

Для серверной части приложения была выбрана технология .Net, а именно её фреймворк *WebApi 2*, который позволяет быстро создавать простой *restful* сервис.

В качестве базы данных выбрана *MSSQL* как входящая в комплект поставки *Microsoft Visual Studio Enterprise*. Для работы с базой данных используется *EntityFramework* с подходом *code first*.

Список использованных источников:

1. Девятко И.Ф., Онлайн исследования и методология социальных наук: новые горизонты, новые (и не столь новые) трудности/ Онлайн-исследования в России 2.0. М.: РИЦ «Северо-Восток», 2010. — С.17-31. — ISBN 978-5-9901939-1-8 https://www.hse.ru/data/2013/01/23/1306499729/Online_Research_in_Russia_2_complete.pdf.

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА PATCHWORK ДЛЯ РЕАЛИЗАЦИИ СИСТЕМЫ ВОДЯНЫХ ЗНАКОВ

Ждан В.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ярмолик В.Н. – д.т.н., профессор

В работе рассматриваются угрозы интеллектуальной собственности, а также способы защиты от них. Особое внимание уделяется системам цифровых водяных знаков, а также использованию стеганографического алгоритма patchwork в рамках его применения для реализации системы водяных знаков.

Для программного обеспечения и других объектов интеллектуальной собственности актуальными являются следующие угрозы: несанкционированное использование, обратное проектирование и несанкционированная модификация.

Одно из направлений для ликвидации данных угроз – системы цифровых водяных знаков. Данное направление использует различные методы: графическая стеганография, текстовая стеганография, обфускация и другие.

Стеганография — это наука, которая занимается вопросами скрытой передачи информации путём сохранения в тайне самого факта передачи. Открытый текст, где скрыта зашифрованная стеганографическим алгоритмом информация, называется контейнером. Графическая стеганография использует в качестве контейнера графические файлы, например растровые изображения [1].

Алгоритм Patchwork является одним из базовых алгоритмов графической стеганографии. В его основе лежит статистический подход. Суть этого алгоритма состоит в следующем. При помощи криптостойкого генератора псевдослучайных чисел, используя заранее известный секретный ключ, выбираются два пикселя изображения. Затем значение яркости одного из них увеличивается или уменьшается на некоторое значение в зависимости от типа встраиваемой информации. Значение яркости другого – уменьшается или увеличивается на ту же величину соответственно. Процесс повторяется около 10000 раз. Значение приращения варьируется от 1 до 5 [2].

Пусть значения яркостей выбираемых на каждом шаге пикселей a_i и b_i , а величина приращения δ . Тогда сумма разностей значений пикселей после n преобразований:

$$S_n = \sum_{i=1}^n [(a_i \pm \delta) - (b_i \mp \delta)] = \pm 2\delta n + \sum_{i=1}^n (a_i - b_i)$$

Математическое ожидание величины $\sum_{i=1}^n (a_i - b_i)$, то есть суммы разности значений пикселей в незаполненном контейнере равно нулю, и его значение в неизменном изображении будет много меньше чем n . Математическое ожидание величины S_n в заполненном контейнере будет равно $\pm 2\delta n$, а вычисленное значение S_n будет иметь большой соответствующий n порядок. В стегодекодере используя ключ проверяется значение S_n . Решение о значении встроенного бита выносится в зависимости от знака S_n . При положительной разности 1, а при отрицательной – 0 [3].

Таким образом, владелец может доказать свои интеллектуальные права, предъявив секретный ключ, который использовался для выборки изменяемых пикселей изображения.

Существует способы улучшения алгоритма для повышения его робастности. Например, можно использовать квадратные или прямоугольные блоки вместо отдельных пикселей изображения. Алгоритм Patchwork является достаточно стойким к операциям усечения, сжатия, изменения гистограммы изображения. Основной недостаток алгоритма – это неустойчивость к геометрическим преобразованиям: сдвигу, повороту, масштабированию. Другим недостатком является малая пропускная способность, что не дает возможности встраивания значительных по размеру цифровых водяных знаков. К примеру, в базовой версии алгоритма для передачи 1 бита скрытой информации необходимо порядка 20 000 пикселей. Таким образом в 20-мегапиксельном изображении можно передать только 1000 бит, то есть около 120 байт скрытого сообщения.

Список использованных источников:

1. Грибунин, В. Г. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев. – М.: СОЛОН-Пресс, 2002. – 261 с.
2. Семёнов К. П. Алгоритмы встраивания цифровых водяных знаков в растровые изображения / К. П. Семёнов, П. В. Зайцев // Информационная безопасность регионов : научно-практический журнал. – 2011. – №1. – С. 46–50.
3. Bender W. Techniques for Data Hiding / W. Bender, D. Gruhl, N. Morimoto, A. Lu // IBM Systems Journal. – 1996. – Vol. 35.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПРОСМОТРА РАСПИСАНИЯ ОБЩЕСТВЕННОГО ТРАНСПОРТА НА ПЛАТФОРМЕ XAMARIN С ИСПОЛЬЗОВАНИЕМ УСТРОЙСТВА ТИПА «УМНЫЕ ЧАСЫ»

Жизневский В.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Трус В.В. – ст. преподаватель каф. ПОИТ

Все большую популярность набирают устройства типа «умные часы». Некоторые современные модели являются многофункциональными устройствами под управлением полноценных операционных систем. Например, линейка Gear от компании Samsung под управлением операционной системы Tizen. Однако полностью свой потенциал «умные часы» раскрывают при работе в связке со смартфоном. В связи с этим большой перспективой обладают комплексные решения, в состав которых входят как приложение для смартфона, так и приложение для часов.

Не редки ситуации, когда необходимо на ходу составлять маршруты внутри черты города, или же иметь быстрый доступ к актуальной информации движения общественного транспорта, например знать ближайшее время прибытия на остановку автобуса по определенному маршруту. И на данный момент есть приложения для смартфонов, которые предоставляют функции для составления маршрутов, просмотра расписаний движения общественного транспорта. Но использование в некоторых случаях часов дает преимущества, т. к. часы являются более мобильным устройством по сравнению с телефоном. Одним из эффективных сценариев использования часов является отображение наиболее актуальной для пользователя информации. Однако редактирование и ввод информации лучше производить на смартфоне, т. к. он обладает более удобными средствами для ввода информации по сравнению со смарт-часами.

В рамках проекта было принято решение о создании мобильного приложения для просмотра расписания общественного транспорта с использованием устройства типа «умные часы». Помимо основного функционала, в которых входит отображение остановок общественного транспорта на карте, просмотр расписания общественного транспорта, добавление расписаний движения общественного транспорта через остановку в избранное, данное приложение будет подразумевать сценарий использования смарт-часов, а именно отображение расписаний движения общественного транспорта через остановку из списка избранного.

В качестве платформы для части приложения, которая будет работать на стороне смартфона, была выбрана платформа Xamarin, т.к. данная платформа позволяет создавать кросс-платформенные приложения для Android и iOS [1]. В качестве целевых устройств типа «умные часы» были выбраны устройства линейки Gear от компании Samsung под управлением операционной системы Tizen. Согласно отчету компании Strategy Analytics за 4-й квартал 2018 устройства компании Samsung входят в тройку по количеству проданных, что говорит о популярности моделей данных часов [2]. Для разработки части приложения, работающего на часах будут использованы инструменты Tizen SDK [3]. Тип приложения будет Tizen Web, т. к. приложения такого типа обладают наибольшей совместимостью с устройствами линейки Gear [4].

В качестве компонента отображения ближайшего времени отправления на смарт-часах будет использован виджет.

Предполагаемый сценарий создания и инициализации виджета: