Evolución y desarrollo de los protocolos criptográficos SSL/TLS para la seguridad web (1994–2025)

Autor
Jefferson Rodríguez Villegas

Ingeniero de sistemas

Director
Edgar Alirio Restrepo Castillo
Víctor Viera Balanta

Facultad de Ingeniería
Ingeniería de Sistemas
Universidad Santiago de Cali
Santiago de Cali - Colombia
2026

Evolución y desarrollo de los protocolos Criptográficos SSL/TLS para la seguridad web 1994–2025

Jefferson Rodríguez Villegas
jefferson.rodriguez01@usc.edu.co

Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería de Sistemas (1)

Resumen

En la época actual un gran porcentaje de la población se comunica por medio de la internet, la cual envía y transporta incalculables cantidades de datos al día que toman un papel vital y estratégico para las personas y las empresa, los cuales están expuestos a múltiples amenazas que varían desde la intervención y modificación por parte de terceros , hasta la destrucción de los mismo .En este estudio se busca presentar el origen y características de los protocolos SSL y TLS , así como exponer algunas de los cambios y actualizaciones que han ofrecido a lo largo de sus respectivos periodos de vida útil , y por ultimo presentar algunas de sus vulnerabilidades más conocidas y una preceptiva de su futuro frente al nuevo entorno de la computación cuántica.

Palabras Clave: Seguridad de la capa de transporte, Secure Sockets Layer, seguridad web, vulnerabilidades, criptografía, criptografía post-cuántica, KEM, ML-KEM

Abstract

In the current era, a large percentage of the population communicates via the internet, which transmits and processes countless amounts of data daily. This data plays a vital and strategic role for both individuals and businesses, who are exposed to various threats ranging from unauthorized intervention and modification by third parties to outright destruction. This study aims to present the origins and characteristics of SSL and TLS protocols, as well as to outline some of the changes and updates they have undergone over their respective lifespans. Finally, it will discuss some of their most well-known vulnerabilities and provide a perspective on their future in the context of emerging quantum computing.

Keywords: Transport layer security, Secure Sockets Layer, web security, Vulnerabilities, Cryptography, Post-quantum cryptography, KEM, ML-KEM

1. INTRODUCCIÓN

Desde la creación y adopción de la web, esta ha desempeñado un papel fundamental en la vida moderna, transformando la forma en que las personas se comunican, trabajan y realizan transacciones. Actualmente, una gran cantidad de actividades cotidianas —desde la interacción social hasta las operaciones bancarias— se desarrollan en entornos digitales que dependen de la transmisión constante de información a través de Internet. En este contexto, la protección de los datos personales y corporativos se ha convertido en un requisito esencial para garantizar la seguridad y la confianza de los usuarios.

SSL(Secure Sockets Layer),es un protocolo diseñado para ofrecer una ruta segura para transmitir información a través de Internet, satisface este requisito a través de procedimientos de cifrado y autenticación.(Devi et al., 2020), fue uno de los primeros protocolos criptográficos y sufrió varias actualizaciones, incluso protocolos usaron su estructura como base para mejorar y corregir errores un ejemplo es TLS (Transport Layer Security). TLS proporciona un canal seguro entre un cliente y un servidor a través de Internet. El canal seguro proporciona autenticación del servidor (y opcionalmente del cliente), confidencialidad e integridad de los mensajes en tránsito(Perugini & Vesco, 2024).

Para la seguridad web los protocolos SSL/TLS se emplean como estándares gracias a su infraestructura de claves públicas y protocolos, para cualquier página web o aplicación, no obstante, a lo largo de su evolución, estos protocolos han enfrentado múltiples vulnerabilidades que han comprometido la información de los usuarios. A pesar de los avances introducidos en versiones más recientes, las amenazas persisten y se espera que nuevos desafíos surjan con el desarrollo de la computación cuántica.

Somos calidad, somos USC

La computación cuántica representa un cambio de paradigma en la tecnología computacional. Los ordenadores cuánticos emplean bits cuánticos, o qubits, que pueden existir simultáneamente en múltiples estados. Esta capacidad única, conocida como paralelismo cuántico, permite procesar una gran cantidad de posibilidades al mismo tiempo, lo que plantea desafíos significativos para los esquemas criptográficos actuales(Kundu et al., 2025).

Con todo lo anteriormente expuesto está en duda si los protocolos SSL/TLS aun serán relevantes en el futuro, por tanto en este trabajo de investigación se buscara indagar en la evolución de los protocolos y evidenciar como han cambiado, que vulnerabilidades han presentado y si se está estudiando tecnologías que se puedan utilizar para poder enfrentar la computación cuántica y permanecer aun como pilares en la cuestión de seguridad, o de lo contrario está destinado a ser reemplazado por algún otro elemento.

2. MATERIALES Y MÉTODOS/METODOLOGÍA

Para este estudio que esta centrado en una revisión descriptiva se buscan referencias factibles y veraces de índole académico y científico, esto se llevara a cabo por medio de los distintos recursos virtuales que la biblioteca digital de la universidad brinda a los estudiantes como ayuda para llevar a cabo consultas de las diferentes fuentes de información y se seguirán una series de pasos para encontrar los documento que emplearan en este documentos así como las condiciones para excluir un documento .

2.1 Fase 1: preguntas de investigación

Se definieron con claridad los objetivos del estudio, en este artículo específicamente son analizar la evolución de los protocolos criptográficos SSL/TLS, identificar vulnerabilidades relevantes, evaluar el impacto de la computación cuántica en su seguridad, e indagar si se han realizado algún estudio frente a su mitigación

2.2 Fase2: definir bases datos

Se utilizaran para la búsqueda bases de datos proporcionadas por la universidad en las cuales se encuentra IEEE Xplore, SpringerLink, ScienceDirect y Scopus , cabe mencionar que Google Scholar también se usara en caso de que no se pueda encontrar el texto por su bases de datos original o en caso de que haya un artículo bastante relevante

2.3 Fase 3: Criterios de búsqueda

Para buscar documentos pertinentes para el articulo se determinan los siguientes criterios de búsqueda

- Búsqueda por palabras claves como Seguridad de la capa de transporte, tls, Secure Sockets Layer,ssl, seguridad web, vulnerabilidades, criptografía, criptografía post-cuántica etc.., se concatenaran con operadores como AND Y OR, (ejemplo: "Secure Sockets Layer" OR SSL OR "Transport Layer Security" OR TLS) AND (vulnerab)),
- El año de publicación debe estar entre el rango de 2020 a 2025(habrá excepciones en cuanto conceptos de protocolos antiguos como ssl 1.0).

