Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Программа «Разработчик профессионально-ориентированных компьютерных технологий»

Выпускная квалификационная работа

«WEB-ИНТЕРФЕЙС ДЛЯ СИСТЕМЫ ЦИФРОВЫХ СЕРТИФИКАТОВ»

Работу выполнил: **Кизилов Дмитрий Михайлович**

Очно-заочная форма обучения

Научный руководитель: к. ф.-м. н., с.н.с. лаб. ОИТ **Намиот Д.Е.**

Москва

2014

ОГЛАВЛЕНИЕ

АННОТАЦИЯ

В работе рассматривается задача подтверждения факта владения мобильным телефоном. Предлагается реализация модели цифровых сертификатов для мобильных телефонов, согласно которой каждый мобильный пользователь может создать некоторую цифровую метку для своего телефона и подписать ее с помощью ссылки на свой профайл в социальной сети. По базе цифровых сертификатов возможен поиск как на основе идентификаторов мобильного телефона, внесенных в базу данных, так и по профайлам социальных сетей.

ВВЕДЕНИЕ

В работе рассматривается модель цифровых сертификатов для мобильных устройств, позволяющая установить факт прав обладания мобильным устройством. Работа является продолжением первой реализации модели [1] (Колосова А. И., Намиот Д. Е., 2013).

Возможность доказать право владения мобильным устройством в случае утери и кражи, является очень важной для владельца мобильного телефона. Обычно, в мобильном телефоне или ином устройстве хранится достаточное количество важной и личной информации, которую может быть сложно восстановить и которая не предназначена для посторонних лиц. Как-то - список контактов, смс и e-mail переписка, логины и пароли от сайтов, фотографии и др., в зависимости от сложности устройства. Поэтому нахождение утерянных аппаратов, равно как и противодействие их использованию, является важной задачей.

В данной работе рассматриваются мобильные устройства с операционной системой Android – смартфоны, планшеты и др. Это, в большей степени, практический выбор, обусловленный удобством программирования. Рассматриваемая модель может быть применена и к другим мобильным операционным системам. У всех мобильных аппаратов имеются различные идентификационные номера. Именно эти уникальные идентификаторы и могут быть использованы для нахождения утерянных телефонов, мониторинга установок некоего приложения, генерации технических средств зашиты авторских прав. Например, мобильные операторы, при наличии определенного оборудования могут полностью или частично прекратить обслуживать украденный телефон, перенаправлять SMS-сообщения с него на другой телефон. Или отследить его место нахождения по GPS [2]. В России подобная практика не так распространена, как в некоторых других странах, однако и у нас есть случаи нахождения украденных телефонов по IMEI номеру.

ПОСТАНОВКА ЗАДАЧИ

Первая реализация модели цифровых сертификатов представлена мобильным android приложением, разработанным на IDE Eclipse с помощью плагина Android Developer Tools [3]. Для аутентификации на Facebook, с последующим сохранением профайла социальной сети в базу данных – MySQL, использовалась библиотека FacebookSDK. Причем данные только добавляются в базу, но не удаляются и не изменяются. Также был реализован тестовый сайт на PHP http://fr30706.tw1.ru/ с общей информацией по сервису, и возможностью поиска записей в базе данных.

Для идентификации мобильных устройств в модели используются IMEI и AndroidId, как наиболее доступные, распространенные и уникальные идентификаторы для Android устройств.

В качестве продолжения разработки были выдвинуты следующие предложения:

1. Реализовать программное API для получения информации из базы данных по запросу.
2. Реализовать возможность распечатать QR-код для конкретного мобильного устройства, содержащий ссылку на сайт с отображением информации по идентификатору со связанным профилем.
3. Реализовать web-интерфейс с адаптивной разметкой и удобным поиском по базе данных с идентификаторами устройств.

ГЛАВА 1. ОСНОВНЫЕ МЕТОДЫ ИДЕНТИФИКАЦИИ МОБИЛЬНЫХ УСТРОЙСТВ

IMEI

IMEI (International Mobile Equipment Identity) - число (обычно 15-разрядное в десятичном представлении), уникальное для каждого использующего его аппарата. Применяется в сотовых телефонах сетей GSM, WCDMA и IDEN, а также в некоторых спутниковых телефонах [4].

IMEI присваивается телефону при заводском изготовлении. Он служит для идентификации устройства в сети и хранится в прошивке аппарата. Как правило, IMEI указывается в четырёх местах: в самом аппарате (в большинстве случаев его можно вывести на экран набором *#06# на клавиатуре), под аккумуляторной батареей, на упаковке и в гарантийном талоне. IMEI играет роль серийного номера аппарата и передаётся в эфир при авторизации в сети. Также IMEI используется для слежения за аппаратами и блокирования краденых телефонов на уровне оператора сотовой связи, что не позволяет в дальнейшем использовать такой аппарат в сети этого оператора, однако не мешает его использованию в других сетях.

В отличие от ESN и MEID, используемых в CDMA и прочих сетях, IMEI используется только для идентификации устройства и не имеет постоянного отношения к абоненту. Вместо него используется номер IMSI, хранящийся на SIM-карте, которую можно вставить практически в любой другой аппарат. Тем не менее, существуют специальные системы, позволяющие одному телефону использовать только одну определённую SIM-карту.

Модель и происхождение телефона описываются первыми восемью цифрами IMEI (так называемый TAC). Оставшаяся часть — серийный номер с контрольным числом в конце. Телефонам, поддерживающим одновременную работу с двумя SIM-картами, присваивается два номера IMEI [5].

Производители постоянно находятся в процессе совершенствования методов защиты программного обеспечения аппарата от изменения IMEI. В современных аппаратах IMEI хранится в однократно программируемой зоне памяти и не может быть изменен программными средствами [6].

В некоторых странах, например в Латвии, Великобритании, Республике Беларусь изменение IMEI является уголовно наказуемым преступлением. Имеется также прецедент попытки уголовного преследования за изменение IMEI в России [7].

MEID

MEID (Mobile Equipment IDentifier) - глобальный уникальный идентификатор подвижного оборудования, работающий в сетях CDMA, и использующий тот же базовый формат, что и IMEI [8].

IMSI

IMSI (International Mobile Subscriber Identity) - международный идентификатор мобильного абонента (индивидуальный номер абонента), ассоциированный с каждым пользователем мобильной связи стандарта GSM, UMTS или CDMA. При регистрации в сети аппарат абонента передаёт IMSI, по которому происходит его идентификация. Во избежание перехвата, этот номер посылается через сеть настолько редко, насколько это возможно — в тех случаях, когда это возможно, вместо него посылается случайно сгенерированный TMSI [9].

В системе GSM идентификатор содержится на SIM- карте в элементарном файле (EF), имеющем идентификатор 6F07. Формат хранения IMSI на SIM- карте описан ETSI в спецификации GSM 11.11. Кроме того, IMSI используется любой мобильной сетью, соединенной с другими сетями (в частности с CDMA или EVDO) таким же образом, как и в GSM сетях. Этот номер связан либо непосредственно с телефоном, либо с R-UIM картой (аналогом SIM карты GSM в системе CDMA) [10].

Длина IMSI, как правило, составляет 15 цифр, но вполне может быть и короче. Например: 250-07-XXXXXXXXXX. Первые три цифры это MCC (Mobile Country Code, мобильный код страны). В примере 250 - Россия. За ним

следует MNC (Mobile Network Code, код мобильной сети). 07 из примера - СМАРТС. Код мобильной сети может содержать две цифры по европейскому стандарту или три по североамериканскому. Все последующие цифры — непосредственно идентификатор пользователя MSIN (Mobile Subscriber Identification Number) [11].

Serial number

Серийный номер можно определить у устройств, не обладающих сервисом телефонии, начиная с операционной системы Android 2.3 ("Gingerbread") и у некоторых телефонов [12].

Android Id

Android Id – это 64-битный номер, который случайным образом генерируется при первом запуске устройства и далее остается неизменным. У устройств с операционной системой более ранних версий, чем 2.2 ("Froyo"), он может не определяться [12].

Mac-Address

MAC-адрес (от англ. Media Access Control — управление доступом к среде, также Hardware Address) — это уникальный идентификатор, присваиваемый каждой единице оборудования компьютерных сетей. Большинство сетевых протоколов канального уровня используют одно из трёх пространств MAC-адресов, управляемых IEEE: MAC-48, EUI-48 и EUI-64. Адреса в каждом из пространств теоретически должны быть глобально уникальными. Не все протоколы используют MAC-адреса, и не все протоколы, использующие MAC- адреса, нуждаются в подобной уникальности этих адресов [13].

