УДК 004.728.5

Аюпова Айгуль Рафисовна, Научный руководитель, Нефтекамского филиала Уфимский Университет Науки и Технологии, г. Нефтекамск

Боковиков Сергей Антонович, Студент 4 курса Нефтекамского филиала Уфимский Университет Науки и Технологии, г. Нефтекамск

АНАЛИЗ ЗАЩИЩЁННОСТИ ПРОТОКОЛОВ TLS 1.3 И QUIC: СРАВНЕНИЕ УСТОЙЧИВОСТИ К ПАССИВНЫМ И АКТИВНЫМ АТАКАМ

Аннотация. В статье представлен сравнительный анализ защищённости двух ключевых протоколов безопасности современных сетей: TLS 1.3 и QUIC. Подробно рассмотрены архитектурные особенности протоколов, модель угроз и их устойчивость к пассивным и активным атакам. Особое внимание уделено таким аспектам, как шифрование транспортных метаданных, защита от анализа трафика, встроенные механизмы противодействия DoS-атакам и возможность бесшовной миграции соединений. Приведён качественный анализ применимости протоколов в различных сценариях (веб-сайты, мобильные приложения, IoT, корпоративные сети) и даны практические рекомендации по их выбору и настройке для обеспечения максимального уровня безопасности. Рассмотрены перспективы перехода на постквантовую криптографию в рамках обоих протоколов.

Ключевые слова: TLS 1.3, QUIC, HTTP/3, информационная безопасность, сетевые протоколы, криптография, анализ трафика, пассивные атаки, активные атаки, DDoS, 0-RTT, постквантовая криптография, PQC.

Abstract. This paper presents a comparative security analysis of two cornerstone protocols of modern network security: TLS 1.3 and QUIC. The architectural features of the protocols, the threat model, and their resilience to passive and active attacks are examined in detail. Particular attention is paid to aspects such as the encryption of transport metadata, protection against traffic analysis, built-in mechanisms for mitigating DoS attacks, and the ability to seamlessly migrate connections. A qualitative analysis of the applicability of the protocols in various scenarios (websites, mobile applications, IoT, corporate networks) is provided, along with practical recommendations for their selection and configuration to ensure maximum security. The prospects for transitioning to post-quantum cryptography within both protocols are considered.

Keywords: TLS 1.3, QUIC, HTTP/3, information security, network protocols, cryptography, traffic analysis, passive attacks, active attacks, DDoS, 0-RTT, post-quantum cryptography, PQC.

Обзор протоколов TLS 1.3 и QUIC

TLS 1.3 и QUIC — два тесно связанных протокола, которые сегодня обеспечивают большую часть защищённого трафика в интернете. Они используют одну и ту же криптографию, но подходят к задаче по-разному: TLS 1.3 — это «классический» слой безопасности поверх TCP, а QUIC — полностью новый транспорт поверх UDP, в который TLS 1.3 встроен изначально[1,2].

TLS 1.3 (RFC 8446)

Архитектура. Протокол состоит из двух частей: handshake (рукопожатие) и record layer (слой записей). Handshake отвечает за согласование параметров, аутентификацию и вывод ключей, а record layer защищает все последующие данные с помощью AEAD-шифрования[2,3].

Главные изменения по сравнению с TLS 1.2. Разработчики сильно упростили и усилили протокол: убрали всё устаревшее (статический RSA, CBC-режимы, компрессию, renegotiation), сделали обязательной forward secrecy через ephemeral (EC)DHE и сократили handshake до одного round-trip (1-RTT). Появились grease values для защиты от downgrade-атак и поддержка Encrypted ClientHello.

0-RTT (PSK) и риски. Для повторных подключений можно использовать pre-shared keys и отправлять «ранние данные» сразу в первом пакете. Это очень быстро, но у 0-RTT нет forward secrecy для этих данных, плюс они уязвимы к replay-атакам. Серверу приходится самостоятельно бороться с повторами (RFC 8446, раздел 8).

QUIC (RFC 9000, 9001, 9002)

QUIC — это транспортный протокол поверх UDP, в который криптография TLS 1.3 встроена нативно. Handshake происходит внутри транспортных пакетов (через CRYPTO-фреймы), а не отдельно, как в обычном TLS.

Ключевые особенности.

— Два вида заголовков: длинные (long header) для Initial, Handshake, 0-RTT и Retry; короткие (short header) — для обычных 1-RTT пакетов.

— Connection ID — специальный идентификатор соединения (до 20 байт), который не привязан к IP-адресу.

— Packet protection: всё содержимое пакета шифруется AEAD, а часть заголовка (включая packet number) дополнительно маскируется (header protection).

— Поддержка 0-RTT и 1-RTT, как в TLS 1.3.

— Connection migration — можно менять IP-адрес (например, с Wi-Fi на мобильный) без разрыва сессии.

— Anti-amplification: сервер в начале не может отправить больше данных, чем в три раза превышает полученное от клиента, пока не убедится в реальности адреса.

Сходства и основные различия

Оба протокола используют одни и те же криптографические примитивы TLS 1.3 (AEAD, HKDF, (EC)DHE). У них есть 0-RTT, forward secrecy и защита от downgrade.

Главные различия:

— TLS 1.3 работает поверх надёжного TCP, QUIC — напрямую по UDP.

— В TLS handshake отдельный, в QUIC он слит с транспортным уровнем.

— QUIC шифрует почти всё с первого пакета и даёт меньше метаданных для анализа.

— QUIC из коробки поддерживает миграцию соединения и лучше борется с потерями пакетов.

В итоге TLS 1.3 — проверенный и универсальный вариант, а QUIC — более современный и «лёгкий» протокол, оптимизированный для скорости и мобильности. Именно эти архитектурные различия сильно влияют на их устойчивость к атакам, о чём пойдёт речь дальше[3].

Модель угроз и классификация атак

Чтобы честно сравнить, насколько TLS 1.3 и QUIC устойчивы к атакам, нужно сначала понять, против кого мы их защищаем. Для этого в криптографии и сетевой безопасности используют классическую **модель угроз Dolev–Yao** (RFC 8446 и RFC 9000 прямо на неё опираются).

В этой модели атакующий — это «активный противник», который полностью контролирует сеть: он может прослушивать все пакеты, изменять их, удалять, подделывать, переупорядочивать и даже создавать новые. Единственное ограничение — он не может взломать саму криптографию (например, подобрать ключ или подделать цифровую подпись без знания секретного ключа). Атакующего делят на два типа:

- Пассивный — просто наблюдает за трафиком, ничего не меняет.
- Активный — вмешивается в соединение (Man-in-the-Middle).

Пассивные атаки

Это атаки «подслушивания» без вмешательства в процесс.

Eavesdropping (простое прослушивание) — попытка прочитать содержимое пакетов. Оба протокола защищают данные AEAD-шифрованием (AES-GCM или ChaCha20-Poly1305), поэтому после handshake данные в принципе недоступны.

Анализ трафика (traffic analysis) — изучение метаданных: размеры пакетов, тайминги, последовательность, fingerprinting соединения. В TLS 1.3 остаётся видимым TCP-заголовок и часть handshake, в QUIC почти всё зашифровано уже с первого пакета (header protection), поэтому анализ сложнее[3,4,5].

Активные атаки

Здесь противник уже «в деле»:

— Man-in-the-Middle (MITM) — перехват и подмена трафика. Оба протокола используют сертификаты и ephemeral ключи, чтобы предотвратить это.

— Downgrade-атаки — заставить клиента/сервер перейти на более старую, слабую версию протокола.

— Replay-атаки — повторное воспроизведение перехваченных пакетов (особенно опасно для 0-RTT).

— Injection и packet reordering — вставка или переупорядочивание пакетов.

— Amplification-атаки — заставить сервер отправить жертве в разы больше трафика, чем пришло от атакующего.

— DoS-атаки — handshake flooding (затопление сервера handshake-запросами), connection reset, stateless reset spoofing.

— Атаки на 0-RTT и session resumption — использование ранних данных для replay или инъекции до завершения полного handshake.

Именно эти классы атак мы и будем разбирать дальше применительно к TLS 1.3 и QUIC. В следующих разделах посмотрим, насколько хорошо каждый

протокол справляется с пассивным и активным противником и где у него остаются слабые места.

Устойчивость TLS 1.3 к пассивным атакам

TLS 1.3 отлично справляется с пассивными атаками — то есть с простым прослушиванием трафика без вмешательства. После завершения handshake данные надёжно защищены, и пассивный наблюдатель (даже с полным доступом к каналу) не может их прочитать. Вот как это работает на практике.

