Internet Engineering Task Force (IETF)
Request for Comments: 8922
Category: Informational
ISSN: 2070-1721

T. Enghardt
TU Berlin
T. Pauly
Apple Inc.
C. Perkins
University of Glasgow
K. Rose
Akamai Technologies, Inc.
C. Wood
Cloudflare

October 2020

A Survey of the Interaction between Security Protocols and Transport Services

Обзор взаимодействия между протоколами защиты и транспортными службами

Аннотация

В этом документе представлен обзор распространённых или заметных сетевых протоколов защиты с акцентом на их взаимодействии и интеграции с приложениями и транспортными протоколами. Цель документа состоит в дополнении работ по определению и каталогизации транспортных услуг путём описания интерфейсов, требуемых для добавления протоколов защиты. Обзор не ограничивается протоколами, разработанными в рамках или контексте IETF[1], и включает протоколы, представленные надмножеством транспортных служб, для которых может потребоваться поддержка.

Статус документа

Документ не содержит какого-либо стандарта (Internet Standards Track) и публикуется с информационными целями.

Документ является результатом работы IETF[2] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[3]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке https://www.rfc-editor.org/info/rfc8922.

Оглавление

1. Введение....2
1.1. Цели....2
1.2. Дополнительные аспекты....2
2. Терминология....3
3. Описания протоколов транспортной защиты....3
3.1. Протоколы защиты данных приложения....4
3.1.1. TLS....4
3.1.2. DTLS....4
3.2. Зависимые от приложения протоколы защиты....4
3.2.1. Secure RTP....4
3.3. Протоколы защиты на транспортном уровне....4
3.3.1. IETF QUIC....4
3.3.2. Google QUIC....4
3.3.3. tcpcrypt....4
3.3.4. MinimaLT....4
3.3.5. CurveCP....5
3.4. Протоколы защиты пакетов....5
3.4.1. IPsec....5
3.4.2. WireGuard....5
3.4.3. OpenVPN....5
4. Транспортные зависимости....5
4.1. Надёжный транспорт для байтовых потоков....5

[1]Internet Engineering Task Force.
[2]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[3]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

4.2. Транспортировка дейтаграмм без гарантий....5
4.2.1. Протоколы дейтаграмм с отображением на поток байтов....5
4.3. Связанные с транспортом зависимости....6
5. Интерфейс с приложениями....6
5.1. Интерфейсы до соединения....6
5.2. Интерфейсы соединения....7
5.3. Интерфейсы после соединения....7
5.4. Сводка интерфейсов, раскрываемых протоколами....8
6. Взаимодействие с IANA....8
7. Вопросы безопасности....8
8. Вопросы приватности....8
9. Литература....8
Благодарности....9
Адреса авторов....10

1. Введение

Услуги и свойства, предоставляемые транспортными протоколами, классифицированы в [RFC8095]. Это документ дополняет выполненную работу и указывает интерфейсы между этими протоколами, а также между транспортными протоколами и приложениями. Документ исследует TLS[1], DTLS[2], IETF QUIC, Google QUIC (gQUIC), tcpcrypt, IPsec[3], SRTP[4] с DTLS, WireGuard, CurveCP, MinimaLT и для каждого протокола в документе приведено краткое описание. Описаны также интерфейсы между этими протоколами и транспортом (раздел 4) и интерфейсы между протоколами и приложениями (раздел 5).

Система транспортных служб раскрывает приложениям интерфейс для доступа к разным (защищённым) транспортным функциям. Протоколы защиты, включённые в этот исследование, представляют расширенный набор функций и возможностей транспортных служб, которые могут потребоваться приложениям как для внутреннего, так и для внешнего применения (через API) [TAPS-ARCH]. Распространённые повсеместно протоколы IETF, такие как (D)TLS, наряду с нестандартными протоколами (например, gQUIC) включены в документ, несмотря на перекрывающиеся функции. Таким образом, исследование не ограничивается протоколами разработанными в сфере действия или контексте IETF. За пределами этого набора остались протоколы, которые не предлагают новых функций. Например, более новые протоколы, такие как WireGuard, применяют уникальные решения, которые могут влиять на ограничения при использовании приложений. Напротив, такие протоколы, как SSH [RFC4253], GRE [RFC2890], L2TP[5] [RFC5641], ALTS[6] [ALTS] не включены в обзор, поскольку они не имеют уникальных интерфейсов.

