I2P

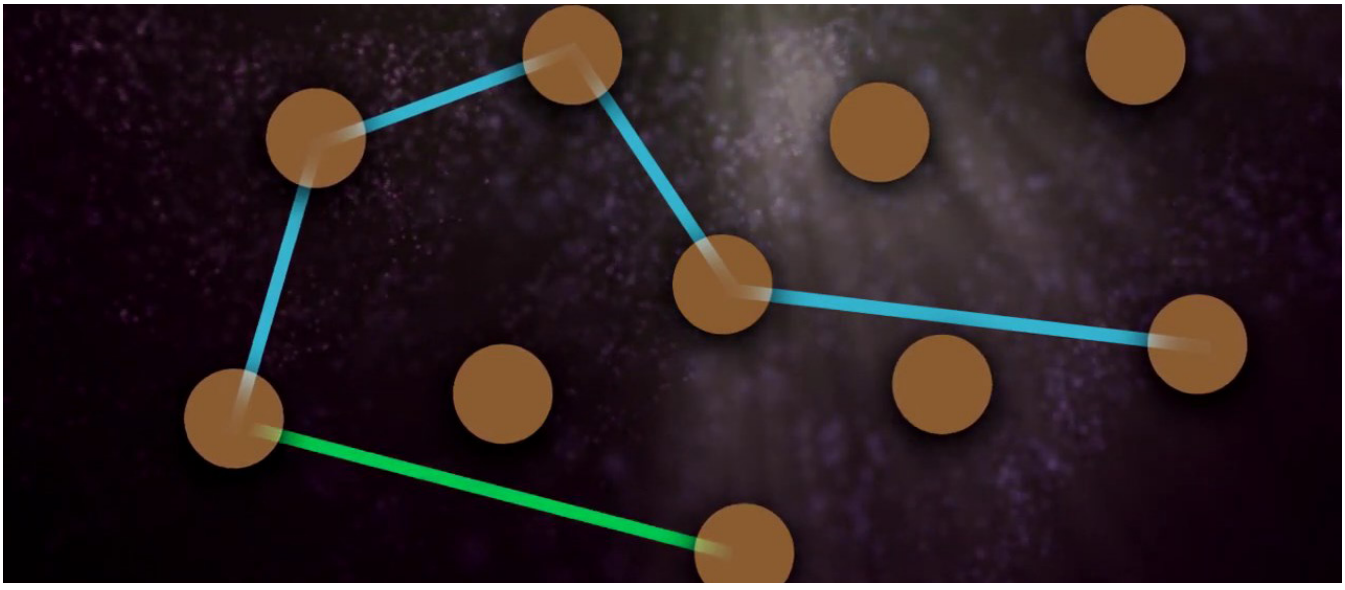
INVISIBLE INTERNET PROJECT

ПРОЕКТ НЕВИДИМОГО ИНТЕРНЕТА: КАК УСТРОЕНА САМАЯ АНОНИМНАЯ СЕТЬ

COMMUNITY

Агрессивный рынок готов заработать на всем, что возможно продать, но популярные предложения в сфере приватной коммуникации являются фикцией и ложью. Это происходит по простым причинам: во-первых, бизнес придумали не меценаты, а корыстные люди; во-вторых, любой предприниматель даже при самых лучших побуждениях подотчетен третьим лицам и в спорных ситуациях свои интересы всегда ставит выше интересов клиента.

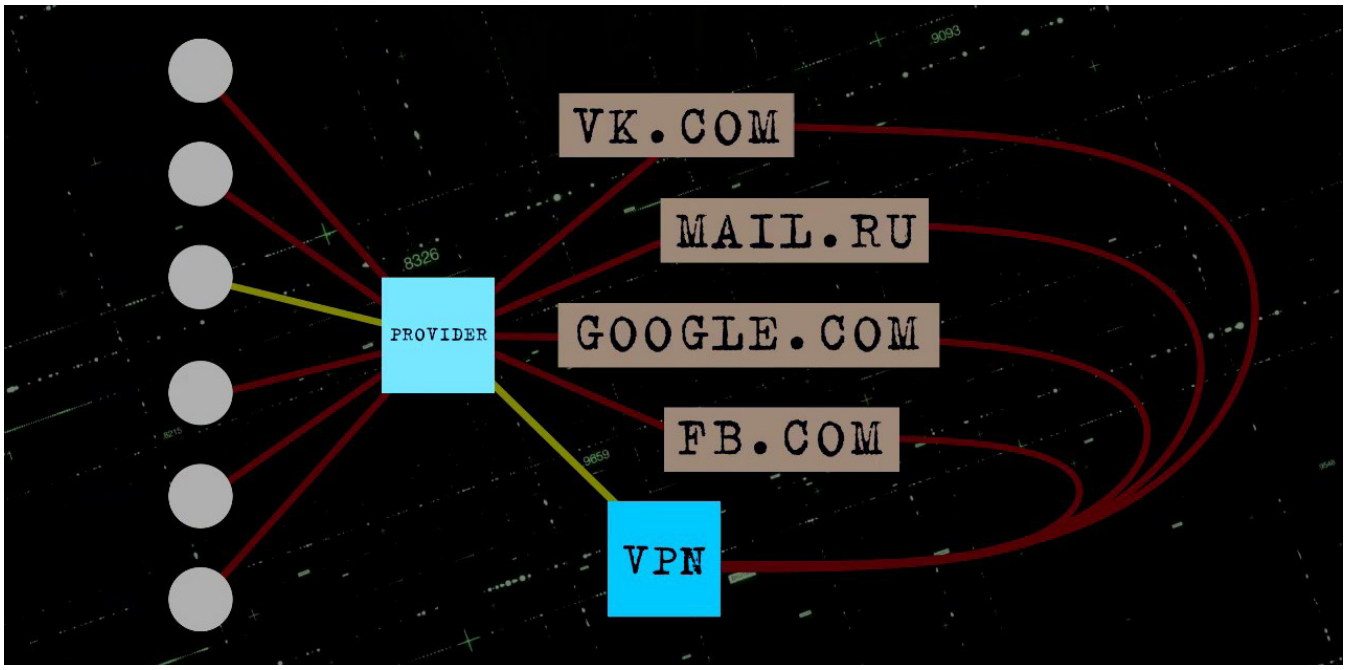
Сейчас пойдет речь про I2P — некоммерческую сингулярность сетевой приватности и анонимности, где никто кроме вас не знает куда и кто передает вашу информацию. Сеть I2P расшифровывается как «Invisible Internet Project» (Проект невидимого интернета) — это оверлейная децентрализованная одноранговая сеть. Оверлейная — значит работает поверх других сетей, например, обычного интернета; децентрализованная — распределенная, не имеющая единой точки отказа: упадет один узел, полсети, или во всей сети останется 3 пользователя — I2P все равно будет функционировать. I2P является одноранговой сетью, так как все участники имеют равные права и возможности: каждый пользователь скрытой сети строит свои туннели через других участников и сам является потенциальным звеном в цепочке другого пользователя. При этом естественная сетевая активность никак не компрометирует абонента перед домашним провайдером или участниками скрытой сети.



Если вы никогда не сталкивались с I2P, эта брошюра — начало вашего пути к по-настоящему приватной и анонимной коммуникации; если вы бывалый энтузиаст скрытых сетей, этот обзор наверняка заполнит пробелы и поможет упорядочить уже имеющиеся знания.

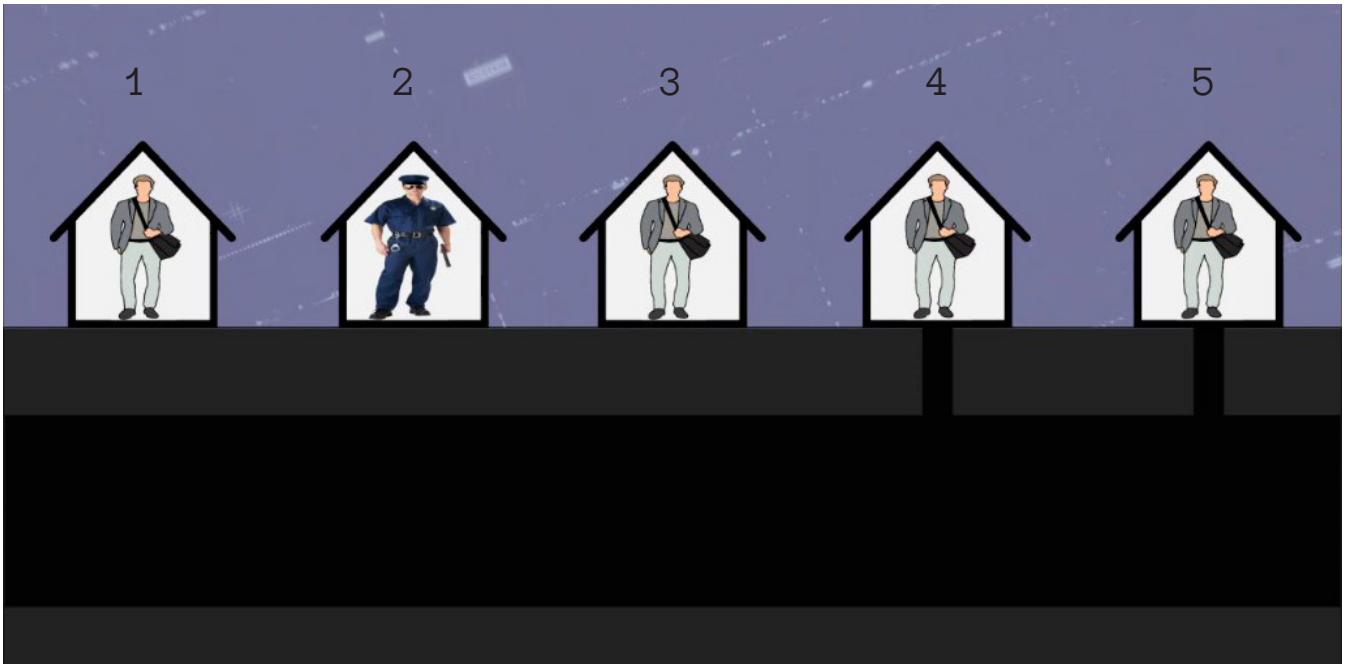
Отличие традиционной сети от скрытой

Традиционный интернет является ничем иным, как разветвленной сетью компьютеров. Он устроен так, что отследить действия пользователей не составляет трудности: интернет-провайдеру доступна история вашего веб-серфинга хотите вы этого или нет. Это может быть и VPN-провайдер, в любом случае о ваших интересах знают некие третьи лица. Благодаря этому глобальная сеть превращается в инструмент манипуляций и однобокой пропаганды.



Скрытые сети, так называемый «даркнет», нацелены на обратное — полную анонимность пользователей.

На примере схематичного городка изобразим различия между обычным интернетом и примитивной скрытой сетью. Пример сильно упрощен для формирования общего понимания.

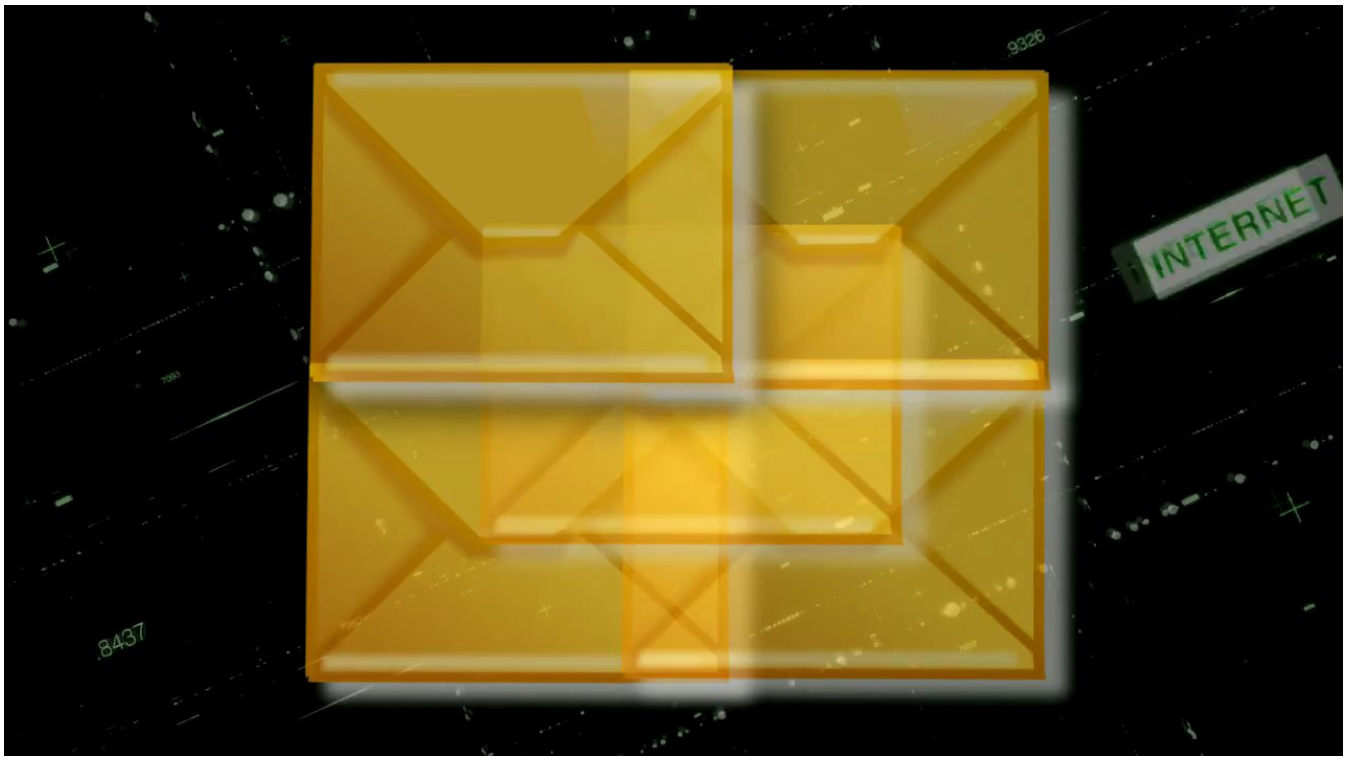


Жители общаются через окна — это обмен информацией по открытым каналам. Обратите внимание на наличие «уполномоченного» соседа (домик №2), который все слышит, и, как мы знаем, записывает. Проведем аналогию шифрования информации с тайным языком: даже если соседи говорят на незнакомом для шпиона языке, как минимум он знает, что определенные дома поддерживают контакт. В случае обычного интернета провайдер хранит информацию о сайтах, которые вы посещаете. Ему не обязательно знать, что именно вы писали на оппозиционном форуме. Один факт посещения подобного ресурса говорит о многом. У двух домов (№4 и №5) имеются связанные подземные туннели — они обмениваются сообщениями без крика на улицу. Шпион не видит, что эти люди общаются.

Важный термин, который понадобится в дальнейшем — это DPI (deep packet inspection). На нашей иллюстрации этим понятием можно охарактеризовать навык шпиона понимать тайное значение услышанных слов. Например, люди будут кричать в окно слово «труба» перед тем, как начать общаться через скрытые туннели. Шпион быстро это поймет и станет рыскать возле домов, откуда донеслось кодовое слово. Это будет называться выявлением при помощи технологии DPI.

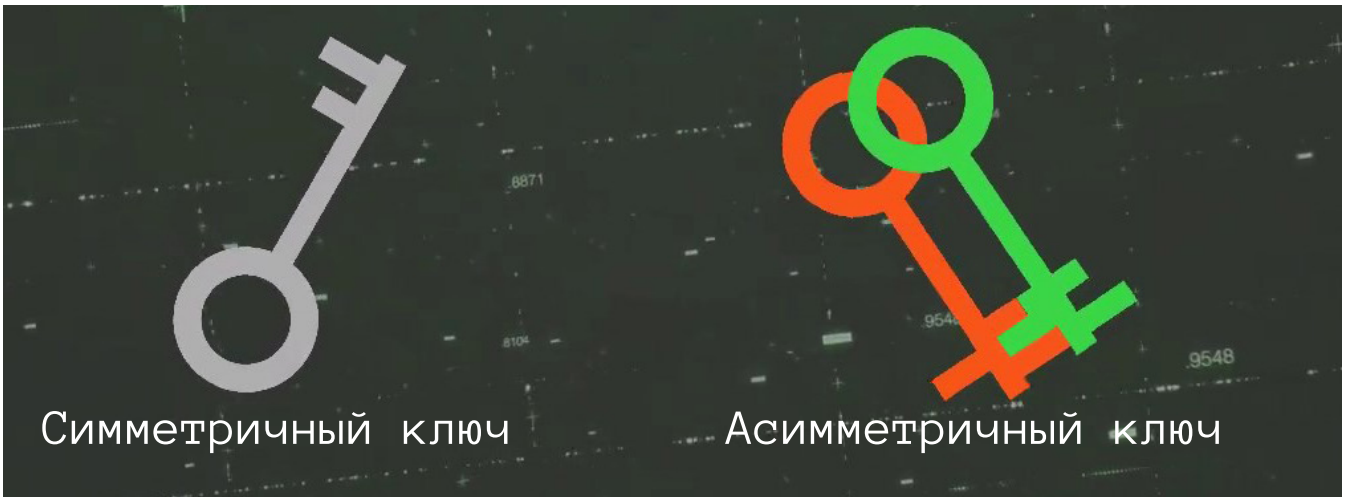
Для представления скрытой одноранговой сети, где все узлы имеют равные права и возможности, сделаем важное замечание: каждый дом может передавать сообщение только ближайшему соседу, прося отправить сообщение дальше. Так получается иллюстрация peer-to-peer сети, в которой каждый узел одновременно выступает получателем, отправителем и транзитным узлом. Очевидны несколько проблем: анонимность отправителя, сохранность и секретность данных на транзитных узлах, а также наличие злонамеренных участников сети, которые могут пытаться анализировать проходящие через них сообщения для определения личности анонимных абонентов.

Сеть I2P изначально была спроектирована с учетом предположения, что все промежуточные узлы являются скомпрометированными или злонамеренными. Туннели I2P являются однонаправленными, т.е. входящий трафик идет по одной цепочке узлов, а исходящий — по другой. Помимо этого все передаваемые сетевые зашифрованные сообщения имеют свойство накладываться друг на друга, сливаясь в информационный шум, что делает бессмысленным попытки прослушать и проанализировать проходящий поток данных.

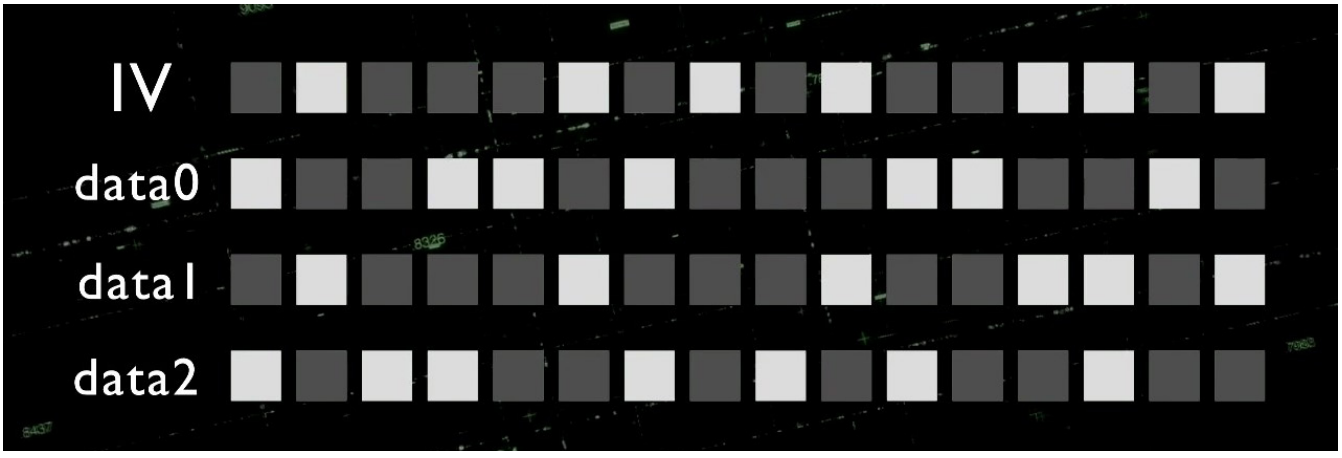


Базовое понимание криптографии

Существуют два основных вида алгоритмов шифрования: симметричный и асимметричный. Эти знания понадобятся, чтобы понимать дальнейший рассказ.



Симметричные алгоритмы более просты и за счет этого работают в разы быстрее: в них для шифрования и расшифровки информации используется один ключ. Слабое место такого подхода — это передача ключа от одного пользователя другому. Если злоумышленник сможет перехватить ключ, он получит доступ к секретной информации. Фактором безопасности в симметричном шифровании (в частности AES — распространенном алгоритме) является вектор инициализации (IV). Вектор инициализации — критически важная составляющая, имитирующая первый блок информации. Так как при AES-шифровании каждый блок из 16 байт влияет на следующий блок, целостность информации критически важна.

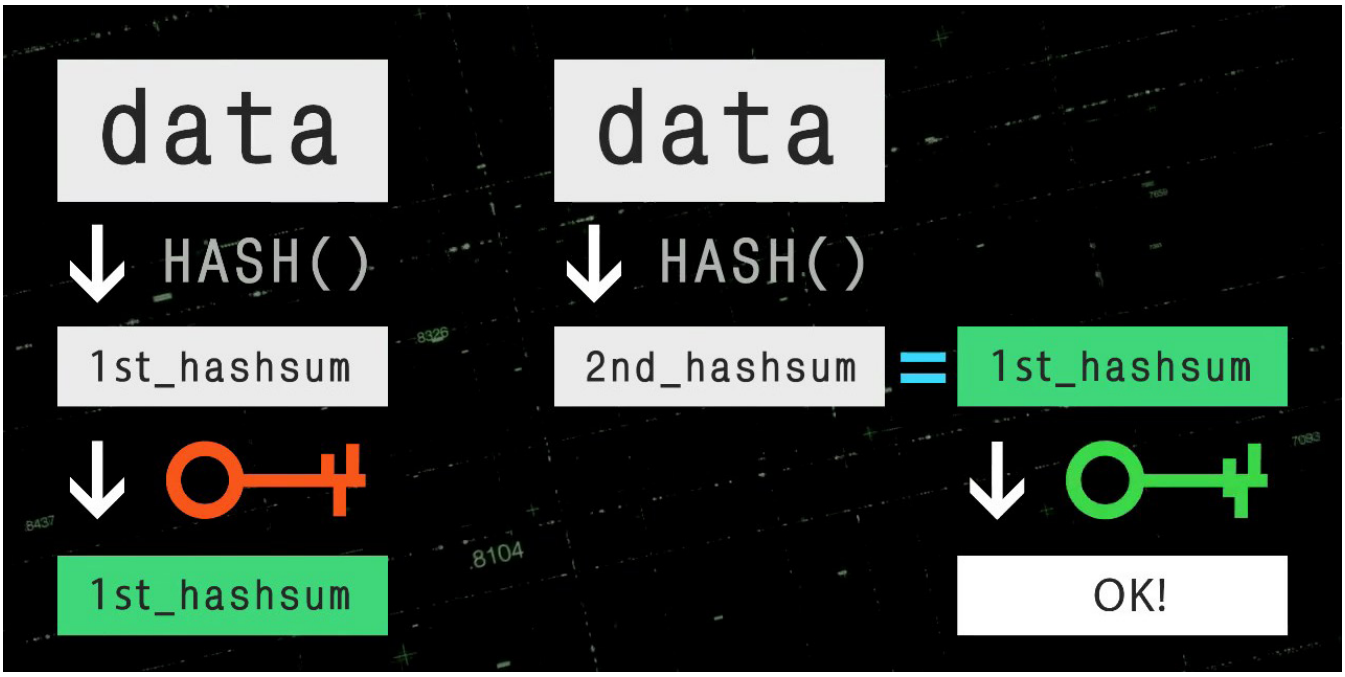


Асимметричное шифрование не подвержено этой угрозе, т.к. у каждого пользователя два ключа: публичный и приватный. Публичный ключ передается открыто и используется для шифрования информации, которая в дальнейшем может быть расшифрована только секретным ключом из соответствующей связки. Асимметричное шифрование используется повсеместно, но сильно проигрывает симметричным алгоритмам по скорости. В случае, когда шифрования много, как в I2P, и нужна максимальная производительность, используются асимметричные алгоритмы согласования ключей. По-простому говоря: пользователи обмениваются публичными ключами, перехват которых ничем не угрожает, и на их основе математическим путем выводят общий симметричный ключ. Это называется согласованием. После согласования пользователи имеют быструю скорость криптографических операций, т.к. используют симметричный алгоритм, где шифрование и расшифровка происходит одним ключом без дополнительных ресурсоемких вычислений, присущих асимметричному шифрованию с открытым и закрытым ключами (также их называют публичный и приватный).

К важным криптографическим операциям относятся хеш-функции. В отличие от шифрования, они не изменяют исходную информацию, и не имеют как таковых ключей.

Пропуская информацию через хеш-функцию, мы получаем уникальную строку. Полученная строка называется хеш-суммой и позволяет проверить целостность информации, т.к. даже самое незначительное изменение информации выдаст абсолютно новую хеш-сумму.

Цифровая подпись является производным совместного использования асимметричных алгоритмов шифрования и хеш-функции: сначала выводится хеш-сумма, а затем секретным ключом в нее закладывается идентификатор ключа подписавшего пользователя. Имея открытый ключ подписавшего, мы самостоятельно проверяем хеш-сумму информации и сравниваем ее с той, которую изначально заложил отправитель. Так проверяется достоверность полученной информации.



Общие принципы работы I2P

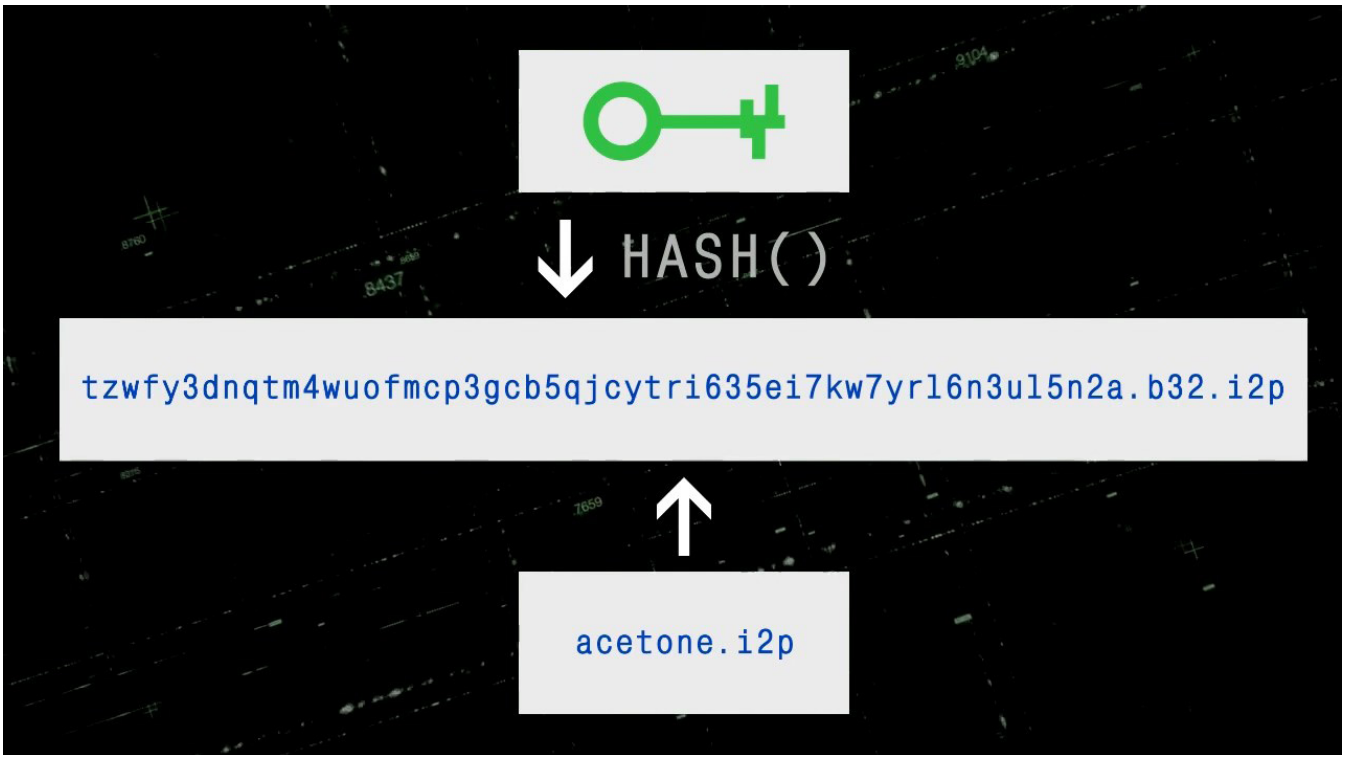
В сети I2P все пакеты зашифровываются на стороне отправителя и расшифровываются только на стороне получателя, при этом никто из промежуточных участников обмена не имеет возможности перехватить исходные данные. По этой причине нет необходимости беспокоиться о том, чтобы прикладные программы обеспечивали шифрование своего трафика. Например, сайту в I2P не нужно использовать TLS-шифрование, чтобы вводимые данные пользователя не были перехвачены, как это делается в обычном интернете, т.е. нет нужды использовать протокол HTTPS.

Также, никто из участников сети не знает, кто на самом деле отправитель и кто получатель, так как статус узла, от которого пришел пакет неизвестен, он может быть отправителем или промежуточным, а следующий, которому нужно этот пакет отправить, может являться получателем или таким же промежуточным узлом. Узнать конечные точки отправителя и получателя промежуточный узел никак не может, так же как не может узнать, что произошло с только что переданным следующему узлу пакетом: обработал ли тот его, или передал куда-то дальше.

В I2P используется концепция роутеров и конечных точек.

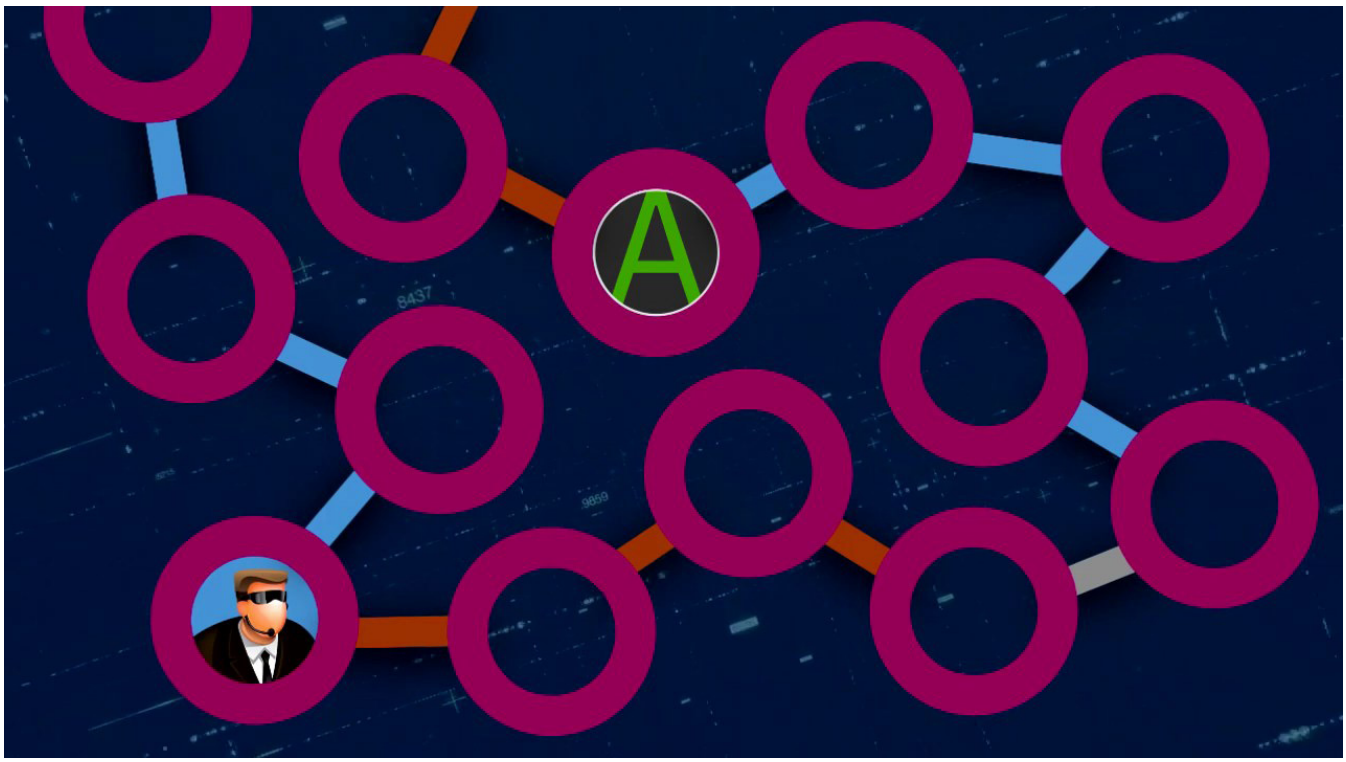
Роутер — это обезличенный участник сети: ни клиент, ни сервер — просто ничем не отличающийся от других транзитный узел. Устанавливая прямое соединение между собой, роутеры видят реальные IP-адреса друг друга, но эта информация не говорит о чем-либо, кроме вероятного использования сети I2P абонентом на другом конце. Если злоумышленник решит выявить всех участников сети, он наткнется на различные погрешности вроде прокси-серверов, и в любом случае не будет иметь фактических данных о какой-либо внутрисетевой активности выявленных пользователей. Устанавливая клиент сети I2P на свое устройство, вы фактически устанавливаете I2P-роутер — обезличенное звено сети.

Конечная точка наоборот является осмысленной сущностью — либо сервером, либо клиентом, но ее реальное местоположение неизвестно. В качестве адресов конечных точек сети I2P используются идентификаторы, выведенные из открытого ключа подписи: хеш-сумма дает уникальную строку фиксированной длины, в конце которой для удобочитаемости подставляется псевдодоменная зона «.b32.i2p» — так получается привычный внутрисетевой адрес. Для использования человекочитаемых доменов в зоне «.i2p», например community.i2p, существуют бесплатные регистраторы, достаточно незамысловатые в использовании. Привязка домена происходит к полному адресу.



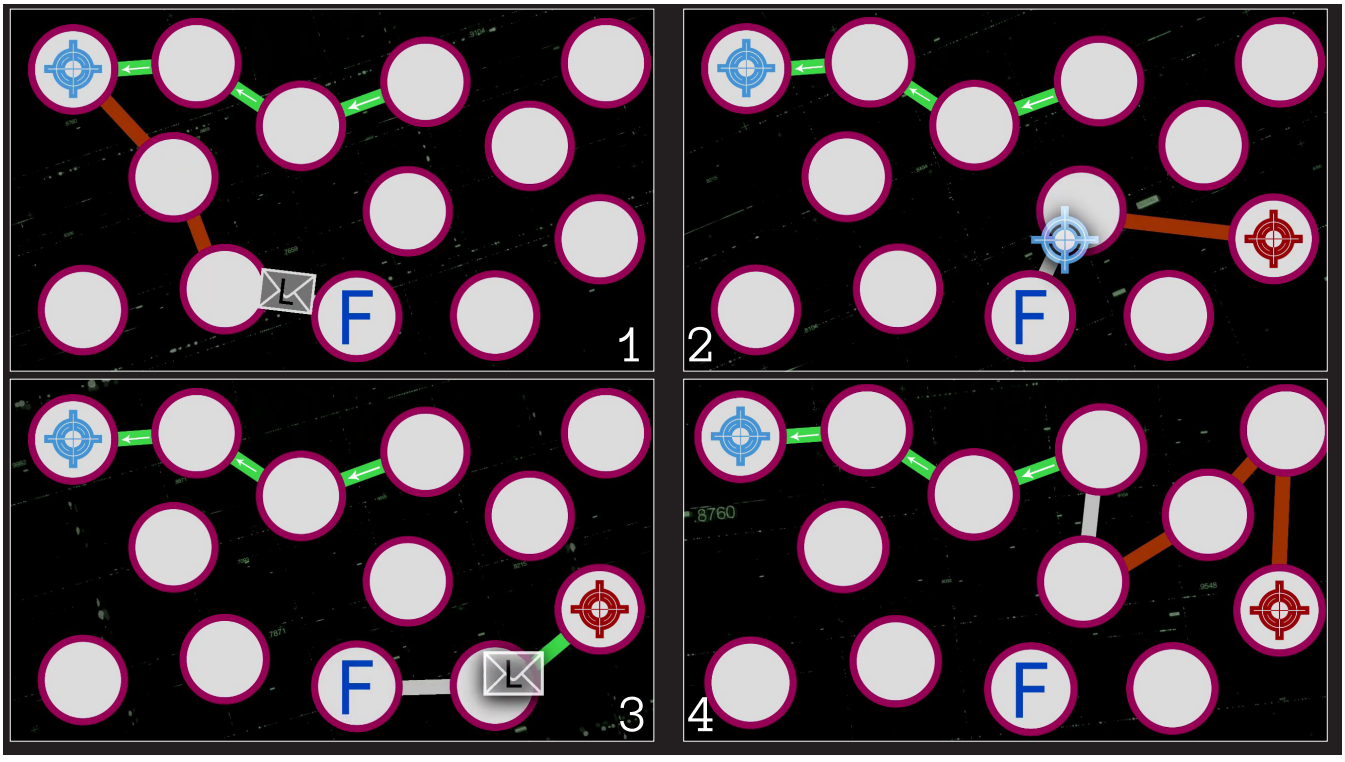
Конечная точка создается администратором роутера, при этом роутер продолжает свою «серую» работу, никому не сообщая об адресах, размещающихся на нем. Роутер, получая информацию, которая предназначается для локального ресурса, не передает ее дальше, но никто из участников сети не может это узнать: роутер-получатель ведет себя также, как обычный транзитный узел.

Транзитные узлы являются частью цепочки серверов, образующих туннель. Туннель можно представить в виде трубы, проходящей через несколько комнат: вода поступает с одного конца и выходит с другого, при этом наблюдатели в промежуточных комнатах видят трубу, могут слышать шум потока внутри, но не знают, что именно по ней течет. В I2P все роутеры по умолчанию могут принимать участие в построении чужих туннелей, но никто не знает что, куда и кому по этим туннелям передается. Создавая точку назначения, пользователь сам определяет длину входящих и исходящих туннелей, а также их количество. Эта информация остается известной только для создателя туннеля: промежуточные звенья ничего не знают о других транзитных узлах и о их общем количестве.



На иллюстрации пустые кружочки — это роутеры. Только на нашей схеме эти кружки находятся рядом, фактически же это компьютеры по всему земному шару. Исходящие туннели пользователя обозначаются коричневым цветом, а входные — синим. Владелец выходного туннеля ничего не знает о входящем туннеле точки назначения, кроме ее конца, через который он посылает информацию запрашиваемому ресурсу.

По умолчанию длина туннелей составляет 3 узла, но в зависимости от потребностей пользователя может составлять 1 транзитный узел или целых 8, и даже 0, тогда будет происходить прямое подключение к туннелю второй стороны. Как видим, администратор конечного ресурса имеет входной туннель в 4 узла. Получив запрос, веб-ресурс отсылает ответ во входной туннель пользователя через свой выходной туннель, т.е. по абсолютно другому пути. Смотря на схему, можно представить насколько сложно пользователю определить реальное местоположение собеседника. Учитывая, что каждые 10 минут происходит смена существующего туннеля на новый с новыми цифровыми подписями и ключами шифрования, деанонимизация кого-либо вовсе кажется фантастикой.



Важная сущность сети I2P — это флудфилы: специальные роутеры, собирающие информацию о сети и обменивающиеся ей друг с другом. Флудфилом может быть любой желающий, задав соответствующий параметр в конфигурационном файле роутера. Все серверные конечные точки сети (т.е. точки, к которым ожидается подключение) автоматически публикуют на флудфилах информацию о себе, которая в совокупности называется лизсетом.

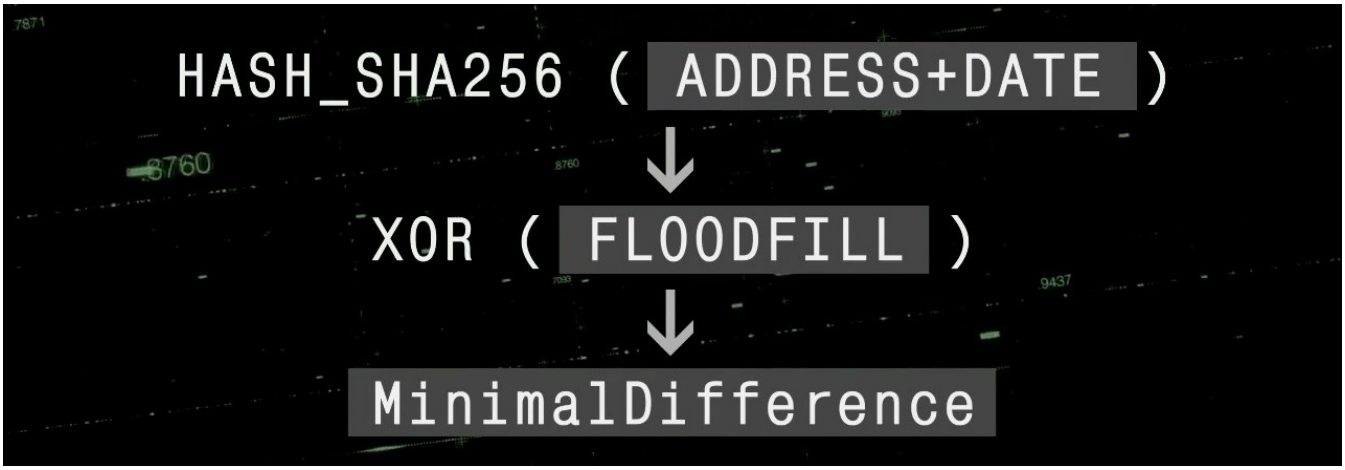
Лизсет включат в себя полный адрес конечной точки, ключи шифрования и список входящих туннелей. Обращаясь к какому-либо адресу в I2P, вы автоматически запрашиваете у случайного флудфила лизсет этого адреса. Если флудфил не знает запрошенный адрес, он сообщает адреса других флудфилов и поиск продолжается. Флудфил является роутером (а не конечной точкой) и принимает подключения напрямую, т.е. не имеет своих входных туннелей. Однако конечные точки обращаются к нему исключительно через анонимные цепочки, тем самым скрывая местоположение публикуемых ресурсов и тех, кто хочет к ним обратиться.

Так как каждая конечная точка имеет в среднем по три входных туннеля, которые меняются каждые 10 минут, обращение к флудфилам происходит лавинообразное, хоть и с небольшим потоком данных. Благодаря этому в сети всегда происходит хаотичное движение служебной информации, образующее «белый шум».

Помимо поиска лизсетов, белый шум генерируется зондированием сети: каждый роутер с небольшой периодичностью опрашивает случайный флудфил, получая от него в ответ три новых роутера (таким образом увеличивая собственный рисунок сети и находя новые флудфилы). Главным источником сетевого шума на роутере является транзитный трафик: он создает большую сетевую активность вне зависимости от конечных точек, расположенных на роутере. Также, будучи транзитным звеном, роутер подмешивает к трафику чужих туннелей полезную нагрузку — трафик своих скрытых сервисов. Чем больше транзитного трафика на роутере, тем абсолютнее секретность его конечных точек: какие-либо действия на скрытом ресурсе, даже DDoS-атаку, нельзя сопоставить с сетевой активностью конкретно взятого сервера. Активный узел сети имеет в своей базе в среднем 5000 активных роутеров и принимает сотни, а то и тысячи абсолютно случайных транзитных подключений. Как говорится: попробуй проанализируй.

Отдельно отметим механизм выбора флудфила пользователем. Это важный вопрос, которому при разработке было уделено

надлежащее внимание. Так как любой из флудфилов может содержать злонамеренный пользователь, стояла задача сделать его попытки анализа и прочих недобросовестных действий не применимыми к конкретному человеку. Флудфил для запроса выбирается следующим образом: берется целевой адрес и сегодняшняя дата. Из этой информации выводится хеш SHA256 — получаются данные той же длины, что и обычный адрес. Затем осуществляется поиск флудфила в локальной базе роутера: используется тот, который при операции «ИСКЛЮЧАЮЩЕЕ ИЛИ» с блоком «целевой адрес + дата» даст наименьшее число. Подобный подход делает выбор узла для служебных запросов непредсказуемым и ежедневно изменяющимся для каждого адреса. В рамках этого материала упоминается «случайный флудфил», однако знайте, что кроется за этими словами.



Недавно мы упомянули DPI — исследование сетевых пакетов с целью выявления типа передаваемой информации. Упомянули не зря, потому что вы должны знать об этой технологии, а также о том, что весь трафик I2P устойчив перед анализом. Трафик сети зашифрован с самого низа, начиная с транспортных протоколов. В I2P используются: **NTCP2**, как крипто-аналог TCP, и **SSU**, как крипто-аналог UDP. I2P-роутер принимает сетевой трафик прикладных программ, обращающихся в скрытую сеть, и оборачивает привычные протоколы в их зашифрованные аналоги. После обработки в сетевых пакетах невозможно выявить что-либо вразумительное, потому что шифруется все, в том числе заголовки и размер. Размер пакетов скрывается

паддингом, т.е. заполнением пакетов случайными данными до определенного размера. Информация о настоящем размере сетевого пакета передается в нем же в зашифрованном виде. При расшифровке на конечном устройстве примешанный «мусор» просто отбрасывается. Таким образом «белый шум» сети, т.е. практически бессмысленная для человека информация смешивается с пользовательской информацией и со стороны абсолютно от нее неотличима.

Описанная организация сети избавляет от потребности использования IP-адресов во внутрисетевой маршрутизации и позволяет всем ресурсам оставаться анонимными, так как никто не знает что располагается на том или ином роутере: просто ли выходной прокси обычного пользователя или веб-ресурс мирового масштаба.

Дополняя и закрепляя все сказанное, подробно опишем процесс от первого запуска роутера до открытия веб-страницы. При первом запуске роутер не имеет каких-либо данных о сети, поэтому происходит обращение к случайному ресид-серверу, который отдает пользователю свою базу известных роутеров. Ресиды держат энтузиасты, их список имеется в свободном доступе. По сути дела, информация от ресида представляет собой zip-архив папки netDb, в которой каждый роутер хранит информацию о сети. Так как передача данных происходит через обычный интернет, во избежание подмены по пути, архивы подписываются цифровым ключом. Публичные ключи всех актуальных ресидов содержатся в самом I2P-роутере. Если по каким-то причинам мы не хотим обращаться к публичным ресид-серверам, можно использовать уже имеющиеся базы сети, например, с других наших роутеров. Обращение к ресиду не происходит, если в папке netDb имеется не менее 25 роутеров. После подключения к нескольким актуальным участникам начинается автоматическое непрерывное расширение рисунка сети.

Построение туннелей

Для создания туннелей, обеспечивающих исчерпывающий уровень приватности, в I2P используется так называемое «чесночное шифрование». «Чеснок» I2P представляет блок информации, включающий несколько «чесночин» по 528 байт. «Чесночины» закладываются в случайном порядке, поэтому их последовательность не несет какой-либо информации. Каждый получатель «чеснока» опознает свой «зубчик» по первым 16 байтам, которые являются частью хеша его адреса. После того, как нужный «зубчик чеснока» найден, содержащиеся в нем инструкции расшифровываются ключом роутера.

Согласно инструкций, весь чеснок может быть передан следующему узлу, где порядок действий повторится. При построении исходящего туннеля, в инструкции последнего получателя предписывается сообщить создателю туннеля о завершении построения цепочки. Так как все туннели однонаправленны, ему сообщается входной туннель. Первые исходящие туннели создаются с возвратом ответа в туннель нулевой длины, т.е. напрямую создателю, однако это редкое явление и факт нулевой длины туннеля известен исключительно его создателю, поэтому компрометации пользователя не происходит.

При создании входящего туннеля картина еще изящнее: через исходящий туннель случайному роутеру передается чеснок, последний получатель которого — сам создатель. С позиции последнего транзитного звена передача «чеснока» создателю туннеля воспринимается как передача следующему транзитному узлу. Именно поэтому транзитные узлы не могут знать длину туннелей и их владельцев.

При построении туннеля длиной не более 4 участников, i2pd формирует «чеснок» из 4 зубчиков, в остальных случаях формируется «чеснок» из 8 частей. Пустые зубчики чеснока заполняются случайной информацией и неотличимы от настоящих.

Протокол I2P не предусматривает туннели длиннее 8 участников, однако на практике 4 транзитных узла считаются исчерпывающим решением.

В «чесноке» содержится информация для каждого участника:

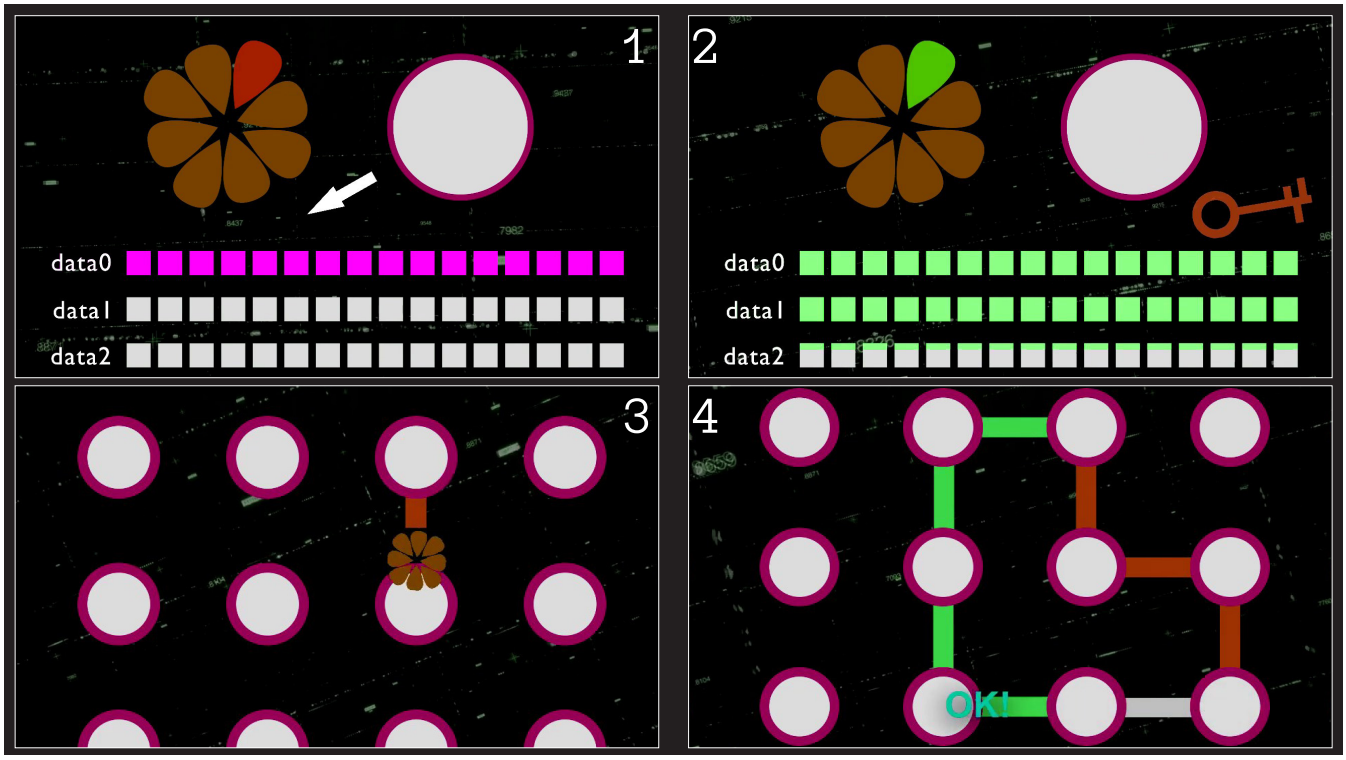
1. Номер туннеля на его роутере (случайные 4 байта).

2. Адрес следующего роутера и номер туннеля на нем (почти как IP-адрес и номер порта в обычной сети).

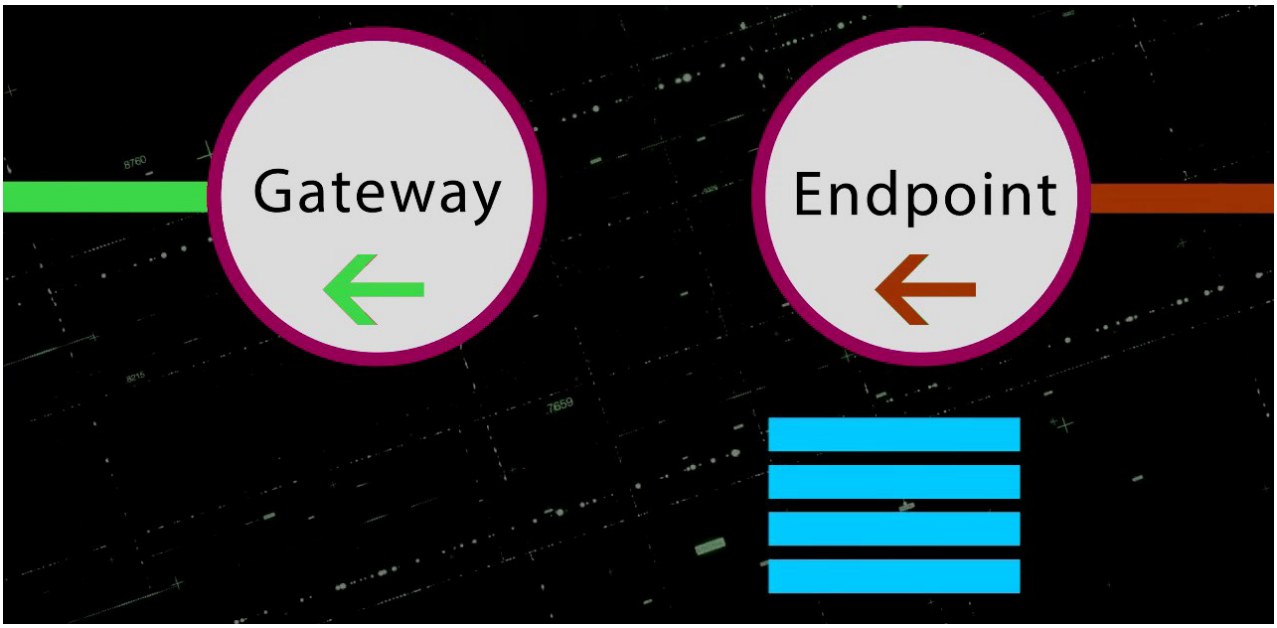
3. Ключ симметричного шифрования и ключ шифрования вектора инициализации (IV). Сам вектор инициализации содержится в начале сообщения.

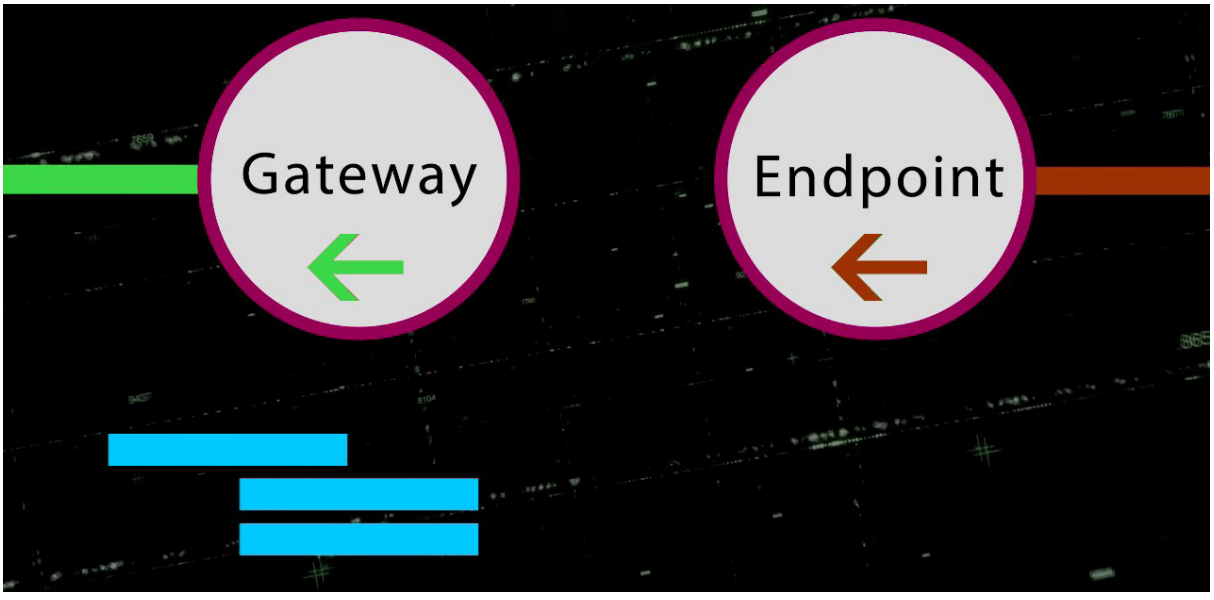
4. Роль узла в цепочке: конечный или промежуточный.

5. Ключ симметричного шифрования и вектор инициализации для ответа. Ими шифруется статус транзитного роутера: готов ли он принять новый туннель. Если в конце окажется, что хотя бы один роутер выразил несогласие, туннель не будет создан и весь процесс создания начнется с нуля.



В зависимости от типа туннеля (входящий или исходящий), последнее звено принимает роль либо «Endpoint», либо «Gateway». Между участниками одного туннеля зашифрованная информация передается блоками по 1 килобайту — это нынешний стандарт протокола. Задача последнего исходящего узла — собрать информацию в более весомый пакет и переслать нужному пользователю на его входящий туннель. Работа роутера входящего туннеля обратная: он разделяет полученную информацию на блоки и отправляет ее на другой конец туннеля. Каждый туннель существует 10 минут. Во избежание разрыва соединения, стороны заранее создают новые туннели и обмениваются их лизсетами.



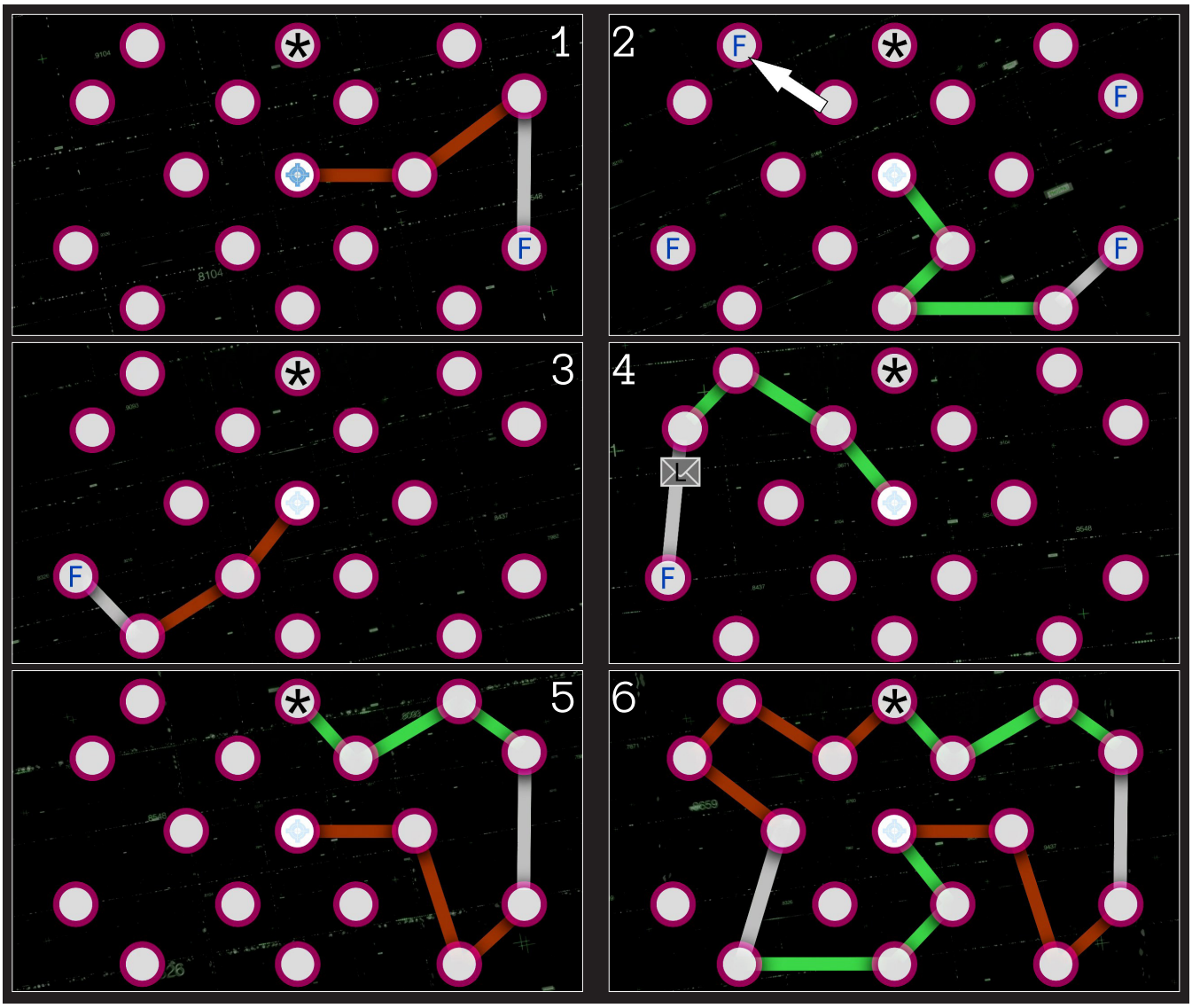


Модель взаимодействия

По умолчанию, i2pd предоставляет для прикладных программ SOCKS и HTTP прокси. Прокси — это посредник. В нашем случае — посредник для выхода в скрытую сеть. По умолчанию SOCKS5 доступен по адресу 127.0.0.1:4447, HTTP-прокси — на порту 4444. Чтобы открыть веб-страницу, размещенную в сети I2P, в браузере необходимо установить соответствующие настройки. В Firefox прокси удобно настраивается через конфигурацию сети, либо через плагин FoxyProxy. Когда мы ввели адрес сайта в настроенном браузере, I2P-роутер получил наш запрос и начал работу.

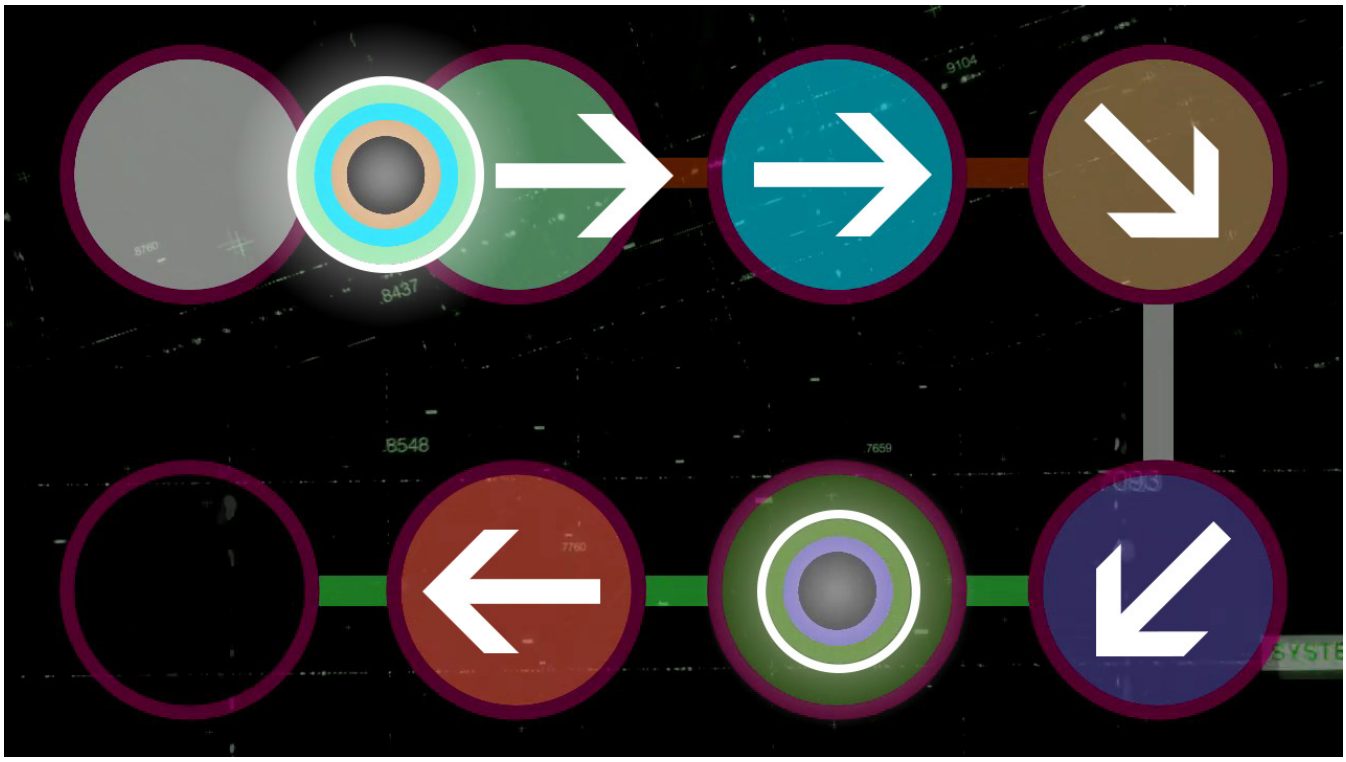
Сначала происходит обращение к случайному флудфилу с запросом лизсета нужной точки назначения. Выходной прокси пользователя является конечной точкой, т.е. скрытой сущностью. Чтобы не раскрыть ее местоположение, обращение к флудфилу происходит через цепочку транзитных узлов. Также мы сообщаем ему данные нашего входного туннеля для ответа. Допустим, что флудфил не знает нужного адреса, поэтому вместо лизсета он возвращает три случайных флудфила из своей базы. Из полученных новых флудфилов случайным образом выбирается один, к которому роутер повторяет прежний запрос. В нашем примере второй флудфил дает нужный лизсет: всю необходимую информацию для связи с конечной точкой, которая включает в себя сведения о входных туннелях и ключах. Входных туннелей, как правило, не меньше трех. Выбрав случайный входной туннель получателя, роутер отправляет через него запрос на сервер. Если все прошло успешно и запрос дошел, сервер отсылает нам веб-страничку. В обычной сети это происходит по тому же пути, откуда пришел запрос, но в I2P дела обстоят совсем иначе. Свой лизсет для ответа конечная точка локального прокси отправляет вместе с нашим запросом. Достучавшись до сервера, мы сообщили ему адрес для ответа: чтобы веб-страница открылась в браузере, серверу необходимо через свой *исходящий* туннель отправить

ответ в наш *входящий*. Приняв ответ через входящий туннель, наш роутер обрабатывает информацию и в нашем браузере открывается заветная страничка!



Пользовательская информация, проходя через туннели, подвергается «луковичному», т.е. многослойному шифрованию. Перед отправкой сообщения в исходящий туннель, роутер последовательно оборачивает ее в несколько слоев шифрования, используя ключи, которые были выданы транзитным узлам туннеля. Каждый узел исходящего туннеля, получая информацию, выданным изначально ключом снимает один слой шифрования и передает информацию дальше. Последний узел исходящей цепочки, Endpoint, снимая последний слой луковичного шифрования, пересылает информацию входящему тунне-

лю. Во входящем туннеле луковичное шифрование происходит зеркально: Gateway шифрует информацию своим ключом, который ему выдал создатель туннеля, и передает ее дальше. Каждый последующий узел входного туннеля добавляет свой слой шифрования. Создатель входящего туннеля, при получении информации, снимает все слои луковичного шифрования ключами, которые были выданы транзитным узлам. Пользовательская информация подвергается сквозному шифрованию при помощи публичного ключа точки назначения, поэтому ни один транзитный узел не располагает какой-либо чувствительной информацией, а конечный получатель, сняв все луковичное шифрование, подвергает информацию расшифровке своим асимметричным ключом и проводит проверку подписи.



Если кто-то попробует вклиниться в туннель и подсунуть свою информацию, выдавая себя за настоящего отправителя, подпись окажется неверной и информация будет отвергнута. После установки сессии, для оптимизации производительности конечные точки используют набор одноразовых симметричных ключей и тэгов к ним, применяя соответствующие ключи и обмениваясь между собой только тэгами.

Практическое использование

Область применения I2P обширна и не ограничивается только веб-сайтами. Вы можете размещать в I2P любые веб-ресурсы: форумы, доски объявлений, гит-репозитории, блоги и персональные одностраничники. Вполне реально организовать даже сетевые пошаговые игры, не требующие низкой задержки при передаче пакетов, например, шахматы или карты. Также найти применение I2P можно в построении выходных туннелей из неопределенного места в глобальную сеть, для полноты картины размещая выходные прокси на анонимно оплаченных серверах. Организуя ssh-доступ к арендуемой машине через I2P, вам всего лишь единожды потребуется зайти на сервер посредством других сетей, чтобы установить i2pd и сконфигурировать туннель, который будет принимать подключения. Единственное ограничение фантазий — скорость передачи данных внутри сети. Надо заметить, что от года к году скорость становится все ближе к привычным показателям обычного интернета благодаря оптимизации программного кода и общему росту количества узлов.

Разработка протокола

Все это интересно, но напрашивается закономерный вопрос: кто за этим стоит? Судя по сложности протокола, можно подумать, что это как и в случае с сетью Tor что-то связанное со спецслужбами и государственным финансированием, но нет. I2P — полностью свободный проект, рожденный и поддерживаемый энтузиастами по сей день. Разработка I2P на языке программирования Java была начата неким JRANDOM после намеренной мистификации теракта 11 сентября, направленной на ограничение прав и свобод. Первый релиз состоялся в 2003 году. Несколько лет спустя JRANDOM отошел от разработки, не оставив о себе вестей, а его место занял пользователь с никнеймом zzz — американец, предположительно из района Нью-Йорка, который на момент выхода брошюры все еще курирует разработку. Так прошло 10 лет:

сообщество росло, I2P-роутер писали как могли, также в это время была написана официальная документация. При поисковом запросе «I2P» первая ссылка, как правило, ведет нас на сайт geti2p.net — официальную страницу того самого I2P-роутера на Java. Но история на этом не заканчивается.

В 2013 году русскоговорящий пользователь orignal с французским никнеймом, судя по всему большой любитель пиратской литературы, обнаружил в онлайн-библиотеке «Флибуста» сообщение о недоступности скачивания книг через клирнет, т.е. обычный интернет. Вместо этого пользователям предлагалось использовать I2P. Однако Ориньяль, как он говорит, не нашел существующей нормальной реализации протокола, а только некое поделие на джаве. Несмотря на то, что разработка «этого поделия» уже велась десять лет, orignal, как искушенный программист, ее не оценил. На тот момент в сети лежало еще с десяток попыток написать i2p-роутер на языках C и C++, в которых не было сделано совсем ничего. Что-то было в продукте под названием «i2pcpp», но до чего-то годного тоже было далеко. И что делает orignal: за три месяца он написал на C++ рабочую программу, пригодную для скачивания книг с Флибусты, и на этом уже хотел остановиться. На дворе был июль 2013 года. Однако некий человек уговорил его изложить свой опыт в статье на Хабре. Дело в том, что orignal пришлось вникать в дебри джава-кода, т.к. официальная документация оказалась не полноценной и имела разрозненную структуру. После публикации первой статьи что-то случилось: наверное, заметив интерес общественности, и почувствовав некоторый азарт, присущий всем программистам, писателям и композиторам, orignal взялся за написание полноценного клиента сети. Дебютный релиз 0.1.0 состоялся 17 октября 2014 года. Для его реализации orignal пришлось собственноручно реализовать на C++ несколько криптографических функций (которые с расширением библиотеки OpenSSL в дальнейшем были заменены на библиотечные). Новый клиент получил название i2pd — Invisible Internet Protocol Daemon. После первого релиза к orignal присо-

единились энтузиасты, которые также начали работу над совершенствованием нового I2P-роутера. Сообщество разработчиков получило имя PurpleI2P.

В первые годы к инициативе i2pd было проявлено много скептицизма сторонних наблюдателей: в PurpleI2P искали руку Кремля, закладки от ФСБ, следы масонов и рептилоидов. Само название Purple в апогее наркотического прихода отдельные индивидуумы сводили к флагу Российской Федерации: смешение красного, синего и белого цветов дает тот самый фиолетовый. Однако orignal, ведущий разработчик i2pd, пояснил, что название PurpleI2P не содержит великой тайны: название было придумано после кружки английского эля с оглядкой на королевскую корону Британской империи. Наверное, это можно считать отсылкой к лидерству i2pd перед изначальным клиентом на Java.

Ориньяль по сей день продолжает публиковать русскоязычные статьи про I2P, рассказывая широким массам о развитии протокола. В команде программистов-энтузиастов появляются новые никнеймы, другие наоборот уходят в перманентный оффлайн. В основном планирование разработки происходит в IRC, в сети ILITA. Русскоязычное сообщество достаточно активно и в IRC на самом деле обсуждается все: от новых фич в I2Pd до климата в Африке и новых мемов.

Каждая версия роутера сменяется следующей, сообщество живет, разработчики постоянно что-то кодят. Чтобы не возникало вопросов что же меняется, если и так все работает хорошо, вкратце скажем о развитии протокола: мало ли параноики решат, что протокол не меняется, а оттачивается только бэкдор от силовых структур. В начале пути, ветки джава-роутера и роутера на C++ развивались не синхронно, согласование нюансов с соседним лагерем происходило с некоторой задержкой. Сейчас разработчики, не смотря на расхождения во взглядах, вместе реализуют значимые решения. Самое заметное изменение за последние годы: переход с транспортного протокола NTCP

на NTCP2. Этот протокол является адаптированным криптоаналогом привычного TCP и используется повсеместно: начиная открытием веб-страницы и заканчивая загрузкой торрента. Главное различие двух протоколов заключается в использовании алгоритма шифрования: в старом NTCP используется Диффи-Хеллман, очень медленный и жадный до системных ресурсов, а в NTCP2 — алгоритм на основе эллиптических кривых — тренд в сфере скоростных криптосистем, превосходящий по криптостойкости и производительности своего предшественника. Это изменение сделало I2P-роутеры более легковесными и полностью пригодными для использования на малых устройствах вроде одноплатных компьютеров и смартфонов (по крайней мере i2pd точно). Замена алгоритма согласования ключей Диффи-Хеллмана новыми решениями происходит и в других частях протокола. Для зрителей, понимающих в криптографии, особо заметим, что сеть I2P постепенно переходит на криптографические протоколы семейства Noise, что тоже является хорошим решением для производительности, анонимности и надежности. I2P-роутеры, как и все программы на свете, подвержены недочетам вроде багов и неполноценной реализации каких-то функций, что влечет дальнейшее развитие, исправление, и новые релизы. Если интересны детали, более подробную информацию можно найти через поисковик и в чатах сообщества.

Почему стоит использовать i2pd на C++ вместо роутера на Джаве? Ответ прост: потому что i2pd архитектурно превосходит поделие на Джаве. I2P — кладезь различной криптографии, реализация которой на Джаве была бы жутко тормозной. Потому в Джава-роутере используются библиотеки, написанные на языке C. И все бы ничего, но каждый вызов этих библиотек — это большие накладные расходы. И как бы разработчики не старались, а тормоза есть. Не такие большие, как могли бы быть, но до скорости нативной (т.е. встроенной) криптографии из i2pd им очень далеко. Также весь трафик джава-роутера локально проходит через фактически лишний протокол I2CP, что, не вникая в детали, сильно портит качество работы, в то время как в i2pd

этот протокол существует исключительно для совместимости с приложениями на джава-роутерах. Для пытливых умов можно сказать более подробно: главная проблема I2CP в том, что через него сессии TCP (которые называются стримами) не могут контролировать доступ к туннелям и лизсетам. Также I2CP лишает пользователей джава-роутера нормальной поддержки UDP-туннелей: их поддержка обеспечивается отдельным приложением streamr. Как уже знает бывалый зритель, протокол UDP используется для быстрой потоковой передачи данных, например, голос собеседника при онлайн-звонке. Сложно переоценить качественную поддержку этого протокола. В I2Pd UDP-туннели поддерживаются без каких-либо проблем и дополнительных программных наслоений.

Помимо более быстрой работы i2pd, этот роутер также потребляет меньше системных ресурсов в сравнении с джаво-версией, т.к. программа на C++ напрямую взаимодействует с системой, в отличие от Джавы, где все крутится внутри виртуальной машины — дополнительной прослойке между операционной системой и программным обеспечением. Дальше только больше: исследуя просторы внутрисетевых площадок, вы не однократно наткнетесь на описание различнейшних проблем с производительностью сети I2P, где источником проблем будет не протокол, а его кривая реализация в популярном клиенте. В рамках тестов со скоростными серверами и сетью, полностью исключающей наличие джава-роутеров, удалось достичь скорости чуть меньше мегабита в секунду. Сегодня такой показатель на фоне средних нескольких десятков килобит кажется невероятным! В видео, по тексту которого составлена эта брошюра, приведен скриншот недавней переписки двух ведущих разработчиков: orignal и zzz. В то время, как через i2pd в тестовом режиме гоняют видеостримы, джава-роутер не позволяет слушать даже онлайн-радио. Чтобы увеличить среднюю скорость сети I2P нужно: во-первых, каждому держать свой узел включенным максимально доступное время, во-вторых, — это должен быть узел сети, работающий на i2pd.

Сравнение с другими сетями

Если вы имеете представление о сети Tor и о меш-сетях вроде Yggdrasil Network, после всего сказанного об I2P, не имеет большого смысла говорить, чем они различаются. Это абсолютно разные концепции. Однако для начинающих читателей приведем несколько основных пунктов.

Начнем с Tor: Во-первых, он сильно привязан к корневым серверам конторы, к которым осуществляется стартовое обращение для построения рисунка сети. В I2P это явно указанные в коде ресиды, которые можно сменить на свои, либо просто использовать уже готовые адресные книги, отталкиваясь от которых роутер продолжит работу, вовсе минуя стартовые узлы. Во-вторых, по мнению компетентных пользователей, Tor изначально и до сих пор принадлежит АНБ, поэтому относительно него не стоит строить иллюзий. В-третьих, Tor имеет более простую архитектуру туннелей и меньше возможностей их конфигурации. В-четвертых, Tor изначально предполагает проксирование в обычный интернет, а в I2P это возможно только при дополнительной конфигурации соответствующих узлов и туннелей к ним. Этот пункт по настроению можно отнести к минусам I2P, но мы не согласимся.

Про Yggdrasil и аналогичные меш-сети: подобные разработки ориентируются на легкость развертки, скорость передачи данных и самоорганизацию. Трафик внутри Yggdrasil искусственно не путается, сеть имеет простой и интуитивно понятный рисунок. Пользователи имеют устойчивые двунаправленные зашифрованные туннели, и чем короче — тем лучше. Как видно, у I2P с Тором совсем мало общего, а с распространенными меш-сетями и того меньше. Разве что I2P может свободно работать через Yggdrasil (поддержка в i2pd с версии 2.36.0). Через Tor тоже, кстати говоря. На этом, пожалуй, все.

Послесловие

Последнее по теме I2P и скрытых сетей в целом: этично ли это? Устоявшееся общественное мнение таково, что любые попытки анонимизации расцениваются, как преступный умысел. Например, в скрытых сетях люди могут размещать то, что будет заблокировано в цензурируемой сети. Это конечно бесит цензоров. Не умолчим и про криминал, который может найти себе пристанище в I2P, все это очевидно. Но что об этом думают разработчики I2P, у которых также могут быть семьи, дети и тяга к справедливости, и как считает большинство криптоанархистов? Если кратко: кухонным ножом можно нарезать хлеб, а можно причинить вред, но из-за возможного злоупотребления кухонные ножи не запрещают. Так же и тут: нельзя бороться с технологиями приватности под предлогом борьбы с преступностью, так как самое распространенное преступление сегодня — это нарушение гражданских прав и свобод: тотальный контроль и продажа личных данных... Подробнее об этом можем порассуждать как-нибудь в следующий раз.

Если вас до сих пор увлекал «нетсталкинг», т.е. романтическое исследование содержимого скрытых сетей, призываем идти дальше — изучать механику работы этих сетей и содействовать их развитию. «Даркнет» (или «дипвеб») по своей сути это не что-то экзотическое; это современный набор необходимых инструментов для свободного распространения информации и личного общения людей без искусственных преград.

Издание подготовлено Свободным Объединением Криптоанархистов.
Настоящая брошюра содержит текст и скриншоты из видео канала «acetonevideo».
Текст и иллюстрации адаптированы для печати неназванными энтузиастами, одним из которых можете быть вы.

COMMUNITY i2p
СВОБОДНОЕ ОБЪЕДИНЕНИЕ КРИПТОАНАРХИСТОВ