- El contenido del título y el abstract(resumen) deben estarán relacionados con alguna de las preguntas de investigación
- Los idiomas de preferencia deberán ser inglés o español (se aceptan otros idiomas si son relevantes)

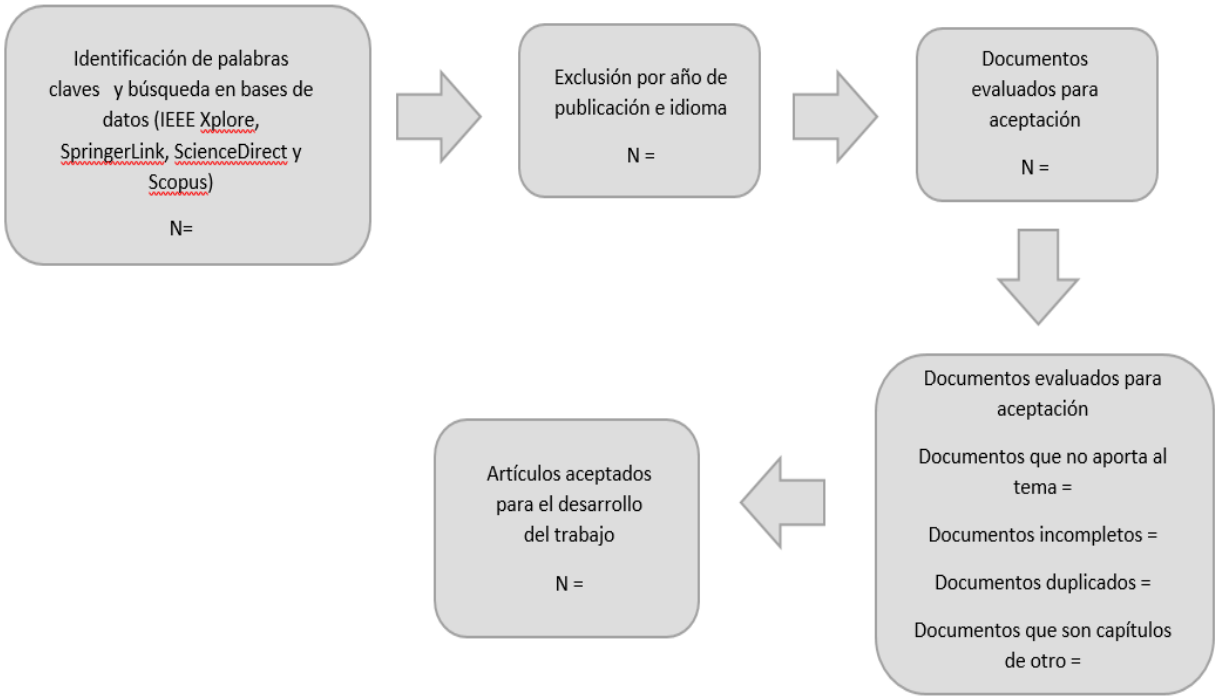
2.4 Fase 4: criterios de aceptación

Se establecen los criterios para aceptar a un documento en el artículo y poder citarlo

- El contenido del documento debe ser relevantes para la investigación y el texto debe estar completo para poder citar
- El documento no puede estar duplicado.

Teniendo en cuenta los pasos se crea la siguiente la figura (Figura 1), la cual deberá ser diligenciada para cumplir con la metodología

Figura 1: seguimiento de pasos

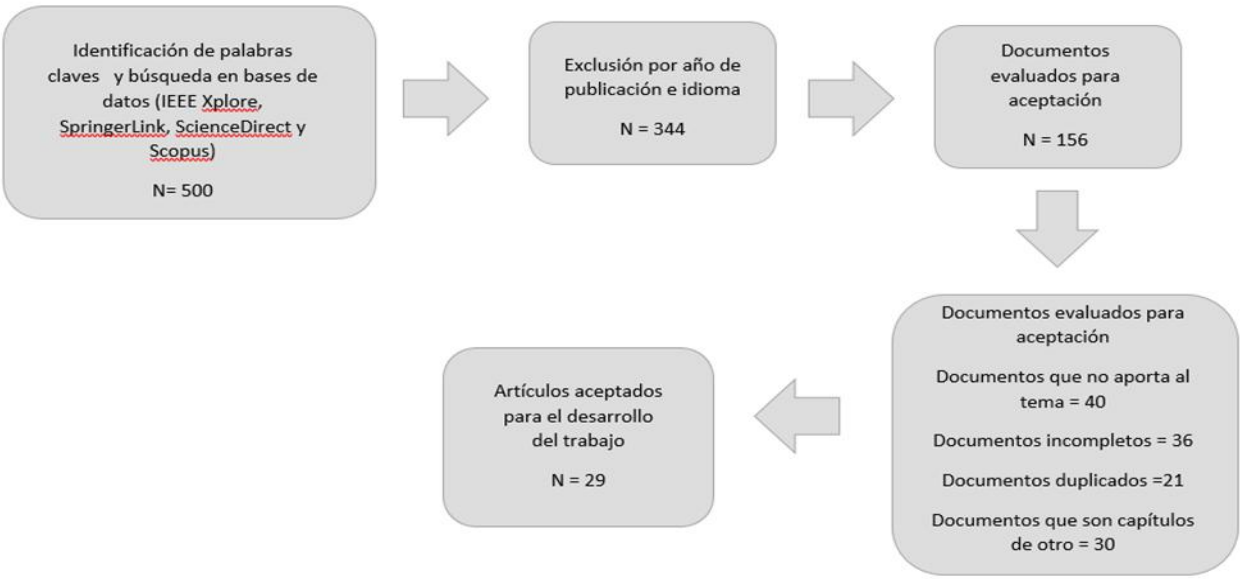


Fuente: propia

3. RESULTADOS

El presente trabajo que sigue lo estipulado en los puntos de la metodología obtiene como resultado la siguiente figura (Figura 2), donde se expresa la cantidad de documentos que sustentaran lo explicado en el mismo.

Figura 2: selección y aceptación de documentos



Fuente: propio

3.1 Protocolo SSL

Los primeros días de internet fueron un momento de experimentación en las cuales se desarrollaron e implementaron nuevas tecnologías debido a las tendencias de ese entonces, una de ellas fue el denominado triada de la seguridad. según (Chuvakin & Williams, 2010):

1. Confidencialidad: Se esfuerza por garantizar que la información se divulgue solo a aquellos que están autorizados a verla.
2. Integridad: Se esfuerza por garantizar que la información no se modifique de maneras no autorizadas. Esto puede referirse a modificaciones realizadas tanto por personas como por procesos
3. Disponibilidad: Se esfuerza por garantizar que los datos estén disponibles para las partes autorizadas de manera confiable y oportuna.

por tanto, En 1994, Netscape creó la solución, Secure Socket Layer (SSL), que ofreció este nivel de seguridad al cliente y a la organización (Devi et al., 2020). El SSL se encuentra por debajo de la capa de Aplicación en el Modelo OSI. Con características de seguridad integradas, como la protección de información sensible que será manejada por los protocolos de la capa de aplicación(Ibrahim, 2024).Nunca se publicó debido a importantes fallos de seguridad. Sin embargo, la versión 2.0 se publicó en 1995 y empezó a utilizarse ampliamente. Pero se encontraron fallos de seguridad en ella y posteriormente fue suplantada por la versión 3.0 de SSL en 1996(Easttom, 2022).

SSL 3.0 noto una mejoría con respecto a su antigua versión en gran medida agracias a los protocolos asociados que le

permiten mantener una mayor capacidad de autenticación por parte de las entidades involucradas en el proceso de intercambio de datos.

