



Universidad Politécnica
de Madrid



Escuela Técnica Superior de Ingenieros Informáticos

Grado en Ingeniería Informática

Trabajo Fin de Grado

Métodos de Hibridación de Claves Criptográficas

Autor: Juan Maruri Pérez
Tutor: Juan Pedro Brito Méndez

Madrid, Julio 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Escuela Técnica Superior de Ingenieros Informáticos
Grado en Ingeniería Informática
Trabajo Fin de Grado

Título: Métodos de Hibridación de Claves Criptográficas

Julio 2025

Autor: Juan Maruri Pérez

Tutor: Juan Pedro Brito Méndez

Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

Resumen

Este trabajo realiza un análisis exhaustivo de los procesos de establecimiento de conexión o *handshakes*, en protocolos de comunicación, así como de los distintos tipos de criptografía implicada, con especial atención a las estrategias de hibridación de claves en el contexto del avance de la computación cuántica.

Posteriormente, mediante un entorno experimental controlado, se evalúa la calidad de los secretos compartidos generados a partir de diversos mecanismos de hibridación. La evaluación se basa en métricas estadísticas. Los resultados muestran cómo determinadas estrategias de hibridación influyen en la solidez criptográfica del sistema, y cómo el uso de funciones de derivación puede corregir deficiencias no resueltas por hibridaciones menos robustas.

El estudio concluye con unas sugerencias sobre la elección del operador de hibridación más adecuado, en función del *pipeline* criptográfico en el que se integre.

Abstract

This work presents a comprehensive analysis of connection establishment processes, or *handshakes*, in communication protocols, as well as the various types of cryptography involved, with particular focus on key hybridization strategies in the context of advances in quantum computing.

Subsequently, within a controlled experimental environment, the quality of shared secrets generated through different hybridization mechanisms is evaluated. The evaluation is based on statistical metrics. The results show how certain hybridization strategies influence the cryptographic strength of the system, and how the use of derivation functions can address deficiencies not resolved by less robust hybridizations.

The study concludes with recommendations on selecting the most suitable hybridization operator, depending on the cryptographic *pipeline* in which it is integrated.

Tabla de contenidos

1. Introducción	1
2. Estado del Arte	5
2.1. Handshakes Criptográficos en Protocolos de Comunicaciones	5
2.2. Criptografía en un Handshake	8
2.2.1. Criptografía Asimétrica	9
2.2.2. Funciones Hash	11
2.2.3. Criptografía Simétrica	15
2.3. Criptografía Post-Cuántica - PQC	16
2.4. Quantum Key Distribution - QKD	19
2.5. Protocolos de Comunicaciones Híbridos	21
2.5.1. Hybrid Key Exchange TLS 1.3 – (TLS-HKE)	23
2.5.2. Hybrid Key Exchange SSH – (SSH-HKE)	24
2.6. Hibridación de Claves	25
3. Desarrollo	29
3.1. Hipótesis	29
3.2. Metodología	30
3.2.1. Casos de Estudio	30
3.2.2. Métricas de Evaluación	31
3.2.3. Herramientas y Tecnologías Utilizadas	34
4. Resultados	35
4.1. Medidas de Tendencia Central	35
4.2. Histogramas de Distribución de Bytes	36
4.3. Tests Estadísticos	37
5. Análisis	41
5.1. Análisis de los Resultados	41
5.2. Extensión del Análisis	45
6. Discusión	47
7. Conclusión	49
Bibliografía	51

Capítulo 1

Introducción

La criptografía es una práctica que se ocupa de elaborar técnicas destinadas a alterar las representaciones lingüísticas de ciertos mensajes con el fin de hacerlos incomprensibles por cualquiera que no esté autorizado a acceder a su contenido. Tiene su origen etimológico en el griego κρύπτος (kryptós, «secreto») y γραφή (graphé, «grafo» o «escritura»), se traduce literalmente como «escritura secreta». Su historia es larga y abundante en anécdotas, una prueba de esto es que el primer método criptográfico, conocido como Escítala, data del siglo V a.C. en la antigua Grecia[1].

En tiempos modernos, la criptografía ha dado un salto significativo con la llegada de Internet en la era digital. En las etapas tempranas de Internet, las comunicaciones se realizaban en texto plano, lo que implicaba que cualquiera que pudiera interceptarlas podía acceder a su contenido, lo que planteaba serios riesgos para la privacidad y la seguridad. Con la expansión de Internet y la introducción del protocolo TCP/IP en 1982, se hizo cada vez más evidente la necesidad de proteger las transacciones y los datos en línea. La invención de la World Wide Web en 1990 aceleró este proceso aún más, destacando la urgencia de desarrollar métodos criptográficos. En 1994, la creación del protocolo Secure Socket Layer (SSL) marcó un hito, ya que permitió cifrar las comunicaciones y garantizar conexiones seguras entre los usuarios y los servidores, cambiando la forma en que la información era transmitida en la web. Con el tiempo, SSL evolucionó al más robusto Transport Layer Security (TLS), que sigue siendo la base de la seguridad en casi todas las actividades en línea.

Hoy en día, nuestras vidas sociales y profesionales están cada vez más integradas en la infraestructura digital. Actividades que antes se realizaban de manera física o en persona, como hacer la compra en el supermercado o conversar con amigos en el parque, ahora se hacen también en internet. El teletrabajo, por ejemplo, es una de las actividades que más se ha acelerado y expandido, siendo una realidad consolidada para aproximadamente el 15% de la población mundial con empleo. Estos ejemplos, y muchos más, son posibles gracias a que confiamos en la seguridad y privacidad que internet nos proporciona. Privacidad y seguridad que son proporcionadas, mayoritariamente, por técnicas criptográficas. Cada vez que realizamos una compra en línea, la criptografía ase-

Capítulo 1. Introducción

gura que nuestros datos bancarios estén cifrados y no sean accesibles aunque sean interceptados. Cuando trabajamos desde casa y usamos herramientas de teletrabajo, como las videollamadas o el acceso remoto a sistemas corporativos mediante VPNs, estos servicios utilizan criptografía para proteger la confidencialidad de la información. Al acceder a nuestras cuentas en redes sociales, la criptografía protege nuestras contraseñas y asegura que nuestra identidad digital esté a salvo. Cuando firmamos documentos oficiales con el DNI electrónico o con el Certificado Electrónico de Ciudadano, dichas firmas, que garantizan que el documento firmado es auténtico y que no ha sido alterado tras su firma, se basan también en métodos criptográficos.

En resumen, la criptografía es un componente crítico sobre el cual descansan muchas de las actividades cotidianas, brindando confianza, seguridad y protección en todos los aspectos de nuestra vida conectada.

Todos estos sistemas comparten un principio fundamental en criptografía moderna su seguridad no depende de la ocultación del algoritmo, como se hacía antiguamente, sino de la solidez de los modelos matemáticos subyacentes. Este enfoque está alineado con el principio de Kerckhoffs[2], según el cual un sistema debe garantizar confidencialidad e integridad incluso si su diseño es público, siempre que las claves, únicos elementos secretos, se gestionen de forma segura. Así, la fortaleza de estos mecanismos depende críticamente de la resistencia computacional de problemas matemáticos y de la longitud y aleatoriedad de las claves.

Para ilustrar esta protección, consideremos un ataque por fuerza bruta, el atacante evita resolver el fundamento matemático y prueba todas las combinaciones posibles de claves. Se estaría enfrentando a un problema de escala descomunal. Por ejemplo, una clave de tamaño estándar de 128 bits tiene aproximadamente $3,4 \times 10^{38}$ combinaciones posibles, lo que hace que un ataque por fuerza bruta sea inviable incluso para los supercomputadores más avanzados, ya que llevaría más tiempo que la edad actual del universo en encontrar la combinación correcta. Esto en el caso de un atacante con una computadora convencional, siguiendo métodos convencionales.

En 1994, Peter Shor, de los Laboratorios Bell, demostró que los ordenadores cuánticos —una tecnología emergente que aprovecha las propiedades físicas de la materia y la energía para realizar cálculos— pueden resolver de manera eficiente el problema de factorización de números enteros y el del logaritmo discreto en varios grupos. Lo hizo mediante el diseño de un algoritmo cuántico, conocido como algoritmo de Shor, específicamente ejecutable por este tipo de computadoras. Este descubrimiento revolucionó el campo de la criptografía al evidenciar que dichos problemas, fundamentales para muchos esquemas criptográficos clásicos, pueden resolverse en tiempo polinómico utilizando un ordenador cuántico. En la práctica, esto hace que un subconjunto de los criptosistemas actuales quede obsoleto. Por lo tanto, un ordenador cuántico lo suficientemente potente pondría en peligro muchas formas de comunicación moderna.

Este descubrimiento ha inspirado un gran interés en la computación cuántica y aún son varias las preguntas sin respuesta. ¿La complejidad cuántica es fundamentalmente diferente de la complejidad clásica? ¿Cuándo se construirán ordenadores cuánticos a gran escala? ¿Qué serie de cambios traerá consigo este nuevo paradigma? ¿Existe forma alguna de resistir a tales computadoras con las computadoras clásicas?

En los veinte años transcurridos desde el descubrimiento de Shor, la teoría de los algoritmos cuánticos se ha desarrollado significativamente. Se han descubierto algoritmos cuánticos que logran una aceleración exponencial para varios problemas relacionados con la simulación física, la teoría de números y la topología. Sin embargo, la lista de problemas que admiten una aceleración exponencial mediante computación cuántica sigue siendo relativamente pequeña. En contraste, se han logrado avances más modestos en la aceleración de algoritmos que resuelven problemas relacionados con la búsqueda, la detección de colisiones y la evaluación de fórmulas booleanas. En particular, el algoritmo de búsqueda de Grover ofrece una aceleración cuadrática en problemas de búsqueda no estructurada. Si bien tal aceleración no vuelve obsoletas las tecnologías criptográficas, puede tener el efecto de requerir tamaños de clave más grandes[3].

Criptosistema	Tipo	Propósito	Contramedida
AES	Clave Simétrica	Cifrado	Se necesitarán tamaños de clave más grandes
SHA-2, SHA-3	Sin Clave	Funciones hash	Se necesitarán salidas más largas
RSA	Clave Pública	Firmas, establecimiento de claves	Ya no será seguro
ECDSA, ECDH	Clave Pública	Firmas, intercambio de claves	Ya no será seguro
DSA (Criptografía de Campo Finito)	Clave Pública	Firmas, intercambio de claves	Ya no será seguro

Cuadro 1.1: Impacto de la Computación Cuántica en Criptosistemas Actuales[3]

Como se muestra en el Cuadro 1.1, los algoritmos actuales de establecimiento de claves dejarán de ser seguros con la llegada de la computación cuántica. Y es esta brecha precisamente la que un nuevo tipo de ataque quiere explotar. Conocido como *harvest now, decrypt later* (cosechar ahora, descifrar después), este tipo de ataque consiste en la recopilación masiva de comunicaciones cifradas con la intención de descifrar su contenido en el futuro. Al recopilar la fase de *handshake*, si la tecnología cuántica madura lo suficiente, es probable que se consiga averiguar la clave secreta establecida en esta fase y, así, se podrá

Capítulo 1. Introducción

acceder al contenido de la comunicación. Este ataque ya está en curso, y dado que muchos datos confidenciales mantienen su valor durante años, es crucial actualizar los protocolos de comunicación para mitigar su impacto[4].

En respuesta a esta amenaza, ha surgido una gran comunidad internacional dedicada a abordar la seguridad de la información en un futuro de computación cuántica. La esperanza es que nuestra infraestructura de clave pública pueda mantenerse segura utilizando nuevos primitivos resistentes a la computación cuántica. Se están investigando diversas alternativas, como la Criptografía Post-Cuántica (PQC) y la Distribución Cuántica de Claves (QKD), ambas son áreas de investigación activa que han recibido un apoyo sustancial alrededor del mundo. Estos esfuerzos han impulsado avances fundamentales, allanando el camino para el despliegue de criptosistemas resistentes a la computación cuántica en entornos reales[3].

Ante este desafío, la presente investigación tiene como objetivo principal analizar las alternativas criptográficas que pretenden actualizar el componente de la infraestructura de clave pública vulnerable al ataque *harvest now, decrypt later*. En otras palabras, **se pretende estudiar y analizar las soluciones existentes que abordan la confidencialidad de los protocolos de comunicación ante la amenaza de la computación cuántica, con el fin de comprender su funcionamiento y valorar su idoneidad.**

Capítulo 2

Estado del Arte

2.1. Handshakes Criptográficos en Protocolos de Comunicaciones

El concepto de *handshake*, tomado del término anglosajón para apretón de manos o saludo, refiere al proceso que establece una comunicación. Históricamente, el apretón de manos humano ha simbolizado buenas intenciones, un ritual social que certifica la ausencia de armas y la disposición al diálogo. En el ámbito técnico, esta ceremonia se traduce en un protocolo estructurado donde los sistemas intercambian mensajes de control para autenticarse mutuamente, acordar los algoritmos criptográficos que utilizarán y generar las claves efímeras que protegerán la posterior transferencia de datos.

El *handshake* constituye una sofisticada secuencia de mensajes que resuelve varios desafíos críticos. Este proceso, aunque invisible para el usuario final, representa la frontera donde se decide la seguridad intrínseca de toda comunicación posterior. Y es precisamente esta fase donde el ataque *harvest now, decrypt later* encuentra su vector de acción ideal: al interceptar los mensajes intercambiados durante el *handshake*, un adversario podría almacenar la comunicación con el fin de descifrarla posteriormente, cuando los ordenadores cuánticos sean capaces de romper los fundamentos matemáticos que sustentan la criptografía subyacente.

Entre los protocolos que implementan mecanismos de *handshake* criptográficos, el más común es TLS/SSL, debido a que figura como el estándar predominante para garantizar comunicaciones web seguras. Pero no es el único. Además de este, existen otros protocolos clave como SSH (acceso remoto seguro), IPsec (redes privadas virtuales), Noise Protocol (redes privadas virtuales modernas como WireGuard), WPA2/WPA3 (protección de redes WiFi), QUIC (transporte web de nueva generación) o Signal Protocol (mensajería privada), que también implementan un handshake como proceso inicial.

Todos realizan en esencia lo mismo, pero lo hacen de formas diferentes y a niveles distintos dentro del modelo OSI, lo que tiene implicaciones tanto en el rendimiento como en la seguridad.

Capítulo 2. Estado del Arte

Protocolo	Descripción
TLS/SSL	Opera en la Capa de Transporte (Capa 4). Establece un canal seguro entre dos partes mediante un <i>handshake</i> con autenticación, intercambio de claves y negociación de suites criptográficas. Protege datos en tránsito, especialmente en aplicaciones web.
SSH	Funciona en la Capa de Aplicación (Capa 7) pero establece el canal seguro a través de la Capa 4. Incluye autenticación y establecimiento de claves. Está orientado al acceso remoto y ejecución de comandos sobre redes inseguras.
IPsec	Opera en la Capa de Red (Capa 3). Asegura el tráfico IP mediante encriptación y establecimiento de túneles seguros, usado en VPNs.
Noise Protocol	Implementado en la Capa de Aplicación (Capa 7) y usado en sistemas como WireGuard. Es rápido, flexible y enfocado en autenticación e intercambio de claves.
WPA2/WPA3	Protocolos de la Capa de Enlace de Datos (Capa 2) que aseguran comunicaciones inalámbricas mediante negociación de claves entre el punto de acceso y el cliente.
QUIC	Funciona en la Capa de Transporte (Capa 4), combina seguridad y rendimiento. Integra el <i>handshake</i> criptográfico directamente con el establecimiento de conexión, optimizado para web moderna.
Signal Protocol	Protocolo de la Capa de Aplicación (Capa 7) usado en mensajería segura. Ofrece intercambio de mensajes cifrados con autenticación sin infraestructura centralizada.

Cuadro 2.1: Resumen de Protocolos y Capas OSI (elaboración propia)

La implicación principal de estas diferencias radica en cómo y dónde se realiza la protección de los datos, y cómo esto afecta al rendimiento y la seguridad. Por ejemplo, los protocolos que operan en capas más bajas tienden a proporcionar protección de extremo a extremo en la red, mientras que los protocolos de capa de aplicación como Signal y SSH permiten una seguridad más granular y dirigida específicamente a servicios como mensajería o acceso remoto. Por otro lado, los protocolos que operan en capas superiores pueden ser más fáciles de implementar en aplicaciones específicas, pero a menudo dependen de protocolos de seguridad en capas inferiores para garantizar una protección completa en la red.