В широковещательных сетях (таких, как сети на основе Ethernet) MAC-адрес позволяет уникально идентифицировать каждый узел сети и доставлять данные только этому узлу. Таким образом, MAC-адреса формируют основу сетей на канальном уровне, которую используют протоколы более высокого (сетевого) уровня. Для преобразования MAC-адресов в адреса сетевого уровня и обратно применяются специальные протоколы (например, ARP и RARP в сетях IPv4 и NDP в сетях на основе IPv6) [13].

Адреса наподобие MAC-48 наиболее распространены; они используются в таких технологиях, как Ethernet, Token ring, FDD I, WiMAX и др. Они состоят из 48 бит, и, таким образом, адресное пространство MAC-48 насчитывает 248 (или 281 474 976 710 656) адресов. Согласно подсчётам IEEE, этого запаса адресов должно хватить по меньшей мере до 2100 года [13].

EUI-48 от MAC-48 отличается лишь семантически: в то время как MAC-48 используется для сетевого оборудования, EUI-48 применяется также для других типов аппаратного и программного обеспечения.

Идентификаторы EUI-64 состоят из 64 бит и используются в FireWire, а также в IPv6 в качестве младших 64 бит сетевого адреса узла. Можно также получить MAC-адрес Wi-Fi или Bluetooth оборудования устройства, но в то же время не рекомендуется использовать его в качестве уникального идентификационного номера, так как не все мобильные устройства имеют Wi-Fi. Если Wi-Fi модуль есть, он должен быть обязательно включен, иначе MAC-адрес не определится. Кроме того, MAC-адрес устройства можно изменить программным путем [13].

ГЛАВА 2. МОДЕЛЬ ЦИФРОВЫХ СЕРТИФИКАТОВ

Идея цифровых сертификатов состоит в создании открытой базы данных, используя которую, каждый владелец мобильного телефона мог бы сохранить идентифицирующие признаки своего аппарата, заверив (а именно, подписав) их ссылкой на собственный профиль в социальной сети. Суть использования ссылки на профайл состоит в том, что в этом случае база данных освобождается от проблем, связанных с хранением персональной информации. В таком случае на просто отсутствует. Вся персональная информация остается в социальной сети. Таким образом, в базе данных сертификатов хранится только открытая ссылка на соответствующий профиль.

Исходя из этого, реализация такой модели должна включать в себя: мобильное приложение для создания сертификата, базу данных для хранения сертификатов и интерфейс к базе данных для поиска. Важным является тот момент, что такой интерфейс должен включать в себя также программный API.

Владелец телефонного аппарата, по собственной инициативе, может бесплатно добавить сертификат для своего телефона в общую базу. База сертификатов публично доступна. Следовательно, очень упрощается процесс проверки владельца телефона. Этот факт, в свою очередь, сможет остановить какой-то значимый процент мобильных абонентов от пользования телефоном, который попал к ним не совсем законным способом. Кроме того, наличие такой базы может оказаться весьма полезным для официального следствия.

Таким образом, основная идея данной модели состоит не в отслеживании потерянного (похищенного) телефона, а во введении возможности проверки владельца телефона в момент использования телефона. При этом, в первую очередь, имеется в виду использование смартфонов в сети Интернет. Самый простой способ применения: приложение во время

авторизации пользователя в социальной сети может проверить, кому именно принадлежит данный телефон.

Само собой разумеется, что нет никаких препятствий операторам связи точно так же использовать ту же самую открытую базу данных для проверки владельцев в момент совершения звонков и отправки SMS.

ГЛАВА 3. АВТОРИЗАЦИЯ ЧЕРЕЗ СОЦИАЛЬНЫЕ СЕТИ

Подавляющее большинство сайтов на сегодняшний день наряду с основной системой авторизации предоставляют пользователям возможность авторизоваться через социальные сети (Social Sign-On). Такой подход очень удобен для посетителя сайта: достаточно нажать кнопку входа и подтвердить доступ [14].

Как только посетитель дает свое разрешение, данные из социальной сети передаются на сайт и записываются в профайл пользователя. Существует несколько способов авторизации, редуцирующих необходимость создания большого числа пользовательских логин-паролей:

1. **Менеджеры паролей** - приложения, которые помогают веб пользователям управлять множеством логин-паролей. Менеджеры паролей упрощают работу пользователей, поскольку им нужно запомнить только один master логин-пароль; однако это не избавляет от необходимости создавать новые логин-пароли для каждого сайта, и необходимости периодически менять пароли.

2. **SSO** - На платформе с SSO, пользователь аутентифицируется только один раз у провайдера идентификации, который содержит информацию о пользователе, которая в свою очередь подтверждает существование этого пользователя в системе. Каждый раз, когда пользователь обращается к приложению, механизм автоматически проверяет, что пользователь правильно аутентифицирован провайдером идентификации. SSO позволяет устранить необходимость для пользователя повторно аутентифицироваться для разных приложений и сохранять различные учетные данные для каждого приложения. SSO помогает повысить производительность пользователей, избавляя пользователя от запоминания огромного количества паролей, а также экономя время, которое пользователь тратит на набор различных логин-паролей. SSO также упрощает администрирование благодаря созданию единых учетных данных вместо их

множества. Это упрощает управление правами вошедшего пользователя, изменяя полномочия при входе/выходе из сервиса, что позволяет быстро интегрировать добавляемые приложения, а при необходимости делегировать права доступа.

3. **WSSO** (Web-based Single Sign-On) - web ориентированное расширение SSO-платформы. WSSO система отделяет роль провайдера идентификации от сервис провайдера. Провайдер идентификации получает информацию идентифицирующую пользователя и аутентифицирует пользователей, в то время как сервис провайдер основывается на результате аутентификации для авторизации пользователя. Главная задача WSSO - позволить пользователям заходить на разные сайты под одним аккаунтом. Используя WSSO, пользователям не нужно запоминать много паролей, им проще вводить свои данные для аутентификации. А контент провайдеры освобождаются от необходимости добавления, обслуживания и защиты аккаунтов у себя на сайте.

Социальные сети, такие как Facebook, Twitter, Вконтакте, Одноклассники предоставляют возможность пользователям регистрироваться и авторизоваться на различных сторонних сайтах используя единый аккаунт, используя их профиль социальной сети [15]. Эти сайты запрашивают у пользователя авторизацию, для доступа и частичного управления их профилем. Такой тип взаимодействия дает возможность сторонним сайтам аутентифицировать пользователей основываясь на их идентификации в Facebook, Twitter, Google+ или иных социальных сетях. К тому же, такие сайты расширяют свой функционал взаимодействия с пользователем посредством возможности ставить лайки, комментировать, делиться контентом.

3.1 Протоколы OAuth и OpenID

Существует несколько моделей авторизации через социальные сети. Наиболее распространены основанные на Oauth и OpenlD.

OAuth - протокол авторизации, опубликованный в 2010[], который дает пользователю доступ к сторонним приложениям, таким как сайты и процессы, запускаемые браузерами, мобильным устройствам, без предоставления информации или данных идентификации. Для использования OAuth в структуре WSSO, социальная сеть хранит идентификатор пользователя и аутентифицирует его, в то время как сторонний сайт работает в качестве проверяющей стороны, которая полагается на аутентификацию идентификатора, чтобы авторизовать пользователя и включить его личные настройки [16].

OAuth 2.0 - протокол авторизации, опубликован в 2012 [17], развитие Oauth, который стандартизирует делегированную авторизацию в Web. Facebook Connect - платформа Social Sign-On соответствующей социальной сети, основана на протоколе OAuth 2.0, который дает возможность сторонним сайтам аутентифицировать пользователей посредством получения доступа к их профилю на facebook. Что используется в модели цифровых сертификатов. Другие популярные социальные сети, такие как Google+ и Twitter, тоже используют этот протокол для улучшения взаимодействия с пользователем.

Использование OAuth 2.0 привлекательно для сервис провайдеров и просто в реализации для разработчиков сервисов. Однако исследование [18] показывает, что он слишком прост, чтобы обеспечить полную безопасность.

В отличие от общепринятых протоколов безопасности, OAuth 2.0 разработан без надежной криптографической защиты, такой как шифрование, цифровая подпись, или использование nonce. Отсутствие шифрования в протоколе требует от удостоверяющей стороны использования SSL, но большинство исследованных сайтов не придерживаются этой практики.

Кроме того, достоверность как запроса авторизации, так и ответа не может быть гарантирована без ЭЦП, а replay-атаку, использующую скомпрометированные SSO учетные данные, трудно обнаружить, если запрос не содержит nonce или timestamp. Кроме того, поддержка client-flow делает протокол уязвимым к широкому спектру атак, поскольку токены доступа передаются через браузер и перенаправляются на удостоверяющий сервер.

По сравнению с server-flow, client-flow по своей сути небезопасен для SSO в целом. Использование OAuth 2.0 без глубокого понимания web-безопасности обычно, приводит к уязвимым реализациям сервисов. Для защиты web-пользователей в существующих реализациях OAuth SSO, предлагается ряд простых и практичных механизмов [18].

Внедрение этих механизмов позволит предотвратить возникновение масштабных брешей в безопасности, которые могли бы поставить под угрозу миллионы учетных записей web-пользователей.

OpenID - другой протокол, который позволяет реализовать удобную для пользователя авторизацию [19]. Пользователь может выбирать провайдера идентификации среди множества сервисов, которые работают посредством OpenID. Особенность OpenID в том, что провайдер не требует предварительного взаимодействия с сайтом или web-сервисом для которого он предоставляет аутентификацию. OpenID - децентрализованный протокол аутентификации.

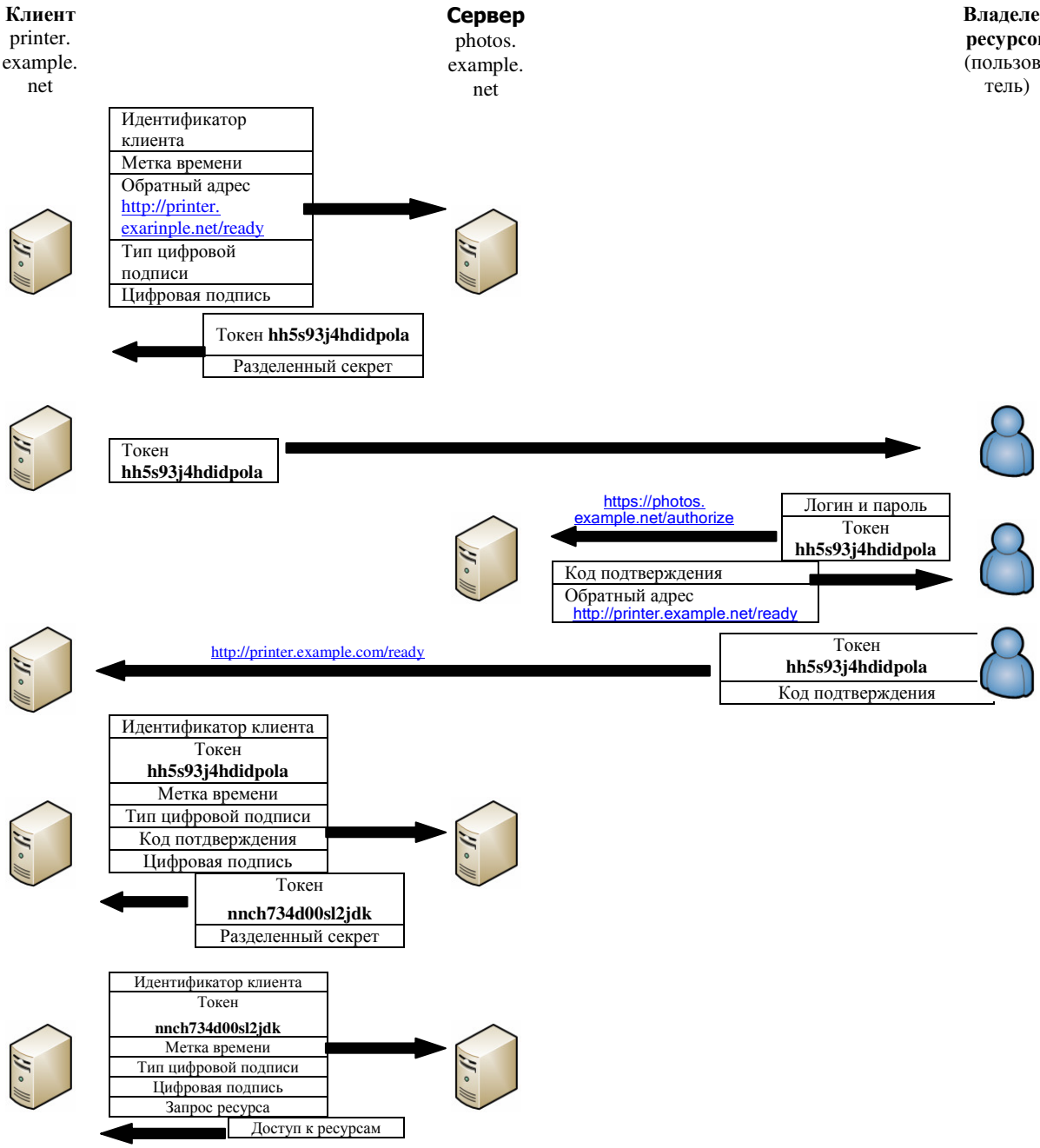
OpenID является средством аутентификации: с помощью этой системы можно удостовериться, что пользователь — именно тот, за кого себя выдает. Какие действия сможет совершать пользователь, прошедший аутентификацию посредством OpenID, определяется стороной, проводящей аутентификацию.

OAuth является протоколом авторизации, который позволяет предоставить права на использование некоторого ресурса (например, API какого-либо сервиса). Наличие прав определяется токеном (уникальным идентификатором), который может быть одним и тем же для разных

пользователей, или же у одного пользователя в разное время могут быть разные токены. Предоставление прав происходит в обмен на предоставление токена.

Этапы работы протокола OAuth в режиме Authorization Code Flow [20]:

1. Клиент посредством протокола https:// отправляет серверу запрос, который содержит идентификатор клиента, метку времени, адрес обратного вызова, по которому нужно вернуть токен, используемый тип цифровой подписи и саму подпись.
2. Сервер подтверждает запрос и отвечает клиенту токеном запроса (Request Token) и частью разделённого секрета.
3. Клиент передает токен владельцу ресурсов (пользователю) и перенаправляет его на сервер для прохождения аутентификации.
4. Сервер, получив от пользователя токен, запрашивает у него его логин и пароль, и, в случае успешной аутентификации, просит пользователя подтвердить доступ клиента к ресурсам (авторизация), после чего пользователь перенаправляется сервером к клиенту.
5. Клиент передает серверу токен (Request Token) посредством протокола TLS и запрашивает доступ к ресурсам.
6. Сервер подтверждает запрос и отвечает клиенту новым токеном доступа (Access Token).
7. Используя новый токен, клиент обращается к серверу за ресурсами.
8. Сервер подтверждает запрос и предоставляет ресурсы.


Клиент printer. example. net
Сервер photos. example. net
Владелец ресурсов (пользова тель)
Идентификатор клиента
Метка времени
Обратный адрес http://printer. exarinple.net/ready
Тип цифровой подписи
Цифровая подпись
Токен **hh5s93j4hdidpola**
Разделенный секрет
Токен **hh5s93j4hdidpola**
https://photos. example.net/authorize
Логин и пароль
Токен **hh5s93j4hdidpola**
Код подтверждения
Обратный адрес http://printer.example.net/ready
http://printer.example.com/ready
Токен **hh5s93j4hdidpola**
Код подтверждения
Идентификатор клиента
Токен **hh5s93j4hdidpola**
Метка времени
Тип цифровой подписи
Код потдверждения
Цифровая подпись
Токен **nnch734d00sl2jdk**
Разделенный секрет
Идентификатор клиента
Токен **nnch734d00sl2jdk**
Метка времени
Тип цифровой подписи
Цифровая подпись
Запрос ресурса
Доступ к ресурсам

3.2 Общая схема авторизации через социальную сеть.

Можно отметить следующие необходимые условия для реализации авторизации через социальную сеть:

1) Регистрация сайта или приложения в социальной сети. Для того чтобы сайт и социальная сеть могли обмениваться данными, присутствие сайта должно быть каким-то образом «обозначено» в социальной сети. С этой целью, в социальных сервисах существует такое понятие как регистрация приложения. При этом происходит связывание сайта с социальной сетью, а полученный идентификатор, ключи приложения используются для управления привилегиями доступа к данным профиля пользователя. Обычно данные публичного профиля доступны по логину через API социальной сети, и этого достаточно для аутентификации пользователя и подписи записи в БД цифровых сертификатов.
2) Подключение механизма авторизации. Организация авторизации через социальную сеть может незначительно различаться в зависимости от устройства или платформы, с которой производится запрос: iOS, Android, Windows Phone, Web (client). Также следует отметить, что возможности работы с API обычно шире при запросах с сервера, зарегистрированного в социальной сети, чем с клиента.