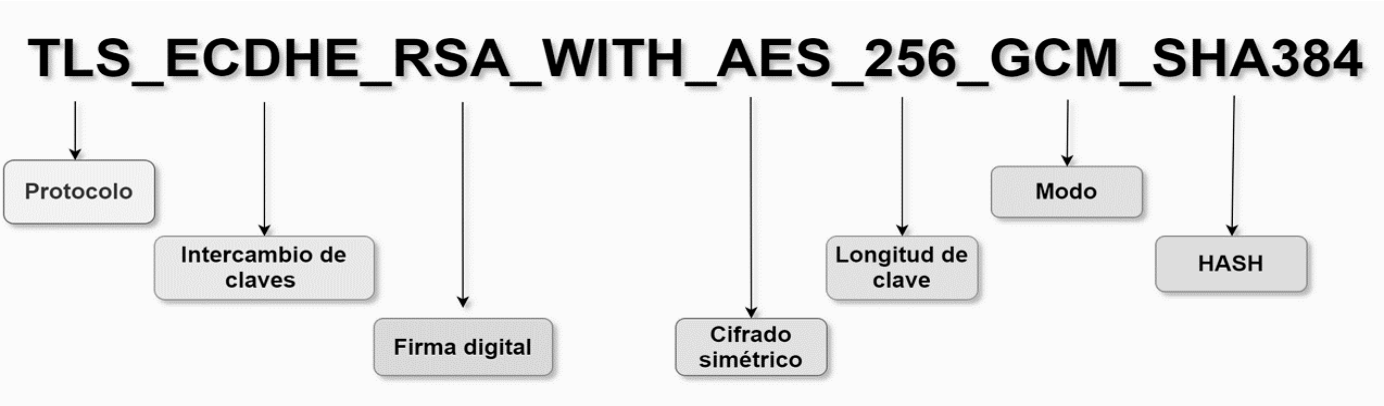
3.1.1 Protocolo Handshake SSL /apretón de manos SSL

Es el primer protocolo que entra en acción después de que la conexión es establecida por el protocolo de capa de transporte(Satapathy & Livingston, 2016).Permite que las entidades (usuarios y servidor) entablen una negociación en la cual se intentan verificar la identidad de las partes involucrado y establecer un canal seguro para la información, este protocolo establece una serie de pasos que los involucrados deben seguir, según(Devi et al., 2020):

4. El cliente envía una señal al servidor de contacto llamada 'CLIENT HELLO'. Este mensaje incluye el número de versión de SSL que es compatible con los consumidores, junto con un número aleatorio de 32 bytes. Junto con las suites de cifrado y el método de compresión del consumidor, este mensaje incluye.
5. El servidor ahora envía un 'SERVER HELLO' al cliente. Este mensaje incluye el número de versión de SSL, un número aleatorio de 32 bytes, el ID de sesión y, por lo tanto, la suite de cifrado y, en consecuencia, el proceso de compresión que admite.
6. Nuevamente, el servidor envía un mensaje 'SERVER KEY EXCHANGE' al cliente. Por ejemplo, la clave pública general para RSA se incluye en este mensaje. Luego, el servidor solicita al cliente la información del certificado para demostrar al cliente si son uno.
7. Una vez que el cliente ha recibido la información, el servidor envía un 'SERVER HELLO DONE' al cliente indicando que las negociaciones originales en el servidor están completas.
8. El cliente puede ahora enviar su información principal al servidor utilizando un mensaje cifrado por clave pública del servidor llamado 'CLIENT KEY EXCHANGE' para que solo el servidor legítimo pueda acceder a los datos del cliente.
9. El cliente envía el 'CHANGE CIPHER SPEC' al servidor para recibir todos los parámetros de enlace cifrados y activar los mismos, ya que el cliente y el servidor envían su información de clave y configuraciones alternativas.
10. Luego, el cliente envía el mensaje 'FINISHED' completado al servidor para confirmar las opciones recientemente aprobadas.
11. El servidor envía un mensaje 'CHANGE CIPHER SPEC' al cliente para reconocer toda la funcionalidad dentro de las conexiones cifradas, de modo que se envíe el mensaje final 'FINISHED' al cliente para verificar todas las opciones.

En todo este proceso se ven involucrado diferentes elementos entre ellos el denominado conjunto de cifrado (cipher suite)(Figura 3)

Figura 3: cipher suite



Fuente:(Romero Benito, 2020)

En la (Figura 3), se encuentran contenido los siguientes elementos, que según(Romero Benito, 2020) son:

1. Protocolo (Protocol): define el protocolo que se utilizará de manera genérica. Será SSL o TLS únicamente
2. Intercambio de claves (Key Exchange): define el algoritmo de intercambio de claves que se utilizará para llegar a la creación de una clave simétrica compartida para la sesión.
3. Firma digital (Digital Signature): define el algoritmo que se utilizará para la validación y autenticación de los datos intercambiados entre cliente y servidor.
4. Algoritmo de cifrado simétrico y longitud de clave (Cipher and key length): estos parámetros indican el algoritmo de cifrado simétrico a utilizar, así como la longitud de clave ha utilizar del mismo.
5. Modo de cifrado (Encryption Mode): este parámetro (opcional) indica el modo en el que funciona el algoritmo de cifrado simétrico.
6. Algoritmo de Hash (Hashing Algorithm): este parámetro indica el algoritmo utilizado para la validación de la integridad de los mensajes para garantizar que no han sufrido alteraciones de ningún tipo.

3.1.2 Protocolo de Registro SSL

El protocolo de capa de registro se utiliza para cifrar los datos que se envían a través de la red utilizando la clave establecida durante el protocolo de apretón de manos. Esta capa maneja los datos reales. Recibe datos de la capa de aplicación, los cifra, los fragmenta a un tamaño apropiado, según lo determine el algoritmo, y los envía a la capa de transporte(Alwazzeh et al., 2020).

El SSL funciona autenticando a los clientes y servidores mediante certificados digitales y encriptando/desencriptando la correspondencia utilizando claves específicas que deben validarse en el centro de certificación de la Autoridad de Certificación (CA). El trabajo de la CA es identificar a las partes de la relación, las direcciones, las cuentas bancarias y la fecha de vencimiento del certificado, y determinar las identidades en función de ellas(Dastres & Soori, 2020).

3.1.3 Protocolo de Alerta SSL

El protocolo de alerta es el encargado de gestionar y notificar los fallos o errores a las entidades involucradas en el momento de la negociación y comunicación que genera el Handshake, Maneja dos bytes que son comprimidos y cifrados al igual que otros mensajes. El primer byte indica el nivel de alerta y tiene dos valores: "1" representa advertencia y "2" representa fatal. Si el mensaje de alerta es fatal, el enlace debe ser abortado y no se puede establecer un nuevo enlace en esa sesión particular por parte de SSL. El segundo byte indica el grado de severidad especificado por un código relacionado con diferentes mensajes de alerta(Satapathy & Livingston, 2016).

3.2 PROTOCOLO TLS

Originalmente, SSL era un protocolo propietario perteneciente y controlado por Netscape. Se hizo tan ampliamente

utilizado que fue liberado a la IETF (Internet Engineering Task Force), que lo renombró como TLS, que significa Seguridad de la Capa de Transporte (lo cual es curioso porque existe dentro de la Capa de Aplicación o, más específicamente, como un "shim" entre la Capa de Aplicación y la Capa de Transporte)(Hughes, 2022).La versión 3.0 de SSL y la recién nombrada TLS 1.0 compartían las características, sin embargo con el tiempo los atacantes encontraron vulnerabilidades lo cual obligó a sacar nuevas versiones del renombrado protocolo. En abril de 2006, se lanzó TLS 1.1, que incorporó mejoras significativas en la Capa de Registro. Posteriormente, en agosto de 2008, se presentó TLS 1.2, que actualizó las funciones de hash, la derivación de claves y los cálculos de MAC, además de habilitar el cifrado autenticado en esta capa. Entre abril de 2014 y agosto de 2018, la IETF dedicó esfuerzos significativos al desarrollo de TLS 1.3. Este rediseño ya no se basa en el modelo de SSL 3.0 y presenta numerosas innovaciones en el handshake, la derivación de claves y la capa de registro(Schwenk, 2022). Transport Layer Security (TLS) se ha convertido en el protocolo estándar de facto para las comunicaciones seguras en los servicios web. Más del 90% del tráfico de Internet se comunica a través de TLS ha evolucionado desde su predecesor, Secure Socket Layer (SSL), hasta su versión más reciente, TLS 1.3, mejorando tanto la seguridad como el rendimiento respecto a versiones anteriores(Lee et al., 2021).

3.2.1 Protocolo Handshake TLS

El protocolo Hansdshake sigue forman parte integra de SSL desde su concepción por tanto todas sus versiones y las de TLS hasta la versión 1.2 mantienen un esquema similar, sin embargo, Desde una perspectiva criptográfica, los principales cambios de diseño en TLS 1.3 incluyen: (1) cifrar algunos mensajes de protocolo de enlace con una clave de sesión intermedia, para proporcionar confidencialidad de los datos del protocolo de enlace, como el certificado del cliente; (5) desaprobar una variedad de algoritmos criptográficos, incluidos el transporte de clave RSA, el intercambio de clave Diffie-Hellman de campo finito, SHA-1, RC4, modo CBC, MAC, luego codificar, luego cifrar; y (6) usar esquemas modernos de cifrado autenticado con datos asociados (AEAD) para proteger los datos de la aplicación(Dowling et al., 2021).

Hay dos modos principales del protocolo de enlace TLS 1.3. Uno es el protocolo de enlace completo, un protocolo de enlace de ida y vuelta (1-RTT), que utiliza certificados de clave pública para la autenticación del servidor y (opcionalmente) del cliente, y (curva elíptica) intercambio de claves efímeras. En este modo, se establecen varias claves de sesión para una variedad de propósitos: para cifrar parte del protocolo de enlace, para permitir la exportación de material de claves a otras aplicaciones, para la reanudación de la sesión y, por supuesto, para cifrar los datos de la aplicación. Este modo recibe su nombre del hecho de que los datos de la aplicación se pueden enviar desde el cliente al servidor con la finalización del protocolo de enlace después de un viaje de ida y vuelta completo, lo que significa que hay un tiempo de ida y vuelta (1-RTT) hasta que se pueda enviar el primer mensaje de la aplicación (sin contar las operaciones de red que no son TLS, como las búsquedas de DNS o el protocolo de enlace TCP de 3 vías)(Dowling et al., 2021).

El otro modo principal del protocolo de protocolo de enlace TLS 1.3 es el modo de reanudación o clave precompartida (PSK), en el que la autenticación se basa en una clave simétrica precompartida, con intercambio opcional de claves (EC)DHE para secreto directo; esto generaliza el protocolo de enlace de reanudación de sesión abreviado de versiones anteriores de TLS(Dowling et al., 2021).Cuando un usuario visita un sitio web que ya ha visitado, no necesita establecer un proceso de protocolo de enlace nuevamente. Este mecanismo se denomina "tiempo de ida y vuelta cero" (0-RTT). El protocolo de enlace TLS más corto ha hecho que TLS v1.3 sea más rápido que TLS v1.2 (Figura 4)(A. P. Singh & Singh, 2022).En caso de que la primera conexión entre un cliente y un servidor. Si el servidor admite 0-RTT, tanto el cliente como el servidor pueden derivar un secreto de reanudación de su clave compartida y parámetros de sesión. El cliente simplemente almacenará este secreto. Naturalmente, el servidor necesita recuperar el secreto de reanudación durante un protocolo de enlace posterior. Hay dos enfoques estándar para esto, cachés de sesión y tickets de sesión, que tienen diferentes ventajas e inconvenientes. Durante el primer protocolo de enlace, el servidor envía al cliente una clave de búsqueda que apunta a una entrada en la caché de sesión del servidor o un ticket de sesión, según la configuración del servidor. Estos enfoques funcionan esencialmente de la siguiente manera:

Cachés de sesión:

El servidor almacena todos los secretos de reanudación de sesiones recientes en una base de datos local y emite a cada

cliente una clave de búsqueda única. Cuando un cliente se vuelve a conectar, incluye esa clave de búsqueda en sus mensajes 0-RTT, lo que permite al servidor recuperar y usar el secreto de reanudación coincidente.

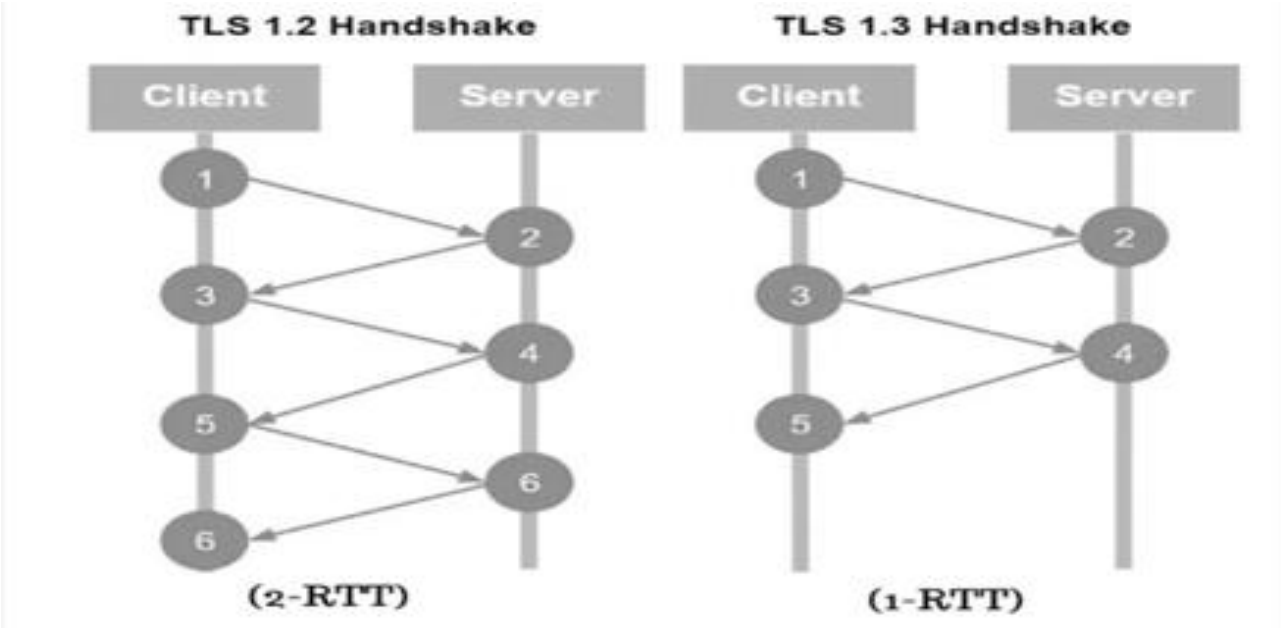
Entradas para la sesión:

El servidor utiliza una clave de cifrado simétrica a largo plazo, denominada clave de cifrado de tickets de sesión (STEK). En lugar de almacenar el secreto de reanudación en una base de datos local, el servidor lo cifra con STEK para crear un ticket de sesión. El cliente almacena el ticket de sesión. Cuando un cliente se vuelve a conectar, incluye ese ticket de sesión en sus mensajes 0-RTT, lo que permite al servidor descifrarlo y recuperar el secreto de reanudación. Tenga en cuenta que se utiliza el mismo STEK para muchas sesiones y clientes(Aviram et al., 2021).

