения зависимостей.

К тому же, Java имеет проработанный API[10]. Функционал API у языка весьма обширный и, что особенно важно, он достаточно явный, так как поставляется в рамках инсталляционного пакета Java. Java предоставляет API

для систем ввода/вывода, сетевой инфраструктуры, различных утилит, XML парсинга, соединения с базой данных и т.д. Для всего остального существуют библиотеки с открытым исходным кодом такие, как Apache Commons, Google Guava и другие.

Помимо этого, у Java большая коллекция библиотек с открытым исходным кодом, которые гарантируют, что язык программирования Java желательно использовать везде. Apache, Google, Guava и другие организации внесли свой вклад в большое количество библиотек, которые позволяют разработку на Java сделать быстрее, легче и рентабельней[10].

Еще одной из самых сильных сторон является отлично развитое сообщество разработчиков. Неважно, насколько хорош язык программирования, потому что если нет сообщества, которое поддерживало бы его, помогало и делилось знаниями, то он бы не выжил. У Java есть множество активных форумов, Stackoverflow, организаций с открытым исходным кодом и т.д. Существуют различные сообщества, которые помогают, начиная от новичков и кончая экспертами в своем деле[10]. Очень многие делятся своим опытом, дают консультации и не стесняются задавать вопросы, которые их волнуют. И эти обсуждения, которые происходят непосредственно между программистами, дают огромную поддержку и уверенность новичкам в Java.

Кроссплатформенность. Java изначально задумывалась как нечто мультиплатформенное. Написано большое количество приложений, крупных проектов, которые используют именно эту платформу[10].

Нельзя не отметить и то, что Java являлся и является полностью бесплатным языком программирования.

Подведем итоги и выделим все плюсы языка/среды программирования Java:

- Легкость и простота изучения;
- Имеет обширно проработанный и функциональный API;

- Отличная поддержка сообщества;
- Большая коллекция библиотек с открытым исходным кодом;
- Кроссплатформенность;
- Является объектно-ориентированным языком программирования;
- Является полностью бесплатным.

Не будем забывать, что Java – это независимая платформа. В 1990-х годах это было главной причиной популяризации Java. Идея независимой платформы просто отлична, а слоган в Java «Напиши один раз – запускай где угодно» был и остается достаточно привлекательным для новой разработки в Java.

Что касается других языков – C++ тоже является популярным, но имеет одну опасную черту: указатели. Это наиболее мощная и наиболее опасная черта C++. Причиной большинства ошибок в коде является именно неправильная работа с указателями. Например, одна из типичных ошибок – просчитаться на единицу в размере массива и тем самым испортить содержимое ячейки памяти, расположенной вслед за ним. Хотя в Java дескрипторы объектов и реализованы в виде указателей, в ней отсутствует возможности работы с ними напрямую. Вы не можете преобразовать целое число в указатель, а также обратиться к произвольному адресу памяти. Если же говорить про C#, то можно сказать, что они очень похожи с Java. Но в отличие от Java, C# - язык относительно новый. Поэтому на Java больше библиотек и API (многие из которых открытые), чем на C#. Также по Java можно найти намного больше литературы, чем по C#.

2.2 Среда разработки

Средой разработки выбрана Eclipse (“затмение”) (рисунок 4). Одна из основных причин – опыт создания предыдущих проектов в этой среде. Так же она является одной из двух полностью бесплатных сред разработки для языка Java с исходным кодом и, как следствие одной из самых популярных сред разработки на Java. Среда очень быстро развивается. Разрешено ее применение в коммерческих целях. Широко используется для программирования на Java.

Первой среди сред разработки Java обеспечила развитые средства для добавления в среду новых возможностей с помощью плагинов. Благодаря этому стала основой (платформой) для многих коммерческих сред разработки — MyEclipse, BEA Workshop, IBM Rational Application Developer, Borland JBuilder и многих других. Существует несколько сайтов репозиториев по умолчанию, в которых содержится большое количество плагинов, а также можно назначать свои адреса репозиториев. Имеется разделение плагинов по категориям в зависимости от области применения. Однако отсутствует сертификация плагинов и подробное описание их назначения и функциональности. Работа с плагинами настраивается через пункт меню Software Updates (Обновления программного обеспечения), находящийся в пункте меню Help[11].

