разрабатываемого приложения, как для популярных настольных ОС, так и для популярных мобильных операционных систем.

Также большинство существующих аналогов не позволяют вести «трансляцию в реальном времени». Одним из плюсов разрабатываемого программного средства будет являться «вещание в прямом эфире» с возможностью обмена мгновенными сообщениями с автором эфира. Это позволит удобно организовать трансляции типа: «преподаватель - студенты» или «руководящее звено - служащие».

Большая доля внимания будет уделена проектированию максимально простого и интуитивно понятного пользовательского интерфейса. Независимо от операционной системы, за основу будет взят единый графический интерфейс. При первом запуске программы пользователю будет предложено ознакомиться с основными возможностями программного средства. Процесс регистрации и авторизации будет максимально упрощен, с возможностью выполнять эти процедуры через различные социальные сети.

Все функции разрабатываемого приложения будут абсолютно бесплатны.

С учетом всего вышеперечисленного можно сформировать список общих требований к разрабатываемому программному средству:

1. Программное средство должно позволять совершать голосовые звонки;
2. Программное средство должно позволять совершать видео звонки;
3. Программное средство должно позволять совершать обмен мгновенными текстовыми сообщениями между пользователями;
4. Программное средство должно позволять совершать обмен файлами между пользователями;
5. Программное средство должно предоставлять возможность записи разговора или видео звонка с последующим сохранением на жесткий диск ПК;
6. Программное средство должно предоставлять возможность трансляции в реальном времени;
7. Программное средство должно предоставлять возможность демонстрации экрана устройства собеседника.

Список использованных источников:

1. Skype [Электронный ресурс]. – 2015. – Режим доступа: http://www.skype.com/ru
2. WhatsApp [Электронный ресурс]. – 2015. – Режим доступа: http://www.whatsapp.com
3. Viber [Электронный ресурс]. – 2015. – Режим доступа: http://www.viber.com/ru
4. RaidCall [Электронный ресурс]. – 2015. – Режим доступа: http://www.raidcall.com.ru

ПРОГРАММНОЕ СРЕДСТВО ЗАЩИТЫ ПОЧТОВЫХ СООБЩЕНИЙ

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

Таразевич С. Ю.

Прохорчик Р. В. – м. т. н., ассистент

В настоящее время электронные средства передачи информации практически вытеснили классические. Одним из самых актуальных вопросов является обеспечение конфиденциальности и целостности передаваемых данных, так как, как правило, данные хранятся на серверах, неподконтрольных пользователям.

Основной проблемой, представляющей угрозу конфиданциальности электронных сообщений, является то, что они хранятся на удаленных серверах, следовательно, пользователь никогда не может быть уверен, что его информация не будет доступна третьим лицам. В последнее время поступает довольно много сообщений об утечках информации с серверов различных крупных компаний, что подтверждает существование проблем с конфиденциальностью при таком подходе к организации хранения данных [2]. Существует много решений, использущих шифрование пользовательской информации на удаленных серверах, но данный подход также не гарантирует конфиденциальность, так как нет уверенности в добросовестности компаний, предоставляющих даные услуги. С другой стороны, у хранения данных на сторонних серверах есть ряд существенных преимуществ: доступ к информации через сеть интернет, не требуются аппаратные ресурсы пользователя. Таким образом, хранение информации на удаленных серверах, при обеспечении ее конфиденциальности, во многих случаях является оптимальным решением.

Для обеспечения конфиденциальности данных при хранении их на удаленном сервере можно использовать шифрование на стороне клиента. При данном подходе данные будут зашифрованы до отправки их на сервер, что, при использовании криптостойкого алгоритма шифрования, позволит защитить их от третьих лиц, не полагаясь на защитные механизмы удаленного сервера.

Для реализации шифрования электронных писем наиболее оптимальным вариантом будет использование симметричного алгоритма шифрования по следующим причинам:

- при наличии ключа любой пользователь сможет расшифровать сообщение, при ассиметричном алгоритме необходимо шифровать сообщение для каждого адресата отдельно с использованием его открытого ключа.

- скорость работы симметричных алгоритмов выше, чем асимметричных [1].

Можно выделить 2 стратегии создания ключей при использовании симметричного алгоритма

шифрования:

- ввод ключа пользователем;
- генерация и хранение ключей на отдельном сервере, которому доверяют все участики процесса обмена информацией.

При вводе ключа пользователем он должен передать этот ключ всем адресатам каким-либо образом. Данный способ прост в реализации, так как не требует отдельного сервера для распределения и хранения ключей. Но у него есть существенный минус, так как пользователи должны передавать друг другу ключи безопасным способом.

Для реализации автоматической генерации ключей, необходимо создать для этого сервер, которому должны доверять все участники обмена информацией. Все пользователи предварительно должны быть авторизованы на этом сервере. В этом случае, перед шифрованием сообщения пользователь делает запрос на сервер для получения ключа. Сервер генерирует ключ и идентификационный номер сообщения, сохраняет их в базе данных и передает пользователю через защищенный канал, использующий протокол HTTPS или асимметричный алгоритм шифрования на уровне приложения. Пользователь, получив ключ и идентификационный номер сообщения, шифрует сообщение, добавляет к нему идентификационный номер и отправляет адресатам, а список адресатов отправляет на сервер. Адресаты, получив сообщение, читают из него идентификационный номер и отправляют запрос на сервер для получения ключа для расшифровки этого сообщения. Сервер, получив запрос на получение ключа, проверяет, имеет ли данный пользователь право на получение данного ключа, если проверка прошла успешно, ключ отправляется пользователю. Пользователь, получив ключ, использует его для расшифровки сообщения.

Данный подход сложнее в реализации, но проще для пользования. Он подходит для корпоративного использования, так как сервер для генерациии и хранения ключей можно разместить на компьютерах в локальной сети, что решает проблему доверия к этому серверу.

Немаловажным фактором является удобность использования программного средства для шифрования информации на клиенте. Обычно люди пользуются определенной программной-клиентом при работе с электронной почтой. Их выбор основан на таких факторах, как функциональность, удобство интерфейса, соответствие другим персональным требованиям. При реализации программного средства как отдельного приложения пользователю придется отказаться от использования привычного почтового клиента, если он хочет воспользоваться функциями шифрования. Для пользователя это неудобно, так как он привык работать с определенным клинтом, он можен сомневаться в качестве реализации функций, присущих обычному почтовому клиенту, в новом приложении.

Таким образом, наиболее удобным для пользователя способом реализации программного средства шифрования на клиенте являются расширения для уже существующих почтовых клиентов. Кроме удобства такой подход также удобен тем, что не нужно реализовывать функционал обычного почтового клиента. Это позволяет упростить разработку и не заниматься поддержкой функциональности, не связанной с шифрованием.

В настоящее время наиболее широко используемыми почтовыми клиентами являются клиенты, реализованные как веб-интерфейсы. Поэтому наиболее целесообразным является разработка расширений для браузеров, которые бы внедряли функции шифрования в почтовые веб-клиенты. При таком подходе существует еще одно немаловажное преимущество – расширения для браузеров разабатываются на языке JavaScript. Таким образом, можно разработать ядро программного средства, которое будет осуществлять шифрование, в виде библиотеки на языке JavaScript и использовать при реализации расширений для различных браузеров.

Список использованных источников:

1. Ярмолик В.Н., Портянко С.С., Ярмолик С.В. Криптография, стеганография и охрана авторского права. – Мн.:Издательский центр БГУ, 2007. – 242 с.
2. Habrahabr [Электронный ресурс]. – Режим доступа: http://habrahabr.ru/company/mailru/blog/230743/, свободный. – Загл. с экрана. – Яз. Русский

МОДУЛЬ ВЫЗОВА И ПРЕДОСТАВЛЕНИЯ УДАЛЕННЫХ СЕРВИСОВ ДЛЯ ПЛАТФОРМЫ RUBY ON RAILS

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

Перминов В. В.

Бахтизин В. В. – к-т. техн. наук, профессор

Сервис-ориентированная архитектура (СОА) – относительно новый архитектурный стиль в разработке программного обеспечения. Множество стандартов по передаче и обработки информации основаны на сервис-ориентированной архитектуре. СОА разделяет задачи на отдельные модули, называемые сервисами, которые могут быть распределены по сети. Сервис – общий интерфейс, который предоставляет