Не включены протоколы, предлагающие лишь аутентификацию, такие как TCP-AO[7] [RFC5925] и IPsec AH[8] [RFC4302]. TCP-AO добавляет аутентификацию для долгосрочных соединений TCP, например, защиту от повторного использования пакетов с помощью кодов MAC[9] на уровне сообщения. TCP-AO обменяет «подписи» TCP MD5, заданные в [RFC2385] и одним из основных применений TCP-AO является защита соединений BGP. AH добавляет на уровне дейтаграмм аутентификацию и защиту целостности, а также защиту от повтора пакетов. Несмотря на эти усовершенствования, ни один из протоколов не относится в категории общего пользования и оба не включают важных свойств, требуемых для новых протоколов защиты, таких как защита конфиденциальности и приватности. Такие протоколы не включены в исследование.

В документ включены лишь протоколы парного взаимодействия (point-to-point), но не групповые протоколы.

1.1. Цели

Этот обзор предназначен для того, чтобы помочь определить более общие интерфейсы между протоколами защиты и транспорта, а также между приложениями и протоколами защиты.

Одной из целей рабочей группы Transport Services является определение общего интерфейса для транспортных протоколов, который позволили бы использующим такие протоколы программам легко приспосабливаться к новым протоколам, обеспечивающим похожий набор функций. Обзор зависимостей протоколов защиты от транспортных протоколов может помочь реализациям при определении набора транспортных протоколов, подходящего для использования преимуществ данного протокола защиты. Например, протокол защиты, предполагающий работу на основе гарантированной доставки потока байтов (например, TLS), ограничивает набор подходящих для него транспортных протоколов.

Определение общих интерфейсов, предоставляемых протоколами защиты приложениям, также позволяет организовать интерфейсы так, чтобы для общих функций могли применяться одни API. Например, многие протоколы защиты, обеспечивающие проверку подлинности, позволяют приложению участвовать в проверке идентичности партнёра. В таких случаях любой интерфейс, использующий стек протоколов защищённого транспорта, должен позволять приложению выполнить такое действие в процессе организации соединения.

1.2. Дополнительные аспекты

Хотя анализ в этом документе похож на анализ транспортных протоколов в [RFC8095], важно подчеркнуть, что применение протоколов защиты требует большего внимания.

В документе не ставится цели обеспечить программам возможность автоматического переключения между разными протоколами защиты даже при эквивалентных интерфейсах с транспортным протоколом и приложением. Даже версии

[1]Transport Layer Security - защита транспортного уровня.
[2]Datagram Transport Layer Security - защита транспортного уровня для дейтаграмм.
[3]Internet Protocol Security - защита протокола IP.
[4]Secure Real-time Transport Protocol - защищенный транспортный протокол в реальном масштабе времени.
[5]Layer 2 Tunneling Protocol - протокол туннелирования на канальном уровне.
[6]Application Layer Transport Security - протокол защиты транспорта прикладного уровня.
[7]TCP Authentication Option - опция аутентификации TCP.
[8]Authentication Header - заголовок аутентификации.
[9]Message Authentication Code - код аутентификации сообщения.

протоколов защиты могут различаться в плане гарантий и уязвимостей. Поэтому любая реализация должна применять лишь тот набор протоколов и алгоритмов, который был запрошен приложениями или правилами системы.

В разных протоколах защиты могут применяться несовместимые толкования идентичности партнёров, аутентификации и криптографических опций. Документ не ставит цели определить общий набор представления для таких концепций.

Рассматриваемые здесь протоколы предоставляют расширенный набор функций и свойств, поддержка которых может потребоваться от транспортной системы. Документ не перечисляет всех транспортных протоколов, реализация которых может потребоваться в транспортной системе, и не требует реализации того или иного протокола.

Транспортные службы могут реализовать любой протокол защищённого транспорта, обеспечивающий описанные функции. При этом может потребоваться раскрытие интерфейса приложению для настройки этих функций.

2. Терминология

Ниже приведены определения используемых в документе терминов для описания ролей и взаимодействий протоколов защищённого транспорта (некоторые термины определены также в [RFC8095]).

Transport Feature - транспортная функция (свойство)
: Сквозная функция обеспечиваемая приложению транспортным уровнем. Примерами являются защита конфиденциальности, надёжная или упорядоченная доставка, работа с потоком или сообщениями.

Transport Service - транспортный сервис (служба)
: Набор транспортных функций, обеспечивающих приложению полное обслуживание, без привязки к протоколу кадрирования.

Transport Services system - система транспортных услуг
: Программные компоненты, раскрывающие приложению интерфейс к разным транспортным службам.

