Организация системы безопасности распределенных социальных сетей

О.Ю.Пескова, А.Г.Богораз, А.С.Власов
Институт компьютерных технологий и информационной безопасности
Южного федерального университета
poy@tsure.ru, bogoraz.a.g@gmail.com, asis.asv@gmail.com

Аннотация

В статье рассматриваются вопросы безопасности социальных сетей. Представлены особенности построения распределенных социальных сетей, которые создавались для повышения уровня конфиденциальности пользовательских данных. В работе описаны наиболее характерные схемы организации подобных сетей на примере двух распределенных социальных сетей – Diaspora и Frendica. Представлена защищенная схема передачи данных в распределенных социальных сетях, основанная на использовании протоколов OverIP.

На данный момент, социальные сети, основанные на полностью децентрализованной распределенной архитектуре, считаются защищенными, так как они построены на базе протоколов и технологий, которые изначально разрабатывались для предоставления максимальной защищенности. Но это заблуждение, поскольку угрозы и уязвимости присутствуют и здесь. Авторами начата разработка архитектуры системы защиты подобных систем.

Ключевые слова: Социальные сети, децентрализованные социальные сети, распределенные социальные сети, персональные данные, Diaspora, Frendica

Введение

Классические социальные сети имеют клиент-серверную архитектуру, и во многом это определяет их достоинства и недостатки, в том числе с точки зрения защиты данных пользователей социальных сетей, поскольку централизованное хранение всех пользовательских данных повышает риск неправомерного использования данных и снижает возможности самого пользователя по управлению ими. В последние годы стали разрабатываться социальные сети нового типа, основанные на архитектуре распределенного типа – распределенные социальные сети (РСС), и цель их создания – прежде всего повышение уровня конфиденциальности пользовательских данных.

В РСС личные данные пользователя хранятся исключительно на его

Сборник научных статей XVIII Объединенной конференции «Интернет и современное общество» IMS-2015, Санкт-Петербург, 23-25 июня 2015 г.

клиентском компьютере. Пользователь целиком и полностью определяет доступ к своим данным для любого другого пользователя распределенной социальной сети. В обычных социальных сетях сервер используется для всех клиентских действий, таких как общение, хранение информации о других пользователях и прочее. В распределенной социальной сети сервер может либо использоваться для соединения пользователей между собой, либо вообще отсутствовать. Таким образом, можно выделить два типа такого вида социальных сетей – клиент-серверные и полностью децентрализованные.

Клиент-серверные распределенные социальные сети работают аналогично обычным социальным сетях, но с серьезными архитектурными отличиями. В распределенных социальных сетях данного типа сервер используется только для соединения клиентов между собой. Преимуществами этого решения можно назвать отсутствие необходимости хранить информацию пользователей и отсутствие доступа к личным данным пользователя. К минусам можно отнести возможность взлома сервера и последующей прослушки пользовательских коммуникаций, а также отслеживание личных данных пользователей, передаваемых между общающимися через сервер пользователями.

Полностью децентрализованные распределенные социальные сети отличаются от клиент-серверных сетей полным отсутствием сервера и строятся по аналогии с классическими пиринговыми (p2p) сетями. Клиент такой сети устанавливается на компьютер пользователя и является как клиентом, так и сервером одновременно.

Преимуществами данного решения можно назвать:

- возможность для пользователя целиком и полностью определять доступ других пользователей к свои личным данным;
- отказоустойчивость – если один клиент будет взломан, сеть продолжит работу;
- невозможность прослушивания коммуникаций пользователей, так как соединение между клиентами шифруется;
- отсутствие возможности отследить личные данные пользователя.

Минусы:

- возможная замедленность работы по сравнению с обычными социальными сетями,
- недостаточно качественная работа на медленных каналах сети Интернет;
- необходимость пропускать через себя большое количество запросов;
- не всегда актуальная информация о состоянии частей системы.

Рассмотрим наиболее характерные схемы организации подобных сетей на примере двух наиболее развитых РСС – Diaspora и Frendica.

1. Распределенная социальная сеть Diaspora

1.1. Архитектура РСС Diaspora

РСС Diaspora состоит из сети взаимосвязанных узлов, которые называются подами. Каждый узел принадлежит конкретному пользователю и работает на своей копии движка, являясь, по сути, отдельным веб-сервером. Пользователи

сети могут создать либо свой собственный под, либо аккаунт на любом из этих серверов, но при этом они будут взаимодействовать со всеми остальными серверами.

Кроме того, каждый пользователь может создать несколько подов для разных целей, например, сформировав личный под, который может быть использован исключительно для общения с семьей, близкими друзьями. Для повышения уровня безопасности такой под можно никогда не объединять с основной сетью.

Пользователи Diaspora сохраняют права на свою информацию и никому их не передают. Движок Diaspora позволяет пользовательским постам быть «публичными» или «ограниченными». Во втором случае посты может прочитать только определённая группа или несколько групп, указанных пользователем. Движок предоставляет возможность в любой момент сохранить все отправленные сообщения и загруженные данные, после чего удалить аккаунт.

Сервера PCC Diaspora взаимодействуют в двух случаях: когда ищут информацию о пользователе с другого сервера и когда отсылают информацию тем, с кем пользователь делится своими публикациями.

Плюсы такого подхода:

- возможность детальной настройки доступа к информации для различных пользователей и групп;
- хранение персональной информации на стороне клиента;
- возможность создания различных подов для одного пользователя.

Но при этом остаются проблемы передачи транзитных данных через под другим клиентам: хотя само соединение между подами и клиентами будет зашифровано, остается актуальной проблема компроментации или подмены пода либо самих данных.

1.2. Криптографическая защита в PCC Diaspora

Поды PCC Diaspora должны быть доступны для поиска пользователей с выдачей информации об их адресах по стандартам протокола Webfinger. Поиск может быть осуществлен и по имени, но только в том случае, если оно уже было добавлено в список имен пода (например, локальные пользователи). Пользователь может запретить выдачу информации о себе при обработке поисковых запросов.

Webfinger — это открытый протокол, который используется для поиска о пользователях, идентифицируя их по URI (Uniform Resource Identificator, Унифицированный Идентификатор Ресурса) [1]. Webfinger-профиль пользователя содержит ссылку на его hcard (HTML vCard), куда включены персональные данные из профиля. Когда пользователь создает аккаунт в поде, PCC ассоциирует его аккаунт со случайным шестнадцатеричным числом, состоящим из не менее, чем 8 шестнадцатеричных цифр, так называемый guid (Global Unique Identifier, Глобальный Уникальный Идентификатор). Под должен использовать guid для создания ссылки hcard и разместить эту информацию в том же месте, что и hcard. В качестве базового алгоритма шифрования используется RSA. При создании пода генерируется пара ключей PGP (Pretty

Good Privacy), которые в дальнейшем используются для подписи и шифрования сообщений.

Ключ обрабатывается следующим образом: берется представление ключа в кодировке ASCII, конвертируется по специальному методу кодирования DER, шифруется с помощью криптографического стандарта PKCS#1 и затем кодируется еще раз по алгоритму base64.

1.3. Обмен сообщениями

Все типы публикаций отправляются с помощью специального протокола Salmon - это протокол отправки сообщений, работающий через HTTP и разработанный для децентрализации комментариев и аннотаций, которые публикуются под новостными сообщениями.

Отправка сообщений – это процесс, состоящий из 3 этапов:

1. Создание сообщения.
2. Создание URL протокола Salman для получателя.
3. Отправка сообщения.

Сообщения, которые пересылаются удаленным пользователям, зашифрованы. Это помогает защитить конфиденциальность сообщений во время пересылки, даже если сообщение по протоколу Salmon будет пересылаться с помощью обычного HTTP-сообщения, а не зашифрованного с помощью SSL. Стоит отметить, что, хотя сообщение будет пересылаться в зашифрованном виде, оно может быть расшифровано подом получателя и затем храниться в открытом виде.

Для поддержки семантики шифрования, PCC Diaspora расширяет так называемый «Магический конверт» протокола Salmon для добавления в него зашифрованного заголовка. Под «Магическим конвертом» протокола Salmon понимается сообщение вместе с подписью этого сообщения, в виде ряда параметров, представленные в виде компактной строки формата JSON или XML. «Конверт» определяет данные, которые должны быть подписаны, тип MIME данных, тип кодирования передачи и подпись.

При создании зашифрованного заголовка используются ключ, подбираемый по симметричному алгоритму AES, и вектор инициализации, подходящие для режима AES-256-CBC. Далее они будут называться «внутренний ключ» и «внутренний вектор инициализации» соответственно. Создается XML-фрагмент <decryption_header>, в котором содержатся данные криптографии, а также полное имя и URI отправителя сообщения.

На следующем шаге создается еще одна пара ключевой информации - AES-ключ и вектор инициализации. Они будут далее называться «внешний ключ» и «внешний вектор инициализации».

Далее, XML-фрагмент <decryption_header> шифруется с помощью внешнего ключа и внешнего вектора инициализации (с помощью использования AES-256-CBC). Этот зашифрованный фрагмент называется зашифрованным текстом.

Внешние ключ и вектор инициализации сохраняются в JSON-объект, который будет называться «пакет внешнего AES ключа». Он шифруется с помощью открытого RSA-ключа получателя. Зашифрованный пакет внешнего AES ключа и зашифрованный текст сохраняются также в JSON-объекте,

который называется «зашифрованный заголовок JSON объекта» и сохраняется в отдельном XML-фрагменте <encrypted_header>.

Фрагмент <encrypted_header> сохраняется для последующего использования. Он будет добавляться «Магический конверт» протокола Salmon для создания расширенного протокола Salmon, используемого PCC Diaspora.

«Внутренний ключ» и «внутренний вектор инициализации» также сохраняются. Они будут использоваться для зашифровывания полезной нагрузки сообщений, которой можно делить с удаленными пользователями.

На последнем этапе создания сообщения из заголовка шифрования и подготовленной полезной нагрузки сообщений формируется «магический конверт». Подпись создается согласно спецификациям, указанным в «Спецификации Магического Конверта». В данном случае, используется алгоритм RSA-SHA256 для подписания базовой строки с помощью закрытого ключа отправителя.

Для создания URL протокола Salmon делается следующее:

1. Берется местонахождение пода получателя.
2. С помощью протокола Webfinger определяется guid получателя
3. Фомируется URL протокола Salmon <pod_url>/receive/users/<guid>.

Суть получения заключается в повторении действий алгоритма отправления сообщения, но в обратном порядке. Полезная нагрузка также содержит указатель на автора сообщения. Через протокол Webfinger получатель может запросить открытый ключ отправителя и проверить электронную цифровую подпись [2].

2. Распределенная социальная сеть Frendica

2.1. Архитектура PCC Frendica

При разработке PCC Friendica акцент делался на обширные настройки конфиденциальности. Используется интеграция с другими социальными сетями, в частности, возможна синхронизация контактов из других социальных сетей, таких как Facebook, Twitter, Diaspora и другие. Может служить в качестве моста для электронной почты [3].

Можно выделить следующие особенности системы в целом:

- децентрализованная архитектура;
- реализация двухсторонних отношений с любой совместимой системой, что позволяет создать масштабируемую интернет-систему из небольших сайтов;
- административный интерфейс с доступом к настройкам сайтов, настройке дополнений, аудиту и мониторингу, управлению пользователями;

Особенности использования профилей:

- публичный профиль, предоставляемый по умолчанию, может быть ограничен;
- может быть создано множество профилей, уникально сконфигурированных каждый для своей целевой аудитории;

- существующий профиль может быть размножен для создания точной копии, с различиями в незначительных деталях;
- интерактивный редактор позволяет выбирать, кто из «Друзей» и какой профиль может просматривать.

С точки зрения конфиденциальности, можно выделить еще ряд особенностей:

- шифрование данных при обмене между серверами (на поддерживаемых сетях);
- разграничение доступа с помощью листов контроля доступа (индивидуальные и групповые листы с возможностью установления запрета и разрешения);
- организация групп частного общения, где все коммуникации доступны только для членов группы;
- полный контроль публикации профиля в сетевых каталогах и настроек видимости профилей, при этом существует возможность поддерживать различную персональность профиля для отдельных пользователей;
- сетевая «электронная почта», которая предоставляет возможность отправки частных сообщений типа «пользователь-пользователь» любому пользователю;
- опциональная возможность автоматического удаления старого контента профиля по истечению определенного времени; контент будет удален со всех серверов Friendica, которые хранили копию, при этом существует возможность скачать и сохранить свои данные [4].

2.2. Базовые механизмы, используемые РСС Friendica

В качестве базового протокола Friendica в основном использует протокол Zot [5] – это веб-фреймворк, разработанный для реализации защищенных распределенных коммуникаций и сервисов. Он отличается от других протоколов такого типа тем, что строит коммуникации на основе децентрализованной идентичности и аутентификации.

Приоритетные задачи, для решения которых создавался Zot [6]:

- создание полностью децентрализованных коммуникаций;
- изоляция от DNS-идентичности с использованием некоторого глобального идентификатора;
- обеспечение мобильности узлов;
- высокая производительность.

В некотором смысле протокол Zot аналогичен стандартному протоколу web-аутентификации OpenID, но является изолированным от идентичности, основанной на DNS.

Например, пользователь User1 хочет поделиться фотографиями из своего блога по адресу «user1.com.xyz» с пользователем User 2, но более ни с кем другим. User2 поддерживает свой собственный семейный сайт по адресу «user2.com.xyz» с несколькими пользователями. Протокол Zot позволяет User1 создать лист доступа, который будет содержать имя «User2». Пользователю User2 необходимо войти в систему только один раз, на своем собственном сайте по адресу «user2.com.xyz», используя свой пароль. User3 может иметь учетную

запись на сайте «user2.com.xyz», но он не будет иметь доступ к фотографиям User1. Дополнительно, протокол Zot позволяет User2 использовать другой сайт - «user2_2.com.xyz», и после входа в систему через этот сайт, он также сможет иметь доступ к фотографиям User1 – не имеет значения, через какой сайт доверенный пользователь войдет в систему.

Для предоставления такой функциональности Zot создает децентрализованный глобальный уникальный идентификатор для каждого узла сети [7]. Этот идентификатор не связан с DNS, обеспечивая необходимую мобильность. Многие существующие децентрализующие коммуникационные протоколы предоставляют возможность связи, но не контроль удаленного доступа и аутентификации. Большинство из них основано на протоколе Webfinger, как в PCC Diaspora, что означает, что в случае примера, описанного выше, User2 будет «распознан», только если он попытается получить доступ к фотографиям User1, войдя в систему через сайт «user2.com.xyz», но не войдя в систему через сайт «user2_2.com.xyz».

Мы будем основываться на утверждении, что DNS-адрес вида user@host (пользователь@компьютер) – это механизм, который позволяет узнать местонахождение пользователя (который находится, соответственно, на компьютере с указанным названием и полученным именем), и коммуникации будут осуществляться через DNS-запись, с использованием протокола TCP и web.

Основной протокол позволит обеспечить уровень абстракции поверх этого для того, чтобы коммуникационный узел мог перейти в другую DNS-локацию и восстановить (или продолжить поддерживать) существующие до этого отношения связей. Побочным эффектом этого требования является необходимость общаться с альтернативными/множественными DNS-локациями и провайдерами услуг, а также поддерживать одну постоянную идентичность. Эта ячеистая сеть будет называться грид (grid); серверы, подключающиеся к этой сети, называются хабами (hub) и могут поддерживать любое число индивидуальных идентичностей.