En un protocolo de enlace 0-RTT posterior, el cliente incluirá en su primer mensaje la clave de búsqueda o el ticket de sesión cifrado, además de un mensaje de intercambio de claves Diffie-Hellman(Aviram et al., 2021).Otro apartado es el certificado. El certificado TLS permite que cualquier persona que se conecte a un sistema sepa que está enviando sus datos al sitio que figura en el certificado. Además, también permite establecer conexiones seguras para que nadie en el medio pueda espiar las comunicaciones(Akram et al., 2020).

Otro aspecto a destacar es la búsqueda de simplificar y reducir ciertos aspectos con respecto a versiones anteriores, TLS 1.2 tenía a su disposición un gran número de conjunto de cifrados que podía usar, pero que eran demasiado complejos que creaba conflictos entre las entidades involucradas y generaba un mayor número de negociaciones para establecer un cifrado, en comparación TLS 1.3 a anula la negociación y solo permite conjunto de cifrados más simples.

Figura 4: comparación de protocolos de enlace



Fuente:(A. P. Singh & Singh, 2022)

3.2.2 Protocolo de Alerta TLS

El protocolo de alerta para TLS en especial la versión 1.3 mantiene el mismo mecanismo con sus contrapartes anteriores y las del SSL, no obstante debido a que distan en pasos con respecto al protocolo handshake se generan nuevas alertas un ejemplo es la alerta User_cancelled que se describe como una terminación anormal de la sesión por parte del usuario

3.2.3 Protocolo de Registro TLS

El Protocolo de Registro TLS es responsable de la seguridad de los datos transmitidos a través de la red, asegurando que se fragmenten, cifren y se mantenga su integridad mediante el uso de MACs y claves criptográficas derivadas. Los componentes del protocolo son según(Schwenk, 2022):

1. Fragmentación: la capa de registro SSL fragmenta el flujo de bytes en diferentes bloques. Estos bloques se denominan registros y no deben tener una longitud superior a 214 = 16384 bytes.
2. Compresión (opcional): después de la fragmentación, se puede aplicar un método de compresión al registro de manera opcional. La compresión tiene como objetivo minimizar el tamaño del registro.
3. Cálculo del MAC: en el siguiente paso, se agrega un código de autenticación de mensajes (MAC) para autenticar los datos, que solo el remitente y el receptor pueden validar.
4. Padding: Cuando se utiliza un cifrador de bloque para el cifrado, la longitud del texto plano debe ser un múltiplo de la longitud del bloque del cifrador. Puede ser necesario añadir algunos bytes adicionales, los llamados bytes de relleno. Para que la capa de registro en el lado receptor reconozca qué bytes han sido añadidos, también debe especificarse el número de estos bytes de relleno
5. Cifrado: Finalmente, el registro se cifra.

3.3 VULNERABILIDADES SSL/TLS

Un defecto en el código del software, un error de configuración del sistema u otra debilidad en el sitio web o la aplicación web, o en cualquiera de sus partes y operaciones, se conoce como vulnerabilidad(Alabduljabbar et al., 2022).Cualquier software puede tener una o varias vulnerabilidades en cualquiera de sus campos o componentes y son explotadas por ataques de terceros no involucrados con el proceso o funcionamiento del software con el objetivo de secuestrar , manipular o destruir la información. Para el caso de ssl/tls a sufrido de vulnerabilidades en cada una de sus versiones y por ende padecidos ataques.

3.3.1 Heartbeat

Heartbeat es un ataque que explota la vulnerabilidad llamada Heartbleed, la cual en si no es un problema del protocolo ssl/tls, si no un fallo de una extensión de una librería que podía usarse junto con el protocolo, La vulnerabilidad se produce por una incorrecta validación de la longitud de entrada en el código. La explotación otorga la capacidad al atacante de recuperar hasta 64 KB de datos de la memoria principal del servidor por cada petición de Heartbeat. El atacante puede enviar múltiples peticiones con las cuales puede recuperar un número ilimitado de información del servidor(Romero Benito, 2020), con esto el atacante puede obtener la información del servidor y el cliente, este ataque perjudico mucho a la versión 1.1 de TLS y todas las versiones de SSL , para mitigarlo se requiere usar una versión superior a la 1.1 como TLS 1.2

3.3.2 Crime

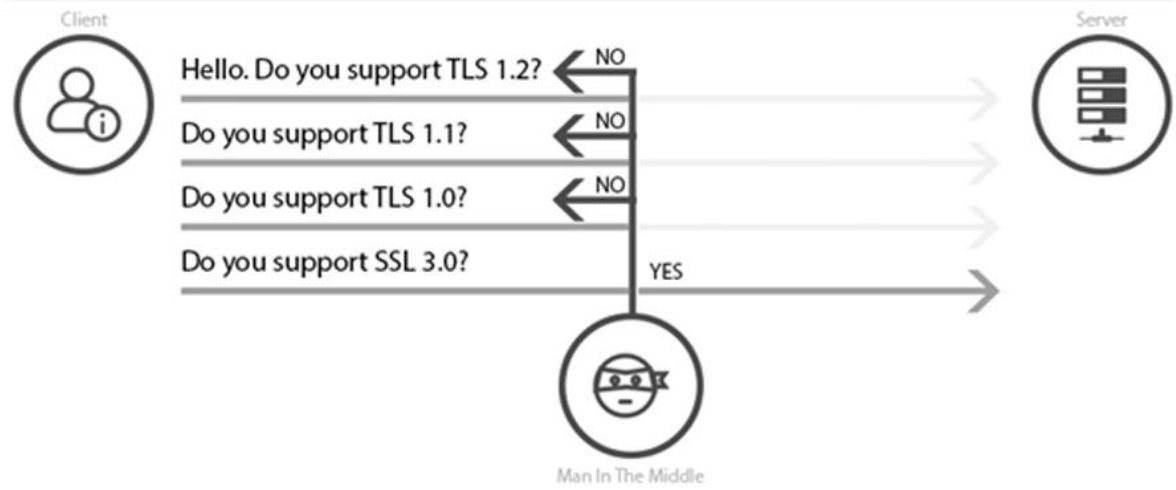
El ataque crime (Compression Ratio Info Leak Mass Exploitation) es un tipo de ataque que busca sacar provecho de la fase de compresión de los datos que hace uno de los protocolos de TLS para obtener la clave y la información La clave se obtiene engañando al navegador y enviando una solicitud comprimida y cifrada al sitio web legítimo, esperando el tamaño de la respuesta HTTP y aumentando el ataque en función de las respuestas HTTP . El hacker repite las técnicas

con diferentes valores hasta que se obtenga la clave(Satapathy & Livingston, 2016).Este ataque requiere de mucho tiempo ya que emplea la fuerza bruta no muy eficaz pero funcional, por su parte este ataque afecta TLS 1.2 y aversiones anteriores debido al uso de la compresión, caso contrario para TLS 1.3..Aproximadamente el 42% de los sitios web se ven afectados por delitos cibernéticos(Alabduljabbar et al., 2022).