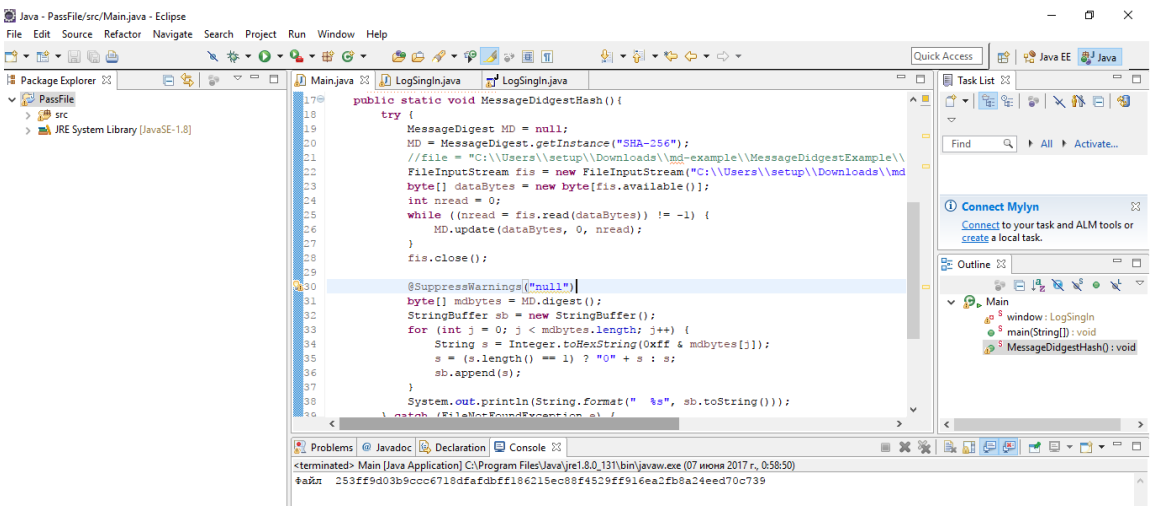

Рисунок 4 – Интерфейс Eclipse Java EE IDE

Среда Eclipse является промышленным стандартом де-факто для разработки программ на языке C++. Eclipse в отличие от старых версий не требует инсталляции на компьютере — запускается непосредственно из набора файлов дистрибутива. Имеет удобный отладчик и встроенный органайзер. При редактировании исходного кода при навигации по классам проекта не обеспечивает переход в исходный код классов JDK, но позволяет легко настроить путь к архиву src.zip из JDK, после чего навигация работает[11]. Подобным же образом можно настроить переход в исходные коды любых используемых классов. Имеет удобное переключение "перспектив" в правой верхней части окна — наборов показываемых на экране окон (один набор служит для редактирования исходного кода, другой — для отладки, и т. д.). Отличается интересной, но несколько раздражающей особенностью редактора исходного кода — при наведении мыши на имя элемента сразу показывается всплывающая подсказка по этому элементу.

Поддерживает работу с языками Java, C, C++, Ruby, Groovy, JavaScript, JRuby, Rails, Grails, Flex, XSL, HTML, CSS, XML, XHTML, технологиями JSP, EJB, AJAX, GWT, Hibernate, JPA, Spring, Web Services, Struts. Поддерживает тестирование с помощью модулей JUnit и импорт проектов JBuilder[11].

2.3 Метод шифрования

SHA-256

Хэш-функции – это функции, предназначенные для «сжатия» произвольного сообщения или набора данных, записанного, как правило, в двоичном алфавите, в некоторую бинарную комбинацию фиксированной длины, называемую сверткой.

Хэш-функции применяются для решения следующих задач:

- построение систем контроля целостности данных при их передаче или хранения;
- аутентификации источника данных.

SHA-256 – специальный алгоритм шифрования, относящийся к семейству криптографических алгоритмов SHA-2, создающих так называемые «отпечатки» сообщений, длина которых может быть произвольна.

SHA-2 разработан Национальным институтом стандартов и технологий (NIST) с целью замены устаревшего алгоритма хеширования SHA-1 с возможными математическими недостатками.

Описание алгоритма

Хеш-функции семейства SHA-2 построены на основе структуры Меркла — Дамгарда.

Исходное сообщение после дополнения разбивается на блоки, каждый блок — на 16 слов. Алгоритм пропускает каждый блок сообщения через цикл с 64 или 80 итерациями (раундами). На каждой итерации 2 слова преобразуются, функцию преобразования задают остальные слова. Результаты обработки каждого блока складываются, сумма является значением хеш-функции. Тем не менее, инициализация внутреннего состояния производится результатом

обработки предыдущего блока. Поэтому независимо обрабатывать блоки и складывать результаты нельзя[12].

Алгоритм SHA-256 работает с данными, разбитыми на куски по 512 бит (64 байт), криптографически их смешивает и выдаёт 256-битный (32 байта) хэш. SHA-256 состоит из относительно простого раунда, повторяющегося 64 раза. Снизу, как раз, и показан такой раунд, принимающий на вход 8 4-байтовых слов — от A до H[13] (рисунок 5).

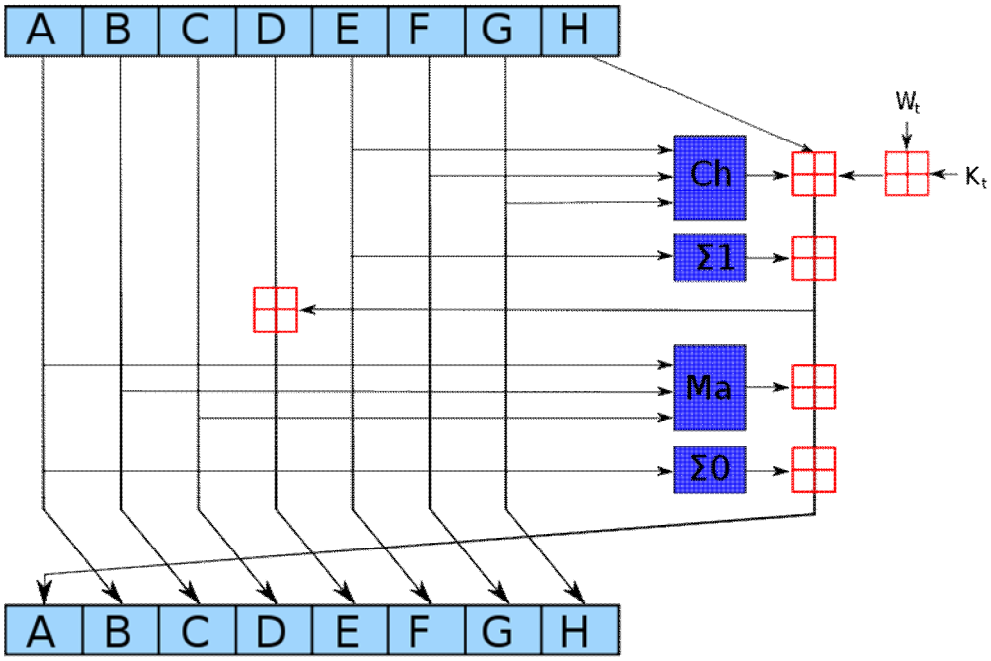

Рисунок 5 – Схема одного раунда SHA-256 для восьми входных слов A-H

Хэш-функция нужна для того, чтобы не хранить сам мастер-пароль в БД, а также для реализации входа с помощью файл-ключа.

Дальше рассмотрим два подходящих для реализации в программе алгоритма шифрования: AES-256 и RSA.

Алгоритм RSA

История алгоритма

RSA (аббревиатура от фамилий ученых Рональда Ривеста (Rivest), Ади Шамира (Shamir) и Леонардо Адлемана (Adelman) Массачусетского технологического института (MIT)) – криптографический алгоритм с открытым ключом, основывавшиеся на вычислительной сложности задачи факторизации больших целых чисел.

Позже они втроем основали компанию RSA Data. В 1989 году RSA, упоминается в RFC 1115 вместе с симметричным шифром DES, давая тем самым старт применения алгоритма в только начавшей свою работу сеть Интернет, а в 1990 году алгоритм принят на вооружение министерством обороны США.

В ноябре 1993 года открыто публикуется версия 1.5 стандарта PKCS1, где описывается применение алгоритма RSA для создания цифровой подписи и шифрования.

Описание алгоритма

В основе RSA лежит задача факторизации произведения двух простых больших чисел. Для шифрования используется простая операция возведения в степень по модулю N. Для расшифровки же необходимо вычислить функцию Эйлера от числа N, для этого необходимо знать разложение числа n на простые множители (В этом и состоит задача факторизации).

В RSA открытый и закрытый ключ состоит из пары целых чисел. Закрытый ключ хранится в секрете, а открытый ключ сообщается другому участнику, либо где-то публикуется.

Генерация ключей RSA. Всё начинается с генерации ключевой пары (открытый, закрытый ключ). Генерация ключей в RSA осуществляется следующим образом[14]:

1. Выбираются два простых числа p и q (такие что p неравно q).

2. Вычисляется модуль N=p*q.

3.Вычисляется значение функции Эйлера от модуля N:□(N)=(p−1)(q−1).

4. Выбирается число e, называемое открытой экспонентой, число e должно лежать в интервале 1<e<□(N), а так же быть взаимно простым со значением функции □(N).

5. Вычисляется число d, называемое секретной экспонентой, такое, что d*e=1(mod□(N)), то есть является мультипликативно обратное к числу e по модулю □(N).

Итак, мы получили пару ключей:

Пара (e,N) - открытый ключ. Пара (d,N) - закрытый ключ.

Шифрование и расшифровывание в RSA. Есть следующий сценарий: Боб и Алиса переписываются в интернете, но хотят использовать шифрование, чтобы поддерживать переписку в секрете :). Алиса заранее сгенерировала закрытый и открытый ключ, а затем отправила открытый ключ Бобу. Боб хочет послать зашифрованное сообщение Алисе:

Шифрование: Боб шифрует сообщение mm, используя открытый ключ Алисы (e,N) :$C=E(M)=M^e \bmod(N)$, и отправляет с Алисе.

Расшифровывание: Алиса принимает зашифрованное сообщение c. Используя закрытый ключ (d,N), расшифровывает сообщение $M=D(C)=C^d \bmod(N)$ (рисунок 6).

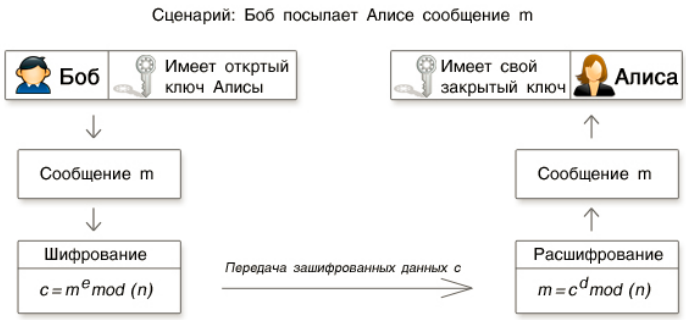

Рисунок 6 – Боб посылает Алисе сообщение m

Алгоритм AES-256

Advanced Encryption Standard (AES), также известный как Rijndael (Рэндал) — симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES. Этот алгоритм хорошо проанализирован и сейчас широко используется, как это было с его предшественником DES. По состоянию на 2009 год AES является одним из самых распространённых алгоритмов симметричного шифрования.

Определения:

- State (форма)– матрица (двумерный массив) байтов, расположенных следующем образом (рисунок 7);

0	4	8	12	...
1	5	9	13	...
2	6	10	14	...
3	7	11	15	...

Рисунок 7 – Матрица байтов

- Round (раунд) – итерация цикла преобразования над State (формой). Количество итераций зависит от длины ключа, чем больше длина ключа, тем больше итераций;
- Round key (раундовый ключ) — уникальный ключ, который применяется в каждом отдельном в раунде;
- S-Box (таблица подстановок) – таблица, которая задает отображение одного байта в другой (биективное отображение) (рисунок 8);

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Например, байт {fe} заменится на {bb}.

Рисунок 8 – S-Box (таблица подстановок)

- Обратная таблица подстановок — аналогично S-Box, задает обратное отображение (рисунок 9);

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Рисунок 9 – Обратная таблица подстановок

Шифрование состоит из следующих функций преобразования (рисунок 10):

- ExpandKey — Функция для вычисления всех раундовых ключей;
- SubBytes — Функция для подстановки байтов, использующая таблицу подстановок;
- ShiftRows — Функция, обеспечивающая циклический сдвиг в форме на различные величины;
- MixColumns — Функция, которая смешивает данные внутри каждого столбца формы;
- AddRoundKey — Сложение ключа раунда с формой.

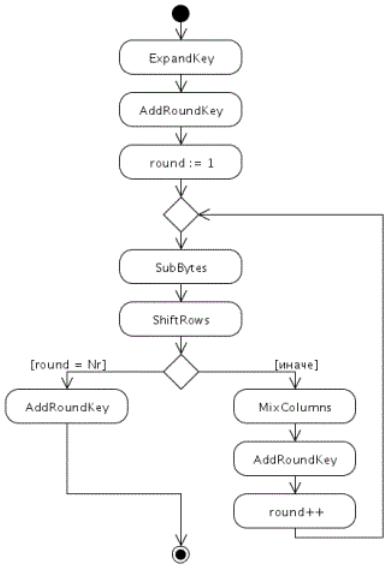

Рисунок 10 – Шифрование в виде блок схемы

Где Nr – количество раундов.

Расшифровывание происходит аналогичным образом, только в обратном порядке[15].

Считается, что используемый в AES ключ длинной в 128 бит – достаточно надежная защита против лобовой атаки, то есть с чисто математической точки зрения подобрать один правильный пароль из всех возможных – трудноосуществимая задача. Несмотря даже на некоторые недостатки AES, взломать защищенную этим алгоритмом информацию практически невозможно.

Любой криптографический алгоритм требует ключ размером в то или иное количество бит, чтобы зашифровать данные. Длина ключа, используемая при шифровании, и определяет практическую целесообразность выполнения полного перебора, ведь информацию, зашифрованную более длинными ключами экспоненциально сложнее взломать, чем с короткими. Наконец, лучше всего за AES говорит статистика: защищенные этим алгоритмом данные никогда не были взломаны. Впрочем, все это работает при размере ключа минимум в 128 бит, поскольку более ранние шифровальные алгоритмы все же не выдерживали испытания на прочность. Несмотря на то, что скорость

вычисления компьютеров увеличивается в геометрической прогрессии согласно закону Мура, 128-битного ключа вполне должно хватить на много лет вперед.

Остановим свой выбор на алгоритме AES-256, так как благодаря описанным преимуществам, AES остается предпочтительным алгоритмом для правительственных организаций, банков и других систем требующих высокий уровень безопасности, по всему миру.

2.4 Метод хранения информации

Защищенная информация имеет вид таблицы, где каждая строка это отдельная запись, предназначенная для одного сервиса, а каждый столбец это параметр записи, а каждый столбец это параметры записи: название, URL, логин, пароль №1, пароль №2, комментарий.

Так как у нас очень простая структура хранения данных, то использовать БД будет нецелесообразно в виду затраты ресурсов и дополнительной сложности разработки.

Разрабатываемая программная система хранения параметров учетных записей использует файл формата .XML

XML (Extensible Markup Language) - это язык разметки документов, позволяющий структурировать информацию разного типа, используя для этого произвольный набор инструкций.

Сегодня XML может использоваться в любых приложениях, которым нужна структурированная информация - от сложных геоинформационных систем, с гигантскими объемами передаваемой информации до обычных "однокомпьютерных" программ, использующих этот язык для описания служебной информации [16].

XML-документ представляет собой обычный текстовый файл, в котором при помощи специальных маркеров создаются элементы данных, последовательность и вложенность которых определяет структуру документа и его содержание. Основным достоинством XML документов является то, что при относительно простом способе создания и обработки (обычный текст может редактироваться любым тестовым процессором и обрабатываться стандартными XML анализаторами), они позволяют создавать структурированную информацию, которую хорошо "понимают" компьютеры.

2.4 Архитектура системы

При запуске исполняемого файла программы выдает диалоговое окно (класс LogSignIn), в котором пользователь вводит логин, и по логину уже определяется какой файл формата .xml подключить к системе, которые хранятся в папке при программе, и какой тип защиты выбран при регистрации: мастер-пароль или файл-ключ. Дальше вводится пароль или же указывается путь к файлу-ключу. Для аутентификации пользователя в файле хранится хэш мастер-пароля или файл-ключа, который сравнивается с прошедшим хэш-функции введенного мастера-пароля или указанного файл-ключа. Если же хэш совпадает, то программа запускает основное окно (класс MainWindow) (рисунок 11).