Адреса, которые распространяются между пользователями, имеют вид user@host и описывают текущие данные местонахождения для заданной идентичности: они основаны на DNS и используются в качестве указателя для того, чтобы найти заданную личность внутри сети. Машинные связи соотнесут этот адрес с соответствующим глобальным уникальным идентификатором (ГУИ). Один ГУИ может быть привязан к любому количеству DNS-локаций. После того, как личность один раз будет подключена к DNS-локации, связь будет осуществляться, основываясь на знании ГУИ и того, какая DNS-локация на данный момент используется.

Для того, чтобы идентичность сохранялась между локациями, должны быть доступны или восстановимы следующие данные:

- ГУИ для этой идентичности;
- закрытый ключ, ассоциированный с этой идентичностью;
- адресная книга контактов для этой идентичности (в случае, если изначальный сервер более не существует).

Эта информация может быть экспортирована из изначального сервера с помощью API, и/или доступна к загрузке с диска или флеш-накопителя. Можно

попытаться провести процедуру восстановления с еще меньшим количеством информации, но это будет аналогично взлому аккаунта и будет требовать подтверждения изменений от ваших друзей.

В целях реализации высокой производительности, в Zot используется только один формат передачи данных – JSON.

Двунаправленное шифрование осуществляется на основе RSA-ключей длиной 4096 бит, выраженных в формате DER/ASN.1 на основе кодирования PKCS#1, с использованием алгоритма AES-256-CBC для блочного шифрования различной длины или больших файлов.

2.3. Создание ГУИ

Создание идентификатора будет основываться на криптографической хэш-функции Whirlpool, для которой входными данными являются URL идентичности и псевдослучайное число, что в итоге дает идентификатор длиной 256 бит. Это число будет представляться в виде строки, закодированной по base64. Базирование идентификатора на DNS-локациях позволяет создать глобальный уникальный указатель, который был бы невозможен, если бы ГУИ был основан исключительно на псевдослучайном числе. Далее такой идентификатор будет именоваться zot_uid.

Так как может быть много DNS-локаций, присоединенных к конкретной zot_uid идентичности, процесс доставки должен быть осуществлен по отношению к ним всем – так как не известно, какая DNS-локация будет активной в конкретный момент времени. Также существует возможность сменить одну локацию на другую. Просмотр информации на текущей локации может предоставлять так называемую «точку перенаправления», которая будет уведомлять о том, что нужно обновить записи и обратиться к новой локации.

Для ассоциирования zot_uid с вторичной (или третичной) локацией, необходимо запросить безопасный обмен. Процесс связи верифицируется с помощью хранимого закрытого ключа, который был скопирован, когда было установлено первое соединение с zot_uid.

В целях устранения путаницы, должен существовать только один четко определенный URL для любого хаба, который будет проиндексирован и ассоциирован с ним. Также необходимо, чтобы все адреса имели тип HTTPS, валидный сертификат, запрашиваемый веб-браузером, и один уникальный компонент host, который будет использоваться для всех видов коммуникаций. Множественные URL могут использоваться локально, но только один уникальный URL должен использоваться для всех видов связи по протоколу Zot внутри сети grid.

Если существует такая возможность, то протокол Zot будет осуществлять передачу между двумя хабами с помощью «транзакций». Это означает, что узел A может послать множественные сообщения узлу B с помощью одной транзакции, сконсолидировав одинаковые сообщения, которые необходимо доставить множеству получателей на одном хабе.

2.4. Передача сообщений и организация поиска

Учитывая ограничения, перечисленные выше, процесс связи по протоколу Zot должен выглядеть как строка индивидуальных сообщений в формате JSON. Они могут быть скомбинированы и перемешаны в пределах одной и той же передачи.

Каждое сообщение должно содержать следующие параметры:

- тип;
- список получателей (опционально).

