Московский Государственный Университет имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

Дипломная работа

Цифровые сертификаты для владельцев мобильных телефонов

Выполнила: Колосова А. И., гр. ВВО

Научный руководитель: к.ф.-м.н., с.н.с. лаборатории ОИТ ВМК МГУ Намиот Д.Е.

Москва 2013

Оглавление

1. Введение

Как правило, у каждого владельца мобильного телефона записано некоторое количество важной информации на нем, которую тяжело восстановить. Как то – список контактов, TODO лист и др., в зависимости от сложности устройства. Поэтому важной задачей является нахождение утерянных аппаратов.

Проблема хищений мобильных телефонов остро стоит во всем мире. Анализ международного опыта позволяет выделить несколько подходов, применяемых к ее решению: от полного бездействия в урегулировании этого вопроса до применения комплекса организационных, технических и правовых мер, к которым относятся блокирование работы мобильного телефона оператором сотовой связи по IMEI; уголовное преследование лиц, изменяющих IMEI похищенного телефона с целью недопущения идентификации похищенного оборудования; разъяснение абонентам операторов сотовой связи последствий использования мобильного оборудования с измененными идентификационными данными[1].

Например, в Великобритании в 2002 году было похищено или утеряно около 700 тысяч телефонов, в 2007 году — более 800 тысяч. В эксплуатацию там была введена система блокирования работы похищенного оборудования по IMEI [1].

В данной дипломной работе рассматриваются мобильные устройства с операционной системой Android – смартфоны и др. У всех подобных аппаратов имеются различные идентификационные номера. Которые могут быть использованы для нахождения

утерянных телефонов, мониторинга установок некоего приложения, генерации технических средств защиты авторских прав (ТСЗАП, DRM – digital rights management). Например, мобильные операторы, при наличии определенного оборудования могут полностью или частично прекратить обслуживать украденный телефон, перенаправлять смс-сообщения с него на другой телефон. Или отследить его место нахождения по GPS. В России подобная практика не так распространена, как в некоторых других странах, однако и у нас есть случаи нахождения украденных телефонов по IMEI номеру.

1.1. Постановка задачи

Задачи этой дипломной работы:

- разработка методики определения идентификационных номеров мобильных аппаратов;

-разработка системы регистрации идентификационных номеров мобильных аппаратов и их владельцев в базе данных.

-разработка сайта с системой взаимодействия с получившейся базой данных.

2. Обзор существующих методов идентификации мобильных телефонов

Существует несколько различных типов идентивикационных номеров для аппаратов с операционной системой Android .

В недавнем прошлом все Android-аппараты обладали сервисами телефонии, поэтому всегда можно было определить уникальный номер IMEI, MEID или ESN.

Но сейчас уже существуют Wifi – only аппараты, музыкальные плееры и др. устройства с операционной системой Android, не обладающие сервисами телефонии. У таких устройств тоже можно определить идентификационные номера.

Ниже указаны все имеющиеся на данный момент типы идентификационных номеров Android-аппаратов.

2.1. 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 [7].

IMEI присваивается телефону во время изготовления на заводе. 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. Опорная сеть GSM хранит IMEI в EIR [2].

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 в практически любой другой аппарат. Однако существуют специальные системы, позволяющие одному телефону использовать только одну определённую SIM-карту.

Согласно СТБ 1356-2002 “Система сухопутной подвижной цифровой сотовой связи общего пользования GSM 900” система сотовой связи состоит из:

1. Подсистемы коммутации, которая включает:

• центр коммутации подвижных служб, выполняющий функции коммутации вызовов

абонентов системы и стационарной сети;

• регистр пользователя — централизованную сетевую базу данных, хранящую информацию обо всех зарегистрированных абонентах сети данного оператора и видах услуг, которые могут быть им оказаны;

• регистр регистрации посетителя — базу данных, хранящую информацию обо всех

абонентах, находящихся в данное время в зоне обслуживания центра коммутации подвижных служб;

• центр аутентификации — базу данных, взаимодействующую с регистром пользователя с целью определения подлинности абонента и недопущения несанкционированного использования сети;

• регистр идентификации оборудования — базу данных, содержащую информацию об оборудовании подвижных станций.

2. Подсистемы базовых станций, которая включает:

• контроллер базовых станций;

• базовые станции, состоящие из приемопередающего оборудования и антенных систем. Базовые станции обеспечивают информационный обмен с подвижными станциями.

3. Подвижных станций, которые состоят из абонентского терминала (именуемого в быту как сотовый телефон) и модуля подлинности абонента (SIM).

4. Подсистемы эксплуатации и обслуживания сети GSM, которая, как правило, включает центр эксплуатации и технического обслуживания — компьютерный центр управления функциями эксплуатации и обслуживания центра коммутации подвижных служб [1].

Исходя из содержания указанного стандарта IMEI идентифицирует конкретный сотовый телефон в системе сотовой связи аналогично тому, как, например, номер кузова автомобиля идентифицирует автомобиль в системе регистрации автотранспортных средств (при указанной аналогии государственный регистрационный номер автотранспортного средства можно сравнить с SIM-картой) [1].

Модель и происхождение телефона описываются первыми 8 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 совершенствуют методы защиты программного обеспечения аппарата от изменения IMEI. В современных аппаратах IMEI хранится в однократно программируемой зоне памяти и не может быть изменён программными средствами [4].

В некоторых странах, например в Латвии, Великобритании, Республике Беларусь изменение IMEI является уголовно наказуемым деянием. Имеется также прецедент попытки уголовного преследования за изменение IMEI в России [1][6].

Структура IMEI и IMEISV.

IMEI (14 десятичных цифр плюс контрольная цифра) содержит информацию о происхождении, модели и серийном номере устройства. Первые 8 цифр составляют модель и место происхождения устройства, и известны как TAC (Type Approval Code). Остальная часть — определяемый производителем серийный номер аппарата, с высчитанной по алгоритму Луна контрольной цифрой в конце. До 2003 года эта цифра обязательно должна была равняться 0. Позже это правило было отменено [5][7].

IMEISV (International Mobile Terminal Identity и Software Version number) состоит из 16 цифр и обеспечивает уникальную идентификацию каждого мобильного телефона и соответствие версии программного обеспечения, инсталлированного в мобильный телефон, разрешенной оператором. От версии программного обеспечения зависят услуги, доступные для мобильного аппарата, а

также способность выполнить речевое кодирование и поэтому данный параметр весьма важен.

По состоянию на 2004 год формат IMEI представляет собой *AA-BBBBBB-CCCCCC-D*, хотя он не всегда может отображаться таким образом. В IMEISV вместо одного контрольного числа используются две цифры версии программного обеспечения, поэтому IMEISV выглядит как *AA-BBBBBB-CCCCCC-EE* [5].

До 2002 TAC состоял только из 6 цифр, оставшиеся 2 цифры составляли код места окончательной сборки (FAC). С 1 января 2003 и до 1 апреля 2004 проходил переходной период, во время которого все коды FAC равнялись цифрам 00. В 2004 FAC прекратил своё существование, а TAC был расширен до 8 цифр [5].

Первые две цифры TAC — это официально зарегистрированный код RBI. RBI всегда десятичен, то есть он меньше чем 0xA0, что позволяет легко отличать IMEI от MEID, начало которого равно или больше, чем 0xA0 [5].

Для примера рассмотрим IMEI 35-209900-176148-1 или IMEISV 35-209900-176148-23:

> **TAC**: 35-2099 — код британского совета по согласованию телекоммуникаций (BABT) и номер модели 2099 (Alcatel One Touch 332)
> **FAC**: 00 — такой код значит что телефон был сделан во время переходного периода, когда FAC был упразднён. Во время существования FAC использовались, в том числе, и следующие коды: 67 — США, 19 или 40 — Великобритания, 78 или 20 —

Германия, 10 или 70 — Финляндия, 30 — Корея, 80 — Китай, 04 — Вьетнам

SNR: 176148 — серийный номер аппарата

CD: 1 — контрольное число

SVN: 23 — номер версии программного обеспечения, которое установлено на телефоне. Цифра 99 зарезервирована.

IMEI нового стиля выглядит немного по другому: 49-015420-323751 (немецкая Nokia 3110 classic) и имеют 8-значный TAC (49-015420) [5].

Вычисление контрольного числа.

Способ 1.

Для расчета последней цифры IMEI необходимо:

1. Сложить все цифры в нечетных положениях;

2. Заменить цифры на четных местах по формуле и сложить их:

0=0

1=2

2=4

3=6

4=8

5=1

6=3

7=5

8=7

9=9

3. К полученному числу прибавить результат, полученный в п.1.;

4. Если полученное число равно нулю либо кратно 10, тогда контрольное число IMEI равно 0. В противном случае

контрольная сумма равна числу, которое нужно прибавить к результату, чтобы получить ближайшее большее «круглое» число.

Методика расчета Check Digit на примере:
Попробуем рассчитать контрольное число CD для IMEI=3**5**4**1**9**0**0**2**3**8**9**6**4**4**3. Для этого нам необходимо выполнить следующие операции с IMEI кодом:

1. Сложить все цифры в нечетных положениях 3,4,9,0,3,9,4 :
3+4+9+0+3+9+4 = 32
2. Заменить цифры на четных местах 5,1,0,2,8,6,4 по формуле 0=>0, 1=>2, 2=>4, 3=>6, 4=>8, 5=>1, 6=>3, 7=>5, 8=>7, 9=>9 :
5,1,0,2,8,6,4 => 1,2,0,4,7,3,8
и сложить их:
1+2+0+4+7+3+8 = 25
3. К полученному числу 25 прибавить результат 32 , полученный в п.1.
25 + 32 = 57
4. "Если полученное число равно нулю либо кратно 10, тогда контрольное число IMEI равно 0" - это не этот случай...
В противном (=нашем) случае контрольная сумма равна числу, которое нужно прибавить к результату, чтобы получить ближайшее бОльшее «круглое» число (т.е. следующий целый десяток).
Следующий целый десяток = 60.
К результату 57 надо прибавить 3 , чтобы получить ближайшее бОльшее «круглое» число 60.

Ответ:

Контрольное число = 3

Способ 2.

1) Удвоить значения цифр на четных позициях (5,1,0,2,8,6,4).

2) Сложить вместе удвоенные числа на четных позициях и не четных позициях, при этом «раскладывая» числа на четных позициях на составляющие числа (например, 14 нужно представить как 1 и 4). Получили 57.

3) Если конечное число заканчивается на 0, то CD = 0. В противном случае CD равно числу, которое нужно добавить к результату, полученному в пункте 2, чтобы получить следующий целый десяток. Следующий десяток после 57 это 60, 60-57 получаем 3. Итого контрольное число равно 3 [5].

2.2. MEID

MEID (Mobile Equipment IDentifier) – глобальный уникальный идентификатор подвижного оборудования, работающий в сетях CDMA, использует тот же базовый формат, что и IMEI [3][8].

MEID был создан на смену ESN [8].

2.3. ESN

ESN (Electronic Serial Number) – уникальный номер для идентификации CDMA мобильных телефонов [3].

2.4. 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) [9].

Длина IMSI, как правило, составляет 15 цифр, но может быть 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) [9].

2.5. 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 [2][10].

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) [10].

Адреса вроде MAC-48 наиболее распространены; они используются в таких технологиях, как Ethernet, Token ring, FDDI, WiMAX и др. Они состоят из 48 бит, таким образом, адресное пространство MAC-48 насчитывает 2^{48} (или 281 474 976 710 656) адресов. Согласно подсчётам IEEE, этого запаса адресов хватит по меньшей мере до 2100 года [10].

EUI-48 от MAC-48 отличается лишь семантически: в то время как MAC-48 используется для сетевого оборудования, EUI-48 применяется 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.

Возможно получить MAC-адрес Wi-Fi или Bluetooth оборудования устройства, однако не рекомендуется использовать его в качестве уникального идентификационного номера, так как не все мобильные устройства имеют Wi-Fi. Если Wi-Fi есть он должен бать обязательно включен, иначе MAC-адрес не определится. Кроме того MAC-адрес устройства можно изменить программным путем [10].

2.6. Serial Number

Серийный номер можно определить у устройств, не обладающих сервисом телефонии начиная с операционной системы Android 2.3 (“Gingerbread”) и у некоторых телефонов [3].

2.7. Android ID

Это 64 битный номер, который случайным образом генерируется при первом запуске устройства и остается неизменным далее. У устройств с операционной системой более ранних версий чем 2.2 (“Froyo”) он может не определяться [3].

3. Построение решения задачи

В своей работе я решила использовать IMEI номер и Android ID, чтобы увеличить надежность идентификация Android аппарата. Так как не у всех устройств определються оба эти номера. Хотябы один из них практически всегда определяется. Мобильные операторы для взаимодействия с телефоном используют IMEI номер.

Получившееся у меня мобильное приложение, определяет оба указанных номера. Далее пользователь может зарегистрироваться в удаленной базе данных. В эту базу данных будут вноситься IMEI номер и Android ID пользователя, а также ссылка на личную страницу пользователя в социальной сети. Таким образом, в случае потери или кражи телефона у одного человека, если этот телефон будет кем-то найден или куплен с рук, новый владелец может проверить по IMEI-базе кому до этого принадлежал телефон. Или полиция может использовать эти номера для того, чтобы найти украденный телефон.

Для получения ссылки на личную страницу пользователя в социальной сети я решила интегрировать мобильное приложение с Facebook. Это единственная социальная сеть у которой есть официальная библиотека для разработки мобильных приложений, со всеми необходимыми функциями работы в фэйсбуке с Android-аппарата.

Взаимодействие с получившейся базой данных осуществляется по средством сайта с базовыми функциями поиска.

4. Описание практической части

Мобильное приложение я решила разработать на Java, в среде Eclipse. Так как именно для этой среды существует плагин официальный Android Developer Tools (ADT), который предоставляет профессиональную среду разработки Android-приложений. Для определения IMEI номера и Android ID воспользовалась классами TelephonyManager и Settings. Для определения ссылки на страницу пользователя в фэйсбуке воспользовалась некоторыми функциями FacebookSDK. В результате при нажатии на кнопку регистрации пользователь перенаправляется в окно авторизации в фэйсбуке. Где он должен ввести свои логин и пароль в фэйсбуке.

Взаимодействие с удаленной базой данных осуществлено через PHP с использованием JSON, в отдельном классе JSONParser. С помощью функций из класса org.apache.http.

JSON (англ. *JavaScript Object Notation*) это текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком. Несмотря на происхождение от JavaScript, формат считается языконезависимым и может использоваться практически с любым языком программирования. Для Java и PHP существуют функции для создания и обработки данных в формате JSON.

Удаленная база данных представляет из себя MySQL базу с 4 столбцами: id, IMEI, AndroidID и Link. Взаимодействие с ней сделала таким образом, чтобы было невозможно добавить строку с каким-то определенным IMEI-номером или Android ID более одного раза.

Для того, чтобы информировать пользователя об успешной отправке запроса с телефона в базу данных, и успешной авторизации в Facebook, я добавила соответствующее всплывающее сообщение (Toast) и функцию добавления фото из профайла пользователя с его личной страницы. На рисунке 1 представлен общий вид после авторизации, получившегося в итоге приложения на Android-эмуляторе. А на рисунке 2 – вид окна авторизации в Facebook.

Рисунок 1.

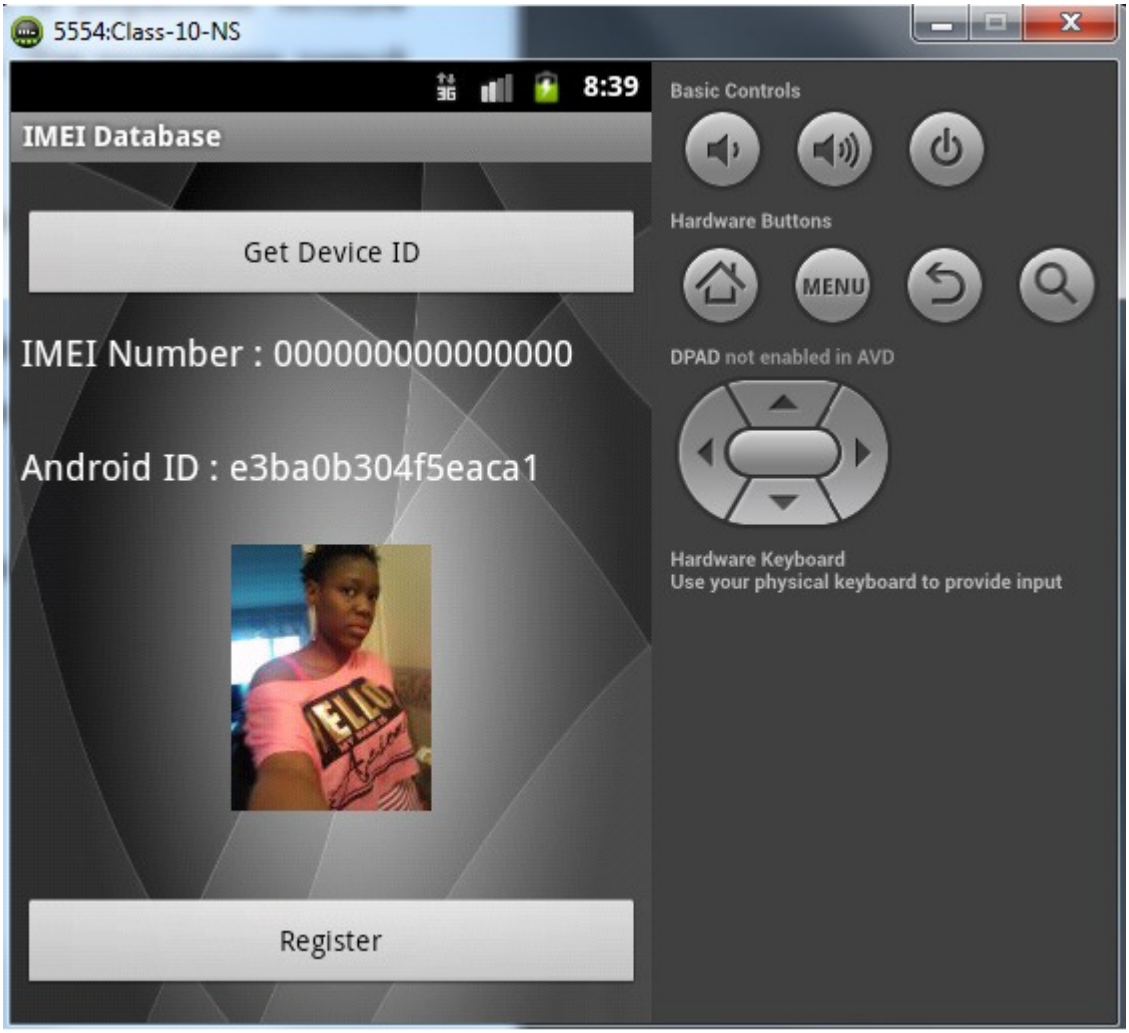
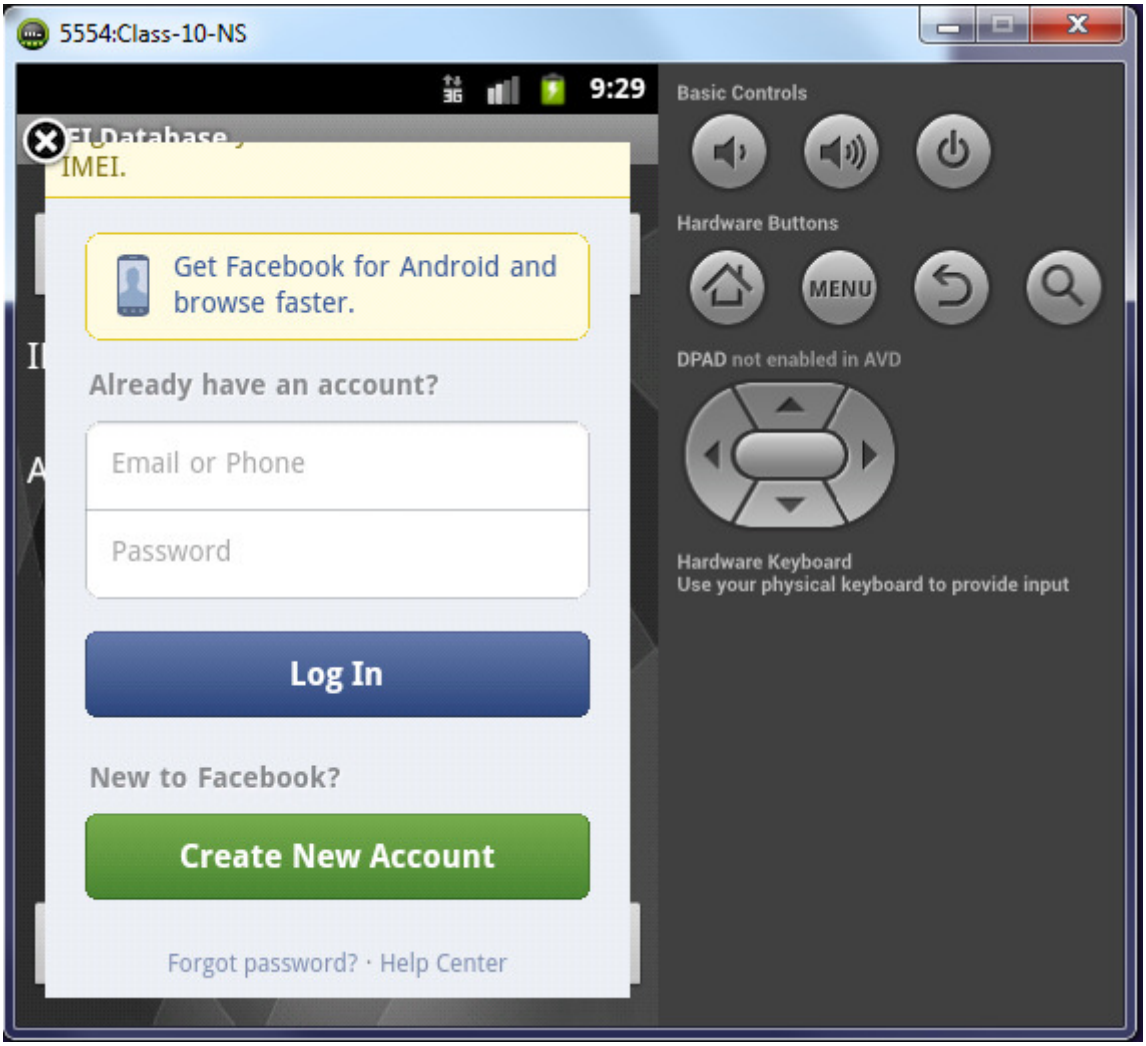


Рисунок 2.

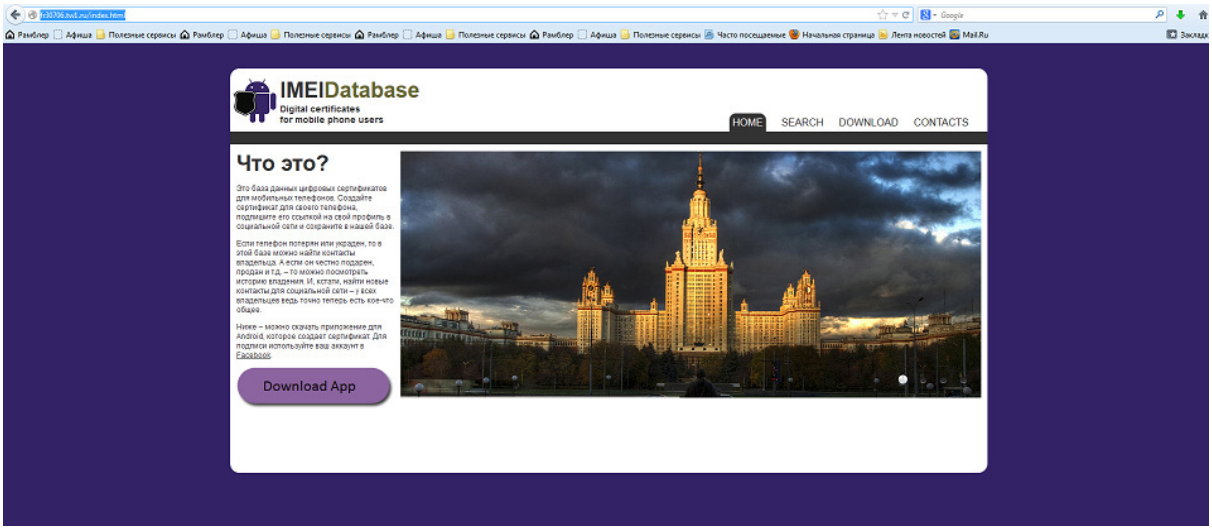


В приложении А представлен листинг получившегося Android-приложения.

Для того чтобы код работал необходимо также добавить в отдельном файле манифеста приложения строки о том, что приложение использует интернет и имеет право определять состояние телефона.

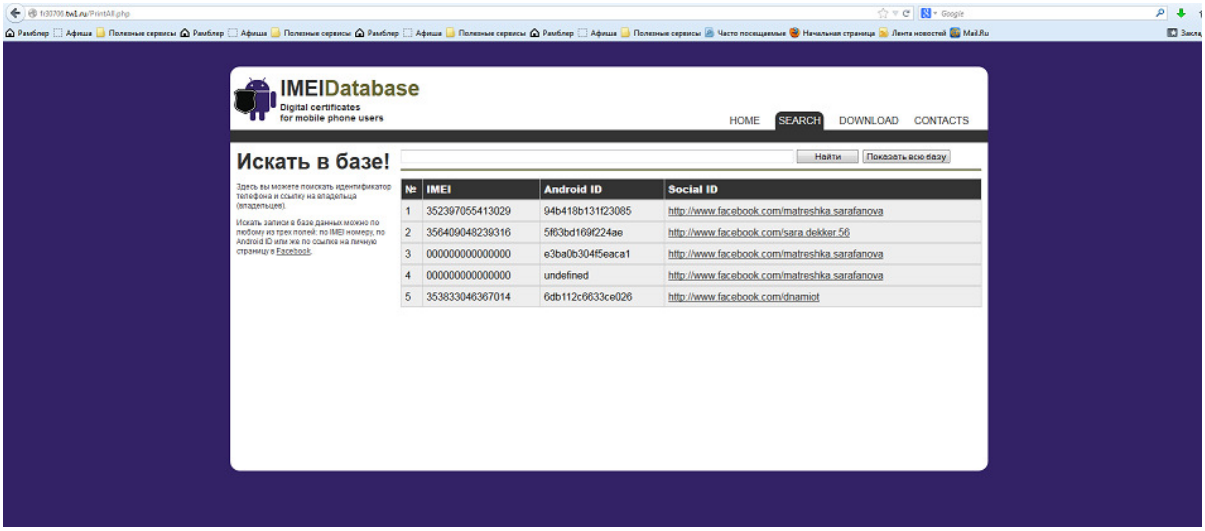
Далее я сделала сайт, который затем разместила в интернете по адресу http://fr30706.tw1.ru/. На рисунке 3 представлен вид главной страницы сайта.

Рисунок 3.



На этом сайте по вкладке «Search» можно перейти на страницу с функциональностью поиска по получившейся IMEI-базе данных. Взаимодействие сайта с базой данных осуществляется через PHP. Искать записи в базе данных можно по любому из трех полей: по IMEI номеру, по Android ID или же по ссылке на личную страницу в Facebook. На рисунке 4 представлен вид страницы поиска.

Рисунок 4.



Также на этом сайте можно скачать последнюю версию приложения для аппаратов с операционной системой Android.

Дизайн сайта разработан с поддержкой технологии Responsive Design, чтобы сайт хорошо отображался, и с ним было удобно работать как на стандартных компьютерных мониторах, так и на маленьких экранах смартфонов. Таким образом, у сайта реализованы 2 разные разметки.

На рисунке 5 представлен вид страницы поиска на экране смартфона. А на рисунке 6 – сравнительный вид страницы «DOWNLOAD» на экране монитора и на Android-эмуляторе.

Рисунок 5.

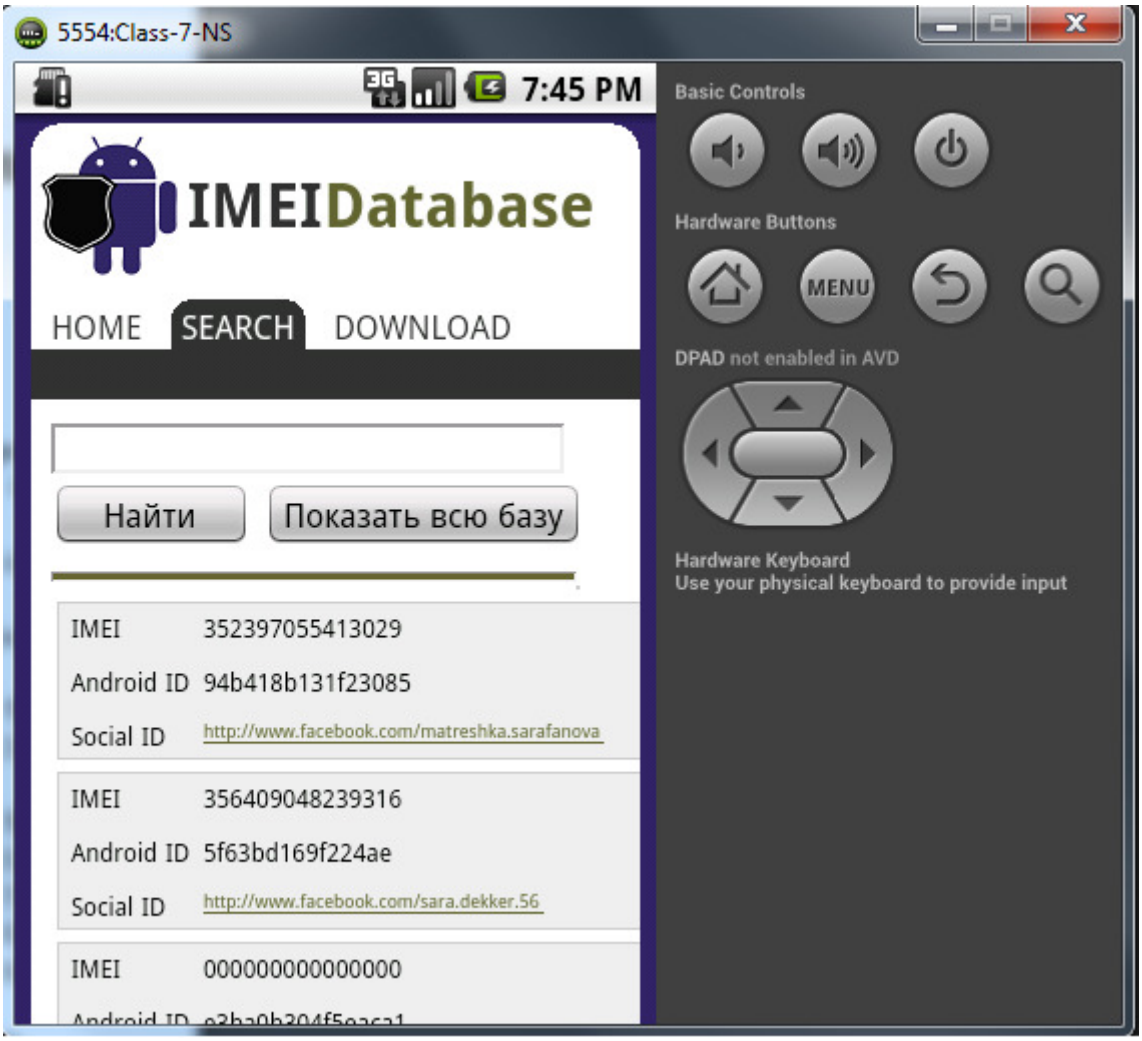
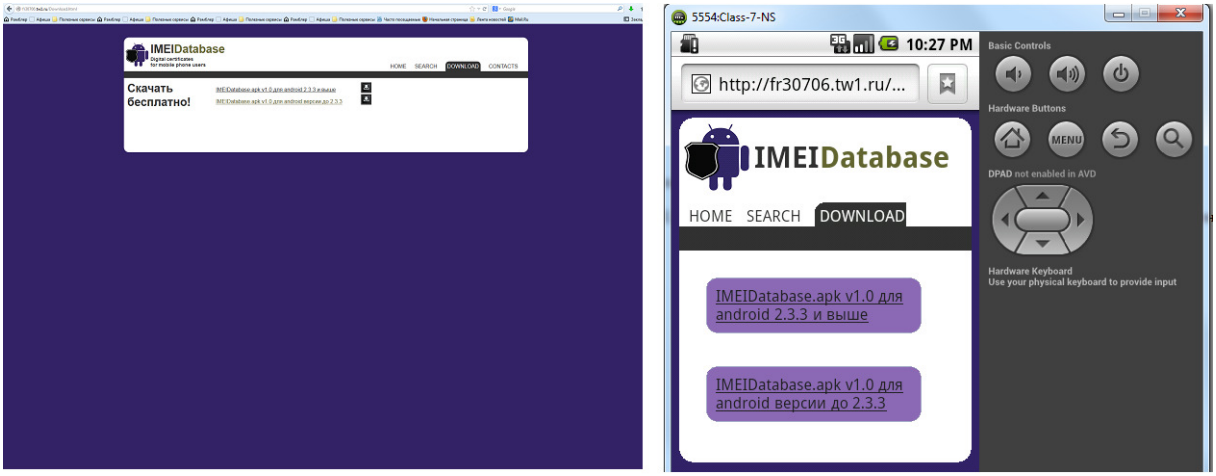


Рисунок 6. Сравнительный вид страницы «DOWNLOAD» на экране монитора и на Android-эмуляторе.



5. Заключение

Разработана и реализована модель цифровых сертификатов для владельцев мобильных устройств. Реализованы клиентские компоненты – мобильное приложение, веб-сайт для поиска по базе данных. И серверный компонент – база данных с интерфейсом для записи и поиска.

С помощью этой системы пользователь может создать сертификат для своего мобильного телефона, подписать его ссылкой на свой профиль в социальной сети и сохранить в базе данных.

Если телефон потерян или украден, то в этой базе можно найти контакты владельца. А если он честно подарен, продан и так далее – то можно посмотреть историю владения.

6. Приложение А

Листинг класса MainActivity.java:

```
package com.imeidatabase.funz;

import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.provider.Settings;
import android.telephony.TelephonyManager;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.facebook.android.DialogError;
import com.facebook.android.Facebook;
import com.facebook.android.Facebook.DialogListener;
import com.facebook.android.FacebookError;
import com.facebook.android.Util;
```

```
public class MainActivity extends Activity {

        Button getIdButton, registrationButton;
        TextView idViewIMEI, idViewAndroidId;
        ImageView pic;
        Facebook fb;
        String link, imeistring, androidId;
        private static String url_create_line =
"http://fr30706.tw1.ru/ADDimei.php";
        JSONParser jsonParser = new JSONParser();

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);

                pic = (ImageView)findViewById(R.id.profile_pic);
                getIdButton =
(Button)findViewById(R.id.button_get_device_id);
                registrationButton =
(Button)findViewById(R.id.button_registrate_in_db);
                idViewIMEI =
(TextView)findViewById(R.id.textView_device_id);
                idViewAndroidId =
(TextView)findViewById(R.id.textView_android_id);
                String APP_ID = getString(R.string.APP_ID);
                fb = new Facebook(APP_ID);

                StrictMode.enableDefaults();

                getIdButton.setOnClickListener(new View.OnClickListener() {

                        @Override
                        public void onClick(View v) {
```

```
                    TelephonyManager telephonyManager  =
(TelephonyManager)getSystemService( Context.TELEPHONY_SERVICE );
                    imeistring = telephonyManager.getDeviceId();
                    idViewIMEI.setText("IMEI Number : " + imeistring +
"\n");
                    androidId =
Settings.Secure.getString(getContentResolver(),
Settings.Secure.ANDROID_ID);
                    idViewAndroidId.setText("Android ID : " +
androidId + "\n");
                    registrationButton.setVisibility(View.VISIBLE);
                }
            });

            registrationButton.setOnClickListener(new
View.OnClickListener() {

                @Override
                public void onClick(View v) {
                    // login to facebook
                    fb.authorize(MainActivity.this, new DialogListener()
{

                        @Override
                        public void onFacebookError(FacebookError
e) {
                            Toast.makeText(MainActivity.this,
e.getMessage(), Toast.LENGTH_SHORT).show();
                        }

                        @Override
                        public void onError(DialogError e) {

                            Toast.makeText(MainActivity.this,
"onError", Toast.LENGTH_SHORT).show();
                        }
```

```
                @Override
                public void onComplete(Bundle values) {

                    JSONObject obj = null;
                    URL img_url = null;
                    try {
                        String jsonUser =
fb.request("me");
                        obj = Util.parseJson(jsonUser);
                    } catch (FacebookError e) {
                        // TODO Auto-generated catch
block
                        e.printStackTrace();
                    } catch (JSONException e) {
                        // TODO Auto-generated catch
block
                        e.printStackTrace();
                    } catch (MalformedURLException e) {
                        // TODO Auto-generated catch
block
                        e.printStackTrace();
                    } catch (IOException e) {
                        // TODO Auto-generated catch
block
                        e.printStackTrace();
                    }
                    link = obj.optString("link");
                    String id = obj.optString("id");
                    String user_name =
obj.optString("name");
                    try {
                        img_url = new
URL("https://graph.facebook.com/"+id+"/picture?type=normal");
                        Bitmap bmp =
BitmapFactory.decodeStream(img_url.openConnection().getInputStream())
;
                        pic.setImageBitmap(bmp);
```

```
                                        } catch (MalformedURLException e) {
                                                // TODO Auto-generated catch
block
                                                e.printStackTrace();
                                        } catch (IOException e) {
                                                // TODO Auto-generated catch
block
                                                e.printStackTrace();
                                        }
                                        Toast.makeText(MainActivity.this,
"Authorized as "+user_name, Toast.LENGTH_SHORT).show();
                                        new CreateNewLine().execute();
                                }

                                @Override
                                public void onCancel() {

                                        Toast.makeText(MainActivity.this,
"onCancel", Toast.LENGTH_SHORT).show();
                                }

                        });

                        }
                });
        }

        @Override
        protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
                super.onActivityResult(requestCode, resultCode, data);
                fb.authorizeCallback(requestCode, resultCode, data);
        }

   class CreateNewLine extends AsyncTask<String, String, String> {
                @Override
                protected String doInBackground(String... arg0) {
```

```
        // Building Parameters
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("IMEI", imeistring));
        params.add(new BasicNameValuePair("AndroidID", androidId));
        params.add(new BasicNameValuePair("Link", link));

        // getting JSON Object
        // Note that create product url accepts POST method
        JSONObject json = jsonParser.makeHttpRequest(url_create_line,
"POST", params);
                        return null;
                }
  }
}
```

Листинг класса JSONParser.java:

```
package com.imeidatabase.funz;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
```

```
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class JSONParser {
        static InputStream is = null;
    static JSONObject jObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    // function get json from url
    // by making HTTP POST or GET mehtod
    public JSONObject makeHttpRequest(String url, String method,
        List<NameValuePair> params) {

      // Making HTTP request
      try {

        // check for request method
        if(method == "POST"){
          // request method is POST
          // defaultHttpClient
          DefaultHttpClient httpClient = new DefaultHttpClient();
          HttpPost httpPost = new HttpPost(url);
          httpPost.setEntity(new UrlEncodedFormEntity(params));

          HttpResponse httpResponse = httpClient.execute(httpPost);
          HttpEntity httpEntity = httpResponse.getEntity();
          is = httpEntity.getContent();

        }else if(method == "GET"){
          // request method is GET
```

```
            DefaultHttpClient httpClient = new DefaultHttpClient();
            String paramString = URLEncodedUtils.format(params, "utf-8");
            url += "?" + paramString;
            HttpGet httpGet = new HttpGet(url);

            HttpResponse httpResponse = httpClient.execute(httpGet);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();
          }

        } catch (UnsupportedEncodingException e) {
          e.printStackTrace();
        } catch (ClientProtocolException e) {
          e.printStackTrace();
        } catch (IOException e) {
          e.printStackTrace();
        }

        try {
          BufferedReader reader = new BufferedReader(new
InputStreamReader(
              is, "iso-8859-1"), 8);
          StringBuilder sb = new StringBuilder();
          String line = null;
          while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
          }
          is.close();
          json = sb.toString();
        } catch (Exception e) {
          Log.e("Buffer Error", "Error converting result " + e.toString());
        }

        // try parse the string to a JSON object
        try {
          jObj = new JSONObject(json);
        } catch (JSONException e) {
```

```
            Log.e("JSON Parser", "Error parsing data " + e.toString());
        }

        // return JSON String
        return jObj;

    }
}
```

7. Список литературы

1. Журнал «Законность и правопорядок», No 3(7)/2008, статья В. Шалькевича, А. Макаревича «Противодействие теневому обороту мобильных телефонов уголовно правовыми мерами» (стр. 36-40).
2. http://ru.wikipedia.org/
3. http://developer.samsung.com/android/technical-docs/How-to-retrieve-the-Device-Unique-ID-from-android-device
4. GSME proposals regarding mobile theft and IMEI security (https://docs.google.com/viewer?a=v&q=cache:0mXtXE_yM3EJ:www.gsmeurope.org/documents/positions/gsme_proposals_mobile_thefts_imei_security.pdf+imei+standard&hl=en&gl=au&pid=bl&srcid=ADGEESgutC2Wv66x8SweH6Tb3AIpZ_e0FtPSpsHeFrswQiPqnm5TgPV440ooDWS_ElQc8aPkeimqNLbd969ngHkpbIbtCcVHQzi_PyYDa0LTFY1m7Pf0Fuh40RUMIpUq4Hf0cAl8ZND4&sig=AHIEtbStnM1cqejWPnMi2PpVLCDSSfJgJQ)
5. GSM Association Non Confidential Official Document IMEI Allocation and Approval Guidelines Version 6.0 (27th July 2011) (http://www.gsma.com/newsroom/wp-content/uploads/2012/03/ts0660tacallocationprocessapproved.pdf)
6. http://www.legislation.gov.uk/ukpga/2002/31/section/1
7. http://www.amta.org.au/pages/amta/FAQs.on.mobile.security
8. 3G Mobile Equipment Identifier (MEID) (3GPP2 S.R0048-A Version 4.0 Date: 23 June 2005)
9. Fred Gaechter "Chairman of IMSI Oversight Committee" (IOC)(GSMNA Doc 036/02) (http://www.ifast.org/files/IFAST22_015_GSMNALetter.pdf)
10. IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture (IEEE Std 802®-2001 (R2007)(Revision of IEEE Std 802-1990))
11. D. Namiot. Network Proximity on Practice: Context-aware Applications and Wi-Fi Proximity. International Journal of Open Information Technologies, 1(3), 2013, pp. 1-4.
12. Namiot, D. “Geo messages”, In Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on (pp. 14-19). IEEE. DOI: 10.1109/ICUMT.2010.5676665