Теория строения скрытых систем

Коваленко Геннадий Александрович

1. Введение

Начиная с классической криптографии, с её зарождения и до перерождения в современную, всегда существовала проблема передачи ключа шифрования между субъектами. Задача частично решилась во второй половине XX века, с появлением асимметричного раздела криптографии, когда атака методом подслушивания стала невозможной [1]. Тем не менее первоначальная проблема всё же остаётся и эксплуатируется по сей день, хоть и в завуалированной форме.

Препятствие, которое стало на решении проблемы, заключается в элементе доверия, в его присутствии и в его же отсутствии. Если не предпринимать никаких мер по установке доверенных связей, то любые соединения априори будут подвержены MITM (man in the middle) атакам. С другой стороны, если устанавливать доверенные соединения, то возможность MITM атак не исчезает, а сокращается. И теперь, вместе с первоначальными субъектами появляются промежуточные, интерстициальные субъекты (серверы, узлы хранения и передачи информации), выбираемые нами, и являющиеся законно установленными атакующими. Лишь ограничивая круг нападения, они не перестают потенциально атаковать. Данным узлам известна вся передаваемая нами и нам информация: увлечения, интересы, хобби, развлечения, сообщения. Далее эта информация (объект), как чаще всего происходит, начинает конвертироваться в рекламу, на основе которой увеличивается капитал всё тех же промежуточных субъектов. В итоге вся проблема полностью переворачивается с ног на голову, и теперь, атакующий вместо того, чтобы искать связь для подслушивания, сам создаёт её, сам становится фундаментом прослушивания в обличии сервера, а люди, в свою очередь, лишь выбирают лицо того, кто за ними будет пристально следить.

Уничтожить такую систему доверия не представляется возможным, из-за появления более общих и разрушительных видов атак, а также из-за невозможности полного искоренения доверия как такового [2, с.267]. Таким образом, остаётся лишь улучшать данную систему, делать так, чтобы сам её механизм стремился к уменьшению мощности доверия*, чтобы собственная её структура представляла защиту объектов и анонимат субъектов. К системам подобного рода относятся анонимные сети и тайные каналы связи.

*Мощность доверия — количество узлов, участвующих в хранении или передаче информации, представленной в открытом виде для данных узлов. Иными словами, такие узлы способны читать, подменять и видоизменять информацию, так как для них она находится в предельно чистом, прозрачном, транспарентном состоянии. Чем больше мощность доверия, тем выше предполагаемый шанс компрометации отдельных узлов, а следовательно, и хранимой на них информации. Принято считать одним из узлов получателя. Таким образом, нулевая мощность доверия будет возникать лишь в моменты отсутствия каких-либо связей и соединений. Если мощность доверия равна единице, это говорит о том, что связь защищена, иными словами, никто кроме отправителя и получателя информацией не владеют. Во всех других случаях мощность доверия будет больше единицы, что говорит о групповой связи (то-есть, о существовании нескольких получателей), либо о промежуточных узлах, способных читать информацию в открытом виде.

2. Анонимные сети

Скрытые, тёмные, анонимные сети – есть сети, грамотно соединяющие и объединяющие маршрутизацию вместе с шифрованием. Маршрутизация обеспечивает критерий анонимности, направленный на субъект, шифрование – критерии конфиденциальности, целостности, аутентификации, направленные на объект. Без маршрутизации легко определяются отправитель/получатель, без шифрования легко определяется передаваемое сообщение. Таким образом, только в совокупности этих двух свойств сеть может являться скрытой.

В современном мире большинство скрытых сетей представляют оверлейные соединения, иными словами соединения, которые основаны на уже существующей сети (например, сети Интернет). Но так или иначе, это не говорит, что скрытые сети не могут существовать сами по себе и быть однородной структурой, так как первоначальная архитектура может быть изначально нацелена на анонимность и безопасность. Именно по историческим причинам, современные сети имеют оверлейные уровни безопасности.

Любая анонимная сеть основывается либо на одноранговой, либо на гибридной архитектуре сети, исключая при этом многоранговую. Многоранговая сеть (или клиент-серверная архитектура) не может быть достаточно анонимной априори, потому как имеет открытую, прямую видимость и непосредственную реальность централизации. Централизация со стороны компьютерных сетей примитивна, но централизация со стороны скрытых сетей неочевидна. В компьютерных сетях под централизацией понимается ограниченное количество серверов, способных обслуживать куда большее количество клиентов. Когда же говорим об анонимных сетях, то централизация в таких системах может иметь свойство приходящее и уходящее, тем самым, измеряя уровень централизации не в виде булевой логики, а в виде процентных соотношений от всей суммарности сети.

В одноранговых системах все пользователи однородны, имеют одинаковые возможности, могут представлять одни и те же услуги маршрутизации. Такие сети легко могут разворачиваться в локальных системах, но с осложнениями работают в глобальном пространстве на основе уже созданных соединений. Сами одноранговые сети могут быть разделены на две категории: децентрализованные и распределённые. Само отличие очень туманно со стороны терминологии, тем не менее, лучшим решением будет расслоение их свойств, для последующего повествования. Распределённые сети можно именовать “истинно децентрализованными” (хоть и противоречиво), где нельзя выделить какой-либо центр или узел связи сразу нескольких других узлов. Характерной чертой такой сети является тот факт, что каждый пользователь одновременно соединяется напрямую сразу с несколькими другими. Децентрализованные же сети можно именовать “слабо централизованными” (хоть также и противоречиво), где к одним узлам сети подключаются сразу несколько других узлов. Характерной чертой такой сети является тот факт, что появляются “неофициальные” узлы, часто используемые другими узлами (узлы-серверы) в качестве последующей маршрутизации.

Гибридная система объединяет свойства многоранговых и одноранговых архитектур, пытаясь взять и удержать как можно больше положительных и меньше отрицательных качеств. Плюсом многоранговых архитектур являются некоторые свойства централизации, как например возможность разделения логики на серверную и клиентскую, более быстрая и/или статичная скорость маршрутизации. Плюсом одноранговых архитектур являются некоторые свойства децентрализации, как собственно возможность достижения и построения анонимности. Гибридные сети не всегда можно однозначно определить, иногда они выглядят и/или ведут себя сродни одноранговым, иными словами, используют минимальное количество свойств многоранговой архитектуры, или наоборот, могут иметь вид многоранговой архитектуры и даже

перерастать в неё полностью. Из всего этого многообразия, гибридные сети являются самой противоречивой системой.

Сама анонимность также не является чем-то однородным, точно определённым, абсолютным, её можно рассматривать как некую градацию, набор стадий, поэтапность. Всего же существует пять стадий и четыре вида анонимности (2,3,4,5 стадии).

1. Первая стадия характеризуется отсутствием анонимности $|A| = 0$*. Примером может являться несуществование связей или прямые соединения между двумя субъектами.

2. Вторая стадия игнорирует анонимность клиент-сервер, обеспечивая только анонимность клиент-клиент. Иными словами, сервер обладает достаточной информацией о клиентах (например, знание IP-адреса, личных данных, интересов). При этом сами клиенты не знают друг друга и потому являются анонимами. Типичным примером могут служить форумы, социальные сети, иначе говоря, большинство сайтов и приложений, построенных на основе клиент-серверной архитектуры. Из этого примера следует, что данная архитектура способна выдавать минимальную мощность анонимности, а именно $|A| = 1$. Второй этап является псевдо-анонимностью.

3. Третья стадия заменяет сетевой адрес криптографическим. Под заменой подразумевается неполное, частичное преобразование, так как сетевой адрес нельзя никак полностью скрыть, удалить, он будет существовать всегда и постоянно. Примером могут являться криптовалюты и блокчейн платформы, где известны лишь криптографические адреса клиентов (публичные ключи, хеши публичных ключей) [3]. Основным и главным отличием третьей стадии анонимности от второй, является тот факт, что сеть может представлять собой гибридный, а следовательно, и раздробленный, неопределённый характер поведения узлов сети, имеющих разные интересы. Из этого образуется и повышенная мощность анонимности, так как каждый узел может перенаправлять, распространять, транслировать информацию другим узлам, не заботясь о сохранении сетевого адреса клиента. $0 < |A| \leq N$, где N - количество узлов в сети. Тем не менее, маршрутизация не играет здесь роли анонимизации субъекта, а необходима лишь для распространения объекта. Получателя в такой системе как такового не существует и потому все действия ориентированы на сохранение информации на всех (или большинстве) узлах сети. Поэтому, третью стадию можно характеризовать игнорированием анонимности (экзотеричностью) со стороны субъекта и её сохранением (эзотеричностью) в передаваемом объекте. Стоит также заметить, что по такому же принципу работают стеганографические [4] методы, поэтому их применение вполне допустимо относить к третьей стадии анонимности.

4. Четвёртая стадия изменяет способ маршрутизации, ограничивая доступ узлов не только к объекту (передаваемой информации), но и к субъектам (отправителю и получателю). Мощность анонимности такого этапа чаще всего статична, тем самым $lim_{|A| \to C}$, где C - заданная по умолчанию константа. Примером данного этапа являются сами скрытые сети, подобия Tor (луковая маршрутизация), I2P (чесночная маршрутизация), RetroShare (turtle маршрутизация) и т.д.

5. Пятая стадия повышает мощность анонимности до теоретического максимума за счёт “слепой” маршрутизации (заливки) [5, с.398]. $lim_{|A| \to N}$, где N - количество узлов в сети. В таком способе не указывается получатель, а криптографический адрес отправителя зашифровывается на моменте отправления. Благодаря этому свойству, безопасность

приложений этого рода полностью (или в бо́льшей мере) зависит от клиентской части. Пятый этап является также представителем скрытых сетей, в качестве примера которых можно привести BitMessage.

Стоит также заметить, что шифрование, определяемое анонимностью, появляется частично лишь в моменты третьей стадии, в то время как во второй - само шифрование является добавочным, дополнительным и второстепенным свойством, служащим лишь и только для защиты клиент-серверной коммуникации.

В третьей стадии, под шифрованием чаще всего понимается только генерация асимметричных ключей и подписание информации. Таким образом, как такового шифрования на этой стадии ещё не происходит, но рождается очень важная концепция — возможность идентификации в одноранговых и гибридных системах, а следовательно, целостность и аутентификация самой передаваемой информации [6, с.223].

На четвёртом этапе добавляется шифрование самой информации (иными словами, рождается свойство конфиденциальности), так как теперь, в отличие от третьего этапа, информация представляет собой суть секретного, скрытого, тайного, а не открытого и общего объекта. Благодаря шифрованию, появляется возможность грамотно использовать маршрутизацию для повышения уровня анонимности в системах с получателем.

В четвёртой стадии существуют так называемые входные (entry) и выходные (exit) узлы, которые имеют дополнительную информацию либо об отправителе, либо о получателе. Так, например, входные узлы имеют IP-адрес отправителя, выходные - IP-адрес получателя. Если данные узлы окажутся скомпрометированными, то наступит этап деанонимизации субъектов. Решений данной проблемы несколько:

1. Первый способ нацелен на увеличение мощности анонимности минимум до трёх. Это обуславливается тем фактом, что даже если входные и выходные узлы окажутся скомпрометированными и будут обладать общей целью, то есть вычислением отправителя и получателя, то им помешает промежуточный узел. Точно также, если входной и промежуточный узел будут скомпрометированы, то им помешает выходной узел для полной деанонимизации субъектов. Всё это выходит из того правила, что мощность анонимности в таких случаях не падает ниже двух. Таким образом, одновременное выявление и отправителя, и получателя является сложной задачей.

2. Второй же способ нацелен на переход к пятой стадии — крайней форме анонимности. В таком случае, целью является сокрытие, удаление, исключение всех возможных связей между отправляемой информацией (объектом) и самим отправителем/получателем (субъектом). После исчезновения всех связей, сама маршрутизация перестаёт быть чем-то реальным и настоящим перерастая тем самым в этап условного и виртуального, где раскрытие даже одного из субъектов, начинает быть сложной задачей. Деанонимизация в таком случае возможна лишь при условии полного контроля сети, по причине сложного обнаружения и последующего восстановления связей с объектом.

Основным и главным отрицательным свойством пятой стадии анонимности является линейное увеличение нагрузки на сеть $O(N)$ со стороны всех пользователей в ней участвующих. Так, например, если сеть состоит из N узлов, то каждый узел должен будет обрабатывать $N-1$ запросов от других узлов. Время жизни пакета (TTL) на пятой стадии не является решением данной проблемы, по причине появления новых связей между отправителем и передаваемой информации, что, следовательно, ведёт к переходу на четвёртый этап.

Большинство атак, направленных на скрытые сети, представляют способы деанонимизации субъектов (как наиболее лёгкий способ), нежели попытки раскрытия, взлома, дешифрования объектов (как наиболее сложный). Так, например, достаточно сильной и сложно искоренимой атакой на одноранговые (а следовательно, и на гибридные) сети является атака Сивиллы. Она базируется на том факте, что главным способом анонимности является элемент маршрутизации, который обеспечивается за счёт передачи информации посредством нескольких узлов. С одной стороны, сутью атаки является замена несвязанных между собой узлов, на узлы подчинённые одному лицу, либо группе лиц с общими интересами, тем самым, атака ориентируется на ($lim_{|A|\to 1}$) уменьшение мощности анонимности до единицы. С другой стороны, в некоторых видах сетей, с мощностью доверия больше единицы, атака может вредить и целостности передаваемой информации, иными словами, подменять и видоизменять её. При повышении количества узлов несвязанных между собой в сети, повышается и сложность реализации атаки Сивиллы, за счёт более равномерного распределения узлов. Из этого также следует, что мощность анонимности будет стремиться к своему константному значению $lim_{|A|\to C}$ (если анонимность представляет четвёртую стадию) или к количеству несвязанных узлов $lim_{|A|\to N}$ (если анонимность представляет пятую стадию). Всё это связано с тем, что атакующие узлы будут конкурировать с обычными узлами за возможность быть посредниками между субъектами передаваемой информации. Чем больше несвязанных узлов и лучше алгоритм распределения, тем меньше вероятность осуществления данной атаки.

Ещё одним способом предотвращения подобной атаки является использование скрытых сетей, основанных на факторе доверия, параметре дружбы. Узлы в таких сетях выстраивают связи между собой, основываясь на субъективности к уровню доверия, так как никакого объективно доверенного, а следовательно, и централизованного, узла не существует. Выстраивая связи друг-к-другу (или friend-to-friend), узлы также начинают выстраивать связи друг моего друга — это мой скрытый друг. Таким образом, друзья друзей не подключаются напрямую и не знают друг друга, но при этом вполне могут обмениваться информацией между собой, что является показателем увеличения размеров сети. Чтобы успешно подключиться к такой сети, необходимо самому стать доверенным узлом, то-есть пользователем, которому кто-либо доверяет. Сложность исполнения атаки, на такой род сети, сводится к сложности встраивания в сеть подчиняемых узлов, как того требует атака Сивиллы, а это, как было описано выше, является проблематичным действием. Единственная проблема friend-to-friend (f2f) сетей заключается в их малой эксплозии, расширении, увеличении масштаба, являясь тем самым следствием причины ручной настройки и установки списка доверенных узлов.

Атака Сивиллы может быть рассмотрена более обще, где вместо встраивания узлов в скрытую сеть, происходит образование первичной сети, на основе которой будет существовать последующая тёмная сеть. Такой вид атаки может существовать лишь при оверлейных соединениях, коим и является сеть Интернет. Если вся тёмная сеть будет воссоздана в первичной сети, подконтрольной одному лицу или группе лиц с общими интересами, то, следовательно, и весь трафик скрытой сети возможно будет анализировать, с момента её появления и до момента её гибели. Подобная атака требует огромных ресурсов и первоначально настроенной инфраструктуры, что в современных реалиях под силу лишь государствам. Предотвратить такой вид атаки крайне сложно, но вполне возможно, если соблюдать два правила:

1. Использовать противоречия государств — вариативные и несогласованные законы, политические и империалистические интересы. Всё это есть моменты, при которых одно государство не будет выдавать информацию о своей сети другому государству. И чем более агрессивно настроены страны по отношению друг к другу, тем менее успешно они могут контролировать свои собственные ресурсы. В таком случае, необходимо строить сеть по

федеративному принципу, чтобы узлы располагались на разных континентах мира, странах и государствах.

2. Использовать изменения информации в процессе её маршрутизации. При таком способе информация будет представлена в полиморфной и самоизменяющийся оболочке, то-есть оболочке шифрованной. Такой подход необходим в моменты, когда информация, приходящая из государства A в государство B, будет снова возвращаться на свою родину A**. В качестве примера можно привести луковую маршрутизацию сети Tor, где само шифрование представлено в виде слоёв, которые каждый раз “сдирают”, снимают при передаче от одного узла к другому.

Существует также и альтернативный вариант противодействия подобной атаке. Он в отличие от вышеописанного не требует этапа с федеративностью, но взамен требует огромное количество информации, приводящую к спаму. Плюсом такого подхода является и то, что его можно использовать в тайных каналах связи как единственно возможный элемент анонимизации субъектов. Для осуществления такого метода идеально подходят сети основанные на пятой стадии анонимности, так как они распространяют информацию методом заливки, что априори ведёт к множественному дублированию, пролиферации. Полиморфизм информации осуществляется способом установки промежуточных получателей и созданием транспортировочных пакетов, представленных в форме множественного шифрования. Как только узел сети принимает пакет, он начинает его расшифровывать. Если пакет успешно расшифровывается, но при этом сама расшифрованная версия является шифрованным экземпляром, то это говорит о том, что данный принимающий узел — это промежуточный получатель, целью которого является последующее распространение “расшифрованной” версии пакета по сети. Так будет происходить до тех пор, пока не будет расшифрован последний пакет, предполагающий существование истинного получателя. Стоит также заметить, что промежуточные получатели при расшифровании пакета могут узнавать криптографический адрес отправителя, именно поэтому стоит отправлять транспортировочные пакеты из-под криптографического адреса псевдо-отправителя.

Пример программного кода [7] для создания транспортировочного пакета:

```
func RoutePackage(sender *PrivateKey, receiver *PublicKey, pack *Package, route []*PublicKey) *Package {
	var (
		rpack = Encrypt(sender, receiver, pack)
		psender = GenerateKey(N)
	)
	for _, pub := range route {
		rpack = Encrypt(
			psender,
			pub,
			NewPackage(ROUTE_MODE, SerializePackage(rpack)),
		)
	}
	return rpack
}
```

Если предположить, что в сети существует всего три узла *{A, B, C}* (где один из них является отправителем — *A*) и сама сеть основывается на пятой стадии анонимности без полиморфизма информации, то в таком случае и при таком условии крайне проблематично определить истинного получателя, пока он сам себя не выдаст ответом на запрос (так как ответом будет являться совершенно новый пакет, отличный от всех остальных). Теперь, если предположить, что

существует возможность полиморфизма информации, то есть вероятность её маршрутизации, то начинается этап слияния свойств получения и отправления, образуя антиципацию. Так, например, если полиморфизм существует, значит будет существовать три этапа: *(A → B ИЛИ A → C) → (B → C ИЛИ C → B) → (B → A ИЛИ C → A)*, но если полиморфизма не существует, то будет два этапа: *(A → B ИЛИ A → C) → (B → A ИЛИ C → A)****. При этом предполагается, что системе известен лишь отправитель информации (инициатор), в то время как получатель не определён. Из этого следует, что если полиморфизм будет являться статичной величиной (то есть, будет всегда существовать или не существовать вовсе), то определение получателя будет являться лёгкой задачей (при условии, что он всегда отвечает инициатору). Но, если полиморфизм будет иметь вероятностную величину, то грань между отправлением и получением будет стираться, сливаться, инвертироваться, что приведёт к разному трактованию анализируемых действий: запрос(1) - ответ(1) - запрос(2) или запрос(1) - маршрутизация(1) - ответ(1). Но в таком случае, возникает свойство гипертелии (сверх окончания), где запрос(2) не получает своего ответа(2), что снова приводит к возможности детерминированного определения субъектов. Теперь, если выровнять количество действий полиморфизма (количество маршрутизации пакета) *k* и количество действий без него *n* (что представляет собой всегда константу *n = 2*), иными словами придерживаться формулы *НОД(k, 2) = 2* (где *НОД* — наибольший общий делитель), то получим максимальную неопределённость, алеаторность при константе *k = 2*, которую можно свести к следующему минимальному набору действий полиморфизма: *(A → B ИЛИ A → C) → (B → C ИЛИ C → B) → (B → C ИЛИ C → B) → (B → A ИЛИ C → A)*. В итоге, все действия можно трактовать двумя полностью самодостаточными процессами: запрос(1) - ответ(1) - запрос(2) - ответ(2) или запрос(1) - маршрутизация(1) - маршрутизация(~1) - ответ(1), что в свою очередь приводит к неопределённости отправления и получения информации со стороны анализа трафика всей сети. И потому, ответ(1) = маршрутизация(1), ответ(2) = ответ(1), а также запрос(2) = маршрутизация(~1). Проблемой здесь является лишь запрос(1), созданный инициатором связи, который будет трактоваться всегда детерминировано. Но при последующих действиях данная проблема всегда угасает, становясь более хаотичной, так как образовывается неопределённость вида запрос(3) = запрос(2), приводящая к неоднозначному выявлению отправителя.

Далее, анализируя вышеописанный подход, можно также заметить, что маршрутизация и ответ являются этапами полностью автоматизированными, в то время как запрос является этапом ручным. Такой момент приводит к явлению, что между ответом и последующим запросом интервал времени ожидания больше, нежели между запросом - ответом, запросом - маршрутизацией, маршрутизацией - маршрутизацией и маршрутизацией - ответом. Это приводит к возможности осуществления атаки методом учёта времени с последующим расщеплением хаотичности, тем самым приводя к однозначности маршрутизации и к возможному выявлению субъектов информации. Предотвратить подобную уязвимость можно двумя противоположными, дифференцированными методами:

1. Симулировать время запроса. Иными словами, маршрутизация и ответ будут подстраиваться под примерное время генерации пакета в сети, способом установления задержки. Чем больше узлов в сети, тем меньше время задержки. Подобный метод следует использовать только в системах с большим количеством узлов, так как с малым количеством, время ожидания маршрутизации или ответа будет достаточно долгим.
2. Симулировать время маршрутизации и ответа. Иными словами, запрос будет подразумевать не только передачу истинной информации, но и передачу ложной, незначимой, пустой информации, в моменты отсутствия настоящего запроса. Подобный метод следует использовать только в системах с малым количеством узлов, так как производится огромное количество спама.

Продолжая анализ, можно выявить некоторые закономерности, приводящие к возможности точного обнаружения состояния пакета, а именно, является ли он запросом или ответом (при этом не разглашая субъектов информации). Исходя из периода *T*, который вычисляется по формуле *НОК(2+k, 2)* (где *НОК* - наименьшее общее кратное), несложно узнать, что период при $k = 2$ будет равен 4. Это в свою очередь говорит о том, что каждое четвёртое действие, начиная с предыдущего запроса, будет также являться запросом (аналогичная ситуация с ответом). Хоть и уязвимостей при данной детерминированности выявлено не было, тем не менее, с консервативной точки зрения, лучшим решением будет повышение периода, если атаки на этой основе будут обнаружены. Повысить период можно несколькими способами:

1. Повысить *k*. Тогда период $T = \begin{cases} 2 + k, k \bmod 2 = 0 \\ 2\,(2 + k), k \bmod 2 \neq 0 \end{cases}$ (не стоит забывать о свойстве гипертелии, если выбор падает на нечётное число).

2. Сделать *k* случайной переменной диапазона *[1;n]*, где $n < N$ (количество узлов в сети). Тогда период $T = НОК(2, 1+2, 2+2, ..., n+2)$.

* Мощность анонимности — количество узлов, выстроенных в цепочку и участвующих в маршрутизации информации, при этом, не будучи никак связанными между собой общими целями и интересами. Из этого следует, что многоранговая архитектура по умолчанию имеет мощность анонимности равной единице (вне зависимости от количества серверов). Нулевая мощность анонимности возникает либо при отсутсвии связей, либо при существовании прямого соединения между субъектами (иными словами при отсутствии какой бы то ни было маршрутизации).

$|A| = |S(N)| - \sum_{i=1}^{W} \begin{cases} 0, W_i = \varnothing \\ |W_i| - 1, W_i \neq \varnothing \end{cases}$, где

$W = E(S(N))$,

N - множество узлов, расположенных в сети,

S - функция выборки множества узлов, участвующих в маршрутизации,

E - функция выборки списка подмножеств узлов, подчиняющихся одному лицу или группе лиц с общими интересами.

** Мощность федеративности — количество государств, не связанных общей военной силой, через территорию которых проходит маршрутизация полиморфной информации. Из этого следует, что если сеть разворачивается лишь в пределах одного государства, то мощность федеративности по умолчанию будет равна единице. Нулевой мощности федеративности не существует.

*** Стрелка в скобках указывает отправителя слева и получателя справа, вне скобок стрелка указывает на изменение структуры пакета, сами же скобки предполагают существование одинакового пакета, операция *ИЛИ* указывает вариативность и параллельность отправления.

3. Тайные каналы связи

Секретные, тайные, эзотерические каналы связи - есть соединения, располагаемые в заведомо замкнутом, незащищённом, враждебном окружении и имеющие характеристики безопасной передачи информации. При этом анонимность, родственная тёмным сетям, не является

основанием секретных каналов связи и, следовательно, может быть отброшена из-за ненадобности или по необходимости. Так, например:

1. Первым, минимальным видом анонимности в тайных каналах связи принято считать третью стадию, то-есть сохранение экзотеричности субъекта и эзотеричности объекта, благодаря использованию криптографических и/или стеганографических методов преобразования информации. При этом, если в секретных каналах связи используются криптографические методы, то они не ограничиваются только идентификацией субъектов (целостностью, аутентификацией), но также и применяют практику шифрования объектов (конфиденциальность).

2. Вторым, максимальным видом анонимности в тайных каналах связи принято считать пятую стадию, при этом пропуская, игнорируя, импутируя четвёртую. Вся особенность такого подхода заключается в невозможности использовать фактическую, реальную маршрутизацию, которую предполагает четвёртая стадия анонимности. Тем самым реальная маршрутизация отдаёт откуп виртуальной, существование которой возможно лишь и только на пятой стадии анонимности. Виртуальная маршрутизация имманентна, сводится к передаче объекта внутри единого, сингулярного приложения, связывающего всех субъектов изнутри. Таким приложением является сервер (или группа серверов с $|A| = 1$), при помощи которого клиенты передают друг другу и принимают друг от друга информацию. Так как приложение располагает полным знанием того, кто является отправителем и кто является получателем, то сам сервер становится создателем сети на основе которой располагается тайный канал связи. При всём этом, такое приложение, в задаче о тайных каналах связи, аналогично и равносильно государству, в задаче о построении анонимных сетей. Всё это ведёт лишь к единственно возможной борьбе за анонимность с приложением-создателем — методом спама (так как способ с федеративностью бессилен и недействен в виртуальном пространстве).

Тайные каналы связи, использующие стеганографию, всегда имеют некий контейнер, в который помещается истинное сообщение. Под контейнером может пониматься ложное, неявное, сбивающее с пути сообщение, которое чаще всего носит нейтральный характер. Из этого также следует, что в зависимости от размера контейнера, зависит и размер самого исходного сообщения, тем самым, стеганографический подход рассчитан на сообщения малых размеров и мало пригоден для передачи целых файлов. Примером контейнера может служить изображение, аудио-запись, видео-файл, то есть всё, что может хранить дополнительную или избыточную информацию, которая останется незаметной для человеческих глаз и ушей. Одним из примеров сокрытия информации может служить замена каждого старшего бита в изображении, битом исходного сообщения. Таким образом, если размер изображения (то есть, контейнера) будет равен 2MiB, то максимальный размер исходного сообщения не будет превышать 256KiB.

Тайные каналы связи, использующие криптографию, по умолчанию можно охарактеризовать третьей стадией анонимности. Если тайный канал разворачивается в заведомо замкнутой и незащищённой, но всё же сети, то это говорит о том, что стадия анонимности не меньше второй. Сами же скрытые каналы данного вида используют идентификацию по криптографическим адресам, а не адресам, заданными системой по умолчанию (никнеймом, телефоном и т.д.), следовательно, стадия анонимности тайных каналов определяется третьим этапом. Далее, если возникает виртуальная маршрутизация между субъектами, то третья стадия начинает переходить в пятую, перешагивая при этом четвёртую. Таким образом, секретные каналы

способны улучшать безопасность уже выстроенной и существующей системы в неизменном для неё состоянии, используя лишь и только её базис в качестве фундамента.

Использование стеганографии, вместе с криптографией, может помочь в случаях, когда имеется вероятность или возможность нахождения скрытого сообщения в контейнере. Тем самым, даже если исходное сообщение было найдено, оно будет иметь шифрованный вид. Здесь стоит учитывать тот факт, что при шифровании размер информации увеличивается (добавляется хеш, подпись, текст дополняется до блока), а из этого уже следует, что максимальный размер исходного сообщения уменьшается.

Существует ещё один, третий способ сокрытия информации, относящийся к криптографическим, но при этом обладающий некоторыми стеганографическими свойствами, качествами, особенностями [8, с.720]. Это не является последовательным объединением, использованием методов, как это было описано выше, а скорее оказывается их слиянием, синтезом, симбиозом. В таком методе истинная информация скрывается в цифровой подписи ложного сообщения на основе общего, согласованного ключа, где главной чертой и исключительностью является стойкость ко взлому, сродни сложности взлома цифровой подписи. При этом, сама подпись - есть контейнер, скрывающий существование сообщения методом аутентификации ложной информации.

Теоретически, тайные каналы связи также могут находиться и в других скрытых каналах, либо анонимных сетях (секретные каналы могут быть воссозданы совершенно в разных системах и ситуациях), тем не менее, подобный подход является очень сомнительным (по причине избыточности накопленных слоёв шифрования), специфичным (по причине редкости практического использования) и затрачиваемым (по причине уменьшения производительности программ, уменьшения ёмкости контейнеров).

4. Протокол безопасной передачи информации

Из всего вышесказанного можно создать легковесный, примитивный, но при этом и безопасный протокол передачи информации, основанный на пятой стадии анонимности, потому как обеспечивает полное сокрытие субъектов в передаваемом объекте при помощи методов шифрования. Протокол является самодостаточным, цельным, монолитным, может быть применим к скрытым сетям и тайным каналам связи [2, с.58][8, с.80].

Участники протокола:

A - отправитель,

B - получатель.

Шаги участника А:

1. $K = G(N), R = G(M),$

   где G - функция-генератор случайных байт,

   N, M - количество байт для генерации,

   K - сеансовый ключ шифрования,

   R - случайный набор байт.

2. $H_M = H(R, M, PubK_A, PubK_B),$

   где H_M - хеш сообщения,

   H - функция хеширования,

   M - исходное сообщение,

   $PubK_A, PubK_B$ - публичные ключи отправителя и получателя.

3. $C_R = [\ E(PubK_B, K), E(K, PubK_A), E(K, M), E(K, R), H_M, E(K, S(PrivK_A, H_M)), W(C, H_M)\ ],$

   где C_R - зашифрованное сообщение,

   E - функция шифрования,

   S - функция подписания,

   W - функция подтверждения работы,

C - сложность работы,
$PrivK_A$ - приватный ключ отправителя.

Шаги участника B:

4. $W(C, H_M) = P(C, W(C, H_M))$,
   где P - функция проверки работы.
   Если $\neq$, то протокол прерывается.
5. $K = D(PrivK_B, E(PubK_B, K))$,
   где D - функция расшифрования,
   $PrivK_B$ - приватный ключ получателя.
   Если расшифрование неверно, то протокол прерывается.
6. $PubK_A = D(K, E(K, PubK_A))$.
7. $H_M = V(PubK_A, D(K, S(PrivK_A, H_M)))$,
   где V - функция проверки подписи.
   Если $\neq$, то протокол прерывается.
8. $H_M = H(D(K, E(K, R)), D(K, E(K, M)), PubK_A, PubK_B)$,
   Если $\neq$, то протокол прерывается.

Данный протокол игнорирует способ получения публичного ключа от точки назначения. Это необходимо по причине того, чтобы протокол был встраиваемым и мог внедряться во множество систем, включая одноранговые сети, не имеющие центров сертификации, и тайные каналы связи, имеющие уже установленную сеть по умолчанию.

Безопасность протокола определяется в большей мере безопасностью асимметричной функции шифрования, так как все действия сводятся к расшифрованию сеансового ключа приватным ключом. Если приватный ключ не может расшифровать сеансовый, то это говорит о том факте, что само сообщение было зашифровано другим публичным ключом и потому получатель также есть другой субъект. Функция хеширования необходима для проверки целостности отправленных данных. Функция проверки подписи необходима для аутентификации отправителя. Функция проверки доказательства работы необходима для предотвращения спама.

Протокол пригоден для многих задач, включая передачу сообщений, запросов, файлов, но не пригоден для передачи поточной информации, подобия аудио звонков и видео трансляций, из-за необходимости подписывать и подтверждать работу, на что уходит много времени. Иными словами, протокол работает с конечным количеством данных, размер которых заведомо известен и обработка которых (то есть, их использование) начинается с момента завершения полной проверки.

Пример программного кода для шифрования информации:

```
import (
	"bytes"
	"encoding/hex"
)
func Encrypt(sender *PrivateKey, receiver *PublicKey, data []byte) *Package {
	var (
		session  = GenerateBytes(N)
		rand     = GenerateBytes(M)
		pubsend = PublicKeyToBytes(&sender.PublicKey)
		hash     = HashSum(bytes.Join(
			[][]byte{
				rand,
				data,
				pubsend,
				PublicKeyToBytes(receiver),
			},
			[]byte{},
```

```
                ))
                sign = Sign(sender, hash)
        )
        return &Package{
                Head: HeadPackage{
                        Rand:    hex.EncodeToString(EncryptS(session, rand)),
                        Sender:  hex.EncodeToString(EncryptS(session, pubsend)),
                        Session: hex.EncodeToString(EncryptA(receiver, session)),
                },
                Body: BodyPackage{
                        Data:    hex.EncodeToString(EncryptS(session, data)),
                        Hash:    hex.EncodeToString(hash),
                        Sign:    hex.EncodeToString(EncryptS(session, sign)),
                        Npow:    ProofOfWork(hash, C),
                },
        }
}
```

Шифрование подписи сеансовым ключом является необходимым, так как взломщик протокола, для определения отправителя (а именно его публичного ключа) может составить список уже известных ему публичных ключей и проверять каждый на правильность подписи. Если проверка приводит к безошибочному результату, то это говорит об обнаружении отправителя.

Шифрование случайного числа (соли) также есть необходимость, потому как, если злоумышленник знает его и субъектов передаваемой информации, то он способен пройтись методом грубой силы по словарю часто встречаемых и распространённых текстов для выявления исходного сообщения.

Пример программного кода для расшифрования информации:

```
import (
        "bytes"
        "encoding/hex"
)
func Decrypt(receiver *PrivateKey, pack *Package) (*PublicKey, []byte) {
        // Check proof.
        hash, err := hex.DecodeString(pack.Body.Hash)
        if err != nil {
                return nil, nil
        }
        if !ProofIsValid(hash, C, pack.Body.Npow) {
                return nil, nil
        }
        // Decrypt session key.
        eskey, err := hex.DecodeString(pack.Head.Session)
        if err != nil {
                return nil, nil
        }
        skey := DecryptA(receiver, eskey)
        if skey == nil {
                return nil, nil
        }
        // Decrypt public key.
        ebpubsend, err := hex.DecodeString(pack.Head.Sender)
        if err != nil {
                return nil, nil
        }
```

```
	bpubsend := DecryptS(skey, ebpubsend)
	if bpubsend == nil {
		return nil, nil
	}
	pubsend := BytesToPublicKey(bpubsend)
	if pubsend == nil {
		return nil, nil
	}
	// Decrypt and check sign.
	esign, err := hex.DecodeString(pack.Body.Sign)
	if err != nil {
		return nil, nil
	}
	sign := DecryptS(skey, esign)
	if sign == nil {
		return nil, nil
	}
	if !Verify(pubsend, hash, sign) {
		return nil, nil
	}
	// Decrypt rand.
	erand, err := hex.DecodeString(pack.Head.Rand)
	if err != nil {
		return nil, nil
	}
	rand := DecryptS(skey, erand)
	if rand == nil {
		return nil, nil
	}
	// Decrypt data.
	edata, err := hex.DecodeString(pack.Body.Data)
	if err != nil {
		return nil, nil
	}
	data := DecryptS(skey, edata)
	if data == nil {
		return nil, nil
	}
	// Check hash.
	check := HashSum(bytes.Join(
		[][]byte{
			rand,
			data,
			PublicKeyToBytes(pubsend),
			PublicKeyToBytes(&receiver.PublicKey),
		},
		[]byte{},
	))
	if !bytes.Equal(hash, check) {
		return nil, nil
	}
	return pubsend, data
}
```

Для улучшения эффективности, допустим при передаче файла, программный код можно изменить так, чтобы снизить количество проверок работы в процессе передачи, но с первоначальным доказательством работы на основе случайной строки (полученной от точки

назначения), а потом и с накопленным хешем из *n*-блоков файла, для *i*-ой проверки. Таким образом, минимальный контроль работы будет осуществляться лишь $\lceil M/nN \rceil + 1$ раз, где *M* - размер файла, *N* - размер одного блока. Если доказательство не поступило или оно является неверным, то нужно считать, что файл был передан с ошибкой и тем самым запросить повреждённый или непроверенный блок заново.

5. Заключение

В данной работе были проанализированы системы, представляющие безопасность и анонимность пользователей (анонимные сети, тайные каналы связи), выявлены их свойства и особенности, объединяющие (стремление к уменьшению мощности доверия) и разделяющие (методы достижения анонимности). Была приведена градация анонимности в компьютерных сетях, базируемая на мощности анонимности. На основе же градации было выявлено само развитие анонимности и необходимые условия для её существования. Были приведены атаки, базируемые не деанонимизации субъектов, и защита от них. В части о тайных каналах связи, разобраны возможные виды анонимности, представлено разделение каналов по используемым средствам (криптографические, стеганографические), а также были приведены их отличия (шифрование, скрытие) и синтезы (последовательное применение, скрытие информации в цифровой подписи). После анализа скрытых систем был представлен протокол безопасной передачи информации (вместе с примерами программного кода), на основе которого могут базироваться в последующем анонимные сети и тайные каналы связи.

6. Список литературы

1. Диффи. В., М. Хеллман. Новые направления в криптографии [Электронный ресурс]. — Режим доступа: https://ee.stanford.edu/~hellman/publications/24.pdf (дата обращения: 19.12.2020).
2. Шнайер, Б., Фергюсон, Н. Т. Практическая криптография / Б. Шнайер, Н. Т. Фергюсон. - М.: Издательский дом «Вильямс, 2005. - 420 с.
3. Накамото, С. Биткойн: система цифровой пиринговой наличности [Электронный ресурс]. — Режим доступа: https://bitcoin.org/files/bitcoin-paper/bitcoin_ru.pdf (дата обращения: 19.12.2020).
4. Шелухин, О.И., Канаев, С.Д. Стеганография. Алгоритмы и программная реализация / О.И. Шелухин, С.Д. Канаев. — М.: Горячая линия - Телеком, 2018. - 592 с.
5. Таненбаум, Э., Уэзеролл, Д. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. — СПб.: Питер, 2017. - 960 с.
6. Рябко, Б. Я., Фионов, А. Н. Криптография в информационном мире / Б. Я. Рябко, А. Н. Фионов. - М.: Горячая линия - Телеком, 2019. - 300 с.
7. Донован, А.А., Керниган, Б.У. Язык программирования Go / А.А. Донован, Б.У. Керниган. — М.: ООО «И.Д. Вильямс», 2018. - 432 с.
8. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы и исходные коды на языке С / Б. Шнайер. — СпБ.: ООО «Альфа-книга», 2018. - 1040 с.