Отсутствие списка получателей будет указывать на то, что сообщение либо является незашифрованным (открытым), либо сообщением уровня узла. Список получателей будет содержать строку zot_uid с индивидуальным ключом расшифрования и соответствующим вектором инициализации IV (Initialization Vector). Ключ расшифрования кодируется с помощью открытого ключа получателя. IV будет шифроваться с помощью закрытого ключа отправителя. Все сообщения должны быть подписаны цифровой подписью отправителя.

Тип сообщения может быть, в частности, следующим:

- публикации (или активность);
- работа с почтой;
- изменение личности;
- аутентификация.

URL используется для проверки хаба на наличие возможностей протокола Zot, а также для просмотра профиля идентичности, включая открытие закрытых ключей, локаций и фото профиля и выяснения основной локации профиля.

Для подготовки просмотра профиля необходимо создать POST-запрос, с помощью которого будет выяснена локация со следующими характеристиками:

- address – указатель на целевую систему (к примеру – Джон) – наличие этого параметра обязательно;
- target – наблюдатель ГУИ по протоколу Zot для оценки разрешений – опциональный параметр;
- target_sig – подпись ГУИ протокола RSA (кодированная по base64);
- key – открытый ключ, необходимый для верификации подписи.

При этом, если цель не будет указана, ограничения будут указываться как для неизвестного или неаутентифицированного наблюдателя.

2.5. Аутентификация (Magic Auth)

Аутентификация (Magic Auth) происходит с помощью специального подпротокола обмена. На удаленном компьютере происходит перенаправление в точку протокола Zot со специальными GET параметрами. В качестве точки в данном случае, будет указан адрес https://example.com/post/name.

Используются 4 параметра:

- auth – адрес вида user@host пользователя, запрашивающего доступ;
- dest – желаемый пункт назначения по URL (urlencoded);
- sec – случайная последовательность, хранящаяся локально, необходимая для фазы верификации;
- version – версия протокола Zot.

Когда происходит получение пакета, Zot отправляет аутентифицируемой личности сообщение типа «auth_check», которое содержит подписанные ГУИ и URL отправителя, ГУИ получателя, подписанное поле secret. Оно шифруется с помощью алгоритма AES-256-CBC и отсылается на исходный сайт, который проверяет, что «secret» имеет то же значение, что и «sec», переданное до этого, а также имеет верную подпись. Если все проверки прошли успешно, то будет отправлено соответствующее сообщение.

После получения и положительной верификации данного пакета, сайт назначения будет перенаправлять на пакеты на оригинальный URL назначения и указывать на успешный удаленный вход [6].

3. Защищенные схемы передачи данных в распределенных социальных сетях

Распределенные социальные сети строятся на использовании протоколов OverIP – это класс протоколов передачи данных, которые упаковываются в стандартные пакеты стека TCP/IP, ничем не отличающиеся от «обычных» пакетов и используемые для передачи по открытым каналам передачи между обычными компьютерами в сети Интернет.

Иными словами, данные отправителя шифруются, упаковываются в протокол OverIP, далее получившиеся пакеты упаковываются в пакеты стека TCP/IP, а затем пересылаются по открытому каналу передачи данных. Получатель, приняв такие пакеты, распаковывает пакеты TCP/IP, затем распаковывает пакеты протокола OverIP, и уже после расшифровывает данные (рис.1).

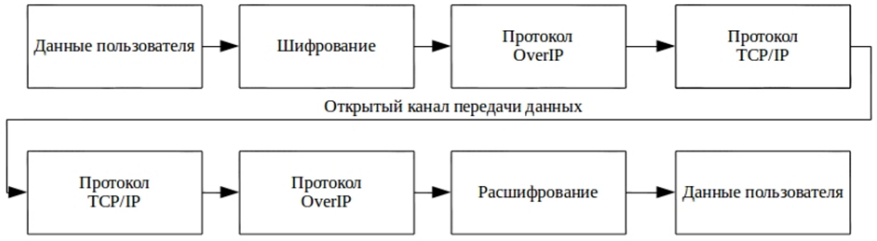

Рис.1. Передача данных с использованием OverIP

Таким образом, при передаче данных по открытому каналу достоверность получателя сообщения будет проверяться трижды:

- на первом этапе – когда сообщение отправляется получателю, проверяется правильность IP-адреса, указанного в заголовке пакета TCP/IP;
- на втором этапе – проверяется получатель сообщения в РСС, который должен знать, каким протоколом OverIP надо воспользоваться при распаковке пакетов;
- на третьем этапе – получатель должен знать ключ, которым необходимо расшифровывать сообщение отправителя.

На данный момент, социальные сети, основанные на полностью децентрализованной распределенной архитектуре, считаются защищенными сами по себе, так как они построены на базе протоколов и технологий, которые изначально разрабатывались для предоставления максимальной защищенности. Но это заблуждение, поскольку угрозы и уязвимости присутствуют и здесь. Авторами начата разработка архитектуры системы защиты подобных систем.

В полностью децентрализованных социальных сетях сервер отсутствует как объект, и поэтому необходимо защищать только клиент, который одновременно является сервером для других клиентов. В случае с клиент-серверными распределенными социальными сетями, вследствие сходства архитектуры между ними и клиент-серверными социальными сетями, методы и средства защиты могут быть просто перенесены с минимальными корректировками. Аналогично необходимо защищать сервер, который можно взломать, и, в случае взлома, прослушать любые коммуникации пользователей.

Архитектура новой РСС строится на основе классических пиринговых сетей, но с доверенным "человеком посредине", который владеет корневым сертификатом. Сеть – одноранговая на уровне «сеанс-адрес», но отдельные узлы могут иметь разные роли, в частности, выполнять функции клиента и сервера. Выделяются 3 группы узлов по отношению к каждому пользователю:

- узлы из контакт-листа («друзья»);
- узлы, доступные через «друзей»;
- незнакомцы в пределах определенного количества человеко-узлов, т.е. глубины связей.

Вход в сеть - по приглашению пользователя либо через узел-посредник.

Для каждого пользователя создается личная карточка, которую можно представить как аналог профиля / персональной страницы в классических социальных сетях или личных данных в программах-мессенджерах типа Skype. Карточка хранится изначально только на узле пользователя, но к ней можно дать доступ (доверить хранение) – полному списку «друзей» либо отдельным «друзьям» (доверенным серверам).

Элементы учетной записи пользователя следующие:

1. Логин - это личная информация, которая не равна номеру/адресу узла и может быть не равна имени/отображаемому нику в сети.
2. Пароль+ контейнер для ключевой информации (например, для своего открытого ключа)
3. Пригласительный билет, который будет нужен всегда при первом входе в сеть с нового устройства.

Сеть динамически дополняемая, поэтому возникает серьезная проблема распределения адресов. На данный момент адрес просто выдается на базе адреса узла, пригласившего новичка в сеть (поля добавляются к родителю: по аналогии с номером сети в Infotecs ViPNet и SNode в Skype).

Каждый узел имеет ближнее окружение – контакт лист и их сертификаты двух уровней, где первый просто позволяет обратиться не по номеру узла, а по имени пользователя и получить ответ без поручителя, а второй уже позволяет дать/получить роль сервера заданного уровня доверия.

Каждый узел имеет дальнее окружение – это доступные узлы «в пределах видимости» (заданной глубины поиска), с нахождением их через общего

сервера («друзей») или через широковещательный поиск – «псевдофлуд» на своем ключе по номеру узла сети.

При посредничестве или поиске узла, глубина поиска не только определяется счетчиком переходов, но и обязательно включает в себя путь передачи с номерами узлов, для возможности проверки. Подмена, пересылка недоверенного запроса или данных, «неответ» по своему ключу (неответ на закрытом ключе) - повод для отказа в доверии.

Пользователь не эквивалентен физическому узлу и может заходить в сеть с различных устройств (проходя соответствующую аутентификацию). Таким образом, каждый узел имеет зеркала: адреса, где есть его учетная информация.

Количество, тип и категория доступных личных данных пользователям различных будут определяться исключительно пользователем-владельцем этих данных. Для максимальной прозрачности системы для пользователя существует возможность просмотра своих личных данных, хранимых других пользователей. Это позволит пользователю более тонко контролировать данные, которые будут храниться у друзей, и тех данных, которые пользователь определит как общедоступные, которые могут храниться у любого пользователя такой сети.

Криптографическая защита сети обеспечивается использованием ряда алгоритмов, в том числе RSA в качестве базового асимметричного криптоалгоритма, и схемы Диффи-Хеллмана для распределения ключевой информации, шифрование трафика предполагает использование симметричных методов шифрования (на данном этапе в качестве основного принят российский стандарт шифрования ГОСТ 28147-89, готовится переход на предлагаемые варианты нового стандарта шифрования).

Заключение

Благодаря развитию человеческого общения, социальные сети стараются гармонично развиваться и подстраиваться под нужды человеческого общения (а иногда и самостоятельно формировать эти нужды). Информационные технологии, к которым принадлежат социальные сети, сами по себе являются динамично развивающейся областью, которая своей эволюцией может отражать как современные тенденции, так и направления возможного дальнейшего развития процессов человеческого общения.

Распределенные полностью децентрализованные социальные сети создавались для того, чтобы дать возможность пользователям управлять своей информацией, а также ее доступностью другим пользователям, что позволяет обеспечить большую защищенность всей структуры. Но и этот тип социальных сетей не может решить все проблемы, связанные с работой в сети Интернет. Авторами начата разработка архитектуры системы защиты подобных систем.

Литература

[1] Distributed social network protocols // Diaspora Project Wiki. URL: https://wiki.diasporafoundation.org/Distributed_social_network_protocols (дата обращения: 20.04.2015).
[2] Federation protocol overview // Diaspora Project Wiki. URL: https://wiki.diasporafoundation.org/Federation_protocol_overview (дата обращения: 20.04.2015).
[3] Friendica - Википедия // Википедия – свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Friendica (дата обращения: 20.04.2015).
[4] Features | Friendica // The internet is our social network | friendica. URL: http://friendica.com/features (дата обращения: 20.04.2015).
[5] View source for Zot Communication Protocol // P2P Foundation. URL: http://p2pfoundation.net/Zot_Communications_Protocol?title=Zot_Communications_Protocol&action=edit (дата обращения: 20.04.2015).
[6] zot – friendica/red Wiki – GitHub // GitHub – Build software better, together. URL: https://github.com/friendica/red/wiki/zot (дата обращения: 20.04.2015).
[7] Zot Communications Protocol – P2P Foundation // P2P Foundation. URL: http://p2pfoundation.net/Zot_Communications_Protocol (дата обращения: 20.04.2015)

The organization of security arrangement of the distributed social networks.

O. Peskova, A.Bogoraz, A.Vlasov
Southern Federal Univercity

The article discusses the security of social networks. The features of distributed social networks that are designed to enhance the privacy of user data are presented. In such networks, the user defines the access to the data for any other user of a distributed social network. The server can either be used only for the connection between users (client-server network) or absent (completely decentralized network). This paper describes the most typical schemes of such networks on the example of two distributed social networks - Diaspora and Frendica. The protected scheme of data transmission on the distributed social networks based on use of the OverIP protocols is shown. At the moment, social networks based on a completely decentralized, distributed architecture are considered to be protected, as they are based on protocols and technologies that were originally developed to provide maximum protection. But this is misleading, because the threats and vulnerabilities are present here. The authors started to develop the architecture to protect similar systems.The architecture is based on the classical peer to peer networks. But individual nodes can have different roles. For each user the personal card is created. The user is not equivalent to the physical node and can access the network from multiple devices (passing the authentication). Each node has a near and far environment.

Keywords: Social networks, decentral social networks, distributed social networks, personal information, Diaspora, Frendica