Конфиденциальность и целостность данных. Все прикладные данные передаются через record layer с помощью AEAD-шифров (AES-GCM, ChaCha20-Poly1305 или AES-CCM). Каждый пакет зашифрован и имеет криптографическую метку аутентификации — изменить или подделать данные незаметно невозможно. Благодаря обязательному forward secrecy (ephemeral (EC)DHE) даже если позже взломают долгосрочный ключ сервера, прошлые сессии останутся в безопасности. RFC 8446 прямо говорит: после установки соединения данные видны только конечным точкам.

Защита handshake. В TLS 1.3 handshake сильно зашифрован по сравнению с предыдущими версиями. Первые два сообщения (ClientHello и ServerHello) идут открыто, но дальше все handshake-сообщения (EncryptedExtensions, Certificate, Finished) уже защищены handshake-ключами. Это значит, что пассивный атакующий видит только базовую информацию о поддерживаемых алгоритмах, но не полную цепочку сертификатов и выбранные параметры[4].

Дополнительно есть расширение Encrypted ClientHello (ECH), которое полностью скрывает SNI (имя сервера) и другие чувствительные поля уже в первом сообщении. К 2026 году ECH активно внедряется крупными провайдерами (Cloudflare, Google), что сильно усложняет пассивный fingerprinting.

Что остаётся слабым местом. TLS 1.3 не прячет всё. Как отмечается в RFC 8446 (раздел о traffic analysis), протокол уязвим к анализу по длине и таймингам пакетов. Видны:

— 5-байтовый заголовок записи (тип, версия, длина);

— размеры пакетов (можно пытаться угадать, какие данные передаются);

— время между пакетами.

Хотя сервер и клиент могут добавлять padding (заполнение) для маскировки длины, это не всегда используется на практике и не спасает от продвинутого анализа трафика (website fingerprinting). TCP-заголовки тоже остаются открытыми, в отличие от QUIC.

В целом TLS 1.3 даёт очень сильную защиту от пассивного прослушивания содержимого, но метаданные (кто с кем общается, сколько и когда) всё ещё могут утечь. Именно поэтому QUIC в следующих разделах выглядит интереснее — он прячет намного больше уже на уровне пакетов[5].

Устойчивость QUIC к пассивным атакам

В отличие от TCP, где часть метаданных передается в открытом виде, QUIC изначально проектировался с прицелом на минимизацию утечек, доступных пассивному наблюдателю. Это достигается комплексом архитектурных решений, направленных на шифрование максимального объема информации.

Шифрование почти всех полей заголовка (Header Protection): В TCP+TLS для наблюдателя остаются видны флаги и порядковый номер пакета. QUIC шифрует большую часть своего заголовка, включая такие поля, как Stream ID и Connection State Information, которые могут раскрывать детали сессии. Для защиты этих полей используется механизм Header Protection, который накладывает маску на определенные части заголовка, делая их нечитаемыми без знания ключей шифрования.

Packet Number Encryption: Номера пакетов QUIC (Packet Number) также шифруются. Это решает сразу две задачи: предотвращает "окостенение" протокола (когда промежуточное оборудование начинает полагаться на их значения) и, что важнее в контексте пассивных атак, не позволяет злоумышленнику легко отслеживать последовательность и коррелировать пакеты при смене клиентом IP-адреса. Это напрямую повышает устойчивость к анализу трафика[5,6].

Более низкая наблюдаемость по сравнению с TLS over TCP: поскольку QUIC является полностью зашифрованным протоколом, он предоставляет гораздо меньше открытых метаданных, чем связка TCP + TLS. Пассивный мониторинг сети сильно затруднен, так как большая часть информации о состоянии соединения скрыта. В сочетании с такими технологиями, как TLS Encrypted Client Hello (ECH), QUIC еще больше повышает приватность, скрывая информацию о целевом сервере в начальном рукопожатии.

Анализ трафика и миграция соединений: несмотря на все меры, QUIC не полностью неуязвим для сложного анализа трафика. Потенциальной уязвимостью является Connection ID (CID). Хотя CID предназначен для того, чтобы избежать привязки к IP-адресам, его долгое использование может позволить связать активность одного и того же клиента на разных сетях. Для противодействия этому рекомендуется частая смена CID, особенно при миграции соединения. Утечки могут возникать и через специфичные для реализаций особенности, например, через маркеры (source-address token), которые могут использоваться для отслеживания пользователей[6].

Устойчивость QUIC к активным атакам

QUIC содержит несколько встроенных механизмов, специально предназначенных для противодействия активным атакам, от DoS до подмены IP.