Если планируется использование разных социальных сетей для авторизации, то необходимо проводить регистрацию приложения или сайта в каждой из них, а также прописывать для каждой из них свое API авторизации (а возможно, и дополнительный код для разных платформ). С целью упрощения этой задачи можно использовать такие сервисы авторизации как, например, Loginza или ULogin. В этом случае доступна авторизация через наиболее известные социальные сети или сервисы посредством устанавливаемого скрипта. Однако теряется возможность более тонко «настраивать» взаимодействие с пользователями в рамках конкретной социальной сети.

ГЛАВА 4. ПОСТРОЕНИЕ РЕШЕНИЯ ЗАДАЧИ

Для реализации серверной части модели цифровых сертификатов был выбран язык программирования Java и среда разработки NetBeans.

В качестве Application Server был выбран Apache Tomcat (версия 7.0.41.0). Также есть возможность развертывания приложения на сервере GlassFish Server 4.0.

Для возможности интеграции с различными базами данных было принято решение использовать библиотеку Hibernate (4.2.2). С помощью этой библиотеки предполагалось реализовать работу с базами данных:

1. Apache Derby 10.10.1.1
2. IBM DB2 10.5

Для функционирования приложения предполагалось разработать структуру модели сервлетов серверной части, управляющих базой данных и клиентским представлением сайта в рамках модели MVC.

В частности, было принято решение отделить клиентскую часть приложения, представленную JSP со статической html/css разметкой и динамическим взаимодействием web-интерфейса, основанного на javascript библиотеках и плагинах, от серверной части, включающей сервлеты, доступные с клиентской стороны посредством технологии ajax, при этом для обмена дынными предполагалось использовать формат JSON.

Для реализации кросс-браузерного интерфейса, с адаптивным дизайном для устройств с различным разрешением экрана (мобильные телефоны, планшеты), было решено использовать фреймворк Bootstrap 3.0. Этот фреймворк также реализует 12-колоночную структуру разметки.

Для представления данных из базы данных на web-странице и реализации удобного интерфейса поиска по ним был выбран jquery плагин DataTables 1.10, интегрированный с фреймворком Bootstrap.

Предполагалось реализовать интерфейс взаимодействия между базой данных и сервлетами модели серверной части посредством библиотек

Hibernate, которые могли бы обрабатывать запросы, поступающие от плагина DataTables.

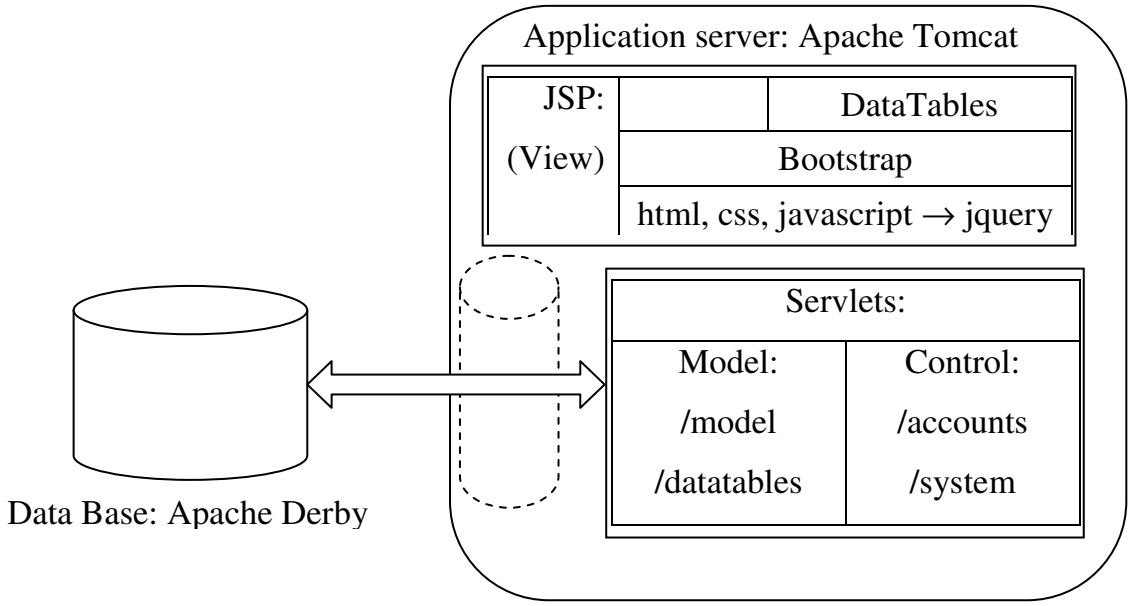
Программное API для получения информации из базы данных решено было реализовать в виде сервлета, доступного по http//, которому на вход подается либо идентификатор, проверяемый в базе данных, либо url профиля в социальной сети. Ответ должен возвращаться в JSON формате, и содержать, в зависимости от запроса, найденную информацию в базе данных.

На момент реализации для создания цифрового сертификата использовались профили Facebook, однако предполагалось учесть возможность использования и других «аутентификаторов»; что должно было найти также отражение и в структуре таблиц базы данных.

QR – код было решено реализовать посредством Google сервиса Google Charts, который возвращает .png файл QR, содержащий url, передаваемый текстом в запросе. При переходе по этому url пользователь попадает на страницу web-интерфейса, отображающую запись из базы данных для устройства.

ГЛАВА 5. ОПИСАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ

В работе рассматривается следующая реализованная конфигурация платформы:



Приложение устанавливается на сервер приложений Apache Tomcat 7.0.41.0 и работает с базой данных Apache Derby 10.10.1.1.

Ядро приложения состоит из классов образующих модель записей «устройство - аутентификатор» и модель DataTables.

Для организации взаимодействия моделей с базой данных и web-интерфейсом были созданы сервлеты управления: для поиска по БД, получения списка записей для DataTables плагина, возможности добавления записей в базу данных.

Взаимодействие модели записей с базой данных организовано посредством Hibernate библиотеки.

Клиентская часть представлена JSP страницей, включающей общую информацию, которая адаптируется под разрешения устройства просмотра сайта посредством фреймворка Bootstrap 3.0. Интерфейс взаимодействия с сервлетами модели реализован посредством http:// запрос-ответов от jquery-

плагина DataTables к сервлету управления, отдающему список запрошенных записей.

5.1 Серверная часть модели цифровые сертификаты

Для серверной части модели цифровые сертификаты я разработал следующую структуру:

1. Model
   a. классы определяющие основные записи в базе данных:
      - **AndroidUser** – идентификатор мобильного устройства, является уникальным и содержит основные поля: IMEI, AndroidID и список url профилей, которыми подписано устройство.
      - **RemouteAuthenticator** – профиль которым подписывается, то или иное устройство. Также уникальный, содержит поля: socialIdUrl – полная ссылка на профиль, username – уникален, netAuthentication – признак определяющий социальную сеть профиля[1].
   b. классы определяющие модель DataTables на сервере:
      - **DataTable**, **DataTableParameters**, **ColumnsMapper**, **SortObject**
   c. вспомогательные классы-утилиты, такие как **ApplicationConfig**

[1] На момент реализации работала привязка только к facebook, однако функционал мобильного приложения, подписывающего мобильные устройства профилями, со временем планируется расширить и профилями других социальных сетей.

2. Control
    a. сервлеты, принимающие на вход параметры или их отсутствие и выдающие ответ в формате JSON, основные из них:
        - **GetAndroidUsersList –** используется для ответов на ajax запросы от объекта DataTable на JSP странице, формирующего возможность интерактивного взаимодействия с содержимым базы данных.
        - **GetAndroidUserDetails –** реализует программное API, возможность получения информации о наличии записи в базе данных по поисковому запросу. Ответ возвращает в JSON формате.
        - **SaveAndroidUser –** помещает новые записи в базу данных, или дополняет существующие записи новыми профилями-подписями.

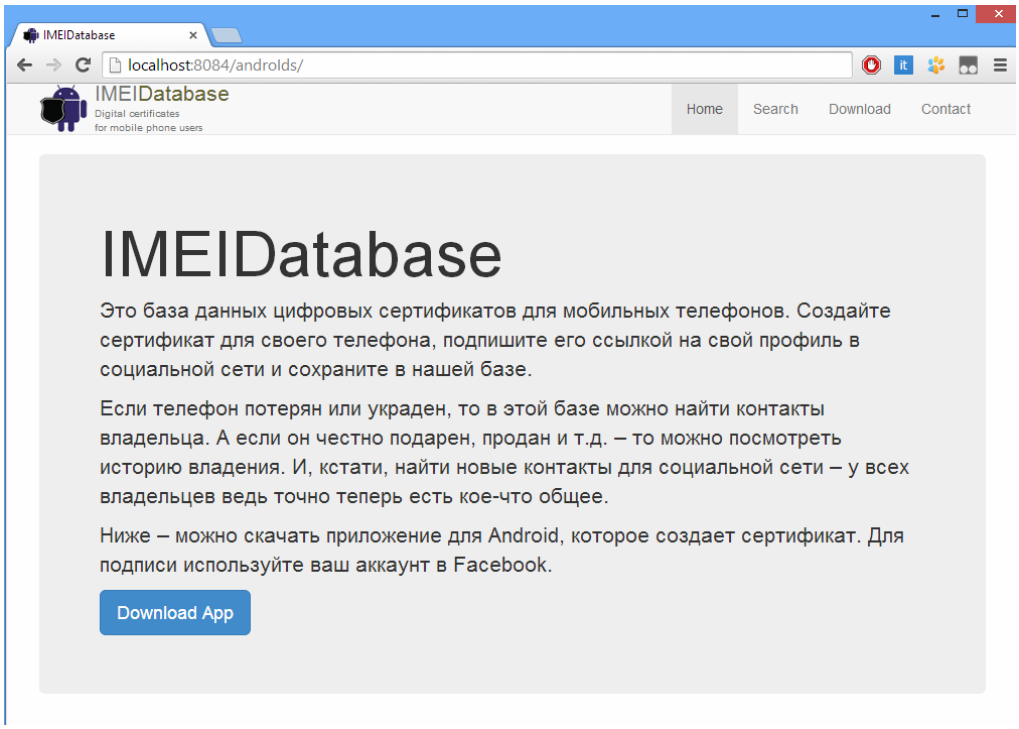
В процессе разработки приложения возник целый ряд технических задач, которые я необходимо разрешал для работы приложения. Например, организация взаимодействия приложения с базой данных посредством библиотеки Hibernate, создание таблиц в базе данных при первом развертывании приложения и ряд других.

5.2 Web-интерфейс системы цифровые сертификаты

Web-интерфейс системы цифровые сертификаты представлен JSP страницей с html/css разметкой, на которой подключены:

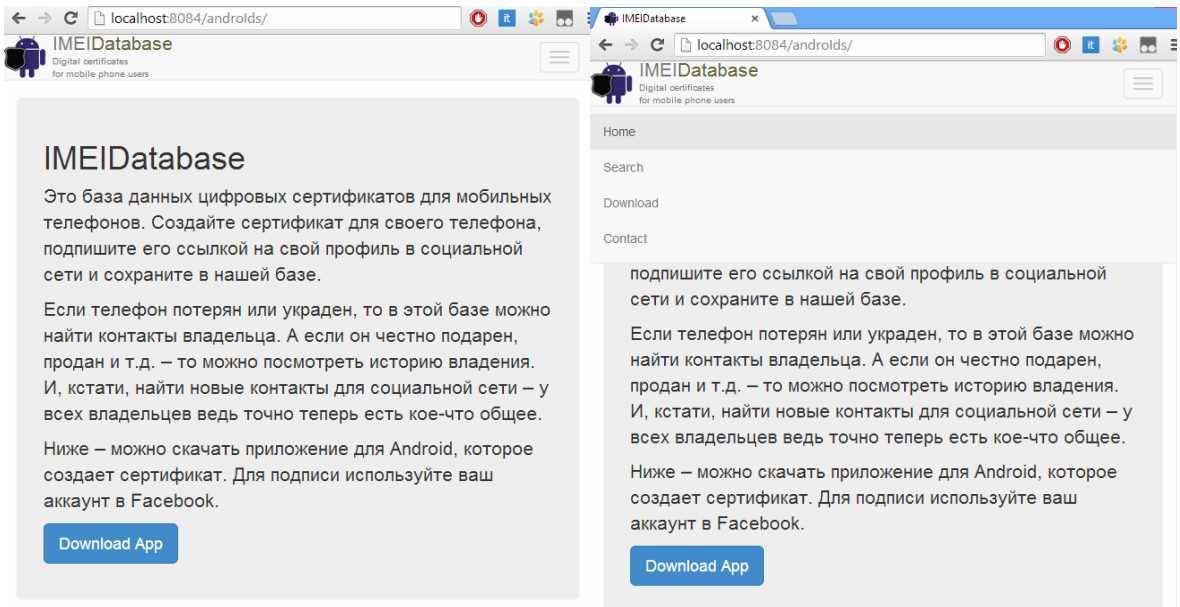
- js библиотека jquery 1.11.0
- фреймфорк Bootstrap 3.0
- jquery плагин DataTables 1.10 интегрированный с Bootstrap
- библиотека яндекс карт
- а также скрипт main.js, определяющий интерфейс взаимодействия с пользователем.

На странице представлены четыре переключаемые вкладки доступные через верхнее меню – Home, Search, Download, Contact:



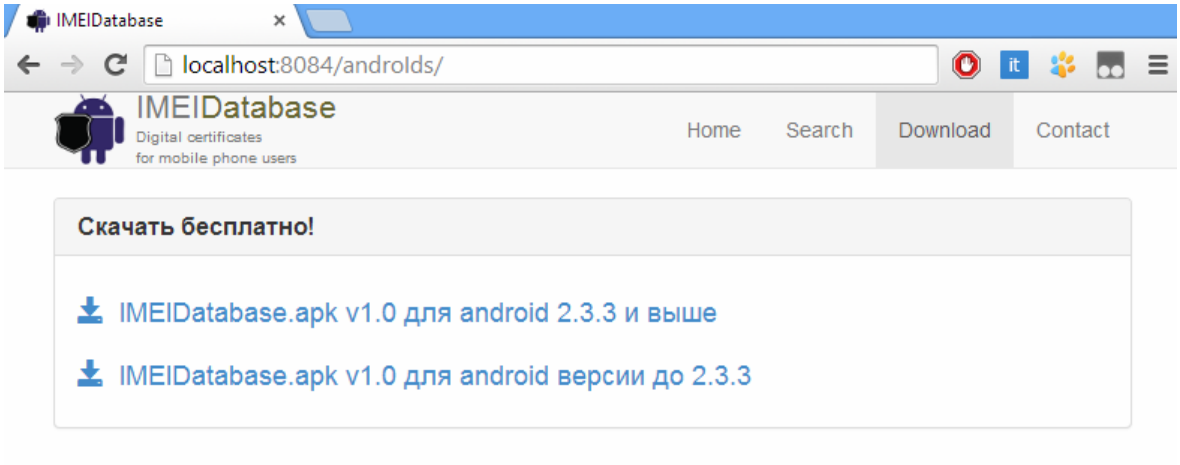
На первой вкладке содержится общая информация по сервису, кнопка «Download App» дублирующая доступ к вкладке на скачивание мобильного приложения.

Благодаря использованию фреймворка Bootstrap, реализована адаптивная разметка изменяющаяся в зависимости от разрешения экрана устройства с которого просматривается сайт:

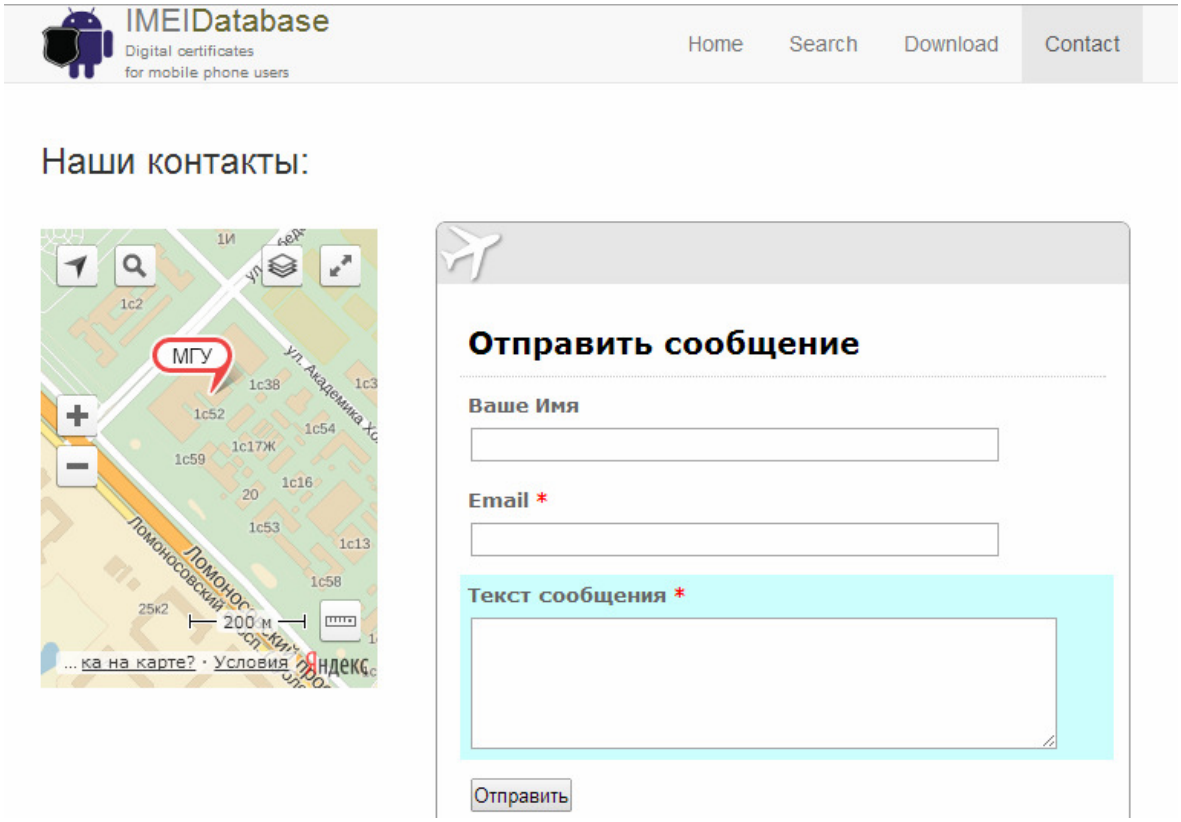


При этом меню остается доступным, но становится компактным.

На вкладке «Downloads» я расположил ссылки на две версии мобильного приложения подписывающего устройство пользователя профилем социальной сети:



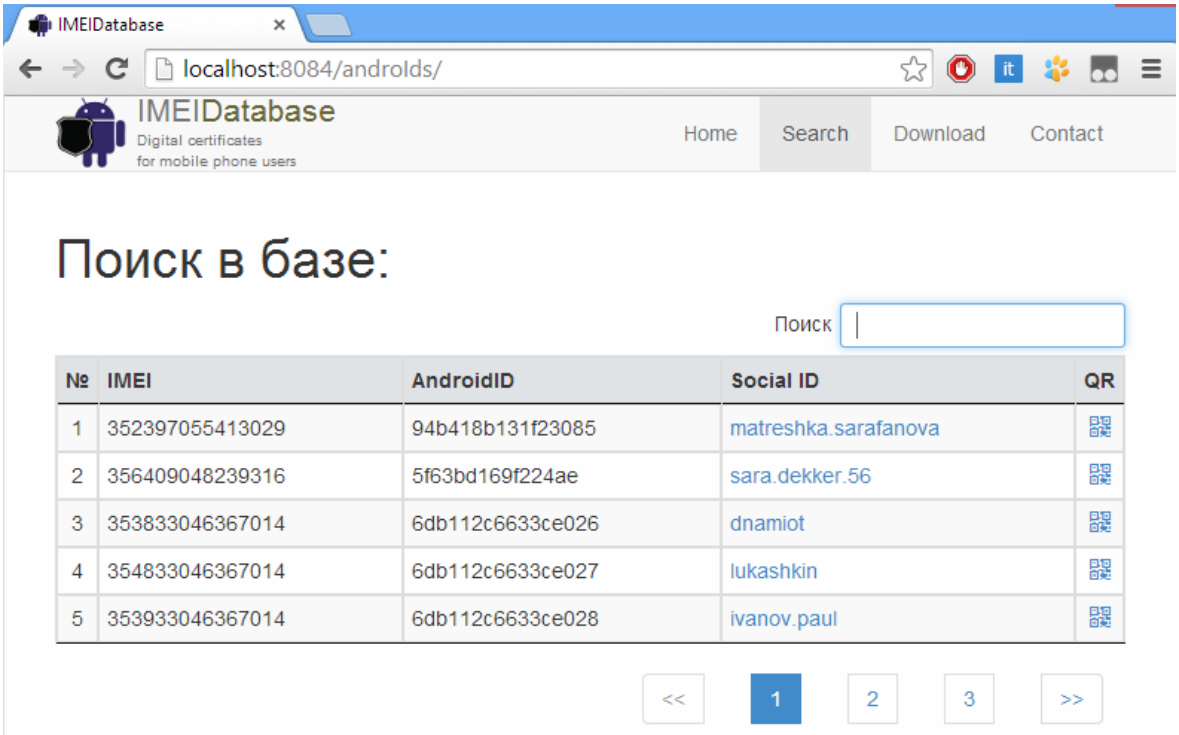
На вкладке «Contact» находятся разворачиваемая яндекс карта и форма обратной связи:



Объект яндекс карты вставлен с помощью подключенного api-maps.yandex.ru 2.1, который я инициализирую из скрипта main.js.

Форма обратной связи реализована посредством сервиса emailmeform.com – этот сервис позволяет создать и настроить подходящую форму обратной связи, которая поддерживается силами сервиса и вставляется посредством скрипта либо, через <iframe>. На этой вкладке форма обратной связи вставлена с помощью тега <iframe>.

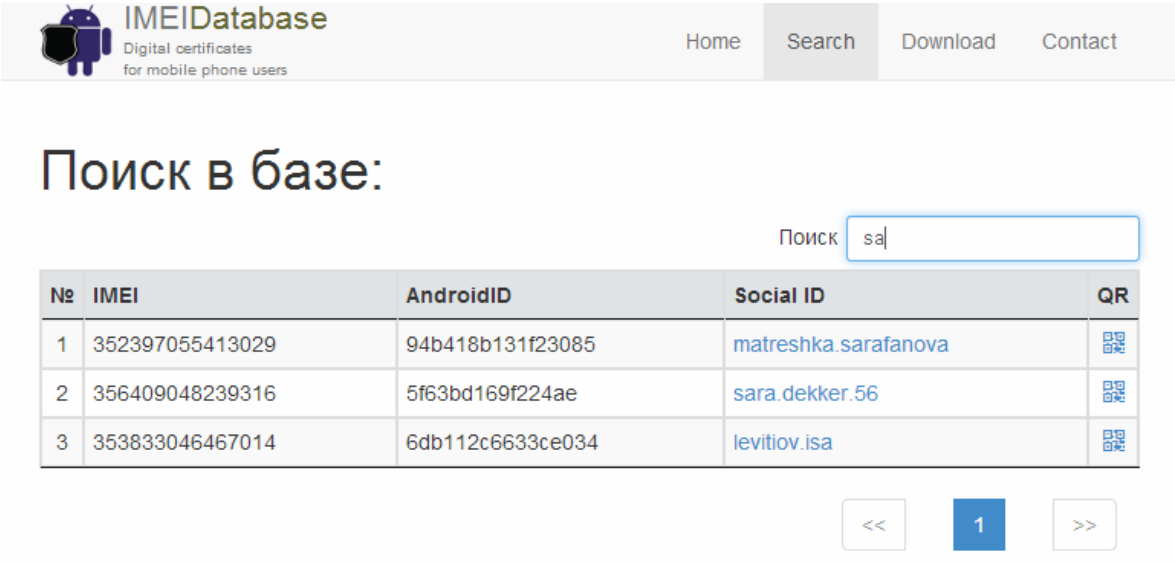
Основной функционал сайта сосредоточен на вкладке «Search»:



Здесь вставлен объект DataTables который имеет вид таблицы содержащей записи отображаемые из базы данных системы цифровые сертификаты. Информация по записям распределена в колонках таблицы:

- **IMEI** – содержит IMEI устройства, при его наличии.
- **AndroidID –** содержит AndroiID
- **Social ID –** представлена ссылкой на профиль в Facebook, которым подписано идентифицированное устройство
- **QR –** кнопка с помощью которой можно распечатать QR-код содержащий ссылку на сайт с записью для данного устройства.

Над таблицей содержится поле ввода для полнотекстового поиска по содержимому колонок:



Под таблицей с записями находится интерфейс постраничной навигации.

Благодаря интеграции DataTables с Bootstrap есть возможность адаптировать отображение таблицы к разрешению устройства, с которого пользователь получает доступ к сайту.

Используя определенный возможности фреймворка Bootstrap, я реализовал следующий вариант отображения таблицы для мобильных устройств.

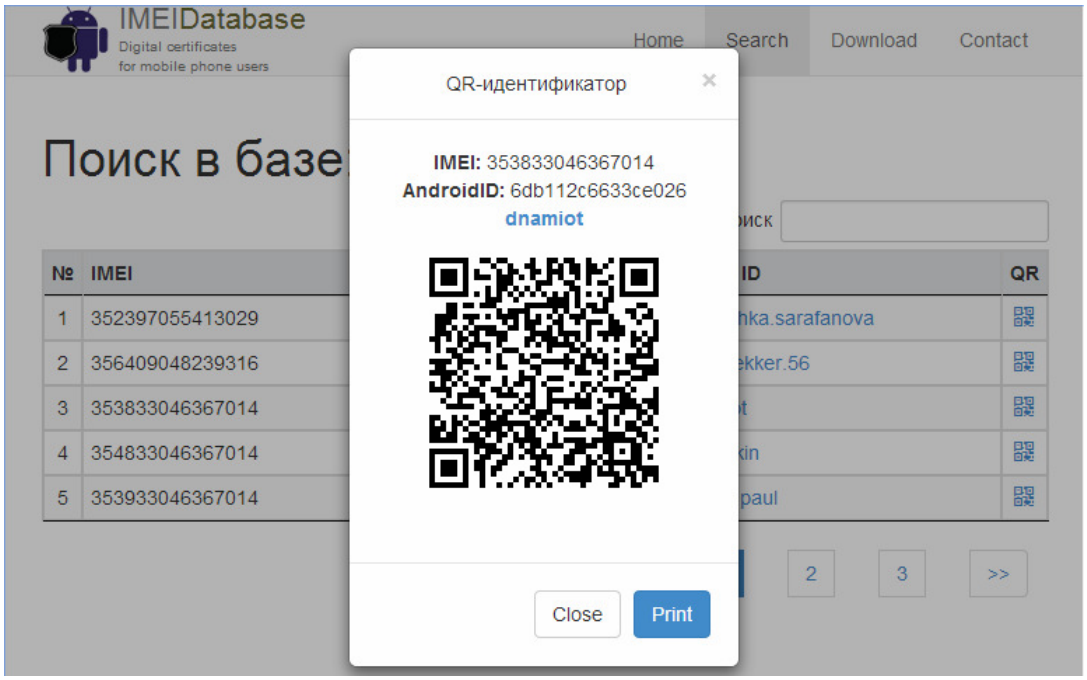
DataTables является настраиваемым объектом, отображение которого можно при необходимости изменить, а интерфейс расширить. Например, можно увеличить количество отображаемых записей. Либо реализовать возможность разворачивать строчку записи для отображения более подробной информации: например, списка профилей которыми подписано данное устройство.

5.3 QR-код записи из системы цифровые сертификаты

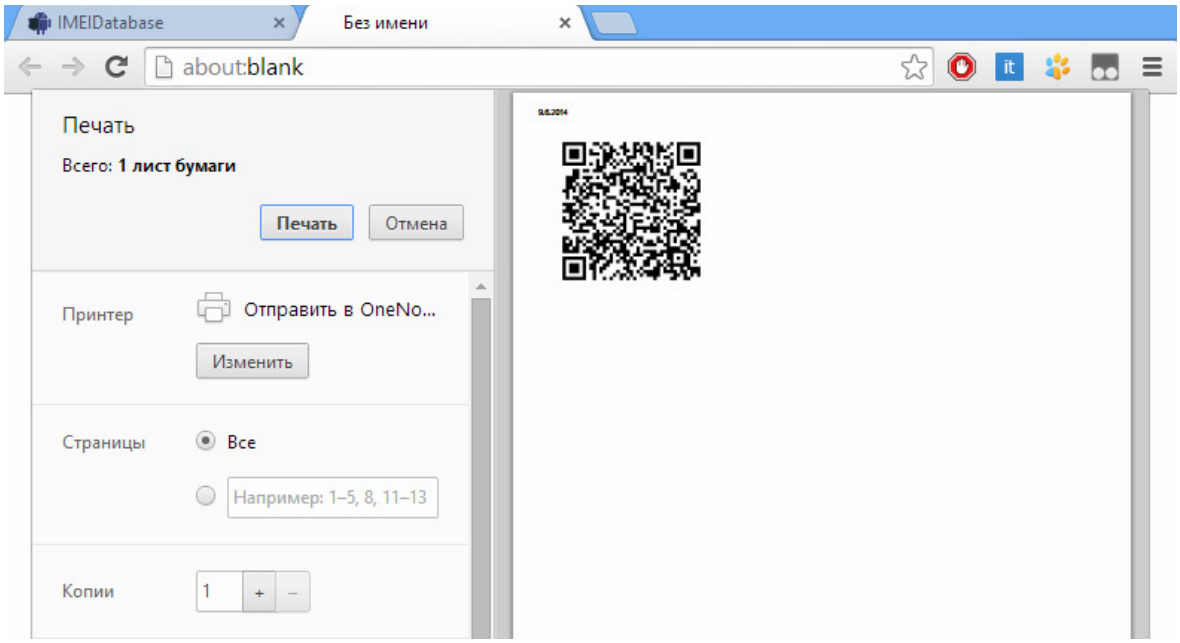
Для реализации получения QR-кода из системы цифровые сертификаты были решены три задачи:

- Создан скрипт, который обрабатывает параметр (imei или androidid) из ссылки на web-интерфейс системы цифровые сертификаты.
- Создание соответствующего url с параметром идентифицирующим мобильное устройство в базе данных.
- Перевод этого url в изображение QR-кода с помощью сервиса Google Charts и организация возможности распечатать полученный QR-код.

Для того чтобы получить QR-код с ссылкой на устройство в системе цифровые сертификаты необходимо нажать на QR-иконку в последней колонке, соответствующей записи. При этом возникает всплывающее окошко:



При нажатии на кнопку «Print» в новой вкладке или окне, в зависимости от браузера, открывается предпросмотр печати QR-кода:



5.4 API системы цифровых сертификатов

Для получения информации из открытой базы данных системы цифровых сертификатов было реализовано программное API. Запрос делается по http:// протоколу к сервлету **GetAndroidUserDetails**, при этом на входе сервлет принимает параметр для поиска, это может быть:

- imei – поиск по IMEI
- andrid – поиск по AndroidId
- socialId – поиск по уникальной составляющей url профиля социальной сети
- all – полнотекстовый поиск по трем выше перечисленным полям

Пример запроса:

<host>/androIds/GetAndroidUserDetails?imei=353833046367014

В ответ на запрос возвращается строка содержащая объект в JSON формате:

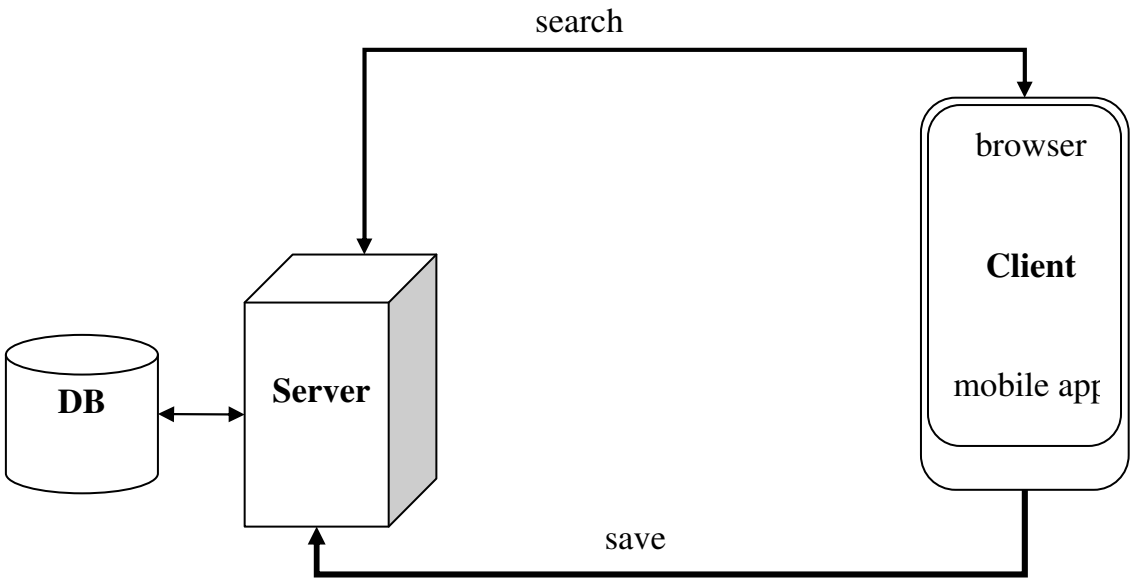
{"res": 1,
"obj": {
"DT_RowId": 1,
"createdOn": "18.05.2014",
"**IMEI**": "352397055413029",
"**AndroidID**": "94b418b131f23085",
"authenticatorsList": [
{
"DT_RowId": 1,
"createdOn": "18.05.2014",
"**socialIdUrl**": "http://www.facebook.com/matreshka.sarafanova",
"userName": "matreshka.sarafanova",
"netAuthentication": "facebook" }
]
}
}

Здесь свойство res – содержит количество возвращенных записей, оно больше 1 для запросов с полнотекстовым поиском, при этом возвращается массив объектов, соответствующих записям в базе данных.

Поскольку в разное время, одно и тоже мобильное устройство может быть подписано несколькими профилями одной социально сети, либо одновременно профилями разных социальных сетей – объект содержит список профилей **authenticatorsList**, которыми подписано устройство. Этот список также представляет собой массив объектов.

ГЛАВА 6. НАПРАВЛЕНИЯ ДАЛЬНЕЙШЕГО РАЗВИТИЯ

В работе представлена реализация одного из компонентов модели цифровые сертификаты, а именно серверной части с web-интерфейсом и базой данных. Другой важной составляющей является мобильное приложение с помощью которого можно подписывать мобильное устройство. Взаимодействие этих компонент можно представить следующей схемой:



Для создания цифрового сертификата устройства в базе данных мобильным приложением необходимо предусмотреть выполнения ряда условий:

- точную идентификацию устройства,
- безопасность мобильного приложения,
- возможность создания новой записи только для данного устройства, которое подписывается профилем пользователя,
- возможность легитимной смены владельца устройства

В связи с этим одной из важнейших задач является безопасное создание новых учетных записей в системе цифровых сертификатов. Здесь возможны два пути решения вопроса:

1) Реализовать протокол безопасного соединения мобильного устройства с базой данных для создания новой записи. При этом должна быть реализована и безопасность самого мобильного приложения, чтобы невозможно было добавлять вымышленные записи в базу данных, как в обход, так и посредством мобильного приложения.
2) Использовать механизм аутентификации мобильного устройства с помощью создаваемого профиля. Здесь необходим предварительный анализ возможностей API, той или иной социальной сети. Можно сказать, что это менее универсальный вариант решения.

Другим направлением развития модели можно назвать анализ возможностей использования иных средств подписи, полученного идентификатора мобильного устройства. Например, это могут быть не только профили социальных сетей, но и аккаунты общедоступных поисковых, почтовых сервисов, также предлагающих API для аутентификации, и авторизации. Многие из них использую протокол OpenID. Сюда же можно отнести вопрос возможностей использования последней версии протокола OpenID Connect в системе цифровые сертификаты.

Довольной серьезной задачей также является разработка и внедрения механизма легитимной смены владельца мобильного устройство. При этом необходимо сохранять историю прав обладания мобильным устройством.

На серверной стороне заложена возможность разграничения профилей, которыми подписывается мобильное устройство, разных социальных сетей или иных сервисов, что требует дальнейшей разработки и реализации.

ЗАКЛЮЧЕНИЕ

В работе представлена модель цифровых сертификатов, описаны проектирование и реализация web-интерфейса для этой модели.

В работе раскрыто решение поставленных задач:

1. Реализация web-интерфейса с адаптивной разметкой и универсальным объектом представления данных DataTables
2. Реализация возможности получения информации из базы данных по поисковым запросам с помощью программного API
3. Возможность получения QR-кода для мобильного устройства, который содержит ссылку на web-интерфейс с информацией по данному устройству.

Кроме того, для реализации первой задачи создана серверная часть, включающая MVC модель системы, интеграцию с базой данных посредством библиотеки Hibernate и серверную часть интерфейса взаимодействия с jquery плагином DataTables.

Также в работе обозначены проблемы и возможности дальнейшего развития реализации модели цифровых сертификатов.

СПИСОК ЛИТЕРАТУРЫ

[1] Колосова А. И., Намиот Д. Е. Цифровые сертификаты для владельцев мобильных телефонов //International Journal of Open Information Technologies. - 2013. - Т. 1. - №. 4. - С. 7-11.

[2] Sonia C. V., Aswatha A. R. SAPt: A Stolen Android Phone Tracking Application //ITSI Transactions on Electrical and Electronics Engineering. - 2013. - Vol. 1. - №.6. - pp. 111-115.

[3] Намиот Д. Е., Колосова А. И. Об определении владельцев мобильного телефона //International Journal of Open Information Technologies. - 2013. - Т. 1. - №. 8. - С. 26-31.

[4] Gosden, P., Allen, A., McDonald, D., & Montemurro, M. (2013). A Uniform Resource Name Namespace for the GSM Association (GSMA) and the International Mobile station Equipment Identity (IMEI).

[5] GSM Association Non Confidential Official Document IMHI Allocation and Approval Guidelines Version 6.0 (27th July 2011) (http://www.gsma.com/ncwsroom/wp-content/uploads/2012/03/ts0660tacallocationprocessapproved.pdf).

[6] В. Шалькевич, А. Макаревич «Противодействие теневому обороту мобильных телефонов уголовно правовыми мерами», Журнал «Законность н правопорядок», № 3(7)/2008, стр. 36-40.

[7] http://www.legislation.gov.Uk/ukpga/2002/31/section/1.

[8] Atarius R. A Uniform Resource Name Namespace for the Device Identity and the Mobile Equipment Identity (MEID). – 2013.

[9] Matsumoto, S., & Sakurai, K. (2013. January). A proposal for the privacy leakage verification tool for Android application developers. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication (p. 54). ACM.

[10] Fred Gaechter "Chairman of IMSI Oversight Committee" (IOC)(GSMNA Doc 036/02) (http://www.ifast.org/files/IFAST22_015_GSMNALetter.pdf).

[11] Abdalla. L, & Venkatesan, S. (2013, April). Scalable addressing of M2M terminals in 4G cellular wireless networks. In Wireless Telecommunications Symposium (WTS), 2013 (pp. 1-6). IEEE

[12] http://developer.samsung.com/android/technical-docs/How-to-retrieve-the-Device-Unique-ID-from-android-device.

[13] IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture (IEEE Std 802®-2001 (R2007)(Revision of IEEE Std 802-1990)).

[14] Gafni, R., & Nissim, D. (2014). To social login or not login? - Exploring factors affecting the decision. //Issues in Informing Science and Information Technology. 11, pp. 57-72. (http://iisit.org/Vol11/IISITv11p057-072Gafni0462.pdf)

[15] Facebook (2013). Facebook for Websites. (https://developeis.facebook.coni.docs/giiides/web/)

[16] Leiba. B. (2012). OAuth web authorization protocol. //IEEE Internet Computing, 76(1), pp. 74-77.

[17] Hardt. D. (2012) The OAuth 2.0 Authorization framework. [RFC 6749] (http://tools.ietf.org/html/rfc6749)

[18] Sun, S. T., & Beznosov, K. (2012). The devil is in the (implementation) details: An empirical analysis of oauth sso systems. Proceedings of the 2012 ACM conference on Computer and Communications Security, 378-390. (http://iisit.org/Vol11/IISITv11p057-072Gafni0462.pdf)

[19] Bellamy-McIntyre. J., Luterroth, C., & Weber, G. (2011). OpenID and the enteiprise: A model-based analysis of single sign-on authentication. //Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. pp. 129-138.

[20] Sun, S. T. (2013). Towards improving the usability and security of Web single sign-on systems. pp. 80-90.