Transport Protocol - транспортный протокол
: Реализация, обеспечивающая одну или несколько транспортных служб с использованием конкретного формата кадрирования и заголовков в линии передачи (проводе). Транспортный протокол обслуживает приложение напрямую или с использованием протокола защиты.

Application - приложение
: Сущность (элемент), использующая интерфейс транспортного уровня для сквозной доставки данных через сеть (возможно, вышележащий протокол или туннель).

Security Protocol - протокол защиты (безопасности)
: Определённый сетевой протокол, реализующий одну или несколько функций защиты, таких как аутентификация, шифрования, генерация ключей, возобновление сессий и приватность. Протоколы защиты могут применяться вместе с транспортными протоколами и в комбинации с другими протоколами защиты.

Handshake Protocol - протокол согласования
: Протокол, позволяющий партнёрам проверить друг друга и безопасно организовать общий контекст криптографической защиты.

Record - запись
: Кадрированные сообщения протокола.

Record Protocol - протокол записей
: Протокол защиты, позволяющий разделить данные на управляемые блоки и защитить их с использованием общего криптографического контекста.

Session - сессия (сеанс)
: Эфемерная защитная ассоциация между приложениями.

Connection - соединение
: Общее состояние двух или более конечных точек, сохраняющееся в передаваемых между этими точками сообщениях. Соединение является временным участником сессии и сессия обычно длится дольше и может включать разные экземпляры соединений.

Peer - партнер
: Приложение конечной точки, участвующее в сессии.

Client - клиент
: Партнер, отвечающий за инициирование сессии.

Server - сервер
: Партнер, отвечающий за отклик на инициирование сессии.

3. Описания протоколов транспортной защиты

В этом разделе даны краткие описания транспорта и защиты различных протоколов, используемых для защиты данных, отправленных через сеть. Эти протоколы сгруппированы по месту их реализации в стеке, что влияет на защищаемые ими части пакета - базовые данные приложения (payload), данные конкретного прикладного протокола, данные приложения и транспортные заголовки, весь пакет IP.

Отметим, что не все протоколы защиты легко классифицировать, например, некоторые протоколы могут применяться разными способами или в комбинации с другими протоколами. Основная причина этого заключается в том, что протоколы защиты каналов часто включают два компонента, указанных ниже.

- Протокол согласования (handshake) отвечающий за согласование параметров, проверку подлинности конечных точек и организацию общих ключей.
- Протокол записей (record), используемый для шифрования трафика с помощью ключей и параметров, предоставленных протоколом согласования.

Для некоторых протоколов (например, tcpcrypt) эти компоненты тесно связаны. И напротив, в IPsec они реализованы в разных протоколах - AH и ESP[1], - являющихся протоколами записи, которые используют ключи, предоставленные протоколом обмена ключами IKEv2[2], другими протоколами согласования или вручную.

[1]Encapsulating Security Payload - защищенные данные инкапсуляции.
[2]Internet Key Exchange Protocol Version 2 - протокол обмена ключами в Internet, версия 2.

Кроме того, некоторые протоколы могут использоваться разными способами. Базовый протокол TLS, определённый в [RFC8446], имеет встроенные протоколы согласования и записей, но TLS или DTLS можно также применять для согласования ключей в других протоколах (например, DTLS-SRTP) или протокол согласования может использоваться с отдельным уровнем записей, как в QUIC [QUIC-TRANSPORT].

3.1. Протоколы защиты данных приложения

Приведённые в этом параграфе протоколы обеспечивают защиту данных приложения (payload), передаваемых с использованием транспорта. Эти протоколы не защищают заголовков, используемых функциями транспортного уровня.

3.1.1. TLS

TLS [RFC8446] является базовым протоколом для организации защищённых сессий между парой конечных точек. Коммуникации в таких сессиях защищены от перехвата, подмены и подставных сообщений. TLS включает тесно связанные протоколы согласования и записей. Протокол согласования служит для аутентификации партнёров, согласования опций протокола, таких как криптографические алгоритмы, и создания ключевого материала для сессии. Протокол записей используется для «присмотра» (marshal), а после согласования служит для шифрования данных между партнёрами. Эти данные могут включать согласующие сообщения и необработанные (raw) данные приложения.

3.1.2. DTLS

Протокол DTLS [RFC6347] [DTLS-1.3] основан на TLS, но отличается тем, что предназначен для работы с протоколами дейтаграмм, такими как UDP, взамен TCP. DTLS меняет протокол, чтобы обеспечить гарантии защиты, эквивалентные TLS, за исключением сохранения порядка и невозможности повторного использования. DTLS разработан максимально похожим на TLS, поэтому здесь предполагается, что перенесены все свойства TLS, кроме отмеченных выше.