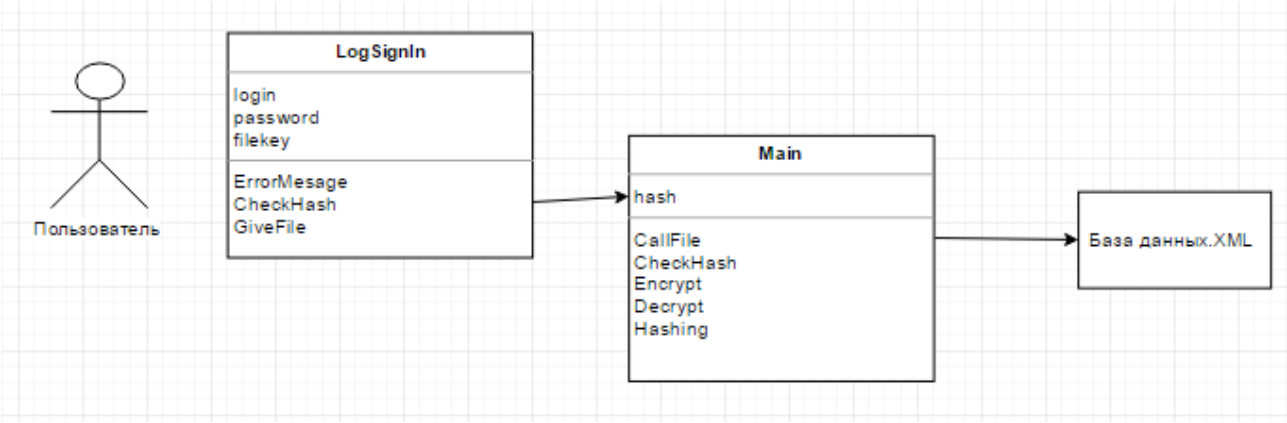

Рисунок 11 – Проверка соответствия пароля

Класс MainWindow выгружает из файла в JList все поля названий из таблицы.

При выборе одного из названий в JList двойным щелчком мыши, программа запрашивает оставшиеся данные этой записи из соответствующих полей в файле базы данных формата .xml. Методы класса Main дешифруют данные оставшиеся данные и выгружают их в главное окно MainWindow, по

очереди заполняя поля. При отсутствии информации в поле, MainWindow не отображает пустой экземпляр поля, а пропускает его.

Основой всей программной системы является класс Main, который содержит классы и методы для шифровки и дешифровки данных, а также хэш-функцию. Он взаимодействует с классами LogSignIn, MainWindow, AddRec (рисунок 12).

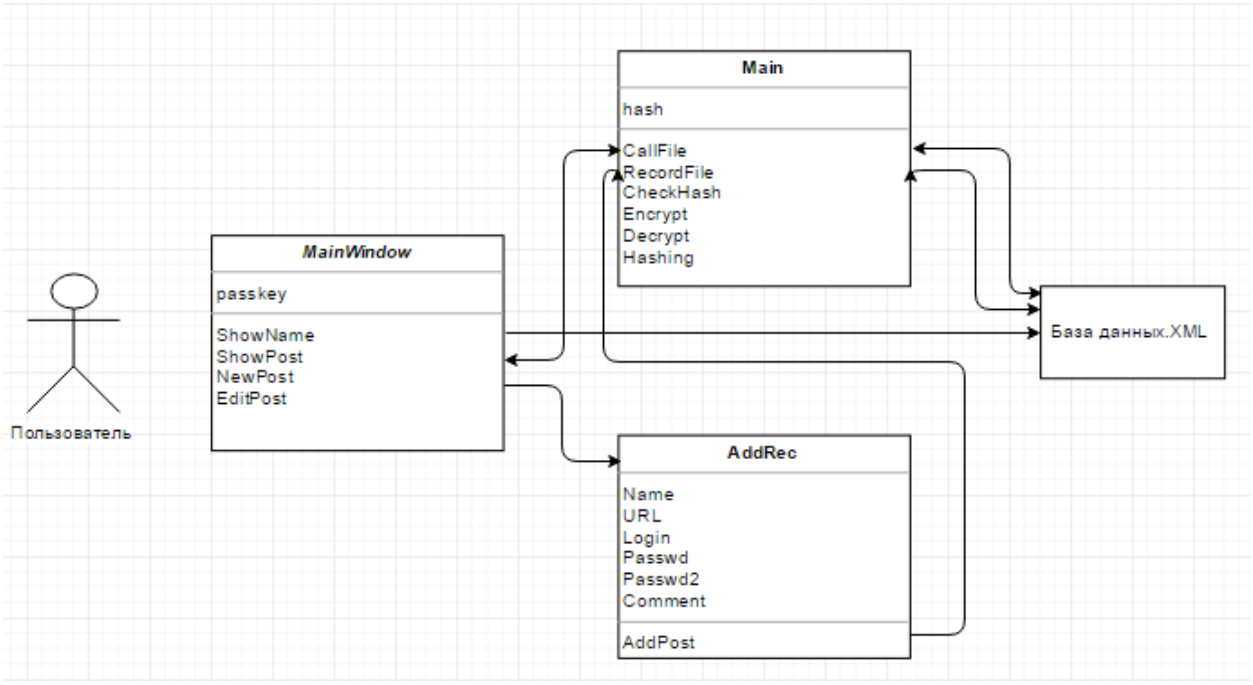

Рисунок 12 – Главное окно, создание записи

2.5 Техническое задание

1.Наименование

Наименование: «Разработка кроссплатформенной программной системы хранения параметров учетных записей»

2.Назначение и цели создания системы

Назначение: Хранение в зашифрованном виде и использование параметров учетных записей пользователя для аутентификации.

Цели создания системы: Исследование методов защиты персональных данных и информации в учебных целях, а также для персонального использования.