Встроенная защита handshake (QUIC-TLS integration): В отличие от раздельных TCP и TLS, QUIC тесно интегрирует TLS 1.3 прямо в свой

транспортный слой. Это означает, что криптографическое рукопожатие, аутентификация и согласование параметров происходят одновременно с установлением транспортного соединения. Такая интеграция не только снижает задержку, но и уменьшает поверхность для атак, делая невозможным раздельное воздействие на транспортный и криптографический уровни.

Stateless Reset, Retry и Anti-amplification: для защиты от DoS-атак QUIC использует несколько ключевых механизмов:

— Retry Packet: Сервер может ответить на Initial Packet от неподтвержденного клиента пакетом Retry. Это заставляет клиента подтвердить свою доступность по указанному адресу, отсекая атаки с поддельных IP-адресов.

— Stateless Reset: позволяет серверу завершить соединение, состояние которого он потерял (например, после перезагрузки), без необходимости хранить информацию о нем. Это предотвращает утечку ресурсов на обработку "мертвых" сессий.

— Anti-Amplification Limit: это критически важная защита от усиления DDoS-атак. Сервер ограничивает объем данных, отправляемых клиенту, до тех пор, пока не подтвердит, что клиент действительно может принимать пакеты по заявленному IP-адресу. Обычно ответ сервера не может превышать размер полученного запроса более чем в три раза. Это правило также применяется к Stateless Reset, где размер ответа ограничен, чтобы сам по себе не стать инструментом усиления.

Connection migration и защита от IP-spoofing: QUIC позволяет клиенту менять IP-адрес без разрыва соединения (например, при переходе с Wi-Fi на мобильную сеть). Механизм Connection Migration защищен от атак с подменой IP-адреса (IP spoofing). Сервер не начнет активно слать трафик на новый адрес до тех пор, пока не получит с него корректный, криптографически защищенный пакет. Это предотвращает атаки, при которых злоумышленник может попытаться перенаправить чужой трафик на жертву, вызвав ее перегрузку [7].

0-RTT в QUIC: отличия от TLS 1.3: Оба протокола поддерживают 0-RTT для быстрого возобновления сессии, но подход QUIC считается более строгим. В QUIC сервер выдает клиенту специальные токены, которые тот должен предъявить для выполнения 0-RTT. Это позволяет серверу проверять валидность запроса, даже если его состояние сессии было утеряно. Хотя полная защита от replay-атак (повторного воспроизведения данных) на уровне протокола невозможна, QUIC предоставляет серверным приложениям больше контекста для принятия решений, снижая риски по сравнению с базовой реализацией в TLS 1.3. Приложениям рекомендуется проектировать обработчики 0-RTT-запросов как идемпотентные.

Защита от packet injection и reordering: QUIC использует строго возрастающие 64-битные номера пакетов (Packet Number), что делает попытки вставки (injection) или удаления пакетов легко обнаруживаемыми. Любое нарушение в последовательности приводит к разрыву соединения или, по крайней мере, к игнорированию поддельных пакетов [7,8].

Известные атаки и их нейтрализация: несмотря на надежную архитектуру, исследователи постоянно находят новые векторы атак, в основном на уровне конкретных реализаций, а не самого протокола.

Optimistic ACK (OACK): Атака, при которой клиент ложно подтверждает получение пакетов, чтобы заставить сервер увеличить скорость отправки и вызвать перегрузку его сети. Современные реализации QUIC внедряют сложные эвристики для обнаружения таких "оптимистичных" подтверждений и ограничения скорости.

Атаки на реализацию: В 2024-2025 годах было обнаружено несколько уязвимостей в популярных библиотеках. Например, атака "Hash DoS" могла вызывать чрезмерное потребление ресурсов CPU в xquic и других библиотеках. Уязвимость "QUIC-LEAK" (CVE-2025-54939) в LSQUIC позволяла удаленно исчерпать память сервера еще до завершения рукопожатия. Также была найдена уязвимость в quiche, связанная с бесконечным циклом при обработке кадров RETIRE_CONNECTION_ID. Все

эти проблемы подчеркивают необходимость своевременного обновления используемых библиотек [8].

Сравнительный анализ (центральный раздел)

В этом разделе сведены ключевые характеристики обоих протоколов для наглядного сравнения их безопасности и архитектурных особенностей.

Таблица 1.

Сравнения по ключевым критериям

Критерий	TLS 1.3 (over TCP)	QUIC	Победитель / Примечание
Конфиденциальность	Полное шифрование данных приложения.	Шифрование данных и большей части транспортных метаданных (включая номера пакетов).	QUIC
Traffic analysis resistance	Низкая. Заголовки TCP (SEQ, ACK, флаги) видны.	Высокая. Заголовки зашифрованы, но утечки возможны через CID и время.	QUIC
MITM / Downgrade	Защищено протоколом. Downgrade до	Встроенная интеграция с TLS 1.3. Более старые	QUIC

Критерий	TLS 1.3 (over TCP)	QUIC	Победитель / Примечание
	TLS 1.2 возможен (если разрешен).	версии не поддерживаются.	
0-RTT replay protection	Защита на основе ClientHello и early_data.	Token-based защита. Более гибкая и строгая модель для сервера.	QUIC
DoS resilience	SYN Cookies (защита от SYN-flood).	Retry, Stateless Reset, Anti-amplification limit.	QUIC
Connection migration	Отсутствует (разрыв при смене IP).	Встроенная поддержка с криптографической проверкой нового адреса.	QUIC
Head-of-Line Blocking	Есть на уровне TCP (страдает весь поток).	Отсутствует на транспортном уровне (независимые потоки).	QUIC
Middlebox interference	Высокая. Многие файрволы "следят" за TCP-соединением.	Низкая. Устойчивость к "окостенению" протокола.	QUIC

Критерий	TLS 1.3 (over TCP)	QUIC	Победитель / Примечание
Сложность реализации в userspace	Низкая. Используется проверенная ОС-реализация TCP.	Высокая. Требует реализации логики контроля перегрузки и потока в приложении.	TLS 1.3 (TCP)
Производительность в lossy-сетях	Страдает из-за Head-of-Line Blocking в TCP.	Лучше. Независимые потоки позволяют не ждать переотправки потерянных пакетов других потоков.	QUIC

Качественное сравнение по сценариям:

— Веб-сайты: здесь преимущество QUIC в снижении задержек и решении проблемы Head-of-Line Blocking дает заметный прирост в скорости загрузки, особенно для ресурсоемких страниц. Это основная движущая сила внедрения HTTP/3.

— Мобильные приложения: QUIC здесь является бесспорным лидером благодаря Connection Migration. Пользователи перестают замечать смену сети (Wi-Fi на 4G/5G), так как соединения с сервером не рвутся. Это критически важно для видеозвонков, стриминга и онлайн-игр.

— IoT (Интернет вещей): для устройств с ограниченными ресурсами и в условиях нестабильной связи QUIC может быть слишком "тяжелым". TLS 1.3 поверх TCP (или даже DTLS 1.3) часто является более прагматичным выбором. Однако для IoT-устройств, которым важна мобильность и быстрая реакция, QUIC может быть оправдан.

— Корпоративные сети: здесь выбор не так однозначен. С одной стороны, QUIC усложняет инспекцию трафика (DPI) из-за более высокого уровня шифрования, что является минусом для служб безопасности. С другой стороны, для сотрудников в удаленном доступе или в командировках его преимущества в виде стабильности соединения и скорости могут перевесить. Часто администраторы сознательно блокируют QUIC, чтобы сохранить возможность анализа трафика через прокси[8].

Практические рекомендации и лучшие практики

Выбор протокола и его настройка должны исходить из конкретных потребностей бизнеса и допустимого уровня риска.

Когда выбирать TLS 1.3 (TCP), а когда QUIC (HTTP/3):

— Выбирайте TLS 1.3 (HTTP/2), если: вам критически важна инспекция трафика (например, в корпоративном прокси), вы работаете в среде с очень стабильным соединением (внутренняя сеть дата-центра), или ваше приложение крайне чувствительно к потреблению CPU на стороне сервера.

— Выбирайте QUIC (HTTP/3), если: ваши пользователи преимущественно мобильные, вы стремитесь к максимальной производительности веб-приложения, хотите минимизировать задержки и обеспечить бесшовный пользовательский опыт при смене сети.

Настройки серверов для максимальной защиты:

— Nginx: для включения QUIC/HTTP/3 используйте директиву listen 443 quic reuseport; и добавьте заголовок Alt-Svc: h3=":443"; ma=86400 в ответах HTTPS. Также важно использовать надежные TLS 1.3 шифры (ssl_ciphers) и отключать устаревшие протоколы.

— Cloudflare: Включение HTTP/3 делается в один клик в разделе Speed> Optimization> Protocol Optimization дашборда. Для более продвинутых пользователей, Cloudflare поддерживает постквантовые алгоритмы (Kyber), которые можно протестировать, используя Firefox 128+ с флагом network.http.http3.enable_kyber.

— Общие рекомендации: регулярно обновляйте используемые QUIC-библиотеки (например, quiche, lsquic, quic-go). Многие недавние уязвимости, такие как Hash DoS или QUIC-LEAK, были исправлены именно в новых версиях.

Мониторинг и обнаружение атак:

— Инструменты: Основным инструментом для анализа трафика остается Wireshark. Для расшифровки QUIC-пакетов в него необходимо загрузить TLS-ключи сессии (через keylog-файл). Для более глубокого анализа метрик соединения и поиска аномалий полезны утилиты из набора qvis, которые визуализируют qlog-файлы, генерируемые многими реализациями QUIC.

— Обнаружение аномалий: в условиях почти полного шифрования, классические сигнатурные методы DPI неэффективны. Фокус смещается в сторону поведенческого анализа и ML-моделей, которые отслеживают аномалии в длительности соединений, размерах пакетов, частоте смены CID и паттернах миграции [9].

Переход на постквантовую криптографию (PQC):

— Стандарты: NIST уже стандартизировал первые квантово-устойчивые алгоритмы: ML-KEM (для обмена ключами) и ML-DSA (для цифровых подписей).

— Внедрение в протоколы: Оба протокола, TLS 1.3 и QUIC, активно готовятся к внедрению PQC. TLS 1.3 имеет расширения для согласования постквантовых алгоритмов в ходе рукопожатия. В QUIC также активно исследуется и тестируется интеграция PQC, в том числе в гибридном режиме, когда классические и постквантовые алгоритмы используются совместно для обеспечения безопасности на переходный период. Крупные игроки, такие как Google (в Chrome) и Cloudflare, уже проводят реальные эксперименты по внедрению PQC в свои сервисы. Однако стоит учитывать, что использование PQC, особенно ML-DSA, может привести к увеличению накладных расходов

на рукопожатие и повышению потребления CPU, что является предметом активных исследований.

Литература:

1. Козырь, Н. С. Анализ и оценка рисков информационной безопасности: учебник для среднего профессионального образования / Н. С. Козырь, В. Н. Хализев. — Москва: Издательство Юрайт, 2026. — 157 с. — (Профессиональное образование). — ISBN 978-5-534-20645-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/590435 (дата обращения: 20.04.2026).
2. HTTP/3: улучшения производительности. [Электронный ресурс]. URL: https://habr.com/ru/companies/slurm/articles/583434/ (дата обращения 10.03.2026).
3. DPI видит всё: шифрование больше не спасает. [Электронный ресурс]. URL: https://www.securitylab.ru/analytics/560399.php (дата обращения 10.03.2026).
4. Анализ уязвимостей в современных протоколах TLS и методы их предотвращения. [Электронный ресурс]. URL: https://iprog.pp.ru/analiz-uyazvimostey-v-sovremennyh-protokolah-tls-i-metody-ih-predotvrascheniya/ (дата обращения 10.03.2026).
5. Рескорла Э. Протокол транспортного уровня безопасности (TLS) версии 1.3. RFC 8446, август 2018 г.
6. HTTPS-инспекция - TLS 1.3, HSTS, QUIC, DOH. [Электронный ресурс]. URL: https://ideco.ru/prikaz_fstek_117_i_metoddokument/tpost/18xo611idr-https-inspektsiya-tls-13-hsts-quic-doh (дата обращения 10.03.2026).
7. HTTP/3 и QUIC: как это работает? [Электронный ресурс]. URL: https://habr.com/ru/companies/otus/articles/725902/ (дата обращения 10.03.2026).

8. TLS-протокол: что это такое и как работает. [Электронный ресурс]. URL: https://thecode.media/tls-protokol/ (дата обращения 10.03.2026).

9. Secure Communication Channel Establishment: TLS 1.3 (over TCP Fast Open) versus QUIC. [Электронный ресурс]. URL: https://iprog.pp.ru/analiz-uyazvimostey-v-sovremennyh-protokolah-tls-i-metody-ih-predotvrascheniya https://www.researchgate.net/publication/351829553_Secure_Communication_Channel_Establishment_TLS_13_over_TCP_Fast_Open_versus_QUIC/ (дата обращения 10.03.2026).