3.2. Зависимые от приложения протоколы защиты

Здесь представлен протокол, обеспечивающий зависимую от приложения защиту для всех данных приложения в конкретных вариантах применения. В отличие от указанных выше протоколов, он не предназначен для использования в приложениях общего назначения.

3.2.1. Secure RTP

Secure RTP (SRTP) - это профиль RTP, обеспечивающий конфиденциальность, аутентификацию сообщений и защиту от повторного использования для пакетов данных RTP и пакетов протокола управления RTCP[1] [RFC3711]. SRTP поддерживать только уровень записей и требует отдельного протокола согласования для управления ключами и контроля идентичности.

Наиболее широко в качестве протокола согласования для SRTP применяется DTLS в форме DTLS-SRTP [RFC5764]. Это расширение DTLS, согласующее применение SRTP как уровня записей и описывающее экспорт ключей для SRTP.

ZRTP [RFC6189] является другим вариантом протокола согласования ключей и контроля идентичности для SRTP. Согласование ключей в ZRTP выполняется с использованием обмена Diffie-Hellman на пути в среде. Алгоритм создаёт общий ключ, который служит для генерации первичного ключа и затравки (salt) для SRTP.

3.3. Протоколы защиты на транспортном уровне

Включённые в этот раздел протоколы обеспечивают защиту для данных приложения и заголовков, используемых транспортными службами.

3.3.1. IETF QUIC

QUIC — это новый стандартный протокол, работающий по протоколу UDP и частично основанный на фирменном протоколе Google (gQUIC) [QUIC-TRANSPORT] (см. параграф 3.3.2). Транспортный уровень QUIC обеспечивает защиту конфиденциальности и целостности. Это требует создания ключей с помощью отдельного протокола согласования. Отображение QUIC на TLS 1.3 [QUIC-TLS] была задано для выполнения такого согласования.

3.3.2. Google QUIC

Google QUIC (gQUIC) - это основанный на UDP мультиплексируемый потоковый протокол, разработанный и развернутый компанией Google на основании опыта развёртывания SPDY - фирменного предшественника HTTP/2. Протокол gQUIC исходно назывался QUIC, но в этом документе используется обозначение gQUIC, чтобы различать этот протокол и IETF QUIC. Запатентованный технический предшественник IETF QUIC - gQUIC исходно включает интеграцию защиты и транспорта данных приложений.

3.3.3. tcpcrypt

Протокол tcpcrypt [RFC8548] является облегчённым расширением TCP для гибкого управления шифрованием. Приложения могут использовать уникальные идентификаторы сессий tcpcrypt для дополнительной аутентификации на уровне приложений. Без этой аутентификации протокол tcpcrypt уязвим для активных атак.

3.3.4. MinimaLT

MinimaLT [MinimaLT] - это основанный на UDP протокол транспортной защиты, разработанный для обеспечения конфиденциальности, взаимной аутентификации, предотвращения DoS[2] и мобильности соединений. Одной из основных целей протокола является усиление имеющихся протоколов для получения данных конфигурации на стороне сервера, используемых для ускоренной организации соединения. MinimaLT использует вариант механизма контроля перегрузок TCP.

---

[1] RTP control protocol - протокол управления RTP.

[2] Denial-of-Service - отказ в обслуживании. *Прим. перев.*

3.3.5. CurveCP

CurveCP [CurveCP] - это основанный на UDP протокол транспортной защиты, основанный в отличие от многих других протоколов защиты, полностью на алгоритмах открытых ключей. CurveCP обеспечивает гарантии для данных приложения как часть протокола.

3.4. Протоколы защиты пакетов

Перечисленные в этом разделе протоколы обеспечивают защиту пакетов IP. Они обычно применяются для туннелей, таких как VPN[1]. Зачастую приложения не взаимодействуют напрямую с этими протоколами, однако приложения, реализующие туннели, используют непосредственное взаимодействие с протоколами.

3.4.1. IPsec

IKEv2 [RFC7296] и ESP [RFC4303] совместно образуют современный стек протоколов IPsec для шифрования и аутентификации пакетов IP при создании туннелей (туннельный режим) или непосредственно в транспортных соединениях (транспортный режим). Этот стек протоколов отделяет протокол генерации ключей (IKEv2) от протокола транспортного шифрования (ESP). Каждый протокол можно применять независимо, но в этом документе они рассматриваются вместе, поскольку это встречается чаще.

3.4.2. WireGuard

WireGuard [WireGuard] - это протокол уровня IP, разработанный как альтернатива IPsec для некоторых вариантов применения. Протокол использует UDP для инкапсуляции дейтаграмм IP между партнёрами. В отличии от большинства протоколов транспортной защиты, опирающихся на PKI[2] для аутентификации партнёра., WireGuard проверяет подлинность партнёров с помощью заранее распространённых открытых ключей, переданных по отдельному каналу (out of band), каждый из которых привязан к одному или множеству адресов IP. Кроме того, как протокол, предназначенный для VPN, WireGuard не предлагает расширяемости, согласования и криптографической ловкости (agility).

3.4.3. OpenVPN

OpenVPN [OpenVPN] - это широко распространённый протокол, разработанный как альтернатива IPsec. Основной целью протокола является организация VPN с простой настройкой и работой с разным транспортом. OpenVPN инкапсулирует пакеты IP или кадры Ethernet в защищённый туннель и может работать по протоколу UDP или TCP. Для организации ключей OpenVPN может использовать TLS в качестве протокола согласования или работать с заранее распространёнными ключами.

4. Транспортные зависимости

Для перечисленных выше протоколов защиты основная зависимость от транспортного протокола определяется представлением данных в форме неограниченного потока байтов или кадрированных сообщений. Среди протоколов, основанных на доставке кадрированных сообщений большая часть работает с транспортном, которому присуще внутреннее кадрирования (например, UDP), но некоторые протоколы определяют своё кадрирование сообщений для работы с транспортом байтовых потоков.

4.1. Надёжный транспорт для байтовых потоков

Перечисленные здесь протоколы зависят от транспорта, обеспечивающего надёжную, упорядоченную доставку потока байтов (обычно это TCP).

Протоколы защиты данных приложения (payload)

TLS.

Протоколы защиты на транспортном уровне

tcpcrypt.

4.2. Транспортировка дейтаграмм без гарантий

Перечисленные здесь протоколы зависят от транспорта, обеспечивающего кадрирование сообщений для инкапсуляции данных протокола. Эти протоколы созданы для работы на основе UDP и не предъявляют требований к надёжности доставки. Работа этих протоколов на основе транспорта с гарантированной доставкой не будет препятствовать функциональности, но может привести к нескольким уровня обеспечения гарантий при инкапсуляции протоколом защиты трафика другого транспортного протокола.

Протоколы защиты данных приложения (payload)

DTLS;
ZRTP;
SRTP.

Протоколы защиты на транспортном уровне

QUIC;
MinimaLT;
CurveCP.

Протоколы защиты пакетов

IPsec;
WireGuard;
OpenVPN.

4.2.1. Протоколы дейтаграмм с отображением на поток байтов

Среди перечисленных выше протоколов, зависящих от транспорта с кадрированием сообщений, некоторые имеют общеизвестные отображения для передачи с использованием транспорта на основе потока байтов, такого как TCP.

[1]Virtual Private Network - виртуальная частная сеть.
[2]Public Key Infrastructure - инфраструктура открытых ключей.

Протоколы защиты данных приложения (payload)

DTLS при использовании как протокола согласования для SRTP [RFC7850];
ZRTP [RFC6189];
SRTP [RFC4571][RFC3711].

Протоколы защиты пакетов

IPsec [RFC8229].

4.3. Связанные с транспортом зависимости

Один из рассматриваемых протоколов (tcpcrypt) имеет прямую зависимость от свойства транспорта, требуемого для его работы. А именно, tcpcrypt предназначен для работы на основе TCP и использует опцию TCP-ENO[1] [RFC8547] для согласования поддержки протокола.

QUIC, CurveCP и MinimaLT поддерживают функции транспорта и защиты. Они зависят от работы по протоколу с кадрированием, подобному UDP, но добавляют свои уровни гарантий доставки и других транспортных услуг. Таким образом, приложение, использующее такой протокол, не может отвязать защиту от транспортной функциональности.

5. Интерфейс с приложениями

В этом разделе описаны интерфейсы, раскрываемые описанными выше протоколами защиты. Эти интерфейсы разделены на pre-connection (до соединения, настройка), connection (соединение), post-connection (после соединения) в соответствии с соглашениями [TAPS-INTERFACE] и [TAPS-ARCH].