3.Требования к структуре и функционированию системы

Требование к структуре: проект должен быть разбит на несколько частей (по функциональности и выполняемым задачам).

Требование к функционированию:

- Для дешифровки данных система должна запрашивать мастер-пароль или путь к файл-ключу;
- Должна быть реализована хэш-функция;
- Должны быть предусмотрены такие функции как: создание, удаление и редактирование учетных записей;
- Шифрование данных алгоритмом AES-256;
- Генератор паролей;
- Возможность хранения двух паролей к учетной записи (при наличие двухфакторной аутентификации).

4.Требование к интерфейсу системы

- Интерфейс должен быть интуитивно понятен;
- Интерфейс должен отображать настраиваемые параметры системы;

- Интерфейс должен быть разделен на две основные части: поле со списком названий и информации о выбранной учетной записи;
- Интерфейс при запуске программной системы должен иметь возможность регистрации и входа.

5.Требование к техническому обеспечению

- Базовый язык реализации – Java;
- Версия компилятора – JDK 1.8;
- Система сборки – Apache Maven используемы в среде разработки;
- IDE - Eclipse Java EE Luna.

3. Реализация и описание программы

3.1 Разработка программы

При создание аккаунта пользователь должен придумать логин и выбрать один из двух видов защиты: мастер-пароль или файл-ключ. Пароль или же файл-ключ является ключом для дешифровки. Для хранения параметров был выбран текстовый формат хранения данных .xml. Для проверки правильности введенных данных при авторизации программа сверяет хэш мастер-пароля или файл-ключа, который находится всегда на первой строке файла.

Шифруются только параметры учетных записей, так первое поле хранит хэш, а второе поле является названием записи, отображаемым в JList. Дешифровка информации производится только после выбора записи в JList.

Интерфейс программы задумано было сделать максимально удобным и интуитивно понятным для пользователя. Решено было сделать добавление параметров через отдельное окно (класс AddRec) с дублированием функционала в таблице основного окна, поскольку такой вариант представляется наиболее удобным и визуально комфортным.

3.2 Руководство пользователя

Установка программы:

1) Windows:

Для установки программы в операционную систему Windows необходимо скачать репозиторий по ссылке https://github.com/stefor4/passkey , запустить файл setup.exe находящийся в корневой папке. Установщик проверит соответствие версии Java. При отсутствии установленных пакетов JDK в системе, установщик перенаправляет на сайт java.com. После установки java в систему, необходимо повторно запустить установщик.

При повторном запуске установщика нужно выбрать путь к папке, в которую будет установлена программа и поставить галочку «Создать ярлык на рабочем столе», если необходимо. В конце выбираем галочку «Запустить менеджер учетных записей», если необходим запуск после установки.

2) Linux (на примере Ubuntu 16.04.1)

Для работы в программе необходимо установить Java.

Введите в терминале: sudo apt-get install openjdk-7-jdk openjdk-7-jre

Введите Y(Yes) для подтверждения установки.

Введите команду для установки программы:

$git clone https://github.com/stefor4/passkey

$./ configure

$ make

$ make install

Авторизация в программе

Для начала работы с программой пользователю необходимо зарегистрироваться. Регистрация проходит в один этап. Пользователь придумаем логин, и выбирает средство защиты: мастер-пароль или файл-ключ. В зависимости от выбора ему нужно либо придумать пароль согласно требованиям (пароль должен быть длинной больше 10 символов, иметь оба регистра и цифры), или же указать путь до файл-ключа (рисунок 13).

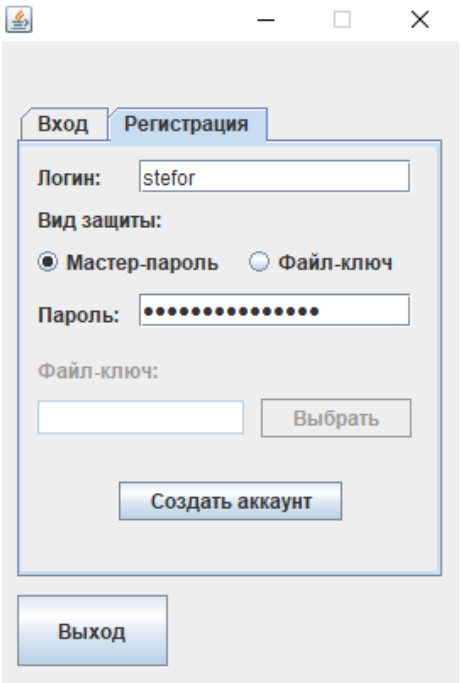

Рисунок 13 – Окно регистрации

При входе в программу нам нужно ввести логин и нажать кнопку «ОК». После нажатия нал подсветятся те поля, которые мы выбрали для защиты. Если это мастер-пароль, то нам будет нужно ввести пароль, если же файл-ключ, то указать путь к файлу (рисунок 14).