3.3.3 Ataque POODLE/POODLE ZOMBI

Un ataque POODLE (Padding Oracle on Downgraded Legacy Encryption) es un ataque de tipo MITM (man-in-the-middle) en el que puede espiar la comunicación entre el cliente y el servidor, así como suplantar a cualquiera de los dos. A continuación, el atacante intenta ejecutar un "ataque de degradación": en este ataque, convence al servidor de usar un protocolo más antiguo como SSL 3.0, provocando la caída de la conexión hasta que el servidor interpreta que el cliente no usa protocolos más recientes como TLS 1.2 (Figura 5). Finalmente, si la comunicación se realiza usando SSL 3.0, el atacante engaña al navegador del usuario para que ejecute un JavaScript que ayuda a descifrar información confidencial. el atacante explota el llamado "oracle de padding"(Matharu, 2021).

Figura 5: Ataque POODLE



Fuente:(Matharu, 2021)

Se creía que el ataque POODLE estaba "muerto"; sin embargo, en febrero de 2019, el investigador Craig Young descubrió dos nuevas vulnerabilidades en el protocolo TLS 1.2 . Según Young, todavía existen productos que no corrigieron completamente el problema del ataque POODLE original, y llamó a esta versión resurgida del ataque "ZOMBIE POODLE" o "POODLE 2.0" .En él un atacante puede inyectar código JavaScript en el navegador de la víctima; una vez que el navegador se ve comprometido, el atacante puede llevar a cabo un ataque de tipo "man-in-the-middle" (MITM) y capturar cookies de sesión y credenciales de usuario de la sesión web(Matharu, 2021).

3.3.4 Ataque de Repetición

Un ataque de repetición se produce cuando un atacante reenvía un mensaje capturado. En la mayoría de los casos, este mensaje se captura antes, antes de que se realice el ataque de reproducción. Es un ataque simple, pero dependiendo del mensaje reproducido, las amenazas pueden ser graves. El atacante puede grabar el mensaje de protocolo de enlace de reanudación 0-RTT, ClientHello, y enviarlo nuevamente al servidor, fingiendo que proviene del cliente legítimo. Dado que este mensaje está autenticado, el servidor puede descifrarlo y extraer los primeros datos de vuelo para ejecutar el proceso necesario de acuerdo con los datos de la aplicación. Esto puede causar un problema grave si los datos de esta aplicación son críticos. Por ejemplo, el atacante puede reproducir una orden de transacción de dinero varias veces, sin necesidad de descifrar o modificar el mensaje. La transacción se puede realizar muchas veces siempre que los mensajes reproducidos lleguen al lado del servidor(Abdelhafez et al., 2023).

3.4 SSL/TLS VS COMPUTACION CUANTICA

3.4.1 Computación cuántica

La computación cuántica plantea un desafío único a la seguridad actual de Internet. La infraestructura de clave pública (PKI) utilizada para generar y distribuir claves de cifrado para el transporte por Internet es particularmente vulnerable a los algoritmos cuánticos que proporcionan una aceleración exponencial en el descubrimiento de claves privadas(Hubermann et al., 2020), Los protocolos asociados de TLS 1.3 como el handshake dependen en gran medida de las PKI para asegurar la confidencialidad de la información, la computación cuántica basa gran parte de su éxito en un nueva unidad denominada qubit. La Internet clásica se basa en el transporte de paquetes que están formados por bits. Los bits son la unidad básica de información en el mundo digital. Es una representación lógica de uno de los dos estados "Verdadero" o "Falso" o "1" o "0". El estado es determinista y las mediciones no cambian el estado de ningún bit. La teoría de la información cuántica utiliza bits cuánticos, abreviados como qubits, como unidades de información. La diferencia fundamental con los bits clásicos es que un qubit siempre está en una superposición de los estados "1" y "0" en un momento dado (Cavaliere et al., 2022), debido a las características del qubit este abre un nuevo paradigma en la computación y da funcionalidad a los denominados algoritmos cuánticos entre ellos uno de los más representativos es el algoritmos de shor.

3.4.2 Algoritmo de Shor

El algoritmo de Shor es un algoritmo cuántico introducido por primera vez por el matemático Peter Shor en 1994. Es un algoritmo de base cuántica que se utiliza para resolver el problema de factorización prima. El problema de factorización prima consiste en determinar los números primos multiplicados para obtener un número compuesto dado(Kumar & Mondal, 2024). Prácticamente toda la criptografía de clave pública implementada en la actualidad se basa en la dificultad de calcular logaritmos discretos o factorización de números enteros. Por lo tanto, el algoritmo de Shor plantea una grave amenaza para la seguridad de la información. Esto afecta por igual a los métodos de intercambio de claves y algoritmos de firma(Gonzalez & Wiggers, 2022) El algoritmo de Shor, por ejemplo, factoriza grandes números en tiempo polinómico, socavando así la premisa de seguridad de RSA, que depende de la dificultad de la factorización prima. El algoritmo de Shor también puede resolver el problema del logaritmo discreto de la curva elíptica (ECDLP) en tiempo polinómico, Shor demostró que las técnicas de cifrado basadas en el problema del logaritmo discreto y el problema de factorización de enteros podrían resolverse fácilmente mediante un control de calidad futuro. Los sistemas criptográficos populares como RSA, DSA y ECC se basan en estos problemas. Como resultado, los protocolos existentes que dependen de ECC y RSA no estarán a salvo de ataques cuánticos(M. Singh et al., 2025).

Kem

3.4.3 Resistencia cuántica TLS

La seguridad de la Distribución Cuántica de Claves (QKD) se basa en dos teoremas de la física cuántica: el teorema de no clonación y el principio de incertidumbre de Heisenberg. El primero establece que es imposible obtener una copia idéntica de un estado cuántico desconocido arbitrario. El segundo explica que la medición de una señal cuántica causa

una perturbación en su estado que no puede ser recuperada. Por lo tanto, cualquier intento de espiar en el canal cuántico alterará inevitablemente el estado de las señales cuánticas intercambiadas, alertando a las partes involucradas sobre la presencia de un espía(Rubio García et al., 2024).La QKD puede ser implementada para TLS en sus protocolos para ser resistente a los algoritmos cuánticos

3.4.4 KEM (Key Encapsulation Mechanism)/ML-KEM (Module Lattice–based Key Encapsulation Mechanism)

Un KEM es un tipo específico de esquema de establecimiento de claves. El establecimiento típico de una clave mediante un KEM involucra dos partes (a las que aquí se denomina Alice y Bob) y consta de las siguientes tres etapas según(Alagic, 2025) (ver Figura 6):

(Generación de claves) Alice genera una clave de desencapsulación (privada) y una clave de encapsulación (pública).

(Encapsulación) Bob utiliza la clave de encapsulación de Alice para generar una clave secreta compartida y un cifrado asociado. El cifrado se envía a Alice.

(Desencapsulación) Alice utiliza el cifrado y su clave de desencapsulación para calcular otra copia de la clave secreta compartida.

Figura 6: ejemplo KEM

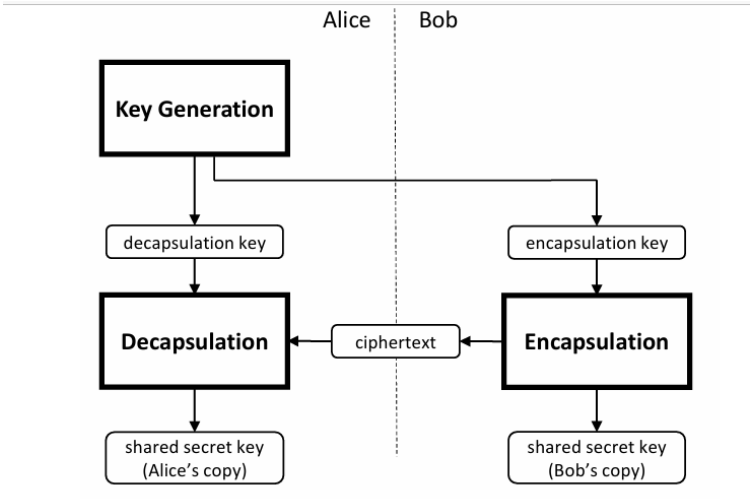


Figura 6 Fuente:(Alagic, 2025)

Cuando un KEM se utiliza como se muestra en la Figura 6, el resultado debe ser una clave secreta compartida que sea aleatoria, desconocida por los adversarios y que coincida para Alice y Bob con alta probabilidad(Alagic, 2025).Basado en lo anterior en 2016, el NIST inició un proceso oficial para desarrollar nuevos estándares criptográficos seguros para la cuántica centrado en el intercambio de claves y las firmas digitales. El proceso de selección pasó por varias rondas y

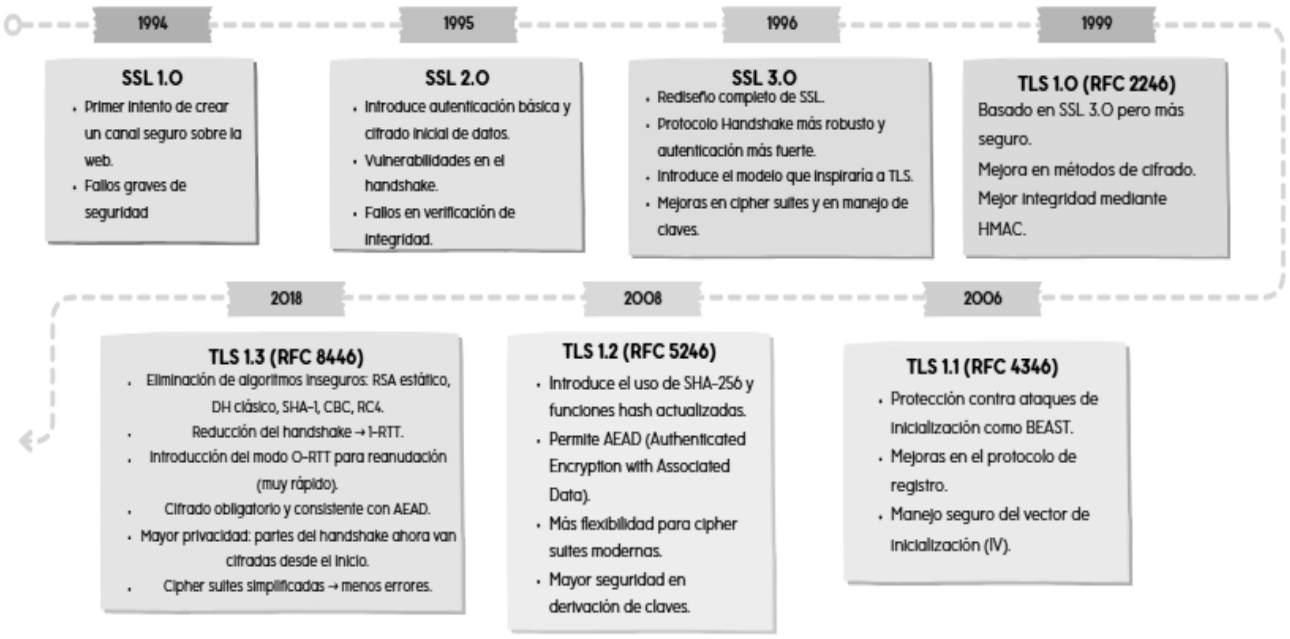
obtuvo aceptación mundial. Actualmente, el proceso se encuentra en la Ronda 4 y Kyber (el cual es un KEM) ya ha sido seleccionado ya que Ofrece una compensación comparativamente eficiente entre seguridad y rendimiento (Athanasopoulos & Mennink, 2023).

ML-KEM es un estándar de mecanismo de encapsulación de claves (KEM) basado en CRYSTALS-Kyber. Su seguridad se basa en la dureza del problema de aprendizaje de módulos con error (MLWE), que implica resolver un sistema de ecuaciones lineales ruidosas en un anillo polinómico. En particular, se construye un esquema de cifrado de clave pública utilizando un sistema de ecuación lineal ruidoso donde la clave pública es una matriz generada aleatoriamente y un vector es la clave privada. Para cifrar un mensaje, se codifica como una ecuación ruidosa adicional, que se puede recuperar utilizando la clave privada. Este esquema se transforma en un KEM utilizando la transformación Fujisaki-Okamoto (FO) , lo que da como resultado un KEM que se cree que proporciona seguridad IND-CCA(Montenegro et al., 2026).ML-KEM se está perfilando como la solución más viable para la futura época cuántica.

4. DISCUCION

Nuestro objetivo era indagar en la evolución de los protocolos SSL y TLS. En base a la investigación se construyó la siguiente (figura 5),

Figura 7: Línea de tiempo



Fuente: Propia

La línea de tiempo (figura 7), evidencia la profunda transformación que han experimentado los protocolos criptográficos SSL y TLS cada uno teniendo diferentes características, esto es consecuencia de las vulnerabilidades de los mismo que se

van descubriendo con el tiempo y que se crean formas de explotarlas, cabe resaltar que sus primeros versiones eran más propensas hacer vulneradas por sus fallos , mientras que las versiones posteriores , sobretodo de TLS, permanecían más debido a todas las mejoras que tenían, en la actualidad la versión 1.3 de TLS es el estándar en las páginas y a pesar de que también tiene problemas se mantiene a día de hoy, sim embargo esto nos hace entrar en el tema de la computación cuántica la cual trae cambiaos para la criptográfica como la conocemos y es la promotora de que se genere un cambio para una nueva versión o por el contrario tal vez un renombramiento del mismo protocolo: Por prevención frente a este cambio algunas agencias estan explorando nuevas tecnologías que , en este artículo se presentaron el KEM y ML-KEM uno de sus grandes exponentes frente a la cuántica, por supuesto no son los únicos pero algunas tecnologías aun estan pasando por fases muy tempranas de exámenes para considerarlos siquiera opciones viables, por tanto en futuras revisiones del tema deberán tenerse si alguna nueva tecnología ha sido presentada y si es mejor en comparación con las ya presentada

5. CONCLUSIONES

El protocolo SSL/TLS son el estándar de la seguridad actualmente, pero en el pasado han enfrentado vulnerabilidades que los han orientado a mejorar y sacar nuevas versiones, asimismo, la inminente llegada de la computación cuántica plantea un punto de inflexión en la criptografía moderna, al amenazar los esquemas de cifrado tradicionales en los que se apoya la infraestructura de seguridad actual. En este sentido, actúa como elemento que induce un cambio y genera que desde hace un par de años se halla investigado contramedidas entre los que destacan los mecanismos post-cuánticos, especialmente los basados en encapsulación de claves (KEM) como ML-KEM, que surgen como una respuesta viable para la preservación de la confidencialidad y la autenticidad de los datos en la era cuántica. Aun así, cabe recalcar que esta es una visión bastante temprana de contramedida y que la computación cuántica aun es un campo bastante amplio y variable que puede traer más obstáculos

Por tanto, se concluye que los protocolos ante la computación cuántica, enfrentan un cambio y su relevancia en el futuro dependerá en gran medida de la integración progresiva de soluciones criptográficas resilientes a la computación cuántica. La transición hacia un entorno post-cuántico no representa una sustitución inmediata, sino un proceso de adaptación que garantizará la continuidad de la confianza digital en un estándar como lo es el protocolo TLS.

REFERENCIAS

Abdelhafez, M. E., Ramadass, S., & Abdelwahab, M. (2023). TLS Guard for TLS 1.3 zero round-trip time (0-RTT) in a distributed environment. *Journal of King Saud University - Computer and Information Sciences*, *35*(10), 101797. https://doi.org/10.1016/j.jksuci.2023.101797

Akram, M., Barker, W. C., Clatterbuck, R., Dodson, D., Everhart, B., Gilbert, J., Haag, W., Johnson, B., Kapasouris, A., Lam, D., Pleasant, B., Raguso, M., Souppaya, M., Symington, S., Turner, P., & Wilson, C. (2020). *Securing web transactions TLS server certificate management* (No. NIST SP 1800-16; p. NIST SP 1800-16). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.1800-16

Alabduljabbar, A., Ma, R., Choi, S., Jang, R., Chen, S., & Mohaisen, D. (2022). Understanding the Security of Free

Content Websites by Analyzing their SSL Certificates: A Comparative Study. *Proceedings of the 1st Workshop on Cybersecurity and Social Sciences*, 19-25. https://doi.org/10.1145/3494108.3522769

Alagic, G. (2025). *Recommendations for Key-Encapsulation Mechanisms* (No. NIST SP 800-227 ipd; p. NIST SP 800-227 ipd). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-227.ipd

Alwazzeh, M., Karaman, S., & Shamma, M. N. (2020). Man in The Middle Attacks Against SSL/TLS: Mitigation and Defeat. *Journal of Cyber Security and Mobility*. https://doi.org/10.13052/jcsm2245-1439.933

Athanasopoulos, E., & Mennink, B. (Eds.). (2023). *Information Security: 26th International Conference, ISC 2023, Groningen, The Netherlands, November 15–17, 2023, Proceedings* (Vol. 14411). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-49187-0

Aviram, N., Gellert, K., & Jager, T. (2021). Session Resumption Protocols and Efficient Forward Security for TLS 1.3 0-RTT. *Journal of Cryptology*, *34*(3), 20. https://doi.org/10.1007/s00145-021-09385-0

Cavaliere, F., Sircar, R. P., & Catuogno, T. (2022). The Future Quantum Internet. En *Quantum Computing Environments* (pp. 75-123). Springer, Cham. https://doi.org/10.1007/978-3-030-89746-8_3

Chuvakin, A. A., & Williams, B. R. (2010). Protecting Cardholder Data. En *PCI Compliance* (pp. 105-135). Elsevier. https://doi.org/10.1016/B978-1-59749-499-1.00011-8

Dastres, R., & Soori, M. (2020). *Secure Socket Layer (SSL) in the Network and Web Security*. *14*(10).

Devi, O. R., Vallabhaneni, S. P., Hussain, M. A., & Kumar, T. K. (2020). Security Analysis on Remote Authentication against Man-in-the-Middle Attack on Secure Socket Layer. *IOP Conference Series: Materials Science and Engineering*, *981*(2), 022015. https://doi.org/10.1088/1757-899X/981/2/022015

Dowling, B., Fischlin, M., Günther, F., & Stebila, D. (2021). A Cryptographic Analysis of the TLS 1.3 Handshake Protocol. *Journal of Cryptology*, *34*(4), 1-69. https://doi.org/10.1007/s00145-021-09384-1

Easttom, C. (2022). SSL/TLS. En *Modern Cryptography* (pp. 285-307). Springer, Cham. https://doi.org/10.1007/978-3-

031-12304-7_13

Gonzalez, R., & Wiggers, T. (2022). KEMTLS vs. Post-quantum TLS: Performance on Embedded Systems. *Security, Privacy, and Applied Cryptography Engineering*, 99-117. https://doi.org/10.1007/978-3-031-22829-2_6

Hubermann, B. A., Lund, B., & Wang, J. (2020). Quantum Secured Internet Transport. *Information Systems Frontiers*, *22*(6), 1561-1567. https://doi.org/10.1007/s10796-020-10086-5

Hughes, L. E. (2022). SSL and TLS. En *Pro Active Directory Certificate Services* (pp. 155-175). Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-7486-6_11

Ibrahim, A. (2024). *Secure Socket Layer: Fundamentals and Certificate verification*. https://doi.org/10.31224/3532

Kumar, M., & Mondal, B. (2024). Study on Implementation of Shor's Factorization Algorithm on Quantum Computer. *SN Computer Science*, *5*(4), 1-13. https://doi.org/10.1007/s42979-024-02771-y

Kundu, S., Gupta, T., Sardar, A., Bandyopadhyay, A., Swain, S., & Mallik, S. (2025). A survey on quantum computing: Transforming cryptography, AI/ML, blockchain, and network communication. *Franklin Open*, *12*, 100371. https://doi.org/10.1016/j.fraope.2025.100371

Lee, H., Kim, D., & Kwon, Y. (2021). TLS 1.3 in Practice:How TLS 1.3 Contributes to the Internet. *Proceedings of the Web Conference 2021*, 70-79. https://doi.org/10.1145/3442381.3450057

Matharu, S. K. (2021). *Exploiting SSL/TLS Vulnerabilities in Modern Technologies*. https://doi.org/10.7939/r3-2q1v-bb10

Montenegro, J. A., Ríos, R., & López-Cerezo, J. (2026). A performance evaluation framework for post-quantum TLS. *Future Generation Computer Systems*, *175*, 108062. https://doi.org/10.1016/j.future.2025.108062

Perugini, L., & Vesco, A. (2024). On the integration of Self-Sovereign Identity with TLS 1.3 handshake to build trust in IoT systems. *Internet of Things*, *25*, 101103. https://doi.org/10.1016/j.iot.2024.101103

Romero Benito, J. C. (2020). *TLS 1.3*. https://openaccess.uoc.edu/handle/10609/126747

Rubio García, C., Rommel, S., Takarabt, S., Vegas Olmos, J. J., Guilley, S., Nguyen, P., & Tafur Monroy, I. (2024). Quantum-resistant Transport Layer Security. *Computer Communications*, *213*, 345-358. https://doi.org/10.1016/j.comcom.2023.11.010

Satapathy, A., & Livingston, J. (2016). A Comprehensive Survey on SSL/ TLS and their Vulnerabilities. *International Journal of Computer Applications*, *153*(5), 31-38. https://doi.org/10.5120/ijca2016912063

Schwenk, J. (2022). Transport Layer Security. En *Guide to Internet Cryptography* (pp. 201-242). Springer, Cham. https://doi.org/10.1007/978-3-031-19439-9_10

Singh, A. P., & Singh, M. (2022). Handshake Comparison Between TLS V 1.2 and TLS V 1.3 Protocol. En *Cyber Security in Intelligent Computing and Communications* (pp. 143-155). Springer, Singapore. https://doi.org/10.1007/978-981-16-8012-0_12

Singh, M., Sood, S. K., & Bhatia, M. (2025). Post-quantum Cryptography: A Review on Cryptographic Solutions for the Era of Quantum Computing. *Archives of Computational Methods in Engineering*. https://doi.org/10.1007/s11831-025-10412-7