Отметим, что не все протоколы поддерживают каждый интерфейс. В таблице параграфа 5.4 приведена сводка интерфейсов, раскрываемых протоколами. В последующих параграфах даны сокращения имён интерфейсов, используемые в этой таблице.

5.1. Интерфейсы до соединения

Конфигурационные интерфейсы служат для настройки протоколов защиты до начала согласования или установки ключей.

Отождествления и секретные ключи (IPK[2])

Приложение может предоставлять своё отождествление, свидетельства (например, сертификаты) и секретные ключи или механизм доступа к ним протоколу защиты для использования в процессе согласования.

- TLS;
- DTLS;
- ZRTP;
- QUIC;
- MinimaLT;
- CurveCP;
- IPsec;
- WireGuard;
- OpenVPN.

Поддерживаемые алгоритмы (обмен ключами, подписи, шифронаборы) (ALG)

Приложение может выбрать алгоритмы, поддерживаемые им для обмена ключами, подписями и шифронаборами.

- TLS;
- DTLS;
- ZRTP;
- QUIC;
- tcpcrypt;
- MinimaLT;
- IPsec;
- OpenVPN.

Расширения (EXT)

Приложение включает или настраивает расширения, согласуемые протоколом защиты, таким как ALPN[3] [RFC7301].

- TLS;
- DTLS;
- QUIC.

Управление сеансовым кэшем (CM[4])

Приложение обеспечивает возможность сохранить и извлечь состояние сессии (квитанции, ключевой материал, параметры сервера), которое можно использовать для восстановления сеанса защиты.

- TLS;
- DTLS;
- ZRTP;
- QUIC;
- tcpcrypt;
- MinimaLT.

Передача полномочий проверки подлинности (AD[5])

Приложение предоставляет доступ к отдельному модулю, обеспечивающему аутентификацию, например, с помощью протокола EAP[6] [RFC3748].

- IPsec;
- tcpcrypt.

---

[1]TCP Encryption Negotiation Option - опция согласования шифрования TCP.
[2]Identities and Private Keys.
[3]Application-Layer Protocol Negotiation - протокол согласования на уровне приложений.
[4]Cache Management.
[5]Authentication Delegation
[6]Extensible Authentication Protocol - расширяемый протокол аутентификации.

Импорт заранее распространённых ключей (PSKI[1])

Протокол согласования или приложение напрямую может представить распределенные заранее общие ключи для использования при шифровании (и проверке подлинности) при взаимодействии с партнёром.

- TLS;
- DTLS;
- ZRTP;
- QUIC;
- tcpcrypt;
- MinimaLT;
- IPsec;
- WireGuard;
- OpenVPN.

5.2. Интерфейсы соединения

Проверка идентичности (IV[2])

В процессе согласования протокол защиты проверяет идентичность своего партнёра. Это может разгружать проверку или выполняться незаметно для приложения.

- TLS;
- DTLS;
- ZRTP;
- QUIC;
- MinimaLT;
- CurveCP;
- IPsec;
- WireGuard;
- OpenVPN.

Проверка адреса получателя (SAV[3])

Протокол согласования может взаимодействовать с транспортных протоколом или приложением для проверки адреса удалённого партнёра., приславшего данные. Это включает обмен cookie для предотвращения DoS-атак. В списке не указаны протоколы, зависящие от TCP, что ведёт к неявному выполнению SAV.

- DTLS;
- QUIC;
- IPsec;
- WireGuard.

5.3. Интерфейсы после соединения

Прерывание соединения (CT[4])

Протоколу защиты можно дать инструкцию разорвать соединение и удалить информацию о сессии. Это нужно некоторым протоколам, например, для предотвращения атак с отсечкой данных приложения, где злоумышленник прерывает работу базового незащищённого протокола, ориентированного на соединения, для разрыва сессии.

- TLS;
- DTLS;
- ZRTP;
- QUIC;
- tcpcrypt;
- MinimaLT;
- IPsec;
- OpenVPN.

Обновление ключей (KU[5])

Протоколу согласования можно дать инструкцию обновить его ключевой материал напрямую в приложении или с помощью протокола записей, инициировав событие завершения срока действия ключей.

- TLS;
- DTLS;
- QUIC;
- tcpcrypt;
- MinimaLT;
- IPsec.

Экспорт общего секретного ключа (SSKE[6])

Протокол согласования может предоставить интерфейс для создания общего секрета, используемого для зависящих от приложения задач.

- TLS;
- DTLS;
- tcpcrypt;
- IPsec;
- OpenVPN;
- MinimaLT.