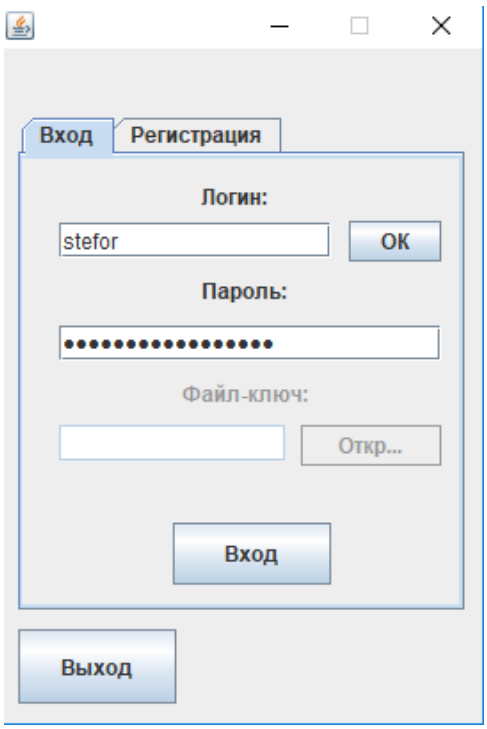

Рисунок 14 – Окно входа

Основное окно программы:

В основном окне программы слева отображаются названия учетных записей, справа расположено поле, в котором отображаются параметры учетных записей пользователя. В поле информации имеются строки: название учетной записи, логин, пароль №1, пароль №2, комментарий. Также имеются кнопка «Добавить» для создания новой учетной записи и кнопка «Редактировать» для редактирования и удаления учетных записей (рисунок 15).

Портативность:

Портативность осуществляется переносом файла формата .xml закрепленного за пользователем с одного устройства, на другое. Файл хранится в папке установленной программы и туда же должен перенесен.

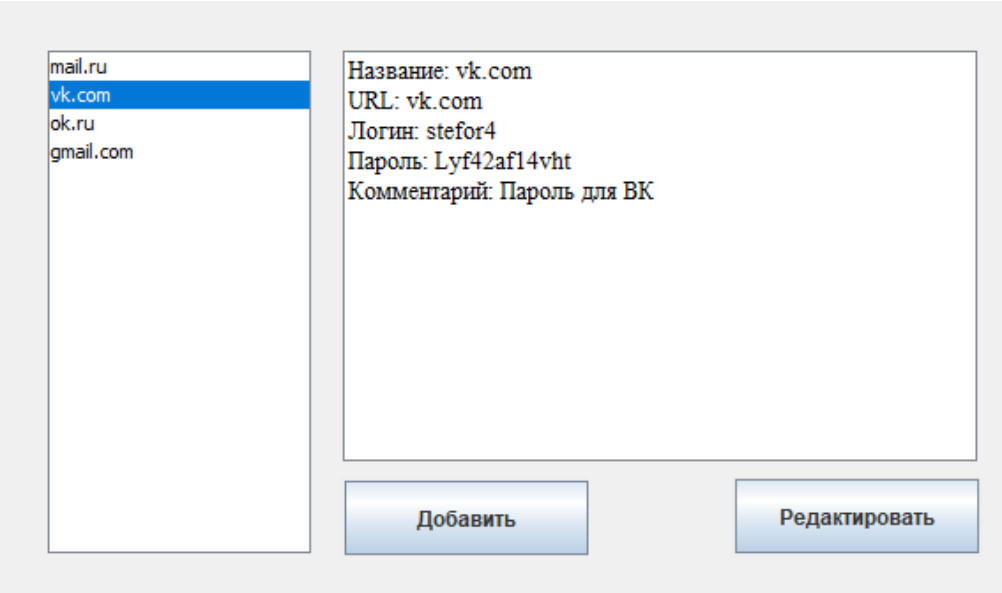

Рисунок 15 – Основное окно программы

Создание записи:

Для добавления новой учетной записи нужно нажать на кнопку «Добавить». Появится окно и в нем прописаны пункты, которые нужно заполнить (обязательные для заполнения помечены звездочкой). Перечислены все те пункты, что и в поле информации в главном окне. Пароль можно сгенерировать одной кнопкой (рисунок 16).

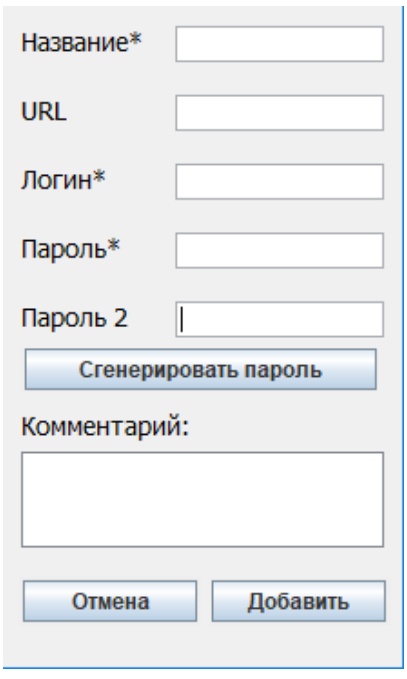

Рисунок 16 – Окно добавления учетной записи

Удаление записи:

Для удаления учетной записи, необходимо нажать кнопку «Редактировать», и в окне редактирования записи нажать «Удалить запись». редактирование производится в этом же окне. Сохранение редактирования происходит после нажатия кнопки «Готово».

Сообщения об ошибках:

В случае неккоректно заполненных полей на этапе авторизации или добавления записи программа выдаст сообщение об ошибках (рисунок 17, 18).

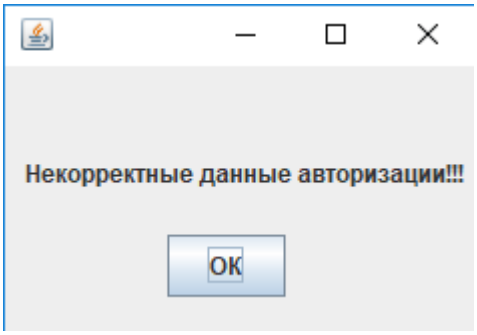

Рисунок 17 – Окно добавления учетной записи

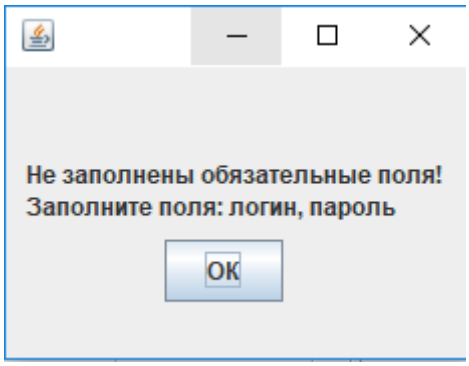

Рисунок 18 – Окно добавления учетной записи

Заключение

Цель ВКР была достигнута – разработана кроссплатформенная программная система хранения параметров учетных записей с использованием алгоритма AES-256. Изучена предметная область, проанализировано несколько программ и алгоритмов шифрования. Так же в ходе работы были выявлены недостатки алгоритма SHA-256 и AES-256, которые будут исправлены в дальнейшей разработке этой программы. Получены теоретические знания, а так же практические навыки по разработке программы с использованием данного алгоритма. Программа разработана для личного использования в исследовательских целях.

Список использованных источников

1. Яремчук С. Менеджеры паролей / С. Яремчук // Мой Компьютер, 2007г. - №03. – 245 стр.

2. Пиковский М. Обзор кроссплаформенных менеджеров паролей [Электронный ресурс]: // Интернет-журнал «Хакер» / М.Пиковский – Режим доступа: https://xakep.ru/2013/10/31/cross-platform-password-managers/ Дата обращения: 20.04.17

3. Популярные менеджеры паролей в сравнение [Электронный ресурс]:// Новостной форум «Хабрахабр» - Режим доступа: https://habrahabr.ru/post/125248/ Дата обращения: 20.04.17

4. Завертайлов В. – Менеджеры паролей – краткий обзор [Электронный ресурс]:// Новостной форум «Хабрахабр» / В.Завертайлов – Режим доступа: https://habrahabr.ru/post/225053/ Дата обращения: 20.04.17

5. Обзор Dashlane 4 [Электронный ресурс]:// Новостной сайт «Comss1» - Режим доступа: https://www.comss.ru/page.php?id=2839 Дата обращения: 20.04.17

6. Горчаков Д. – Keepasss vs Dashlane vs LastPass. Выбираем лучший менеджер паролей. [Электронный ресурс]:// Интернет-журнал «Лайвхакер» / Д.Горчаков – Режим доступа: https://lifehacker.ru/2014/01/10/keepass-vs-dashlane-vs-lastpass-vybiraem-luchshij-menedzher-parolej/ Дата обращения: 20.04.17

7. Обзор и обоснование выбора инструментальных средств разработки программного комплекса для автоматизированной системы управления. [Электронный ресурс] – Режим доступа: http://megaobuchalka.ru/1/25141.html Дата обращения: 10.05.17

8. O'Grady S. The RedMonk Programming Language Rankings: January 2017 [Электронный ресурс] ://Аналитическая компания RedMonk / S. O'Grady –

Режим доступа: http://redmonk.com/sogrady/2017/03/17/language-rankings-1-17/ Дата обращения: 10.05.17

9. TIOBE Index for June 2017 [Электронный ресурс] – Режим доступа: https://www.tiobe.com/tiobe-index/ Дата обращения: 10.05.17

10. Харланчук С. – 10 причин, почему стоит начать учить язык программирования Java и почему он лучший [Электронный ресурс] ://Международная бизнес-школа «Digitov» / С. Харланчук – Режим доступа: https://digitov.com/article/10-reasons-why-you-should-start-learning-the-Java-programming-language Дата обращения: 10.05.17

11. Монахов В. В. Язык программирования Java и среда NetBeans. 3-е издание: учебное пособие / В. В. Монахов - СПб.: БХВ-Петербург, 2011. — 704 с.

12. SHA-2 [Электронный ресурс]:// Свободная энциклопедия «Wikipedia» - Режим доступа: https://ru.wikipedia.org/wiki/SHA-2 Дата обращения: 25.05.17

13. Майним Bitcoin с помощью бумаги и ручки [Электронный ресурс]:// Новостной форум «Хабрахабр» - Режим доступа: https://habrahabr.ru/post/258181/ Дата обращения: 25.05.17

14. Алгоритм шифрования RSA на пальцах [Электронный ресурс]:// Портал посвященный вопросам информационной безопастности «Technology Box» - Режим доступа: http://teh-box.ru/informationsecurity/algoritm-shifrovaniya-rsa-na-palcax.html Дата обращения: 25.05.17

15. Алгоритм шифрования AES для самых маленьких [Электронный ресурс]:// Портал посвященный вопросам информационной безопасности «Technology Box» - Режим доступа: http://teh-box.ru/programming/algoritm-shifrovaniya-aes-dlya-samyx-malenkix.html Дата обращения: 25.05.17

16. Язык XML - практическое введение. Часть 2 [Электронный ресурс]:// ИТ-форум «Citforum» - Режим доступа: http://citforum.ru/internet/xml2/part1.shtml Дата обращения: 25.05.17