Es evidente que estos protocolos guardan diferencias en su diseño y aplicación final, pero tienen puntos en común en cuanto a los hitos alcanzados durante el proceso de *handshake*. Todos estos protocolos siguen una estructura general similar. Aunque se toma como referencia TLS[5].

2.1. Handshakes Criptográficos en Protocolos de Comunicaciones

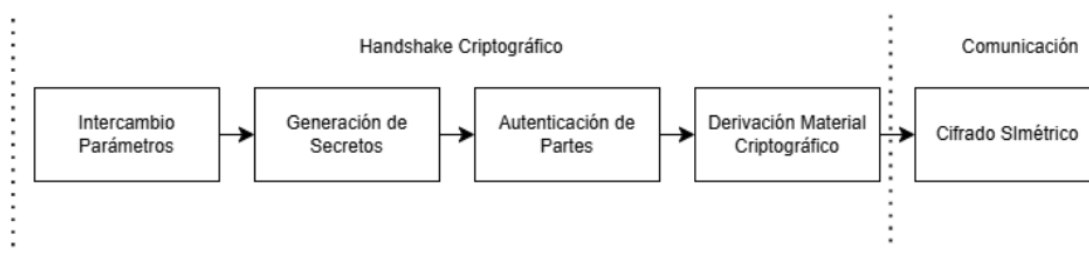


Figura 2.1: Hitos de un Handshake Criptográfico Genérico (elaboración propia)

Como se puede ver en la Fig. 2.1, el proceso de *handshake* se compone de varios hitos clave: primero se produce un intercambio de claves, parámetros y/o suites criptográficas; seguido del establecimiento de secretos compartidos efímeros. A continuación, se lleva a cabo la autenticación de las partes involucradas, tras lo cual se procede a la derivación de las claves de sesión y a la generación del material criptográfico necesario. Finalmente, la comunicación se cifra utilizando dichas claves derivadas mediante técnicas de cifrado simétrico.

En el siguiente apartado, se analizan los fundamentos criptográficos que sustentan las operaciones realizadas en cada uno de estos hitos.

2.2. Criptografía en un Handshake

Extendiendo la definición de la introducción, la criptografía es la disciplina que estudia métodos y técnicas que pretenden garantizar la seguridad de la información en presencia de adversarios. Su propósito es proteger datos frente a accesos no autorizados, alteraciones o falsificaciones, asegurando que solo las partes legítimas puedan acceder y manipular la información de manera confiable.

Cualquier sistema criptográficos digital se diseña siguiendo una serie de principios fundamentales:

- **Confidencialidad:** Garantiza que la información sólo pueda ser leída por las partes autorizadas. Se logra mediante técnicas de cifrado que hacen que los datos sean ininteligibles para quienes no poseen la clave adecuada.
- **Integridad:** Asegura que la información no ha sido alterada de manera no autorizada durante su almacenamiento o transmisión. Se implementa mediante funciones hash y códigos de autenticación de mensajes.
- **Autenticación:** Permite verificar la identidad de los participantes en una comunicación, asegurando que los datos provienen de una fuente legítima. Se logra mediante firmas digitales y protocolos de autenticación.
- **No repudio:** Impide que una entidad pueda negar haber realizado una acción determinada, como el envío de un mensaje o la firma de un documento digital.

Estos principios fundamentales de la criptografía[6] constituyen el pilar sobre el que se construyen los sistemas seguros de comunicación. En el marco del presente análisis, **el ataque considerado constituye una amenaza que compromete exclusivamente el principio de confidencialidad**, sin comprometer los demás pilares de la seguridad. Su objetivo único y exclusivo es acceder al contenido de las comunicaciones interceptadas, sin buscar alterar los datos en tránsito (integridad), suplantar a los participantes legítimos (autenticación) o negar la autoría de los mensajes (no repudio).

Dado que la autenticación no se ve afectada, es posible simplificar el diagrama de hitos representado en la Fig. 2.1, centrando el análisis únicamente en los elementos comprometidos. Además, la fase de intercambio de parámetros, al realizarse frecuentemente en texto plano, tampoco representa un hito relevante desde el punto de vista de la amenaza considerada. Como resultado, el proceso puede resumirse en el nuevo esquema mostrado en la Fig. 2.2, donde se ilustran únicamente los hitos críticos afectados por el ataque.

2.2. Criptografía en un Handshake

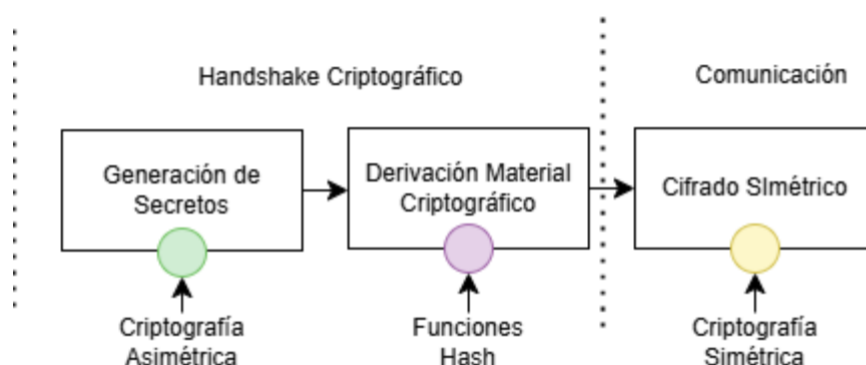


Figura 2.2: Hitos y Tipos de Criptografía de un Handshake (elaboración propia)

En estos tres hitos se emplean diferentes paradigmas de criptografía ; la generación de secretos se realiza mediante **criptografía asimétrica**, la derivación de material criptográfico emplea **funciones hash**, y el cifrado de los datos se lleva a cabo utilizando **criptografía simétrica**.

En el cuadro próximo se detallan algoritmos específicos implicados en protocolos de comunicación según la etapa del *handshake*.

Protocolo	Establecimiento de Claves	Derivación de Material Criptográfico	Cifrado de Datos
TLS	ECDH	HKDF-SHA384	AES, ChaCha20
SSH	ECDH	HMAC-SHA2	AES, ChaCha20
IPsec	ECDH	SHA-384	AES
Noise Protocol	ECDH	Blake2s	AES, ChaCha20
WPA3	ECDH	SHA-384	AES
QUIC	ECDH	SHA-256	AES
SIGNAL	X3DH	HMAC-SHA256	AES

Cuadro 2.2: Algoritmos Utilizados según Protocolo e Hito (elaboración propia)

2.2.1. Criptografía Asimétrica

La criptografía asimétrica es un paradigma criptográfico en el que se emplea un par de claves relacionadas matemáticamente: una clave pública y una clave privada. Esta relación permite que lo que cifra una, solo lo pueda descifrar la otra. La clave pública puede compartirse libremente y se usa para cifrar datos. En cambio, la clave privada debe mantenerse en secreto y se emplea para descifrar los mensajes cifrados con la clave pública, aunque puede tener otras aplicaciones, como autenticación o firma[7].

Este enfoque resolvió uno de los mayores desafíos en los protocolos de comunicación: la distribución segura de claves, permitiendo establecer canales cifrados

sin necesidad de intercambiar una clave secreta previamente. Sin embargo, el cifrado y descifrado asimétrico es computacionalmente costoso, por lo que no es eficiente para cifrar una comunicación completa de esta forma. En su lugar, se usa la criptografía asimétrica sólo para establecer un secreto compartido, el cual se usará posteriormente como clave en un esquema de cifrado simétrico, mucho más eficiente para manejar grandes volúmenes de datos.

Para lograr este intercambio inicial seguro, actualmente se emplean esquemas de intercambio de claves basados en criptografía asimétrica, como el protocolo Diffie-Hellman (DH) y su versión optimizada mediante criptografía de curvas elípticas, conocida como ECDH. La seguridad de estos esquemas asimétricos se basa en la dificultad computacional de ciertos problemas matemáticos. Por ejemplo, RSA se fundamenta en la factorización de números primos grandes —una técnica cada más en desuso—, mientras que ECDH se apoya en el problema del logaritmo discreto sobre curvas elípticas, que ofrece un nivel de seguridad equivalente con claves más pequeñas y mayor eficiencia.

Diffie-Hellman DH

El protocolo Diffie-Hellman (DH) es un algoritmo criptográfico que permite a dos partes (por ejemplo, Alice y Bob) establecer un secreto compartido sobre un canal de comunicación inseguro (como Internet), incluso si un atacante intercepta todos los mensajes intercambiados. Fue propuesto en 1976 por Whitfield Diffie y Martin Hellman, revolucionando la criptografía al introducir el concepto de intercambio de claves seguras sin necesidad de compartirlas previamente. DH se basa en un problema matemático difícil de resolver: el logaritmo discreto. Para entender por qué este problema es tan crucial en criptografía, imaginemos una operación aparentemente sencilla: si sabemos que $3^x = 81$, calcular el exponente x , en este caso 4, a partir del resultado, en este caso 81 es trivial. Sin embargo, en el caso general, $g^a b \bmod p$, cuando $g^a b$ es un número primo de 2048 bits, es decir, un número de más de 600 dígitos decimales, resolver este problema se vuelve computacionalmente inviable. No existen atajos conocidos que permitan despejar el exponente de manera eficiente[8].

Elliptic-curve Diffie-Hellman ECDH

Este protocolo es una variante del protocolo Diffie-Hellman (DH) que utiliza la aritmética de curvas elípticas sobre campos finitos en lugar de la exponenciación modular. Su objetivo sigue siendo el mismo que en DH, permitir que dos partes establezcan un secreto compartido a través de un canal inseguro.

La diferencia fundamental entre ambos protocolos radica en el problema matemático sobre el que se sustentan. A diferencia de DH tradicional, ECDH se apoya en el problema del logaritmo discreto en curvas elípticas. Este cambio aporta una ventaja significativa en términos de seguridad y eficiencia, ya que el problema en curvas elípticas es considerablemente más difícil de resolver con el mismo tamaño de clave. Como consecuencia, ECDH logra el mismo nivel de seguridad que DH con claves mucho más pequeñas, reduciendo el uso de memoria, el consumo de energía y el tiempo de cómputo. Por ejemplo, una clave de 256

bits en ECDH ofrece una seguridad equivalente a la de una clave de 3072 bits en DH tradicional, lo que lo convierte en una opción óptima para dispositivos con recursos limitados.

El funcionamiento de ECDH sigue una estructura similar a DH. Ambas partes acuerdan previamente una curva elíptica y un punto generador sobre un campo finito. A partir de ahí, cada una genera una clave privada, un número aleatorio que se multiplica por el punto generador para obtener su correspondiente clave pública. Posteriormente, intercambian sus claves públicas a través del canal inseguro y, utilizando su propia clave privada, realizan una nueva multiplicación para obtener un secreto compartido. Dado que ambas operaciones conducen al mismo resultado, ambas partes logran derivar la misma clave secreta sin que un atacante pueda reconstruirla a partir de los valores públicos[9].

Estos mecanismos resultaban computacionalmente inviables de revertir para atacantes convencionales. Sin embargo, **el advenimiento de la computación cuántica**, como se ha comentado, altera radicalmente este panorama. Esta vulnerabilidad **afecta a los esquemas asimétricos más modernos**, como ECDH, considerado hasta ahora como el último bastión de la criptografía asimétrica clásica.

2.2.2. Funciones Hash

Una vez completado el primer hito del *handshake*, el intercambio de claves sustentado por criptografía asimétrica, ambos agentes de la comunicación proceden al segundo hito, derivar material criptográfico adicional para establecer una conexión segura. Este proceso se aprovecha de las características de las funciones hash criptográficas.

Una función hash es una operación matemática que transforma una entrada de datos de cualquier tamaño en una salida de longitud fija, conocida como valor hash o resumen. Estas funciones juegan un papel fundamental en la seguridad, ya que permiten generar claves derivadas de manera determinista pero irreversible, asegurando la integridad y confidencialidad de la comunicación.

Es por esto que las funciones hash son fundamentales en muchas aplicaciones de seguridad informática y se utilizan ampliamente en criptografía, almacenamiento de contraseñas y verificación de la integridad de los datos.

Las propiedades fundamentales de una función hash incluyen:

1. **Compresión:** Los datos de entrada de longitud variable se mapean a un dominio con datos de longitud fija.
2. **Irreversibilidad:** No debe ser posible reconstruir los datos originales a partir del valor hash.
3. **Bajo Costo:** El cálculo del valor hash debe ser eficiente en términos computacionales.
4. **Uniforme:** Todas las salidas han de ser equiprobables.

5. **Determinismo:** La misma entrada ha de generar la misma salida siempre.
6. **Resistencia a colisiones:** Debe ser computacionalmente difícil encontrar dos entradas distintas que generen el mismo valor hash.
7. **Difusión:** Un pequeño cambio en los datos de entrada debe resultar en un valor hash completamente diferente.

Estas propiedades hacen que las funciones hash[10] sean de gran utilidad y tengan una amplia gama de aplicaciones. Existen varios algoritmos de hash ampliamente utilizados. Algunos de los más relevantes incluyen se agrupan en SHA-2, que es una familia de funciones hash que incluye SHA-224, SHA-256, SHA-384, SHA-512, etc. Estos algoritmos son ampliamente utilizados en aplicaciones de seguridad, como en la generación de firmas digitales, certificados SSL, y en la creación de direcciones en criptomonedas como Bitcoin. Se exponen dos tipos de funciones hash modernas.

SHA-3: es el último estándar de la familia SHA y se basa en un diseño completamente diferente al de SHA-2. Utiliza el algoritmo Keccak, que tiene un enfoque basado en funciones de absorción de estado interno y es resistente a algunos tipos de ataques que afectan a SHA-2. Aunque es más reciente, se ha adoptado como estándar en aplicaciones donde se necesita mayor seguridad[11] .

BLAKE2: es una función hash criptográfica más rápida que SHA-2, pero igual de segura. Ha sido diseñada con la eficiencia en mente, lo que la hace útil en entornos de alto rendimiento, y es resistente a los ataques de colisión. En un handshake criptográfico, se emplean funciones hash en varias ocasiones. Primero, se utilizan para garantizar la integridad, ya que permiten que una parte calcule el valor hash del mensaje recibido y verifique si ha sido alterado. Sin embargo, como hemos señalado previamente, este aspecto no se ve comprometido por el tipo de ataque en cuestión[11] .

Como se ha menciona, durante el proceso de *handshake*, las operaciones de hashing criptográfico son esenciales para la derivación de secretos compartidos. Las propiedades de estas funciones simulan comportamientos aleatorios, lo que las hace ideales para su aplicación en funciones pseudoaleatorias (Pseudo Random Functions, PRF) y funciones de derivación de claves (Key-Derivation Functions, KDF).

Pseudo Random Functions PRFs

Una función pseudaleatoria (PRF) es el bloque fundamental para los procesos de derivación de claves. Generalmente se define como una función, computable en tiempo polinómico que recibe un índice o semilla y una entrada, y siempre que la entrada sea elegida aleatoriamente y no sea conocida por un observador, la salida de la PRF es computacionalmente indistinguible de una función aleatoria definida en el mismo dominio. Como PRFs se establecen las construcciones HMAC, CMAC y KMAC. Aunque el uso de CMAC no es recomendado debido a que su uso requiere medidas de seguridad adicionales.[12]

2.2. Criptografía en un Handshake

El *Keyed-Hash Message Authentication Code*, HMAC, es una operación originalmente diseñada como *Message Authentication Code*, MAC, y su propósito era garantizar integridad y autenticidad en los mensajes. Con el tiempo le han surgido más aplicaciones, una de ellas es su uso como método pseudoaleatorio. La razón de esto es la estructura interna de HMAC, basada en combinaciones de operaciones hash[13].

$$\text{MAC}(\text{text}) = \text{HMAC}(K, \text{text}) = H(K_0 \oplus \text{opad}) \parallel H((K_0 \oplus \text{ipad}) \parallel \text{text})$$

Como se puede ver en la Fig.2.3, HMAC además del texto de entrada, como una función hash, utiliza unos valores constantes *opad* e *ipad*, y una clave.

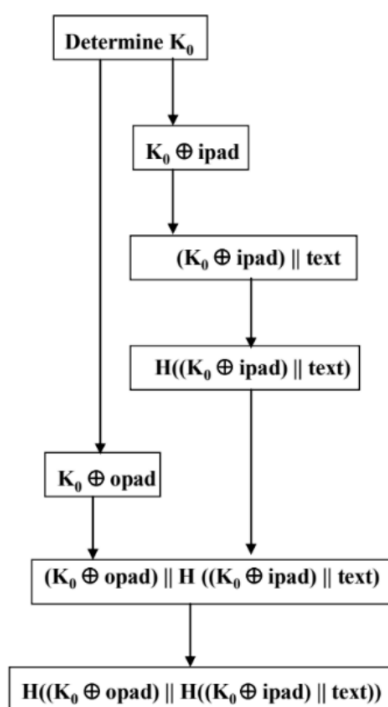


Figura 2.3: Diagrama de Operaciones Internas de HMAC [13]

Según la documentación de HMAC, H es cualquier función hash de las que hemos comentado previamente.

Key-Derivation Functions KDFs

Una función de derivación de claves (KDF) es un componente básico y esencial de los sistemas criptográficos. Su objetivo es tomar una fuente de material de claves y derivar de ella una o más claves secretas criptográficamente robustas. Estas funciones hacen uso de función pseudoaleatorias (PRFs) internamente para derivar el material de clave. Dependiendo de la longitud del material de

clave requerido, puede ser necesario realizar múltiples invocaciones de la PRF durante la construcción de la KDF. Existen diversas opciones, pero uno de los métodos más utilizados y extendidos es el HKDF[12] .

HMAC-based Extract-and-Expand Key Derivation Function, HKDF, esta función sigue el paradigma de *Extract & Expand* para generar material criptográfico de clave. Esta construcción se compone de dos módulos lógicos: el primero se encarga de extraer una clave pseudoaleatoria de longitud determinada que concentre la entropía esparcida en el material de clave de entrada; el segundo módulo expande esta clave en varias claves pseudoaleatorias, que constituyen la salida de la HKDF[14]. La PRF utilizada es HMAC.

El motivo detrás de este enfoque radica en que, en muchas aplicaciones, el material de claves de entrada no siempre está distribuido de manera uniforme, y un atacante puede tener conocimiento parcial o incluso control sobre el material de entrada (como un valor calculado en un intercambio de claves Diffie-Hellman o la recolección de entropía en ciertos entornos). Por esta razón, la fase de extracción tiene como objetivo concentrar la entropía dispersa del material de entrada en una clave pseudoaleatoria corta y robusta criptográficamente.

En algunos casos, el material de entrada ya puede ser una clave pseudoaleatoria de alta calidad, lo que hace innecesaria la fase de extracción. En estos casos, se puede proceder directamente a la etapa de expansión. En esta fase, la clave pseudoaleatoria se expande a la longitud deseada, y el número y la longitud de las claves generadas dependerán de los algoritmos criptográficos específicos que necesiten las claves derivadas.

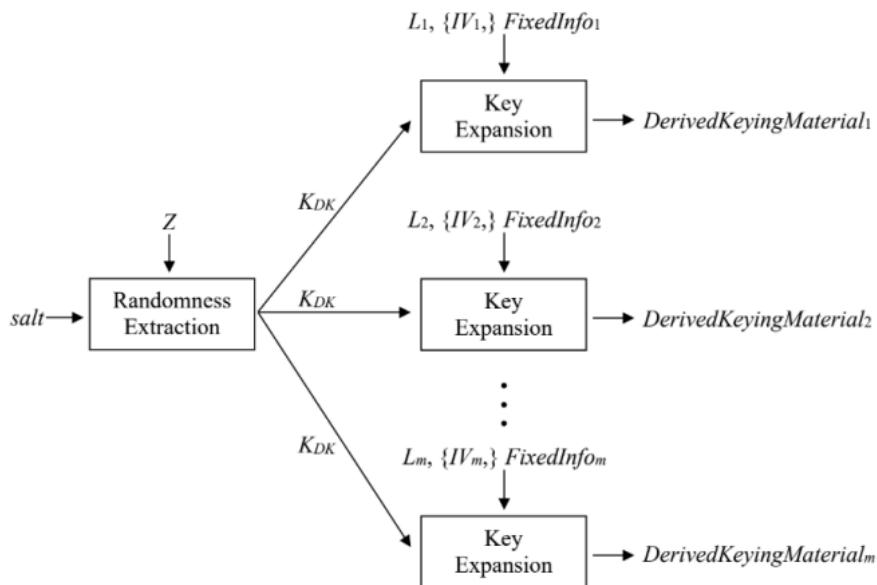


Figura 2.4: Diagrama de Operaciones Internas de HKDF [14]

2.2.3. Criptografía Simétrica

Hasta ahora se han visto los primeros hitos del *handshake* criptográfico y ejemplos de las funciones más comunes, ECDH, HMAC y HKDF. Una vez generado el material criptográfico, el último paso de un *handshake* es el cifrado de la conexión.

Aquí entra la criptografía simétrica[15], paradigma criptográfico en el que el cifrado y el descifrado se realizan con la misma clave secreta. Su principal ventaja radica en su eficiencia computacional, ya que sus algoritmos suelen ser más rápidos y menos exigentes, en términos de recursos computacionales, que los de la criptografía asimétrica. La seguridad de este enfoque depende tanto de la confidencialidad de la clave como de la solidez matemática del algoritmo empleado.

Como se ha mencionado las claves utilizadas se establecen previamente mediante un intercambio asimétrico y su posterior derivación mediante procesos que involucran funciones hash. Es ahora cuando surgen dos opciones: optar por un algoritmo de cifrado de bloque como AES o optar por un algoritmo de cifrado de flujo como ChaCha20.

Advanced Encryption Standard, AES

AES[16] es un algoritmo de cifrado simétrico que opera con bloques de 128 bits y soporta claves de hasta 256 bits. Cifrado de bloque significa que el proceso de cifrado divide los datos de entrada en pedazos del mismo tamaño y los cifra de forma independiente.

La forma de hacerlo combina elementos de una red de sustitución permutación con 10, 12 o 14 rondas, dependiendo del tamaño de la clave. AES es altamente eficiente en hardware y software, con soporte en procesadores modernos que incluyen instrucciones especializadas para mejorar su velocidad. Es considerado uno de los algoritmos más seguros y es ampliamente utilizado

ChaCha20

ChaCha20[17] es un algoritmo de cifrado de flujo diseñado para proporcionar alta seguridad y eficiencia en hardware y software. Cifrado de flujo indica que el cifrado se realiza bit a bit o byte a byte de forma continua. Utiliza una clave de 256 bits y opera mediante una serie de operaciones aritméticas y lógicas simples, lo que lo hace rápido y resistente a ataques de canal lateral. Es utilizado en dispositivos con recursos limitados donde AES puede ser menos eficiente. Su resistencia a ataques criptográficos y su velocidad lo han convertido en una alternativa segura y recomendada en protocolos modernos.

2.3. Criptografía Post-Cuántica - PQC

Una vez se han revisado los principios criptográficos fundamentales involucrados en las fases de un *handshake* criptográfico genérico, a la par que se han identificado los algoritmos comunes que ejecutan estas tareas. Además, se ha señalado la criptografía asimétrica como el componente crítico y el eslabón débil de la cadena, vulnerable a la computación cuántica.

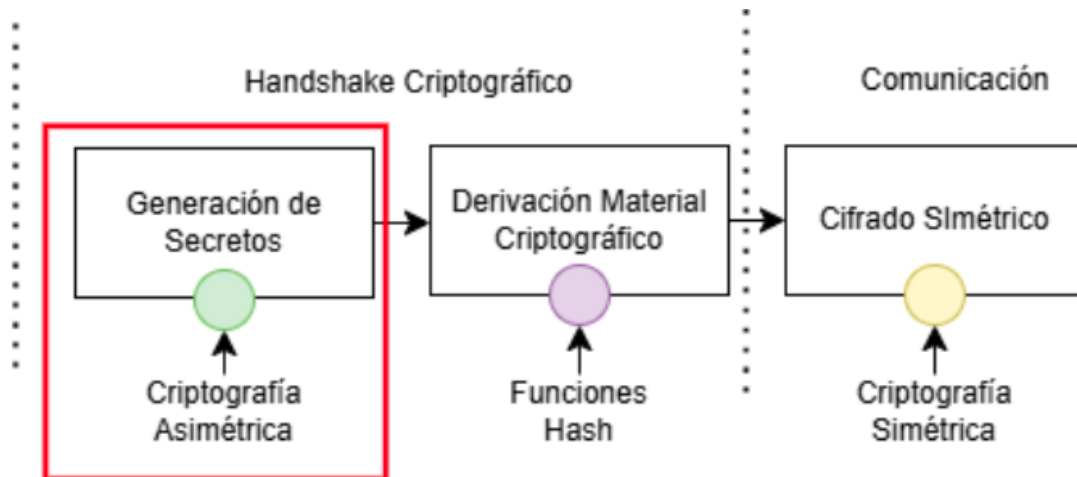


Figura 2.5: Hito Crítico Identificado (elaboración propia)

Es hora de examinar soluciones para reforzar este eslabón débil en la seguridad de las comunicaciones. Una de las principales alternativas es la criptografía post-cuántica, PQC[18], un paradigma que agrupa algoritmos criptográficos diseñados específicamente para resistir los ataques de los futuros ordenadores cuánticos.

Los principios fundamentales de la criptografía post-cuántica están enfocados en desarrollar algoritmos que sean resistentes a los ataques de los ordenadores cuánticos mientras mantienen su eficiencia en los entornos de hardware y software clásicos. Los principales objetivos incluyen:

- **Compatibilidad con Infraestructura Actual:** Asegurarse de que los nuevos algoritmos post-cuánticos sean compatibles con la infraestructura criptográfica existente, como las infraestructuras de clave pública (PKI) y los protocolos de comunicaciones actuales. Esto implica que los nuevos sistemas deben ser interoperables y fácilmente integrables en sistemas ya desplegados.
- **Resistencia a Ataques Cuánticos:** Los algoritmos deben ser diseñados para ser resistentes a los ataques basados en ordenadores cuánticos, como el algoritmo de Shor y otros métodos cuánticos que podrían romper la seguridad de los sistemas tradicionales.

- **Eficiencia en HW y SW:** Los algoritmos deben ser eficientes tanto en términos de hardware como de software. Esto es crucial para garantizar que su adopción sea viable en una variedad de dispositivos, desde grandes servidores hasta dispositivos con limitaciones de recursos.

En el contexto de la criptografía post-cuántica, el National Institute of Standards and Technology (NIST) es clave para la evolución y adopción de estos nuevos algoritmos. En 2016, el NIST inició un proceso de estandarización de algoritmos resistentes a ataques cuánticos, culminando en la selección de varios esquemas principales y de respaldo[19] .

Algoritmos estandarizados, publicados el 13 de agosto de 2024:

- **ML-KEM (Module-Lattice-Based Key Encapsulation Mechanism):** Antes conocido como *CRYSTALS-Kyber*. Estándar principal para cifrado general (FIPS 203[20]). Basado en problemas de retículos (*lattice*), ofrece claves pequeñas y operaciones rápidas, ideal para intercambio de claves seguras.
- **ML-DSA (Module-Lattice-Based Digital Signature Algorithm):** Antes denominado *CRYSTALS-Dilithium*. Estándar principal para firmas digitales (FIPS 204). También basado en retículos, proporciona robustez y eficiencia en autenticación.
- **SLH-DSA (Stateless Hash-Based Digital Signature Algorithm):** Previamente llamado *SPHINCS+*. Estándar secundario para firmas digitales (FIPS 205). Basado en funciones hash, sirve como alternativa con un enfoque matemático distinto.
- **FN-DSA (FFT over NTRU-Lattice-Based Digital Signature Algorithm):** Antes conocido como *FALCON*. Basado en retículos NTRU y transformadas rápidas de Fourier (FFT). Proyectado como cuarto estándar (FIPS 206), enfocado en firmas digitales, con publicación estimada para finales de 2025.
- **HQC (Hamming Quasi-Cyclic):** Respaldo para cifrado general, complementando a ML-KEM. Basado en códigos correctores de errores, ofrece un enfoque alternativo a los retículos. Su estándar está en desarrollo, con borrador esperado para 2026 y finalización en 2027.

De entre estos algoritmos, sólo serviría para reemplazar el intercambio de claves clásico el algoritmo asimétrico ML-KEM[20]. Este esquema, seleccionado por el NIST como estándar para la criptografía post-cuántica, se basa en la dureza computacional del problema del aprendizaje con errores en módulos, una variante del problema de retículas que ha demostrado ser resistente tanto a ataques clásicos como cuánticos. ML-KEM encapsula una clave simétrica mediante el uso de una clave pública, permitiendo así un intercambio seguro sin necesidad de revelar directamente el secreto.

A diferencia de ECDH, que se fundamenta en el problema del logaritmo discreto en curvas elípticas y utiliza operaciones algebraicas sobre curvas definidas so-

Capítulo 2. Estado del Arte

bre cuerpos finitos, ML-KEM opera sobre estructuras de retículas modulares y emplea aritmética de enteros en anillos polinomiales. Esta diferencia estructural no solo implica un cambio en la base matemática, sino también en la forma de implementación: ML-KEM requiere mayor espacio para las claves y los mensajes intercambiados, pero ofrece seguridad frente a adversarios cuánticos, lo que lo convierte en una opción viable para la futura infraestructura de comunicaciones seguras.

2.4. Quantum Key Distribution - QKD

La segunda alternativa al intercambio de claves criptográficas, es la Distribución Cuántica de Claves (QKD)[21], un sistema basada en los principios de la mecánica cuántica que permite el intercambio de claves criptográficas con una característica única: cualquier intento de interceptación es detectable, asegurando así la confidencialidad.

Esta tecnología se apoya en propiedades cuánticas fundamentales, utilizando principalmente la transmisión de fotones individuales a través de fibras ópticas. Los protocolos empleados garantizan que cualquier manipulación no autorizada de la comunicación sea inmediatamente detectable, proporcionando un nivel de seguridad inalcanzable con métodos clásicos.

Así, mientras la criptografía post-cuántica busca adaptar los sistemas existentes, QKD representa un salto tecnológico que aprovecha las leyes de la física cuántica para redefinir los estándares de seguridad en las comunicaciones. Veamos algunos de los protocolos.

Protocolo BB84

El protocolo BB84, propuesto por Charles Bennett y Gilles Brassard en 1984, es el primer y más utilizado protocolo de distribución cuántica de claves. Su seguridad se basa en el teorema de no clonación y el colapso de la función de onda.

- **Codificación en bases cuánticas:** Se basa en la transmisión de bits cuánticos, qubits, codificados en diferentes bases de polarización de fotones. El emisor elige al azar una base para cada qubit enviado, y el receptor mide en una base aleatoria.
- **Detección de ataques:** Si un atacante intenta interceptar y medir los qubits, inevitablemente introduce errores en la transmisión, debido al principio de incertidumbre de Heisenberg. Los extremos de la comunicación pueden detectar esta interferencia comparando públicamente una parte de sus datos sin revelar la clave.
- **Aplicabilidad y estado actual:** Es el protocolo de QKD más implementado y probado, con múltiples implementaciones comerciales y pruebas en redes ópticas de telecomunicaciones. Se han logrado transmisiones de BB84 a cientos de kilómetros mediante fibra óptica y a distancias aún mayores mediante satélites.

Protocolo E91

Propuesto por Artur Ekert en 1991, el protocolo E91 se basa en el fenómeno del entrelazamiento cuántico en lugar de la transmisión de qubits individuales, lo que introduce un nivel de seguridad aún mayor.

- **Uso del entrelazamiento cuántico:** En lugar de enviar directamente qubits, ambos extremos reciben pares de fotones entrelazados generados por una fuente externa. Según la mecánica cuántica, cualquier intento de medición por parte de un atacante afectará instantáneamente el sistema, alterando los resultados esperados.
- **Verificación mediante desigualdades de Bell:** Para detectar la presencia de un agente no autorizado, verifican si las correlaciones entre sus mediciones cumplen las desigualdades de Bell. Si se detectan desviaciones, significa que un tercero ha intentado interceptar la clave.
- **Ventajas y desafíos:** Conceptualmente, E91 ofrece una seguridad más sólida que BB84, ya que no requiere confianza en los dispositivos de los participantes. Sin embargo, es más difícil de implementar debido a los desafíos en la generación y distribución confiable de pares de fotones entrelazados.

El desarrollo de la Distribución Cuántica de Claves avanza hacia soluciones tecnológicas innovadoras que podrían superar sus limitaciones actuales. Una de las líneas más prometedoras es la creación de repetidores cuánticos y redes cuánticas, que permitirían extender significativamente las distancias de transmisión segura y sentar las bases para infraestructuras cuánticas a escala global.

La convergencia con la criptografía post-cuántica emerge como otra dirección estratégica, donde la combinación de QKD con algoritmos resistentes a computación cuántica podría ofrecer soluciones de seguridad integrales para la era post-cuántica. Paralelamente, las investigaciones se centran en mejorar la eficiencia operativa mediante el desarrollo de fuentes de fotones más avanzadas, detectores de mayor sensibilidad y sistemas ópticos optimizados, lo que podría reducir sustancialmente los costes de implementación.

Pese a sus ventajas, la QKD presenta importantes retos técnicos y prácticos. Su implementación está actualmente limitada por factores de distancia y velocidad, siendo viable principalmente en redes de fibra óptica con alcances restringidos. Los costes asociados a su despliegue son significativos, ya que requiere hardware especializado como fuentes de fotones y detectores de estado cuántico.

Otra limitación importante es que la infraestructura QKD no es escalable a nivel global, lo que impide que pueda reemplazar por completo a la criptografía asimétrica tradicional sin avances tecnológicos significativos en redes cuánticas.

En conclusión, la Distribución Cuántica de Claves es una de las soluciones más avanzadas para la seguridad de las comunicaciones en un mundo post-cuántico. Sin embargo, su aplicación generalizada enfrenta barreras técnicas, económicas y de infraestructura que deben ser resueltas antes de que pueda ser adoptada a gran escala.

2.5. Protocolos de Comunicaciones Híbridos

Ante la necesidad de reforzar el eslabón crítico los *handshakes* criptográficos, se conocen dos enfoques viables: la criptografía post-cuántica (PQC) y la Quantum Key Distribution (QKD). La transición hacia nuevas soluciones debe llevarse a cabo con especial precaución, garantizando que el proceso de actualización sea seguro y compatible con los sistemas actuales. En una transición gradual y de bajo riesgo, tiene sentido que la migración a nuevas soluciones incluya inicialmente soluciones híbridas. Igual que se está haciendo con la migración de los coches de combustión a los coches eléctricos, con el uso de motores híbridos que combinan ambos sistemas.

Actualmente existen investigaciones que exploran aproximaciones a la hibridación criptográfica. Como fundamento, debemos recordar que todo sistema criptográfico se compone de dos elementos esenciales: el algoritmo y la clave. Este estudio propone ver la hibridación de una forma simple que puede diferir de enfoques más complejos. No se trata de fusionar algoritmos de distintos paradigmas en un único sistema, lo que equivaldría prácticamente a crear soluciones desde cero, sino de emplear cada paradigma de manera independiente y posteriormente combinar sus fortalezas a través de la combinación de las claves resultantes.

Este enfoque puede incorporar el uso de los nuevos estándares QKD o PQC con los estándares clásicos en los protocolos comprometidos. Mediante una solución híbrida se puede proteger al sistema contra una posible falla criptográfica o de implementación en uno de los algoritmos componentes subyacentes. Esto sería vital en caso de que alguno de los nuevos estándares PQC resultase inseguro. De esta forma no se pondría en riesgo el sistema y, como mínimo, este resultaría tan seguro como el más seguro de los algoritmos subyacentes. Además, el enfoque híbrido proporciona una vía para adaptar el uso de PQC de forma que se cumpla la compatibilidad retroactiva y se respete la interoperabilidad. En este contexto, podría funcionar como una etapa intermedia que facilite la transición hacia una segunda fase, donde las herramientas criptográficas se basen exclusivamente en algoritmos PQC, una vez estos sean maduros y la infraestructura global esté preparada.[4]

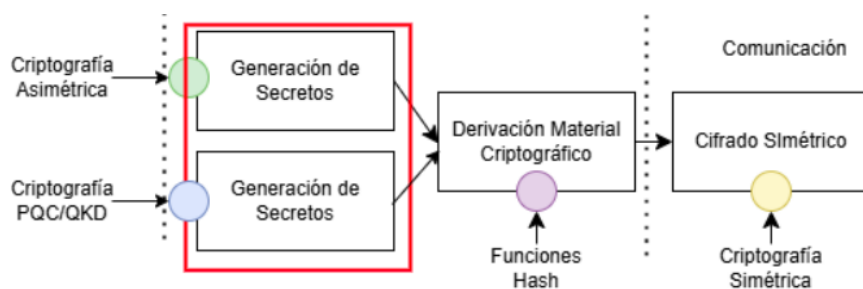


Figura 2.6: Intercambio Híbrido (elaboración propia)

En la figura anterior, se observa cómo el nuevo bloque de intercambio de claves se incorpora de manera paralela al esquema existente. A través de una operación de hibridación entre ambas claves —la clásica y la post-cuántica— se genera un secreto híbrido único, que permite que las etapas posteriores del protocolo continúen funcionando como lo hacían previamente.

A partir de este punto, el estudio se centra específicamente en este enfoque de intercambio de claves híbrido, concebido como una estrategia criptográfica innovadora que combina las fortalezas de diferentes paradigmas criptográficos. Combina la criptografía clásica asimétrica, la criptografía post-cuántica (PQC) y la distribución cuántica de claves (QKD), para generar una clave final híbrida. Esta clave híbrida aprovecha las propiedades de seguridad de los paradigmas involucrados, asegurando la protección de la información frente a amenazas tanto clásicas como cuánticas.

El proceso de intercambio de claves híbrido sigue varios pasos:

1. **Generación de Claves Independientes:** Se generan dos claves por separado, una mediante un algoritmo clásico asimétrico (por ejemplo, RSA o ECC) y otra utilizando un algoritmo post-cuántico (como Kyber o NTRU), o mediante un proceso de QKD para garantizar la resistencia cuántica.
2. **Combinación de Claves:** Una vez generadas las claves, se combinan para formar una clave única híbrida. Esta combinación puede realizarse de diversas maneras. Concatenando las claves y luego aplicando un hash, o mediante la operación XOR entre las claves. Pero siempre creando una nueva clave final que tiene las propiedades de seguridad de ambos paradigmas.
3. **Uso de la Clave Híbrida:** La clave híbrida generada, se puede derivar y utilizar para cifrar y descifrar la información, utilizando algoritmos de cifrado simétrico como AES. Al combinar las propiedades de ambos paradigmas, la clave híbrida garantiza una mayor resistencia a los ataques cuánticos y clásicos.

El intercambio de claves híbrido presenta importantes ventajas como la seguridad redundante (donde si un paradigma es comprometido por ataques cuánticos, el otro mantiene la resistencia de la clave combinada), la transición gradual a la criptografía post-cuántica (permitiendo integrar nuevas soluciones sin reemplazar abruptamente los sistemas clásicos) y una mayor robustez al combinar distintos enfoques criptográficos, añadiendo capas adicionales de protección contra vulnerabilidades. Sin embargo, este modelo también enfrenta desafíos técnicos, como la gestión de diferencias en los tamaños de claves y la mayor complejidad operativa al coordinar múltiples paradigmas. Pese a estos retos, el enfoque híbrido se posiciona como una solución adaptable y resistente para salvaguardar las comunicaciones en la era cuántica, equilibrando seguridad presente y futura sin sacrificar la funcionalidad de los sistemas existentes.

Actualmente existen diversas propuestas para protocolos que establecen conexiones que siguen la mencionada estrategia de hibridación de claves simétricas generadas paralelamente y su posterior hibridación.

2.5.1. Hybrid Key Exchange TLS 1.3 – (TLS-HKE)

La primera propuesta es el borrador *Hybrid Key Exchange*[22] para TLS 1.3, actualmente en su versión 12 y catalogado como *work in progress*, fué publicado por el IETF el 14 de Enero de 2025 y se encuentra en fase de revisión, con validez hasta el 18 de Julio de 2025.

Este borrador sirve como referencia para entender las decisiones que parte de la comunidad está tomando en cuanto a medidas para contener los flancos abiertos por la llegada de la computación cuántica. Gran parte de la comunidad está optando por la integración de lo que están llamando *next generation algorithms* en la infraestructura global. Este término hace referencia, mayoritariamente, a post-quantum algorithms. Aunque de forma prudente eligen este nombre en caso de que finalmente fuesen otros los elegidos.

De todos modos, el rumbo definitivo aún está por determinar, pero es razonable anticipar que, siendo TLS el protocolo más ampliamente utilizado para asegurar comunicaciones en todo tipo de entornos, podría ser uno de los primeros en actualizarse e incorporar las nuevas técnicas. Esto podría posicionarlo como un modelo a seguir por otros protocolos, marcando la pauta en este crucial proceso de transición. En cuanto al contenido del borrador, en la sección *Motivation* se explica que:

“...el intercambio híbrido de claves combina algoritmos tradicionales y next generation, ofreciendo seguridad post-cuántica preliminar mientras se mantiene la protección de los métodos clásicos. Los next generation algorithms, al ser nuevos, generan menor confianza que RSA o Diffie-Hellman, lo que puede retrasar su adopción total. Sin embargo, el intercambio híbrido mitiga riesgos de descifrado retroactivo...”

Esta cita expone las razones existentes para querer adoptar este tipo de esquemas híbridos. Por un lado, se presenta como una solución frente a los ordenadores cuánticos y al ataque mencionado. Por otro lado, el enfoque híbrido da cierta seguridad para aquellos usuarios escépticos que miren de forma desconfiada a estos jóvenes algoritmos que no han sido expuestos al escrutinio público durante un tiempo comparable al de sus antecesores clásicos. De esta forma, no se renuncia a la seguridad ofrecida hasta ahora por estos protocolos y, en el mejor de los casos, incluso la seguridad total del sistema es mayor.

La propuesta se basa en un enfoque híbrido que utiliza la concatenación de secretos generados por diferentes algoritmos. Para ello, sugiere ampliar la negociación de *cipher-suites* para incluir los nuevos estándares PQC, de forma que se tenga que elegir uno clásico y uno PQC. Para posteriormente emplear de forma paralela un intercambio ECDH y un intercambio KEM, generar dos claves simétricas independientes y concatenarlas para formar un único secreto híbrido denominado *shared_secret*. Este secreto híbrido se incorpora al *Key-Schedule* de TLS1.3, que es el algoritmo HKDF mencionado en 2.2.2, reemplazando la clave clásica (EC)DH con la clave híbrida generada, pero la rutina no presenta ningún cambio. Este método introduce reducidas variaciones al protocolo, respetando en gran medida su diseño original y manteniendo la compatibilidad con versio-

nes anteriores.

$$shared_secret = (MyECDH.shared_secret \parallel MyPQKEM.shared_secret)$$

2.5.2. Hybrid Key Exchange SSH – (SSH-HKE)

Otra propuesta de estructura similar es el borrador *PQ/T Hybrid Key exchange*[23] para SSH, actualmente en su versión 6 y catalogado como *work in progress*, también publicado por el IETF, Publicado el 14 de Abril de 2025, este borrador se encuentra en revisión y permanecerá vigente hasta el 16 de Octubre de 2025. En cuanto al contenido del borrador, en la sección *Introduction* se explica que:

“Si bien hoy en día no se dispone de ordenadores cuánticos de tamaño suficiente, un adversario podría registrar la comunicación cifrada enviada entre el cliente y el servidor en una sesión SSH y descifrarla más tarde cuando se disponga de ordenadores cuánticos de tamaño suficiente. Este tipo de ataque se conoce como el ataque harvest now, decrypt later...”

Nuevamente, el propósito del borrador es ofrecer una alternativa a la situación que venimos investigando durante el presente estudio.

La propuesta, sin profundizar en los detalles del protocolo SSH, adopta nuevamente un enfoque híbrido basado en la concatenación de secretos generados por algoritmos clásicos y PQC. Estos secretos concatenados se someten posteriormente a una operación *hash* para derivar el secreto híbrido. Este método conserva la esencia del protocolo SSH, introduciendo cambios mínimos para su adaptación a los nuevos estándares. La siguiente fórmula presenta el resultado de la hibridación. Es importante destacar su similitud estructural con la propuesta presentada en TLS-HKE.

$$K = \text{HASH}(K_{PQ} \parallel K_{EC})$$

Este secreto híbrido, posteriormente, se utiliza en la rutina de derivación propia del protocolo, sin añadir cambios, al igual que en la propuesta TLS. En el documento los autores destacan que este método se ha inspirado en el método del borrador TLS-HKE, lo que confirma lo previamente anticipado; TLS a menudo sirve de modelo para otros protocolos. En contraste con la primera propuesta, que no recomienda combinaciones concretas de protocolos, esta propuesta si define tres combinaciones. Las tres de ellas con una curva elíptica y un mlkem.

- mlkem768nistp256-sha256
- mlkem1024nistp384-sha384
- mlkem768x25519-sha256

2.6. Hibridación de Claves

En la documentación técnica anglosajona, la operación encargada de combinar dos o más claves criptográficas se conoce como *key combiner*, aunque en este trabajo nos referiremos a ella como hibridación de claves.

En las dos propuestas presentadas en los apartados 2.5.1 y 2.5.2, la metodología empleada es esencialmente la misma, lo que implica que la robustez del sistema resultante debería ser comparable en ambos casos. No obstante, la operación de hibridación de claves introduce una diferencia significativa que puede constituir un punto crítico en el sistema. Aunque es un mecanismo relativamente reciente, ya está siendo objeto de atención por parte de la comunidad científica.

En enero de 2025, el Instituto Nacional de Estándares y Tecnología (NIST) publicó el documento SP 800-277[24], que recoge una serie de recomendaciones sobre mecanismos de encapsulación de claves, KEM. Entre sus apartados, destaca uno dedicado específicamente a orientar sobre su uso seguro en aplicaciones prácticas, que es el que resulta de mayor interés en este trabajo.

Transcribiendo el contenido del apartado, el NIST señala que combinar múltiples esquemas de establecimiento de claves en un único sistema puede resultar beneficioso durante el proceso de transición hacia la criptografía post-cuántica. Estos sistemas híbridos se basan en la ejecución paralela de distintos esquemas y en la posterior combinación de sus salidas mediante una operación de hibridación de claves. Como ejemplo, el NIST menciona el esquema X-Wing[25], una propuesta híbrida PQ/T (Post-Quantum/Tradicional) que emplea ML-KEM junto con la curva elíptica X25519. Esta configuración coincide con una de las tres opciones presentadas en la propuesta de SSH-HKE 2.5.2.

A raíz de la recomendación de utilizar esquemas híbridos, el documento especifica que **todas las operaciones aprobadas para la derivación de claves también pueden aplicarse en la hibridación de claves**. Para profundizar en este aspecto, remite a dos documentos clave del propio NIST:

- **Key Derivation Methods** en la SP 800-56C
- **Recommendation for Cryptographic Key Generation** en la SP 800-133

En el caso de X-Wing, este sistema híbrido adopta un enfoque similar al de las propuestas TLS-HKE y SSH-HKE, con algunas diferencias específicas. En particular, al igual que en SSH-HKE, se aplica una función hash a la concatenación de las claves compartidas. Sin embargo, X-Wing también concatena el *ciphertext* y las claves públicas elípticas de ambos agentes antes de aplicar el hash. Como puede observarse, todas estas propuestas siguen estrategias parecidas a lo largo del proceso, pero difieren en el funcionamiento de la operación de hibridación.

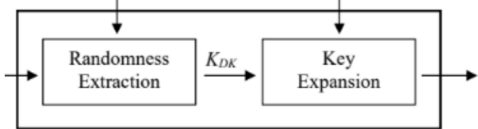
Key Derivation Methods - SP 800-56C

Esta recomendación[26] especifica dos categorías de métodos de derivación de claves que pueden emplearse, según sea necesario, para derivar material clave a

Capítulo 2. Estado del Arte

partir de un secreto compartido Z generado durante la ejecución de un esquema de establecimiento de claves especificado. Originalmente diseñado para los estándares SP 800-56A[27] y SP 800-56B[28] . Esta recomendación permite usar un secreto compartido híbrido $Z = H|T$. Se definen H y T como secretos generados por diferentes fuentes. Una fuente puede ser un algoritmo clásico, 56A o 56B, y la otra fuente PQC.

Se definen dos categorías de métodos de derivación:

One-Step Key Derivation	Two-Step Key Derivation
Las implementaciones de estos KDFs de un solo paso dependen de la elección de una función auxiliar H . Puede ser cualquier función hash simple o una construcción como PRF o KDF	Dos pasos: extracción y expansión. Cada fase recibe diferentes valores de <i>salting</i> . Básicamente, se deriva una clave con una porción de valores <i>salt</i> , y el resto de los valores se usa para ir expandiendo claves a partir de la clave intermedia. 

Cuadro 2.3: Métodos Derivación de Claves según Número de Pasos[26]

El segundo método es, en esencia, el método HKDF, utilizado en la rutina de claves de TLS y especificado en el RFC 5869[14].

El primer paso tiene como objetivo concentrar la entropía dispersa en una clave más corta, pero criptográficamente más fuerte. La segunda fase expande esa clave pseudoaleatoria, generando tantas derivaciones como se necesiten.

Aunque también se indica que la fase *Extract* no es obligatoria y se puede usar la fase *Expand* directamente si la clave ya tiene la entropía distribuida homogéneamente.

Recommendation for Cryptographic Key Generation - SP 800-133

Esta recomendación[29] se centra en la generación de claves criptográficas. Estas claves pueden ser generadas siguiendo los siguientes métodos:

- 1. A través del procesamiento matemático de la salida de Generadores de Bits Aleatorios (RBG).
- 2. Mediante la derivación de una clave desde una contraseña.
- 3. Mediante la derivación de una clave a partir de otra clave.

2.6. Hibridación de Claves

Los dos primeros métodos no tienen aplicación en este estudio. El último es ampliamente utilizado para la obtención de nuevas claves durante una sesión. Se utiliza por SSH y TLS en sus respectivas rutinas de claves. Los métodos recomendados están remitidos a los documentados en el *NIST SP 800-108*, que se expone en el siguiente punto.

Además de los tres métodos de generación de claves, la presente recomendación es la única en abordar la combinación de claves, conocida por este estudio como hibridación. Estos procesos permiten, opcionalmente, incluir otros datos en la combinación.

Las metodologías aprobadas para combinar las claves componentes y otros datos son:

Concatenación	XOR	HMAC
$K = K_1 \mid \dots \mid K_n$ Se puede truncar K para obtener la longitud requerida.	$K = K_1 \oplus \dots \oplus K_n$ Todas las K_i deben tener la misma longitud. Se puede aplicar XOR con otros valores D_i no secretos.	$K = \text{HMAC}(s, K_1 \mid \dots \mid K_n \mid \dots)$ HMAC debe ser una implementación conforme al FIPS 198[13]. Se puede truncar K a la longitud requerida.

Cuadro 2.4: Operaciones de Combinación de Claves[29]

Key Derivation Using Pseudorandom Functions SP 800-108

Esta recomendación[12] se centra en establecer métodos para generar claves adicionales a partir de una clave criptográfica preexistente, generalmente mediante una función que toma como entrada dicha clave junto con otros datos auxiliares, como identificadores de contexto o salting. En particular, define una familia de funciones denominadas PRF (Pseudo-Random Functions), que producen salidas indistinguibles de aleatorias para entradas distintas. Sobre esta base, introduce una segunda familia denominada KDF (Key Derivation Functions), que utilizan internamente dichas PRFs para derivar material criptográfico útil para otros procesos como el cifrado, la autenticación o la integridad.

Ambas definiciones ya fueron presentadas en el apartado 2.2.2, ya que —como establece el propio NIST— estos procedimientos son lo suficientemente flexibles como para ser utilizados tanto en procesos de derivación de claves como en esquemas de hibridación, que es el contexto que actualmente nos ocupa. Esta versatilidad hace que los mecanismos de derivación definidos en esta publicación sirvan de base común para diferentes aplicaciones criptográficas, incluidas las estrategias híbridas de seguridad post-cuántica.

Capítulo 3

Desarrollo

3.1. Hipótesis

Las propuestas híbridas expuestas, TLS-HKE y SSH-HKE, comparten una estructura similar: incorporan, de forma paralela al algoritmo clásico utilizado tradicionalmente, un algoritmo post-cuántico. Sin embargo, uno de los aspectos que ha despertado especial atención en estas propuestas es la operación de hibridación empleada para combinar los secretos generados por cada algoritmo. Esta operación se perfila como un punto crítico cuya correcta implementación resulta esencial para garantizar la seguridad del sistema híbrido.

En el caso de TLS-HKE, esta hibridación se realiza mediante una simple concatenación de los dos secretos, sin la inclusión de ningún campo adicional como un nonce, salt o byte de separación. En cambio, la propuesta para HKE-SSH añade un paso adicional al someter la secuencia resultante de la concatenación a una función hash. Por último X-Wing, va un paso más allá y concatena el *ciphertext* y las claves públicas elípticas de ambos agentes antes de someterlo a la operación de hash.

Estas diferencias en el diseño llaman la atención y plantean preguntas relevantes: ¿por qué uno de los protocolos aplica un hash adicional y el otro no?, ¿tiene esto un efecto significativo —positivo o negativo— sobre la seguridad o eficiencia del protocolo?. Por otro lado, ¿es una variable influyente la longitud de la cadena que se introduce a la operación de hash?. Si fuese trivial elegir una de las opciones debido a que no produce un impacto apreciable, ¿podría considerarse innecesaria ?.

Estas preguntas adquieren aún más contexto si se tiene en cuenta que el resultado de la hibridación se incorpora posteriormente en la rutina de derivación de claves. ¿Influye esto en la decisión, o no, de someter a la concatenación de las claves a un hash?

La hipótesis de este trabajo postula que, a priori, **aplicar únicamente la concatenación de secretos como operación de hibridación influye negativamente en la seguridad del sistema**, en comparación con las alternativas.

3.2. Metodología

Con el fin de analizar el impacto que puede tener la operación de hibridación en la fortaleza criptográfica de un secreto compartido, este estudio adopta una metodología experimental con enfoque cuantitativo y controlado. El objetivo principal es evaluar si distintas formas de hibridar claves influyen significativamente en la calidad criptográfica del resultado final.

Para ello, se han definido cuatro casos de estudio. Tres de ellos representan las principales opciones de hibridación exploradas hasta el momento, mientras que el cuarto se incluye con el objetivo de facilitar la comparación y el análisis de los resultados. Todos los casos comparten un entorno experimental homogéneo, en el que únicamente se modifica el mecanismo de hibridación, manteniéndose constantes el resto de los parámetros criptográficos y de ejecución. Esto permite aislar el efecto de la hibridación y evaluar su impacto con precisión.

La evaluación realizada en este estudio se centra exclusivamente en métricas criptográficas de carácter estadístico, dejando fuera el análisis de eficiencia computacional. La razón principal es que el coste computacional asociado a las distintas operaciones de hibridación resulta despreciable dentro del *pipeline* completo de un proceso de *handshake*. Aunque pueden existir diferencias de rendimiento entre los algoritmos criptográficos individuales empleados, las operaciones de combinación se ejecutan localmente, sin generar una carga significativa ni afectar a la latencia del establecimiento de conexión, que es el parámetro más crítico en protocolos de comunicación. Por tanto, el foco del análisis recae exclusivamente en la seguridad criptográfica, entendida aquí como la cantidad de entropía presente en las claves resultantes. Las métricas empleadas permiten cuantificar el grado de aleatoriedad, uniformidad o dispersión de los datos generados, así como detectar posibles sesgos o patrones estructurales que puedan comprometer la calidad del secreto compartido.

El entorno experimental ha sido programado íntegramente en Python por el autor[30], ejecutado en una máquina local y documentado en un repositorio público, lo que garantiza un control total del proceso y la reproducibilidad de los resultados.

A continuación, se detallan los casos de estudio evaluados, las métricas empleadas y las herramientas utilizadas en el proceso experimental.

3.2.1. Casos de Estudio

Los cuatro casos analizados presentan una estructura análoga, basada en un esquema híbrido que combina una clave clásica con una clave post-cuántica. Para facilitar la evaluación comparativa, se ha optado por mantener constante tanto el esquema clásico como el post-cuántico en todos los casos, ya que el objetivo del estudio es analizar el efecto de los distintos métodos de hibridación.

En el plano clásico, se ha seleccionado la curva elíptica X25519 debido a su amplio uso y alta eficiencia. En cuanto al esquema post-cuántico, se ha elegido Kyber-768, dado que es la opción recomendada tanto por X-Wing como por SSH-

HKE cuando se utiliza en combinación con X25519. Asimismo, todos los casos emplean el mismo algoritmo hash (SHA-256) y la misma función de derivación de claves: HKDF.

Caso	Claves	Hibridación	Derivación
TLS-HKE	ECDH (X25519) + Kyber768	Concatenación directa de los dos secretos	HKDF
SSH-HKE	ECDH (X25519) + Kyber768	Concatenación + hash SHA-256	HKDF
X-Wing	ECDH (X25519) + Kyber768	Concatenación de las claves + claves públicas, seguido de la función hash SHA-256	HKDF
HMAC-HKE	ECDH (X25519) + Kyber768	Concatenación + HMAC	HKDF

Cuadro 3.1: Estructura de los Casos de Estudio (elaboración propia)

Para cada caso, se han realizado mediciones de entropía en tres puntos específicos del flujo de ejecución del *handshake*:

1. En primer lugar se mide la entropía de las claves generadas por separado, antes de la operación de hibridación. Esto permite evaluar la calidad individual de las claves clásicas y post-cuánticas. Dado que los algoritmos de generación de claves son comunes a todos los casos, los resultados en este punto son compartidos.
2. En segundo lugar, se mide la entropía después de aplicar el método de hibridación correspondiente en cada caso.
3. Finalmente, se analiza la entropía tras aplicar la rutina de derivación común a todos los esquemas, HKDF.

Este enfoque permite comparar las diferentes estrategias de hibridación en términos de robustez criptográfica y además observar cómo evolucionan las propiedades estadísticas del secreto compartido a lo largo de las distintas etapas del proceso.

Para la evaluación experimental, se han generado dos conjuntos de 500.000 claves criptográficas, cada una de 32 bytes. El primero compuesto por claves ECDH y el segundo por claves PQC.

3.2.2. Métricas de Evaluación

Como se ha mencionado previamente, las métricas de evaluación que se han utilizado miden exclusivamente la resistencia criptográfica desde un punto de vista estadístico.

Medidas Estadísticas de Tendencia Central

Se evalúan indicadores clásicos de tendencia central y dispersión.

- **Frecuencia absoluta:** cuántas veces aparece cada valor de byte en los datos. Permite identificar sesgos o irregularidades visuales.
- **Media:** mide el valor promedio de aparición de cada byte.
- **Desviación estándar:** cuantifica la dispersión de las frecuencias de aparición respecto a su media. Una desviación baja indicaría una distribución más uniforme, mientras que una alta podría reflejar sesgos.

Entropía de Shannon

En términos generales, la entropía de un sistema cuantifica su nivel de aleatoriedad o dispersión. En el contexto de la información, la entropía representa la cantidad promedio de información contenida en cada símbolo emitido por una fuente de datos. Cuanto más impredecibles son los símbolos, mayor es la entropía.

La entropía de Shannon[31] se define mediante la siguiente fórmula:

$$H(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i)$$

donde $p(x_i)$ es la probabilidad de aparición del símbolo x_i . En este trabajo, al analizar la entropía a nivel de byte, se considera un alfabeto de 256 símbolos (uno por cada posible valor de byte). La probabilidad $p(x_i)$ se estima empíricamente a partir de la frecuencia relativa de aparición de cada byte i en los datos observados:

$$p(x_i) = \frac{f_i}{N}$$

donde f_i es el número de veces que aparece el byte i en la cadena de datos analizada, y N es el número total de bytes.

Dado que la probabilidad teórica de aparición de un byte en una distribución perfectamente uniforme es $\frac{1}{256}$, el valor máximo teórico de entropía de Shannon sería de 8 bits. Este valor corresponde a una cadena de 256 bytes en la que cada uno de los posibles valores de byte (de 0 a 255) aparece exactamente una vez. En este caso, la entropía alcanza su valor máximo de 8 bits. Por tanto, cuanto más se aproxime el valor de entropía de una cadena criptográfica a 8, más uniforme y aleatoria será su distribución.

Tests Estadísticos

Estas métricas se han complementado con dos herramientas estadísticas más rigurosas para comparar distribuciones:

- **Test Chi-cuadrado:** compara las frecuencias observadas con las esperadas bajo una distribución uniforme. Una estadística alta indica que hay diferencias significativas entre ambas, lo cual sugiere que la distribución no es uniforme.
- **Distancia de Jensen-Shannon:** mide la similitud entre dos distribuciones de probabilidad. Cuanto menor sea esta distancia, más cercana está la distribución empírica a la teórica uniforme.

Aunque estos exámenes son útiles no son suficientes, existen casos en los que generadores de claves criptográficos crean claves siguiendo patrones repetitivos como el siguiente, donde se generan los bytes en orden creciente:

1. **Clave 1:** 00 01 02 ...7F
2. **Clave 2:** 80 81 82 ...FF
3. **Clave 3:** 00 01 02 ...7F (y se repite el ciclo)

Si se analiza un conjunto suficientemente grande de estas claves, se puede ver que cada byte aparece el mismo número de veces. La frecuencia de aparición de los bytes es perfectamente uniforme. La entropía de Shannon se acerca al valor máximo (8 bits). La desviación típica de las frecuencias es prácticamente cero. El test Chi-cuadrado y la Jensen-Shannon dan resultados perfectos.

Se estaría ante un caso en el que la distribución estadística parece aleatoria pero es completamente predecible. Las claves siguen un patrón determinista, monótono y predecible. En este caso es muy obvio pero pudiese ser más complejo. Esto significa que la entropía estadística es alta pero la entropía criptográfica es nula. Es un generador altamente predecible y por tanto las claves no son robustas.

Para evitar esta circunstancia, se propone un último test:

- **Test de Correlación:** mide correlaciones entre bytes sucesivos siguiendo un análisis de frecuencia condicional o un modelo de Markov de primer orden.

Para realizar este test, se construye una matriz condicional de la forma $p(\text{byte}_j | \text{byte}_i)$, que representa la probabilidad de que aparezca el byte j dado que previamente haya aparecido el byte i .

El uso conjunto de todas las métricas ha permitido evaluar no sólo cuán aleatoria es la distribución de bytes, sino también detectar patrones, anomalías o deficiencias en los generadores de claves desde un enfoque tanto informacional como estadístico.

3.2.3. Herramientas y Tecnologías Utilizadas

Para llevar a cabo la evaluación experimental de este trabajo, se han utilizado las siguientes tecnologías, bibliotecas criptográficas y herramientas estadísticas. A continuación, se detallan los elementos más relevantes del entorno de desarrollo y análisis:

- **OpenSSL (versión 3.5):** se ha utilizado como base para la implementación de los algoritmos criptográficos clásicos, específicamente para el establecimiento de claves mediante ECDH sobre la curva X25519, así como para la gestión de funciones de derivación de claves como HMAC, HKDF y SHA-256. Esta biblioteca se ha empleado mediante su versión adaptada a Python: `cryptography`[32][33].
- **liboqs – Open Quantum Safe:** se ha utilizado esta biblioteca para incorporar algoritmos post-cuánticos. En concreto, se ha integrado el esquema Kyber768 como algoritmo de intercambio de claves post-cuántico. Al igual que en el caso anterior, se ha usado su versión adaptada a Python: `oqs-python`[34][35].
- **Python:** ha sido el lenguaje principal para la implementación del entorno de pruebas y análisis. Se han utilizado bibliotecas como `NumPy`, `SciPy` y `matplotlib` para el tratamiento de datos y la elaboración de análisis estadísticos.

Capítulo 4

Resultados

4.1. Medidas de Tendencia Central

Para cada conjunto de claves, se ha calculado la **entropía global**, que refleja el grado de aleatoriedad del conjunto completo; la **entropía media individual**, que representa la entropía promedio por clave.

Se incluyen también el número total de bytes procesados, la **media teórica esperada** de las frecuencias de aparición de byte (en una distribución uniforme) y la **desviación estándar**. La siguiente tabla resume estos indicadores clave para cada uno de los esquemas analizados:

Esquema	Entropía Global	Entropía Media Ind.	Total Bytes	Media	Desviación Estándar
X25519	7.9993	4.8816	16 M	62.500	1989.06
ML-KEM768	8	4.8817	16 M	62.500	259.31
TLS-HKE	7.9998	5.7656	32 M	125.000	2012.64
SSH-HKE	8	4.8818	16 M	62.500	232.03
X-Wing	8	4.8817	16 M	62.500	264.10
HMAC-HKE	8	4.8817	16 M	62.500	254.74
TLS-HKE-HKDF	8	4.8819	16 M	62.500	244.46
SSH-HKE-HKDF	8	4.8818	16 M	62.500	250.68
X-Wing-HKDF	8	4.8818	16 M	62.500	247.05
HMAC-HKE-HKDF	8	4.8817	16 M	62.500	236.53

Cuadro 4.1: Indicadores Clave según Caso de Estudio e Hito del Handshake (elaboración propia)

4.2. Histogramas de Distribución de Bytes

Los histogramas han sido obtenidos a partir de las frecuencias absolutas de aparición de los bytes generados. Representando visualmente la distribución de aparición de cada uno de los 256 valores de byte posibles.

Cada barra indica cuántas veces aparece un byte específico en el conjunto de claves. En una distribución idealmente uniforme, todas las barras deberían estar próximas a una misma altura.

Estos gráficos permiten observar la homogeneidad en la generación de claves de ambos sistemas, así como identificar cualquier desviación o patrón inesperado.

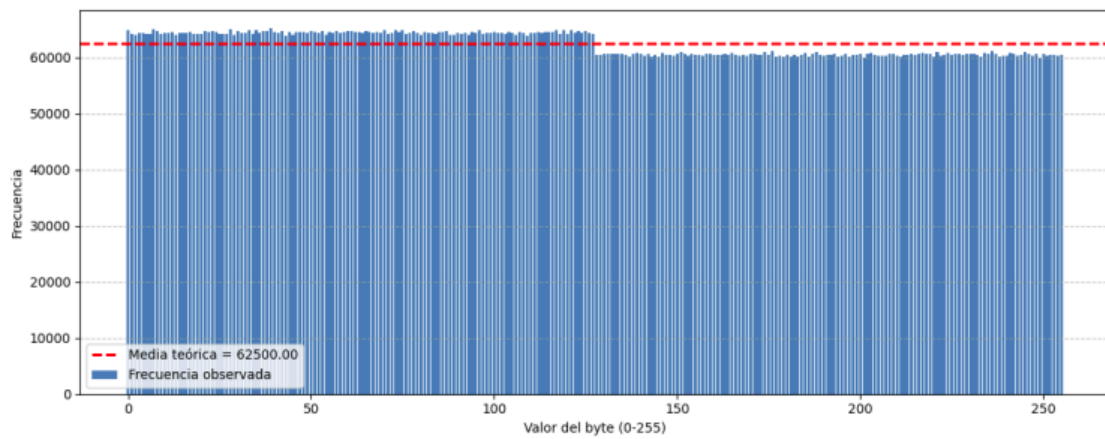


Figura 4.1: Distribución de Frecuencias de Aparición de un Byte en ECDH (elaboración propia)

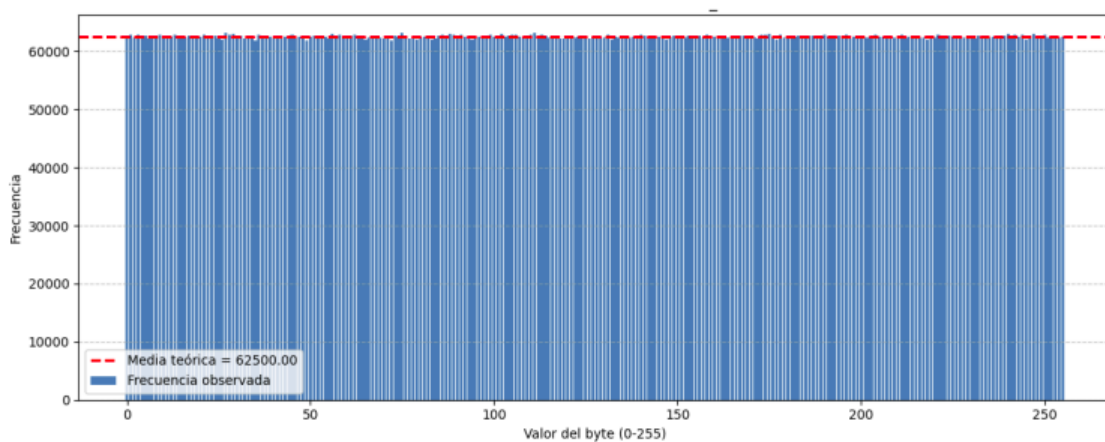


Figura 4.2: Distribución de Frecuencias de Aparición de un Byte en KEM

Las Figuras 4.1 y 4.2 muestran los histogramas correspondientes a las claves iniciales comunes a todos los casos de estudio. Estas claves son posteriormente procesadas a través de los 4 métodos definidos.

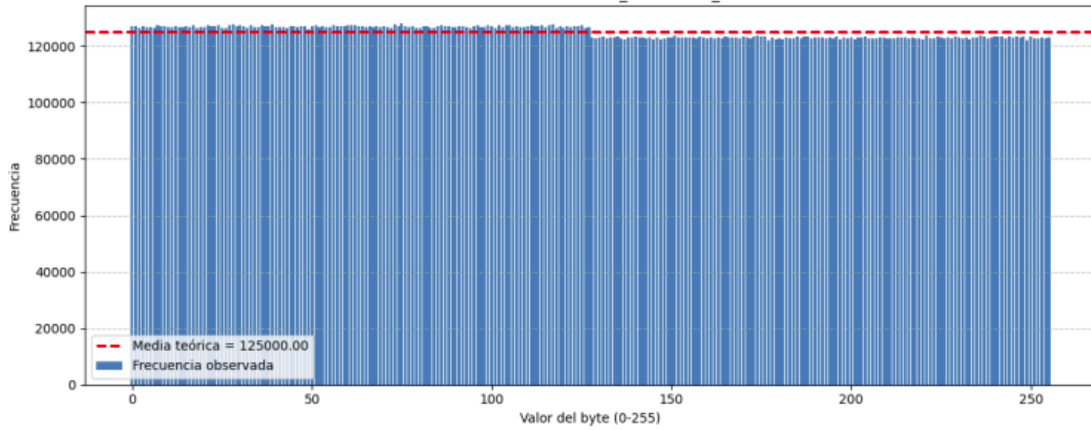


Figura 4.3: Distribución de Frecuencias de Aparición de un Byte en Concatenación TLS-HKE (elaboración propia)

Se presenta exclusivamente el histograma correspondiente a la variante de concatenación directa TLS-HKE, Figura 4.3, ya que es la única que muestra una distribución que arroja información relevante. Este caso ilustra un sesgo en la distribución de frecuencias similar a la figura 4.1, mientras que el resto de combinaciones no presenta dicho patrón.

Por este motivo, no se incluyen los histogramas de las demás variantes híbridas (SSH-HKE, X-Wing, HMAC-HKE) ni de sus versiones derivadas con HKDF, dado que todos ellos presentan distribuciones visualmente idénticas a la observada en la Figura 4.2. Estas distribuciones son completamente uniformes y sin patrones apreciables.

4.3. Tests Estadísticos

Chi² y Distancia Jensen-Shannon

Para evaluar rigurosamente la distribución de los bytes generados por los distintos mecanismos criptográficos, se han aplicado dos pruebas estadísticas complementarias:

- **Test Chi-cuadrado:** compara las frecuencias observadas de cada valor de byte (0-255) con las esperadas bajo una distribución uniforme ideal. Un valor elevado de la estadística χ^2 y un p-valor bajo indican que la distribución observada difiere significativamente de la uniforme.
- **Distancia de Jensen-Shannon (JSD):** mide la divergencia entre la distribución empírica y la distribución uniforme, proporcionando un valor entre

Capítulo 4. Resultados

0 y 1. Cuanto menor sea la JSD, mayor es la similitud entre ambas distribuciones.

La siguiente tabla resume los resultados obtenidos para cada fichero de claves analizado. En ella se incluyen los valores de la estadística χ^2 , su p-valor asociado, y la distancia de Jensen–Shannon:

Esquema	χ^2	p-valor	JSD
X25519	16205	0.0000	0.0135
ML-KEM768	275	0.1813	0.0018
TLS-HKE	8295	0.0000	0.0068
SSH-HKE	220	0.9419	0.0016
X-Wing	285	0.0906	0.0018
HMAC-HKE	265	0.3080	0.0017
TLS-HKE-HKDF	244	0.6659	0.0017
SSH-HKE-HKDF	257	0.4463	0.0017
X-Wing-HKDF	250	0.5767	0.0017
HMAC-HKE-HKDF	229	0.8761	0.0016

Cuadro 4.2: Resultados del test Chi-cuadrado y distancia de Jensen–Shannons según Hito del Handshake (elaboración propia)

Test de Correlación

En este último apartado, se presentan gráficos de correlación que permiten analizar la dependencia lineal entre pares de bytes dentro de cada clave. El objetivo es detectar posibles patrones o relaciones no aleatorias entre diferentes bytes, lo cual podría indicar una falta de independencia entre los bytes generados.

Para cada conjunto de claves (ECDH, KEM, concatenación, hash, HMAC, X-Wing y sus variantes derivadas), se ha calculado la matriz de correlación entre un byte y el próximo para todos los de bytes. Estas matrices se representan gráficamente como mapas de calor.

Del mismo modo que con los histogramas, no se presentan los mapas de calor de correlación entre bytes para las variantes híbridas ni para sus versiones derivadas con HKDF, ya que no aportan información adicional relevante. En todos los casos analizados, los mapas muestran patrones visualmente aleatorios y sin estructuras discernibles, de forma prácticamente idéntica a lo observado en la Figura 4.5.

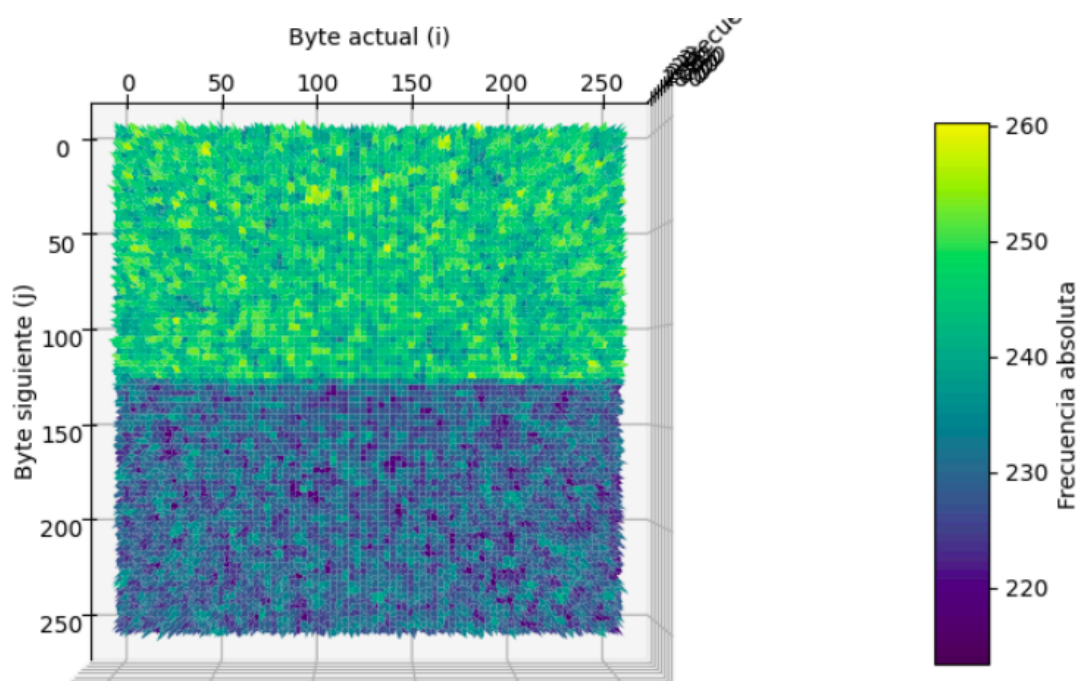


Figura 4.4: Mapa de Calor de Correlación entre Bytes ECDH (elaboración propia)

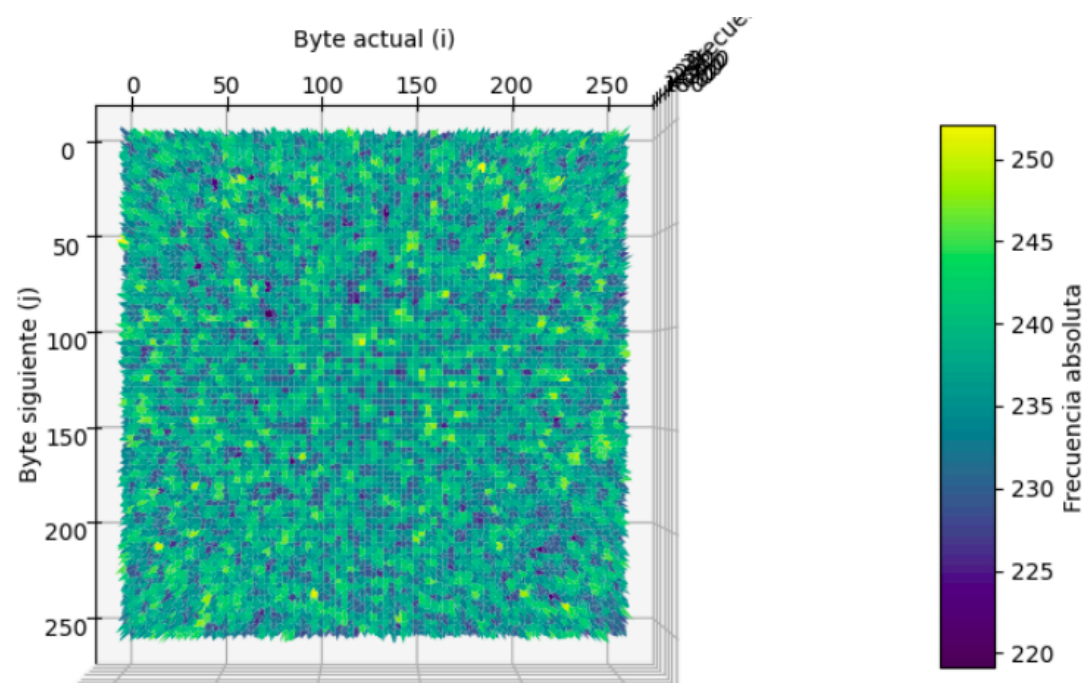


Fig 20. Heatmap de Correlación entre Bytes KEM

Figura 4.5: Mapa de Calor de Correlación entre Bytes KEM (elaboración propia)

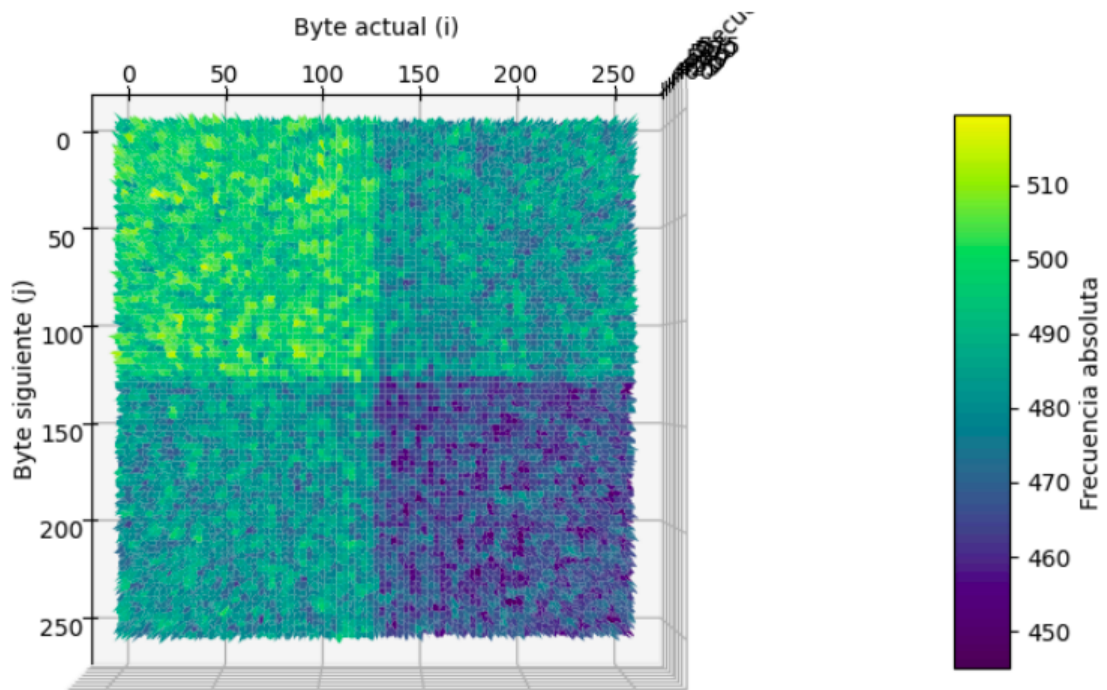


Figura 4.6: Mapa de Calor de Correlación entre Bytes Concatenation TLS-HKE (elaboración propia)

Capítulo 5

Análisis

5.1. Análisis de los Resultados

Antes de proceder a responder a las preguntas planteadas en la hipótesis es preciso hacer un comentario de los resultados. En cuanto a la tabla del Cuadro 4.1 se observan varias tendencias relevantes:

Desviación Estándar

Aquí se aprecian diferencias significativas. X25519 y TLS presentan una desviación estándar alta (1989,06 y 2012,64 respectivamente), señal de una variabilidad más marcada y menos uniforme en la aparición de ciertos bytes. En cambio, los métodos híbridos y sus versiones derivadas muestran desviaciones mucho más contenidas (en torno a 230–260), lo que apunta a distribuciones más homogéneas.

Entropía Global cercana a 8

Todos los conjuntos de 500.000 claves presentan una entropía, a nivel de conjunto, superior a 7.993 en todos los casos. Esto sugiere que son distribuciones muy cercanas a la ideal en la que todos los valores de byte (0–255) aparecen con la misma probabilidad.

Entropía Media Individual

Se observa que todos los tipos de claves, de 16B, tienen valores de entropía medios de alrededor de 4.88 bits. En cambio, las claves de TLS-HKE, de 32B, tienen un valor de 5.76 bits, reflejando la influencia que tiene el tamaño de la clave en la entropía máxima que puede tener una única clave. Estos valores concuerdan con los valores esperados que deben situarse en la curva observada en la siguiente figura.

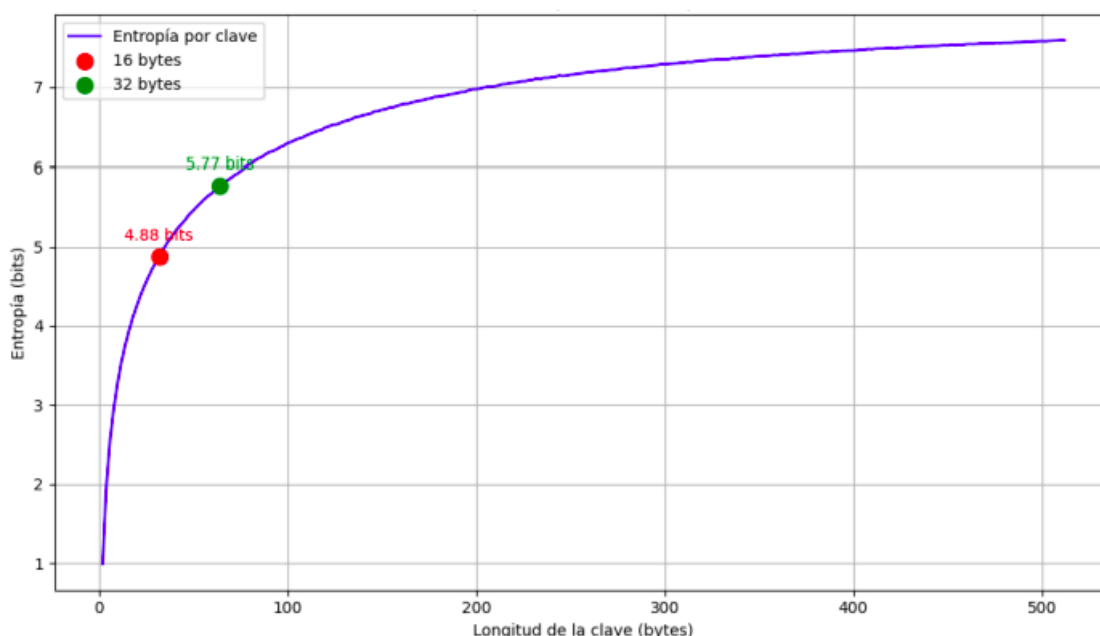


Figura 5.1: Entropía Máxima Teórica según Tamaño de Clave (elaboración propia)

Histogramas

Los resultados obtenidos a partir de los histogramas de distribución de bytes muestran que la mayoría de los tipos de claves presentan una aleatoriedad uniforme, lo cual es un comportamiento deseable en contextos criptográficos. Sin embargo, se observan excepciones notables en los casos de X25519 y TLS.

En la Figura 4.1, correspondiente a X25519, se aprecia un patrón característico: los primeros 128 posibles valores de byte tienen frecuencias de aparición por encima de la media esperada, mientras que los últimos 128 valores aparecen con frecuencias por debajo de la media esperada. Esta asimetría sugiere una desviación respecto de una distribución verdaderamente uniforme. Este mismo patrón se reproduce en la Figura 4.3, correspondiente a las claves concatenadas del modelo TLS-HKE. Dado que la mitad de la clave en TLS-HKE proviene directamente de X25519, el sesgo observado no es un fenómeno aislado, sino un comportamiento heredado de la distribución no uniforme introducida por X25519.

Correlación entre Bytes

Los mapas de calor que representan la frecuencia con la que un byte sigue a otro refuerzan los resultados obtenidos previamente en los histogramas y en la tabla de estadísticas. En general, la mayoría de los tipos de claves presentan distribuciones aparentemente aleatorias en estos mapas, lo cual sugiere que no hay patrones evidentes en la generación de bytes sucesivos.

Pero como ya se ha descubierto, destacan como excepciones los casos de X25519 y TLS. En la Figura 4.4, correspondiente a X25519, el mapa de calor se divide visualmente en dos zonas: la mitad superior muestra una concentración de colores cálidos (amarillos), que indican frecuencias por encima de la media, mientras que la mitad inferior se tiñe de colores fríos (azules), lo que señala frecuencias por debajo de la media. Esta asimetría sugiere un sesgo estructural en la distribución de bytes consecutivos y respalda el patrón observado en el histograma respectivo de esta clave.

Por su parte, la Figura 4.6, que representa TLS-HKE, muestra un patrón más complejo dividido en cuatro cuadrantes. Dos de ellos, situados en la diagonal, presentan una distribución aparentemente uniforme. Sin embargo, los otros dos cuadrantes muestran una clara asimetría: uno predominantemente amarillo y el otro azul. Este comportamiento se comprende al considerar que la clave TLS-HKE está compuesta por la concatenación de dos mitades: una proveniente de KEM —que presenta una distribución uniforme— y otra de X25519, que hereda el sesgo observado en los primeros 128 bytes.

Tests de Uniformidad

Por último, se ha evaluado la calidad estadística de la distribución de los bytes en las distintas claves mediante la prueba de chi-cuadrado (χ^2) y la distancia de Jensen-Shannon (JSD). Las claves generadas con X25519 y TLS presentan los valores más altos en la prueba de chi-cuadrado:

- **X25519:** $\chi^2 = 16,205,26$, p-valor $\approx 0,0000$, JSD = 0.0135
- **TLS-HKE:** $\chi^2 = 8,295,89$, p-valor $\approx 0,0000$, JSD = 0.0068

Estos resultados permiten rechazar con total confianza que x25519 y TLS-HKE siguen una distribución uniforme, indicando una desviación estadísticamente significativa. En términos de JSD, ambos también presentan las mayores distancias respecto a la distribución ideal, lo que refuerza la evidencia de sesgo ya identificada visualmente en histogramas y mapas de calor.

En contraste, el resto de métodos, incluyendo ML-KEM768 y todas las variantes híbridas (SSH, HMAC, X-Wing), presentan valores de χ^2 mucho más bajos y p-valores claramente no significativos (todos superiores a 0,09), lo que impide rechazar la hipótesis de uniformidad. Los valores de JSD en estos casos se sitúan en torno a 0,0016–0,0018, indicando una distribución de bytes mucho más próxima al ideal teórico.

Una vez aplicada la rutina de derivación de claves HKDF a todas las combinaciones híbridas, se observa una mejora generalizada en la distribución estadística de los bytes. En particular, la clave originalmente más sesgada (TLS-HKE) muestra una mejora drástica, reduciendo su χ^2 de 8.295 a 244,79, con un p-valor de 0,6659. Todas las variantes derivadas con HKDF alcanzan p-valores elevados (entre 0,44 y 0,87) y mantienen JSD bajos (0,0016–0,0017), confirmando que

la derivación reduce eficazmente las carencias estructurales de las claves originales.

En este conjunto de resultados, HMAC - HKDF destaca como la clave con la mejor distribución estadística global ($\chi^2 = 229,16$, $p = 0,8761$), seguida de cerca por SSH - HKDF y X-Wing - HKDF, con diferencias prácticamente indistinguibles entre sí.

¿Qué sucede con X25519?

Los análisis realizados revelaron un patrón de sesgo en las claves generadas mediante la curva X25519, que además ha arrastrado el modelo TLS-HKE debido a su concatenación simple; los primeros 128 bytes tienden a aparecer con una frecuencia mayor a la media, y los últimos 128, menor a la media. Este fenómeno ha sido confirmado tanto en histogramas de distribución de bytes como en mapas de calor de correlación, y también se refleja claramente en las pruebas estadísticas de uniformidad, como chi-cuadrado y Jensen-Shannon Divergence (JSD).

Al parecer este comportamiento es esperado según información encontrada en el RFC 7748 *Security Considerations for Elliptic Curve Cryptography*[36]. La curva X25519 implementa una serie de optimizaciones diseñadas para mejorar el rendimiento y la seguridad práctica. Una de estas características clave es el uso del llamado *clamping* sobre las claves privadas, que fija determinados bits del escalar antes de realizar la operación de multiplicación escalar en la curva. Esta operación restringe el espacio efectivo de claves privadas y puede introducir ciertos patrones o regularidades en los valores resultantes, especialmente en las claves públicas o compartidas.

Además, X25519 opera sobre un campo de 255 bits y tiene un cofactor de 8, lo que implica que algunos puntos pueden tener orden reducido, afectando así la contribución efectiva del secreto privado. Aunque esto no compromete la seguridad criptográfica del protocolo, sí puede influir en la distribución estadística de los bytes generados, como se ha evidenciado en los resultados experimentados.

Asimismo, el RFC 7748[36] expone otra curva, la X448, que no presenta estos mismos efectos. X448 trabaja sobre un campo de 448 bits y tiene un cofactor menor (4), lo que reduce la probabilidad de que puntos de orden reducido influyan significativamente. Además, aunque también utiliza una forma de *clamping*, esta es menos restrictiva debido al mayor tamaño del escalar, lo que amplía el espacio efectivo de claves y distribuye mejor la entropía. Los resultados estadísticos sobre claves generadas con X448 muestran una distribución de bytes mucho más uniforme, sin sesgos detectables, y con valores de chi-cuadrado y JSD bien dentro de los rangos esperados para datos aleatorios. Esto sugiere que la mayor longitud, el menor cofactor y el diseño interno de X448 contribuyen a una salida estadísticamente más equilibrada.

5.2. Extensión del Análisis

Para completar este estudio de manera rigurosa y comparativa, se ha decidido extender el experimento de análisis de uniformidad, inicialmente aplicado a X25519, a otras dos curvas ampliamente utilizadas: X448 y NIST P-256.

Para ello, se han generado claves ECDH utilizando las tres curvas y se han sometido a las mismas pruebas estadísticas de distribución: chi-cuadrado (χ^2) y Jensen-Shannon Divergence (JSD).

Esta ampliación del análisis busca determinar si los patrones de sesgo observados en X25519 son específicos de dicha curva o si también pueden manifestarse en otras implementaciones ECDH. Además, permite establecer una comparación directa del comportamiento estadístico entre estas tres curvas, lo que contribuye a una evaluación más precisa del nivel de aleatoriedad en sus salidas. Los resultados obtenidos permiten fundamentar mejor las decisiones de diseño en contextos criptográficos donde la distribución uniforme de los bytes es un factor relevante.

Esquema	Chi ²			p-valor			JSD		
	X25519	X448	P-256	X25519	X448	P-256	X25519	X448	P-256
ECDH	16205	240	276	0.0000	0.7334	0.1702	0.0135	0.0012	0.0018
ML-KEM768	275	258	246	0.1813	0.4327	0.6373	0.0018	0.0017	0.0017
TLS-HKE	8295	243	272	0.0000	0.6798	0.2185	0.0068	0.0010	0.0012
SSH-HKE	220	258	267	0.9419	0.4341	0.2805	0.0016	0.0017	0.0017
X-Wing	285	305	266	0.0906	0.0167	0.3038	0.0018	0.0019	0.0017
HMAC-HKE	265	228	264	0.3080	0.8853	0.3253	0.0017	0.0016	0.0017
TLS-HKE-HKDF	244	260	232	0.6659	0.3880	0.8461	0.0017	0.0017	0.0016
SSH-HKE-HKDF	257	223	241	0.4463	0.9234	0.7257	0.0017	0.0016	0.0016
X-Wing-HKDF	250	230	205	0.5767	0.8665	0.9900	0.0017	0.0016	0.0015
HMAC-HKE-HKDF	229	218	257	0.8761	0.9547	0.4363	0.0016	0.0016	0.0017

Cuadro 5.1: Comparativa Estadística de Chi², p-valor y JSD para Distintas Curvas (X25519, X448, NIST P-256) (elaboración propia)

Como se observa, los peores resultados se dan en los esquemas que emplean **X25519**, especialmente en el caso base de ECDH (sin hibridación), que arroja un valor extremadamente alto de χ^2 (16205) y una JSD de 0.0135. Estos valores indican una fuerte desviación respecto a una distribución uniforme. En cambio, tanto X448 como NIST P-256 presentan valores de χ^2 y JSD mucho más reducidos y p-valores elevados, lo que sugiere una mayor aleatoriedad y homogeneidad en sus salidas.

Este patrón se mantiene en los esquemas híbridos. En **TLS-HKE**, que concatena directamente la clave ECDH con la de Kyber768, el sesgo de X25519 persiste claramente, aunque algo atenuado (por ejemplo, $\chi^2 = 8295$). En comparación, TLS-HKE con X448 o P-256 se comporta de forma mucho más equilibrada. Lo

mismo se observa en las demás variantes de hibridación: **SSH-HKE**, **X-Wing** y **HMAC-HKE**, donde el impacto negativo de X25519 es notable frente a las otras curvas.

Sin embargo, una vez aplicada la función de derivación **HKDF**, la situación cambia radicalmente. En todos los casos derivados —incluidos aquellos basados en X25519— los valores de χ^2 y JSD caen a niveles bajos, y los p-valores aumentan considerablemente. Esto sugiere que HKDF actúa como un mecanismo de “normalización estadística”, suavizando los sesgos introducidos en etapas anteriores del proceso. Por ejemplo, **TLS-HKE-HKDF** con X25519 reduce su χ^2 a 244 y su JSD a 0.0017, situándose en el mismo rango que sus equivalentes con X448 o P-256.

En conjunto, los resultados evidencian que la elección de la curva ECDH tiene un efecto significativo sobre la distribución estadística del secreto híbrido. Mientras que X448 y P-256 presentan un comportamiento consistente y balanceado, X25519 introduce desviaciones estadísticamente detectables que pueden propagarse si no se corrigen. La aplicación de HKDF demuestra ser eficaz para neutralizar estos efectos, homogeneizando la calidad criptográfica final independientemente de la curva base empleada.

Capítulo 6

Discusión

A lo largo de este punto se responden a las preguntas planteadas en el punto 3.1.

El análisis estadístico realizado sobre los cuatro métodos de hibridación de claves ha permitido obtener evidencias que responden directamente a las preguntas planteadas en la hipótesis. Si nos centramos exclusivamente en la operación de hibridación, y comparamos la concatenación simple empleada en TLS-HKE con alternativas que aplican una función hash (como en SSH-HKE) o construcciones equivalentes con mayor complejidad estructural (como X-Wing o HMAC), podemos concluir que **sí existe un impacto positivo al aplicar funciones hash sobre los secretos generados.**

Los resultados muestran que la concatenación por sí sola no aporta propiedades criptográficas adicionales al secreto híbrido. Su seguridad depende exclusivamente de la calidad individual de los secretos que se combinan. Esto no significa que sea una opción inválida, pero sí implica que es fundamental conocer con precisión la entropía y uniformidad de las claves de entrada. Si una de ellas presenta debilidades criptográficas o distribuciones sesgadas, estas deficiencias se propagan al resultado final sin corrección alguna.

Por el contrario, cuando la operación de hibridación incluye una función hash, incluso en presencia de claves de partida con cierta debilidad estadística, el resultado es significativamente más robusto desde un punto de vista criptográfico. Estas técnicas no solo homogenizan la distribución de los bytes, sino que también mitigan posibles patrones estructurales que puedan surgir de los algoritmos de origen, mejorando así la seguridad del secreto híbrido de forma consistente.

Por otro lado, en relación con las siguientes preguntas planteadas en la hipótesis, es importante recordar que la operación de hibridación no constituye el paso final del proceso. Como se ha visto, en un handshake criptográfico la clave híbrida resultante se somete a una rutina de derivación, como HKDF, que aplica transformaciones adicionales para obtener las claves finales.

Los resultados obtenidos al aplicar HKDF de forma uniforme sobre las claves

híbridas generadas por los cuatro métodos evaluados revelan un aspecto clave: **HKDF actúa como un mecanismo de normalización que refuerza la entropía y mejora la uniformidad, corrigiendo las deficiencias que puedan haber persistido tras la fase de hibridación.**

Esto sugiere que, en determinados contextos, podría considerarse una estrategia como la empleada en HKE-TLS: **aceptar la concatenación simple como método de hibridación, beneficiándose de su bajo coste computacional, aún a costa de una menor resistencia criptográfica en ese punto, consciente de que la posterior rutina de derivación compensará dicha debilidad.** No obstante, esta elección debe realizarse con pleno conocimiento del riesgo que implica, ya que la robustez final del sistema dependerá de la calidad combinada de ambas fases. Esta estrategia solo sería recomendable en caso de utilizar una rutina de derivación como HKFD, ya que en cualquiera otro caso el sistema sería inseguro.

Por último, para responder a la pregunta sobre cuál es la variable que más influye en la calidad estadística de la operación de hibridación —la longitud de la cadena de entrada al hash o el número de veces que se aplica el hash—, se pueden comparar tres construcciones específicas: SSH-HKE, que aplica un hash simple sobre una cadena corta; X-Wing, que también usa un hash simple pero sobre una cadena más larga (incluyendo las claves públicas); y HMAC, que aplica múltiples iteraciones del hash sobre la misma cadena que usa SSH-HKE.

Los resultados extraídos de las pruebas χ^2 , p-valor y JSD sobre estas tres variantes, en combinación con las curvas X25519, X448 y NIST P-256, revelan que el impacto del número de aplicaciones del hash es más relevante que la longitud de la cadena de entrada. Por ejemplo, en el caso de X448, la variante HMAC obtiene los mejores resultados estadísticos ($\chi^2 = 228$, p-valor = 0.8853, JSD = 0.0016), superando tanto a SSH-HKE ($\chi^2 = 258$, p-valor = 0.4341) como a X-Wing ($\chi^2 = 305$, p-valor = 0.0167), a pesar de que estas últimas trabajan sobre entradas de diferente longitud. Este patrón se repite con las otras dos curvas: para X25519, HMAC ($\chi^2 = 265$, p = 0.3080) se comporta de forma más estable que X-Wing ($\chi^2 = 285$, p = 0.0906), y también muy similar a SSH ($\chi^2 = 220$, p = 0.9419), pero sin requerir un esquema de entrada más complejo.

Del mismo modo, con NIST P-256, HMAC mantiene una distribución muy uniforme ($\chi^2 = 264$, p = 0.3253), ligeramente mejor que X-Wing ($\chi^2 = 266$, p = 0.3038) y comparable a SSH-HKE ($\chi^2 = 267$, p = 0.2805). Estos resultados indican que aplicar un esquema de hashing iterativo como HMAC ofrece beneficios estadísticos más consistentes que simplemente alargar la cadena de entrada, incluso cuando esta incluye componentes adicionales como claves públicas o ciphertexts.

En conclusión, **si se busca mejorar la calidad estadística de la clave híbrida, la estructura de la función hashing (como ocurre en HMAC) tiene mayor peso que la complejidad o longitud de los datos que se introducen en ella.**

Capítulo 7

Conclusión

A lo largo de este trabajo se han analizado los desafíos que plantea el surgimiento de la computación cuántica para la infraestructura global de clave pública. En particular, se ha hecho hincapié en el escenario conocido como *harvest now, decrypt later*, que explota vulnerabilidades latentes en los protocolos de comunicación actuales, especialmente durante sus fases de *handshake*. Si bien este análisis se ha centrado principalmente en entornos de comunicación, las implicaciones del problema son extrapolables a otros ámbitos de la seguridad informática.

Se ha llevado a cabo una revisión detallada de la documentación técnica y académica producida por grupos de investigación activos en este campo, muchos de los cuales proponen soluciones híbridas como enfoque de transición hacia una criptografía resistente a ataques cuánticos. Dentro de estas propuestas se ha identificado una estrategia común que parece consolidarse como una solución práctica y viable. Sin embargo, se ha detectado un elemento diferenciador especialmente relevante: la operación utilizada para combinar los secretos generados por los algoritmos clásicos y post-cuánticos, es decir, la operación de hibridación de claves. Estas diferencias metodológicas suscitaron una serie de preguntas que dieron lugar a una hipótesis de trabajo, la cual ha guiado el diseño y ejecución de un experimento orientado a obtener datos concretos que permitan evaluar el impacto real de estas decisiones técnicas.

Desde una perspectiva personal, este trabajo ha sido una oportunidad para iniciarme en este campo de estudio y conocer los nuevos desafíos a los que nos enfrentamos. Inicialmente, ciertas elecciones de diseño en los protocolos me parecieron potencialmente arriesgadas; sin embargo, a medida que se desarrolló el análisis, comprendí que dichas decisiones responden a criterios sólidos y están respaldadas por expertos de primer nivel en el ámbito de la criptografía. Este proceso me ha ayudado a valorar la importancia del análisis crítico externo y del escrutinio público como herramienta para validar propuestas técnicas. A lo largo de la historia de la criptografía, numerosos algoritmos y protocolos han sido fortalecidos (o descartados) gracias a este tipo de revisión abierta.

Si bien este estudio no ofrece una contribución significativa, quizás ayude a

Capítulo 7. Conclusión

responder a otro investigador las mismas preguntas que yo me he planteado

Bibliografía

- [1] Wikipedia contributors. «Criptografía». (2024), dirección: <https://es.wikipedia.org/wiki/Criptograf%C3%ADa>.
- [2] Wikipedia contributors. «Principios de Kerckhoffs». Accedido el 29 de junio de 2025. (2024), dirección: https://es.wikipedia.org/wiki/Principios_de_Kerckhoffs.
- [3] National Institute of Standards and Technology, «Report on Post-Quantum Cryptography», National Institute of Standards y Technology, Gaithersburg, MD, USA, NIST Internal Report NIST.IR.8105, 2016. dirección: <https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf>.
- [4] D. Moody, R. Perlner, A. Regenscheid, A. Robinson y D. Cooper, «Transition to Post-Quantum Cryptography Standards (NIST IR 8547, Initial Public Draft)», National Institute of Standards y Technology, Gaithersburg, MD, USA, NIST Internal Report 8547, ipd, nov. de 2024. DOI: 10.6028/NIST.IR.8547.ipd. dirección: <https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf>.
- [5] IBM. «An overview of the ssl/tls handshake». (2025), dirección: <https://www.ibm.com/docs/en/ibm-mq/9.3.x?topic=tls-overview-ssl-tls-handshake>.
- [6] IBM. «Cryptography». (2025), dirección: <https://www.ibm.com/es-es/topics/cryptography>.
- [7] IBM. «Cifrado asimétrico». (2025), dirección: <https://www.ibm.com/es-es/think/topics/asymmetric-encryption>.
- [8] Wikipedia contributors. «Intercambio de claves de Diffie-Hellman». Accedido el 29 de junio de 2025. (2024), dirección: https://es.wikipedia.org/wiki/Intercambio_de_claves_de_Diffie-Hellman.
- [9] Wikipedia contributors. «Elliptic-curve Diffie-Hellman». Accedido el 29 de junio de 2025. (2024), dirección: https://es.wikipedia.org/wiki/Elliptic-curve_Diffie-Hellman.
- [10] Wikipedia contributors. «Función hash». Accedido el 29 de junio de 2025. (2024), dirección: https://es.wikipedia.org/wiki/Funci%C3%B3n_hash.

- [11] *FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, Federal Information Processing Standard (FIPS), Accessed: 29 June 2025, 2015. dirección: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>.
- [12] National Institute of Standards and Technology, «NIST SP 800-108 Rev. 1: Recommendation for Key Derivation Using Pseudorandom Functions», National Institute of Standards y Technology, Special Publication 800-108 Rev. 1, 2023, Accessed: 29 June 2025. dirección: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1-upd1.pdf>.
- [13] *FIPS 198-1: The Keyed-Hash Message Authentication Code (HMAC)*, Federal Information Processing Standard (FIPS), Accessed: 29 June 2025, 2008. dirección: <https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.198-1.pdf>.
- [14] H. Krawczyk, «HMAC-based Extract-and-Expand Key Derivation Function (HKDF)», Internet Engineering Task Force, RFC 5869, 2010, Accessed: 29 June 2025. dirección: <https://datatracker.ietf.org/doc/html/rfc5869>.
- [15] IBM. «Cifrado simétrico». Accedido el 30 de junio de 2025. (2025), dirección: <https://www.ibm.com/es-es/think/topics/symmetric-encryption>.
- [16] *FIPS 197 (Update 1): Advanced Encryption Standard (AES)*, Federal Information Processing Standard (FIPS), Accessed: 30 June 2025, 2019. dirección: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf>.
- [17] I. Ferguson y D. Bernstein, «ChaCha20 and Poly1305 for IETF Protocols», Internet Engineering Task Force, RFC 8439, 2018, Accessed: 30 June 2025. dirección: <https://datatracker.ietf.org/doc/html/rfc8439>.
- [18] National Institute of Standards and Technology. «Call for proposals — post-quantum cryptography standardization». Accedido el 30 de junio de 2025. (2016), dirección: <https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals>.
- [19] D. Moody, R. Perlner, A. Robinson, W. Whyte y M. Boldyreva, «Report on Post-Quantum Key Exchange Round 1 (NIST IR 8545)», National Institute of Standards y Technology, Gaithersburg, MD, USA, NIST Internal Report 8545, jun. de 2025, Accessed: 30 June 2025. dirección: <https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8545.pdf>.
- [20] *FIPS 203: Personal Identity Verification (PIV) of Federal Employees and Contractors*, Federal Information Processing Standard (FIPS), Accessed: 30 June 2025, 2017. dirección: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf>.
- [21] Wikipedia contributors. «Distribución cuántica de claves». Accedido el 30 de junio de 2025. (2024), dirección: https://es.wikipedia.org/wiki/Distribuci%C3%B3n_cu%C3%A1ntica_de_claves.

- [22] IETF TLS Hybrid Design Team, «Draft—Hybrid Public Key Encryption for TLS», Internet Engineering Task Force, Internet-Draft draft-ietf-tls-hybrid-design-12, mayo de 2024, Accessed: 1 July 2025. dirección: <https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design-12>.
- [23] IETF SSHM ML-KEM Hybrid KEX Working Group, «Draft—ML-KEM Hybrid Key Exchange for SSH», Internet Engineering Task Force, Internet-Draft draft-ietf-sshm-mlkem-hybrid-kex-00, mayo de 2025, Accessed: 1 July 2025. dirección: <https://datatracker.ietf.org/doc/html/draft-ietf-sshm-mlkem-hybrid-kex>.
- [24] National Institute of Standards and Technology, «NIST SP800-227 IPD: Recommendation for the Transition to Post-Quantum Cryptography (Initial Public Draft)», National Institute of Standards y Technology, Special Publication 800-227 IPD, jul. de 2024, Accessed: 1 July 2025. dirección: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-227.ipd.pdf>.
- [25] M. Barbosa, D. Connolly, J. D. Duarte et al., «X-Wing: The Hybrid KEM You’ve Been Looking For», *IACR Communications in Cryptology*, vol. 1, n.º 1, págs. 1-22, 2024, Accessed: 1 July 2025, ISSN: 3006-5496. DOI: 10.62056/a3qj89n4e. dirección: <https://doi.org/10.62056/a3qj89n4e>.
- [26] National Institute of Standards and Technology, *NIST SP 800-56C Revision 2: Recommendation for Key-Derivation Methods in Key-Establishment Schemes*, Special Publication, Accessed: 1 July 2025, Gaithersburg, MD, USA, ago. de 2020. DOI: 10.6028/NIST.SP.800-56Cr2. dirección: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf>.
- [27] National Institute of Standards and Technology, *NIST SP 800-56A Revision 3: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, Special Publication, Accessed: 1 July 2025, Gaithersburg, MD, USA, sep. de 2020. dirección: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf>.
- [28] National Institute of Standards and Technology, *NIST SP 800-56B Revision 2: Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography*, Special Publication, Accessed: 1 July 2025, Gaithersburg, MD, USA, ago. de 2020. dirección: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Br2.pdf>.
- [29] National Institute of Standards and Technology, *NIST SP 800-133 Revision 2: Recommendation for Cryptographic Key Generation*, Special Publication, Accessed: 1 July 2025, Gaithersburg, MD, USA, ene. de 2024. dirección: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf>.
- [30] J. M. Pérez, *Hybridoperatorsexam: Hybrid key exchange experimental code*, <https://github.com/JuanMaruriPerez/HybridOperatorsExam>, Accedido el 1 de julio de 2025, 2025.

BIBLIOGRAFÍA

- [31] Wikipedia contributors, *Entropía (información)* — *Wikipedia, La enciclopedia libre*, Accedido el 1 de julio de 2025, 2025. dirección: [https://es.wikipedia.org/wiki/Entrop%C3%ADa_\(informaci%C3%B3n\)](https://es.wikipedia.org/wiki/Entrop%C3%ADa_(informaci%C3%B3n)).
- [32] The OpenSSL Project. «Openssl: Cryptography and ssl/tls toolkit». Accedido el 1 de julio de 2025. (2025), dirección: <https://openssl-library.org/>.
- [33] Cryptography Developers. «Cryptography – python package for cryptographic recipes and primitives». Accedido el 1 de julio de 2025. (2025), dirección: <https://cryptography.io/en/latest/>.
- [34] Open Quantum Safe. «Liboqs: A c library for quantum-safe cryptographic algorithms». Accedido el 1 de julio de 2025. (2025), dirección: <https://openquantumsafe.org/liboqs/>.
- [35] Open Quantum Safe, *Liboqs-python: Python bindings for liboqs*, <https://github.com/open-quantum-safe/liboqs-python>, Accedido el 1 de julio de 2025, 2025.
- [36] D. J. Bernstein, T. Lange y P. Schwabe, «RFC 7748: Elliptic Curves for Security, including X25519 and X448», Internet Engineering Task Force, RFC 7748, 2016, Accessed: 1 July 2025. dirección: <https://datatracker.ietf.org/doc/html/rfc7748>.

Anexos