Завершение срока действия ключа (KE[7])

Протокол записей может сообщать о завершении срока действия ключей (по времени или объёму зашифрованных данных). Это взаимодействие часто ограничено сигналами между уровнями согласования и записей.

- IPsec.

---

[1]Pre-Shared Key Import.
[2]Identity Validation.
[3]Source Address Validation.
[4]Connection Termination.
[5]Key Update.
[6]Shared Secret Key Export.
[7]Key Expiration.

События мобильности (ME[1])

Протокол записей может сообщать о своём переносе на другой транспорт или интерфейс по причине мобильности соединения, что может приводить к сбросу адреса и проверки состояния, а также вызывать смену состояния, например, использование нового идентификатора соединения (Connection Identifier или CID).

- DTLS (только версия 1.3 [DTLS-1.3]);
- QUIC;
- MinimaLT;
- CurveCP;
- IPsec [RFC4555];
- WireGuard.

5.4. Сводка интерфейсов, раскрываемых протоколами

В таблице знаком + указаны интерфейсы, раскрываемые каждым из протоколов.

Таблица 1.

Протокол	*IPK*	*ALG*	*EXT*	*CM*	*AD*	*PSKI*	*IV*	*SAV*	*CT*	*KU*	*SSKE*	*KE*	*ME*
TLS	+	+	+	+		+	+		+	+	+		
DTLS	+	+	+	+		+	+	+	+	+	+		+
ZRTP	+	+		+		+	+		+				
QUIC	+	+	+	+		+	+	+	+	+			+
tcpcrypt		+		+	+	+			+	+	+		
MinimaLT	+	+		+		+	+		+	+	+		+
CurveCP	+						+						+
IPsec	+	+			+	+	+	+	+	+	+	+	+
WireGuard	+					+	+	+					+
OpenVPN	+	+				+	+		+		+		

6. Взаимодействие с IANA

Этот документ не требует действий со стороны IANA.

7. Вопросы безопасности

В этом документе кратко описаны имеющиеся протоколы транспортной защиты и их интерфейсы. Документ не предлагает изменений или рекомендация по использованию этих протоколов. Не делается каких-либо заявлений о свойствах защиты или приватности сверх гарантируемых рассматриваемыми протоколами. Например, утечка метаданных через временные побочные каналы и анализ трафика могут поставить под угрозу любой из рассматриваемых здесь протоколов. Приложениям, использующим интерфейсы защиты, следует учитывать такие ограничения при выборе конкретной реализации протокола.

8. Вопросы приватности

Анализ влияния функций на снижение или рост защиты приватности намеренно не включён в документ. Все рассмотренные протоколы защиты в целом повышают уровень приватности за счёт шифрования для снижения утечек информации. Однако то или иное количество метаданных сохраняется каждым протоколом в открытом виде. Например, сертификаты клиентов и серверов передаются в открытом виде для TLS 1.2 [RFC5246], но шифруются в TLS 1.3 [RFC8446]. Обзор свойств приватности или их отсутствия может быть дан в отдельном документе.

9. Литература

[ALTS] Ghali, C., Stubblefield, A., Knapp, E., Li, J., Schmidt, B., and J. Boeuf, "Application Layer Transport Security", <https://cloud.google.com/security/encryption-in-transit/application-layer-transport-security/>.

[CurveCP] Bernstein, D., "CurveCP: Usable security for the Internet", <https://curvecp.org/>.

[DTLS-1.3] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-dtls13-38, 29 May 2020, <https://tools.ietf.org/html/draft-ietf-tls-dtls13-38>.

[MinimaLT] Petullo, W., Zhang, X., Solworth, J., Bernstein, D., and T. Lange, "MinimaLT: minimal-latency networking through better security", DOI 10.1145/2508859.2516737, <https://dl.acm.org/citation.cfm?id=2516737>.

[OpenVPN] OpenVPN, "OpenVPN cryptographic layer", <https://openvpn.net/community-resources/openvpn-cryptographic-layer/>.

[QUIC-TLS] Thomson, M. and S. Turner, "Using TLS to Secure QUIC", Work in Progress[2], Internet-Draft, draft-ietf-quic-tls-31, 24 September 2020, <https://tools.ietf.org/html/draft-ietf-quic-tls-31>.

[QUIC-TRANSPORT] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress[3], Internet-Draft, draft-ietf-quic-transport-31, 24 September 2020, <https://tools.ietf.org/html/draft-ietf-quic-transport-31>.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <https://www.rfc-editor.org/info/rfc2385>.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <https://www.rfc-editor.org/info/rfc2890>.

[1]Mobility Events.
[2]Опубликовано в RFC 9001. *Прим. перев.*
[3]Опубликовано в RFC 9000. *Прим. перев.*

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/info/rfc3748>.

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <https://www.rfc-editor.org/info/rfc4253>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>. [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, DOI 10.17487/RFC4555, June 2006, <https://www.rfc-editor.org/info/rfc4555>.

[RFC4571] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, DOI 10.17487/RFC4571, July 2006, <https://www.rfc-editor.org/info/rfc4571>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5641] McGill, N. and C. Pignataro, "Layer 2 Tunneling Protocol Version 3 (L2TPv3) Extended Circuit Status Values", RFC 5641, DOI 10.17487/RFC5641, August 2009, <https://www.rfc-editor.org/info/rfc5641>.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <https://www.rfc-editor.org/info/rfc5764>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC6189] Zimmermann, P., Johnston, A., Ed., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", RFC 6189, DOI 10.17487/RFC6189, April 2011, <https://www.rfc-editor.org/info/rfc6189>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc7301>.

[RFC7850] Nandakumar, S., "Registering Values of the SDP 'proto' Field for Transporting RTP Media over TCP under Various RTP Profiles", RFC 7850, DOI 10.17487/RFC7850, April 2016, <https://www.rfc-editor.org/info/rfc7850>.

[RFC8095] Fairhurst, G., Ed., Trammell, B., Ed., and M. Kuehlewind, Ed., "Services Provided by IETF Transport Protocols and Congestion Control Mechanisms", RFC 8095, DOI 10.17487/RFC8095, March 2017, <https://www.rfc-editor.org/info/rfc8095>.

[RFC8229] Pauly, T., Touati, S., and R. Mantha, "TCP Encapsulation of IKE and IPsec Packets", RFC 8229, DOI 10.17487/RFC8229, August 2017, <https://www.rfc-editor.org/info/rfc8229>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8547] Bittau, A., Giffin, D., Handley, M., Mazieres, D., and E. Smith, "TCP-ENO: Encryption Negotiation Option", RFC 8547, DOI 10.17487/RFC8547, May 2019, <https://www.rfc-editor.org/info/rfc8547>.

[RFC8548] Bittau, A., Giffin, D., Handley, M., Mazieres, D., Slack, Q., and E. Smith, "Cryptographic Protection of TCP Streams (tcpcrypt)", RFC 8548, DOI 10.17487/RFC8548, May 2019, <https://www.rfc-editor.org/info/rfc8548>.

[TAPS-ARCH] Pauly, T., Trammell, B., Brunstrom, A., Fairhurst, G., Perkins, C., Tiesel, P. S., and C. A. Wood, "An Architecture for Transport Services", Work in Progress, Internet-Draft, draft-ietf-taps-arch-08, 13 July 2020, <https://tools.ietf.org/html/draft-ietf-taps-arch-08>.

[TAPS-INTERFACE] Trammell, B., Welzl, M., Enghardt, T., Fairhurst, G., Kuehlewind, M., Perkins, C., Tiesel, P. S., Wood, C. A., and T. Pauly, "An Abstract Application Layer Interface to Transport Services", Work in Progress, Internet-Draft, draft-ietf-taps-interface-09, 27 July 2020, <https://tools.ietf.org/html/draft-ietf-taps-interface-09>.

[WireGuard] Donenfeld, J., "WireGuard: Next Generation Kernel Network Tunnel", <https://www.wireguard.com/papers/wireguard.pdf>.

Благодарности

Авторы благодарны Bob Bradley, Frederic Jacobs, Mirja Kühlewind, Yannick Sierra, Brian Trammell, Magnus Westerlund за их вклад и отклики на этот документ.

Адреса авторов

Theresa Enghardt

TU Berlin

Marchstr. 23

10587 Berlin

Germany

Email: ietf@tenghardt.net

Tommy Pauly

Apple Inc.

One Apple Park Way

Cupertino, California 95014

United States of America

Email: tpauly@apple.com

Colin Perkins

University of Glasgow

School of Computing Science

Glasgow

G12 8QQ

United Kingdom

Email: csp@csperkins.org

Kyle Rose

Akamai Technologies, Inc.

150 Broadway

Cambridge, MA 02144

United States of America

Email: krose@krose.org

Christopher A. Wood

Cloudflare

101 Townsend St

San Francisco,

United States of America

Email: caw@heapingbits.net

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru