¿Universidad Central del Ecuador

Facultad de Ingeniería y Ciencias Aplicadas

Carrera de Computación

Criptografía y Seguridad de la Información

Docente

Ing. Giovanny Moncayo

Tema

Infraestructura De Clave Pública (PKI)

Grupo 1

Stalin Acurio

Ángelo Silva

Edison Enríquez

Kelly Ledesma

Periodo

2025 - 2026

TABLA DE CONTENIDO

INFRAESTRUCTURA DE CLAVE PÚBLICA (PKI): ANÁLISIS EXHAUSTIVO DE ARQUITECTURA, PROTOCOLOS Y GESTIÓN DE IDENTIDAD DIGITAL

1. Introducción: La Crisis de la Confianza en el Ciberespacio

La seguridad de la información, en su concepción más fundamental, ha evolucionado desde la protección física de los activos hacia la gestión lógica de la confianza en entornos distribuidos y hostiles. En el mundo analógico, la identidad se valida mediante la presencia física, documentos tangibles emitidos por estados soberanos y características biométricas inherentes. Sin embargo, la transición hacia una economía y sociedad digital ha eliminado estos vectores de confianza tradicionales, creando un vacío que debe ser llenado por construcciones matemáticas y procedimentales rigurosas. La Infraestructura de Clave Pública (PKI, por sus siglas en inglés) no es simplemente una herramienta tecnológica; representa el marco sociotécnico crítico que permite la escalabilidad de la confianza digital, garantizando la autenticidad, integridad, confidencialidad y el no repudio en redes abiertas como Internet
Históricamente, la criptografía se basaba en secretos compartidos (criptografía simétrica). Si bien algoritmos como el DES (Data Encryption Standard) y posteriormente el AES (Advanced Encryption Standard) ofrecían confidencialidad eficiente, enfrentaban un obstáculo logístico insalvable conocido como el "problema de distribución de claves". Para que dos entidades, Alicia y Beto, pudieran comunicarse de forma segura, debían haber intercambiado previamente una clave secreta por un canal seguro. En una red global con miles de millones de usuarios, la necesidad de intercambiar claves secretas de a pares resulta combinatoriamente imposible y operativamente insegura. La invención de la criptografía asimétrica o de clave pública en la década de 1970 resolvió la mecánica del intercambio de claves, pero introdujo un nuevo problema: la autenticación de la clave pública misma. ¿Cómo puede Beto saber que la clave pública que recibe realmente pertenece a Alicia y no a un atacante que intercepta la comunicación?

La PKI surge como la respuesta sistémica a este desafío. Al introducir una tercera parte de confianza —la Autoridad de Certificación (CA)—, la PKI vincula de manera irrefutable una clave criptográfica con una identidad del mundo real (persona, organización o dispositivo). Este informe técnico tiene como objetivo proporcionar un

análisis exhaustivo y corregido del estado del arte de la PKI, expandiendo las definiciones básicas para abarcar el ecosistema completo, categorizando taxonómicamente los algoritmos involucrados —incluyendo el rol vital de los algoritmos simétricos como AES en protocolos híbridos como TLS— y detallando la anatomía de los estándares X.509 y los mecanismos de revocación como CRL y OCSP. Asimismo, se abordan las correcciones terminológicas y bibliográficas necesarias para alinear la documentación académica con el rigor de la industria y los estándares internacionales actuales, como los definidos por la UIT-T y el NIST.

2. Definición y Distinción Conceptual: Tecnología vs. Infraestructura

Una de las confusiones más prevalentes en la literatura introductoria es la equiparación de la "criptografía de clave pública" con la "Infraestructura de Clave Pública". Es imperativo establecer una distinción clara para comprender el alcance real del ecosistema PKI.

2.1 La Tecnología: Criptografía Asimétrica

La "tecnología" se refiere a los fundamentos matemáticos y algorítmicos que hacen posible las operaciones criptográficas. Se basa en funciones unidireccionales con trampa (trapdoor one-way functions), problemas matemáticos que son fáciles de calcular en una dirección, pero computacionalmente inviables de revertir sin información especial (la clave privada).

Esta capa tecnológica incluye:

- La generación de pares de claves (pública y privada).
- Los algoritmos de cifrado y descifrado.
- Los algoritmos de firma digital y verificación.

La tecnología es necesaria, pero no suficiente. Un par de claves generado en aislamiento no tiene valor de identidad; es anónimo. Sin un marco que gobierne quién posee esas claves, la tecnología solo ofrece privacidad, no autenticidad.

2.2 La Infraestructura: El Ecosistema PKI

La PKI es el "ecosistema" integral. Se define formalmente como el conjunto de hardware, software, personas, políticas y procedimientos necesarios para crear, gestionar, distribuir, usar,

almacenar y revocar certificados digitales y gestionar el ciclo de vida de la criptografía de clave pública.

A diferencia de la mera tecnología, la infraestructura incorpora la gobernanza. El ecosistema PKI opera bajo documentos legales y técnicos estrictos:

1. **Política de Certificación (CP - Certificate Policy):** Un documento estratégico que define *qué* reglas sigue la PKI. Establece la aplicabilidad de los certificados a una comunidad particular y/o clase de aplicación con requisitos de seguridad comunes. Por ejemplo, una CP puede dictar que la identidad del usuario debe ser verificada presencialmente.
2. **Declaración de Prácticas de Certificación (CPS - Certification Practice Statement):** Un documento operativo que detalla *cómo* la CA implementa los requisitos de la CP. Por ejemplo, la CPS especificará que las claves de la CA se almacenan en Módulos de Seguridad de Hardware (HSM) con certificación FIPS 140-2 de nivel 3.

La robustez de una PKI no reside únicamente en la longitud de las claves (tecnología), sino en la seguridad física de la CA, la diligencia de la Autoridad de Registro (RA) al verificar identidades y la eficiencia de los mecanismos de revocación. Por tanto, la PKI debe entenderse como un sistema de gestión de confianza a gran escala, no solo como un método de cifrado.

2.3 Componentes del Ecosistema

El ecosistema PKI se compone de entidades funcionales que interactúan bajo el marco de las políticas mencionadas:

- **Autoridad de Certificación (CA):** El ancla de confianza. Emite y firma certificados.
- **Autoridad de Registro (RA):** La interfaz de verificación. Valida la identidad del solicitante antes de autorizar la emisión.
- **Repositorios:** Directorios (LDAP, HTTP) donde se publican certificados y listas de revocación.
- **Entidades Finales (Suscriptores):** Usuarios, servidores o dispositivos identificados en el certificado.
- **Partes Confingentes (Relying Parties):** Quienes consumen el certificado para validar una firma o establecer una conexión segura

3. Taxonomía de Algoritmos Criptográficos en PKI

Para una implementación segura y conforme a los estándares modernos, es necesario categorizar los algoritmos no como una lista plana, sino según su función criptográfica específica. Además, es crucial reconocer que la PKI moderna rara vez funciona solo con algoritmos asimétricos; en protocolos como TLS, la PKI facilita un entorno híbrido donde los algoritmos simétricos juegan un papel fundamental en el rendimiento.

3.1 Algoritmos Asimétricos (Cifrado de Clave y Firma)

Estos algoritmos constituyen la base de la identidad digital. Su seguridad depende de la dificultad computacional de problemas matemáticos específicos.

- **RSA (Rivest-Shamir-Adleman):**
  - *Fundamento:* Basado en la dificultad de la factorización de números enteros (el producto de dos números primos grandes).
  - *Estado Actual:* Sigue siendo el algoritmo más desplegado. El NIST, en su publicación especial SP 800-78-5 (Revisión de Julio 2024), especifica que para usos de verificación de identidad personal (PIV) y seguridad general, se requieren claves de al menos 2048 bits. Claves de 3072 bits o superiores son recomendadas para datos que requieren protección a largo plazo (más allá de 2030).
  - *Uso:* Cifrado de claves de sesión, firmas digitales.
- **ECC (Elliptic Curve Cryptography - Criptografía de Curva Elíptica):**
  - *Fundamento:* Basado en la estructura algebraica de curvas elípticas sobre campos finitos y el problema del logaritmo discreto de curva elíptica (ECDLP).
  - *Ventaja:* Ofrece una seguridad equivalente a RSA con claves significativamente más cortas. Una clave ECC de 256 bits proporciona una seguridad comparable a una clave RSA de 3072 bits, lo que resulta en un menor consumo de energía y ancho de banda, ideal para dispositivos móviles e IoT.
  - *Curvas Estándar:* NIST P-256, P-384.

3.2 Algoritmos de Hash

Las funciones hash criptográficas toman una entrada de cualquier tamaño y producen una

salida de tamaño fijo (digest). Son esenciales para la firma digital, ya que firmar un mensaje completo es ineficiente; en su lugar, se firma el hash del mensaje.

- **SHA-2 (Secure Hash Algorithm 2):**
  - La familia estándar actual. Incluye variantes como SHA-256, SHA-384 y SHA-512.
  - *Función:* Garantizar la integridad. Es computacionalmente inviable encontrar dos mensajes diferentes que produzcan el mismo hash (resistencia a colisiones). Si un solo bit del mensaje cambia, el hash resultante cambia drásticamente (efecto avalancha).
- **SHA-3:**
  - Estándar más reciente (Keccak), diseñado como alternativa de seguridad en caso de que SHA-2 sea vulnerado.

3.3 Esquemas de Firma Digital

Un algoritmo asimétrico puro (como RSA de libro de texto) no es seguro para firmas por sí solo. Se requiere un esquema de relleno (padding) para prevenir ataques.

- **RSA-PSS (Probabilistic Signature Scheme):**
  - *Descripción:* Es el esquema de firma recomendado sobre el antiguo PKCS#1 v1.5. Introduce un componente aleatorio (salt) en el proceso de codificación antes de aplicar la clave privada RSA.
  - *Importancia:* Proporciona una prueba de seguridad formal (reducción de seguridad) al problema RSA subyacente y es más robusto contra ataques teóricos. El documento base debe actualizarse para reflejar RSA-PSS como la práctica recomendada.
- **ECDSA (Elliptic Curve Digital Signature Algorithm):**
  - La variante de firma para claves ECC. Ampliamente utilizado en TLS y criptomonedas.

3.4 Algoritmos Simétricos en el Contexto de TLS (El Eslabón Híbrido)

Una corrección crítica al documento original es la inclusión de algoritmos simétricos. La PKI se utiliza para autenticar a las partes (handshake), pero el cifrado del

tráfico de datos masivo se realiza mediante algoritmos simétricos debido a su velocidad (órdenes de magnitud más rápidos que los asimétricos).

- **AES (Advanced Encryption Standard):**
    - *Contexto:* En una conexión HTTPS (TLS), una vez que el servidor envía su certificado (RSA/ECC) y el cliente lo valida, ambas partes negocian una "clave de sesión" temporal. Esta clave se usa con AES (generalmente AES-128 o AES-256) para cifrar la comunicación real.
    - *Modos de Operación:* En TLS 1.3, se prefieren los modos de cifrado autenticado con datos asociados (AEAD), como AES-GCM (Galois/Counter Mode), que proporciona confidencialidad e integridad simultáneamente.
    - *Conclusión:* Sin AES, la PKI sería inutilizable para la web moderna debido a la latencia que introduciría el cifrado asimétrico de todos los datos.

Categoría	**Algoritmo Principal**	**Función en el Ecosistema PKI**
Asimétrico	RSA (2048+ bits), ECC (P-256)	Generación de identidad, intercambio de claves, firma de certificados.
Hash	SHA-256, SHA-384	Integridad de datos, generación de huellas digitales para firmas.
Esquema de Firma	RSA-PSS, ECDSA	Mecanismo seguro de aplicación de la firma digital (Hash + Asimétrico + Padding).
Simétrico	AES (GCM/CBC)	Cifrado del túnel de datos (TLS) tras la autenticación PKI.

4. Protocolos y Estándares: La Estructura de la Interoperabilidad

La PKI funciona gracias a que todos los actores (navegadores, servidores, sistemas operativos) hablan el mismo idioma. Este idioma está definido por estándares internacionales, principalmente de la UIT-T (Unión Internacional de Telecomunicaciones) y la IETF (Internet Engineering Task Force).

4.1 X.509 Versión 3: Anatomía Detallada del Certificado

El estándar ITU-T X.509 define el formato de los certificados de clave pública. La versión 3 (v3) es la implementación universal que introdujo flexibilidad mediante el uso de "extensiones". Es fundamental ampliar la descripción de este estándar más allá de una mención superficial.

Un certificado X.509 v3 consta de tres partes principales:

1. **tbsCertificate (To Be Signed):** La información del certificado.
2. **signatureAlgorithm:** El identificador del algoritmo usado por la CA para firmar.
3. **signatureValue:** La firma digital propiamente dicha.

Campos Principales (Core Fields)

- **Version:** Indica la versión del formato. Para certificados modernos con extensiones, este valor debe ser 2 (que corresponde a v3, ya que se cuenta desde 0).
- **Serial Number:** Un entero positivo único asignado por la CA a cada certificado que emite. Es crítico para la revocación, ya que las CRLs identifican los certificados revocados únicamente por este número.
- **Signature Algorithm ID:** Identifica el algoritmo (ej. sha256WithRSAEncryption).
- **Issuer (Emisor):** El Nombre Distinguido (Distinguished Name - DN) de la CA que firma y emite el certificado. Sigue el formato X.500 (ej. CN=DigiCert Global Root CA, O=DigiCert Inc, C=US).
- **Validity (Validez):** Define el periodo de vida útil con dos fechas: Not Before (inicio) y Not After (expiración).
- **Subject (Sujeto):** El DN de la entidad propietaria de la clave pública (ej. CN=www.ejemplo.com, O=Empresa, L=Quito, C=EC).
- **Subject Public Key Info:** Contiene el algoritmo de la clave pública (ej. RSA) y la clave pública misma en formato binario (DER).

Extensiones Críticas (X.509 v3 Extensions)

Las extensiones son lo que permite a la PKI definir políticas de uso y restricciones. Se dividen en críticas (si el sistema no reconoce la extensión, debe rechazar el certificado)

y no críticas.

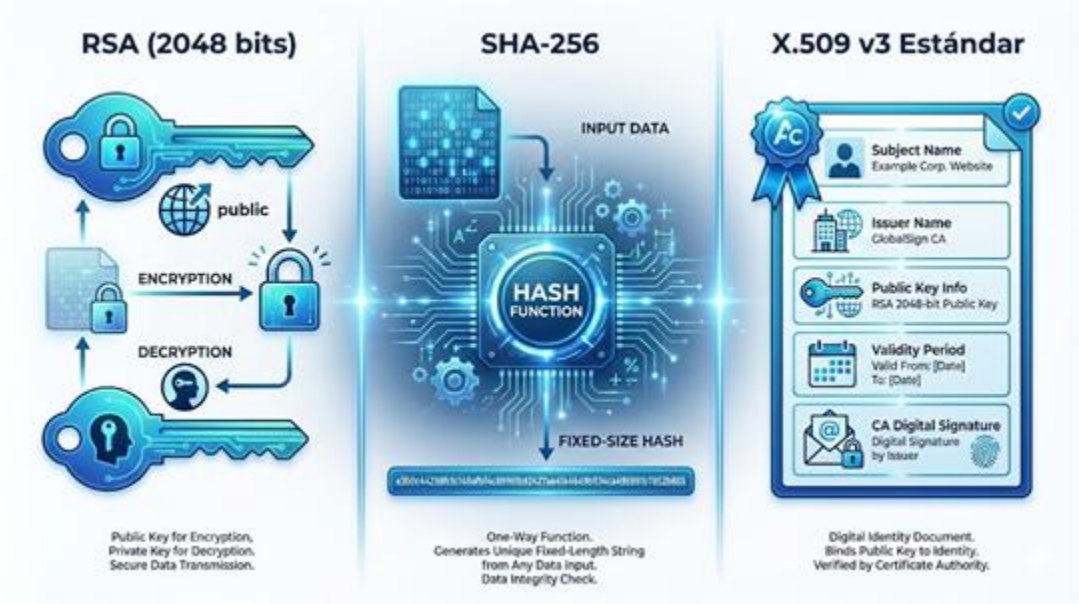
- **Basic Constraints (Restricciones Básicas - OID 2.5.29.19):**
    - Esta es quizás la extensión más importante para la seguridad estructural. Contiene un indicador booleano cA.
    - cA=TRUE: Indica que el certificado pertenece a una Autoridad de Certificación y puede ser usado para firmar otros certificados.
    - cA=FALSE: Indica que es una entidad final (usuario/servidor) y no puede firmar otros certificados.
    - *PathLenConstraint:* Si cA=TRUE, este campo opcional limita cuántos niveles de CAs subordinadas pueden existir debajo de esta CA.
    - *Importancia:* Si esta extensión no se verifica correctamente, un usuario común podría actuar como una CA y emitir certificados falsos para cualquier sitio web (ej. google.com), rompiendo la seguridad de Internet.
- **Key Usage (Uso de la Clave - OID 2.5.29.15):**
    - Define las operaciones criptográficas permitidas para la clave pública contenida. Es un mapa de bits (bitmask).
    - digitalSignature (bit 0): Para autenticación de entidades y validación de integridad de datos.
    - nonRepudiation (bit 1): Para prevenir que el firmante niegue la acción (común en firma de documentos legales).
    - keyEncipherment (bit 2): Para cifrar claves de sesión (usado en RSA key exchange).
    - keyCertSign (bit 5): Exclusivo para certificados de CA, permite validar firmas en otros certificados.
    - cRLSign (bit 6): Permite firmar Listas de Revocación de Certificados.
- **Extended Key Usage (Uso Extendido de la Clave - EKU - OID 2.5.29.37):**
    - Refina aún más el propósito. Ejemplos típicos:
        - serverAuth: Autenticación de servidor TLS (certificados SSL).
        - clientAuth: Autenticación de cliente TLS (acceso a VPN/Web).
        - codeSigning: Firma de código ejecutable.
        - emailProtection: Correo seguro (S/MIME).
- **Subject Alternative Name (SAN - OID 2.5.29.17):**

- Permite vincular múltiples identidades a un solo certificado (ej. nombres DNS, direcciones IP, URIs, emails).
- *Nota Actual:* Los navegadores modernos (Chrome, Firefox) han dejado de usar el campo Subject: Common Name para la validación de dominios y requieren que el nombre de dominio esté presente en la extensión SAN.

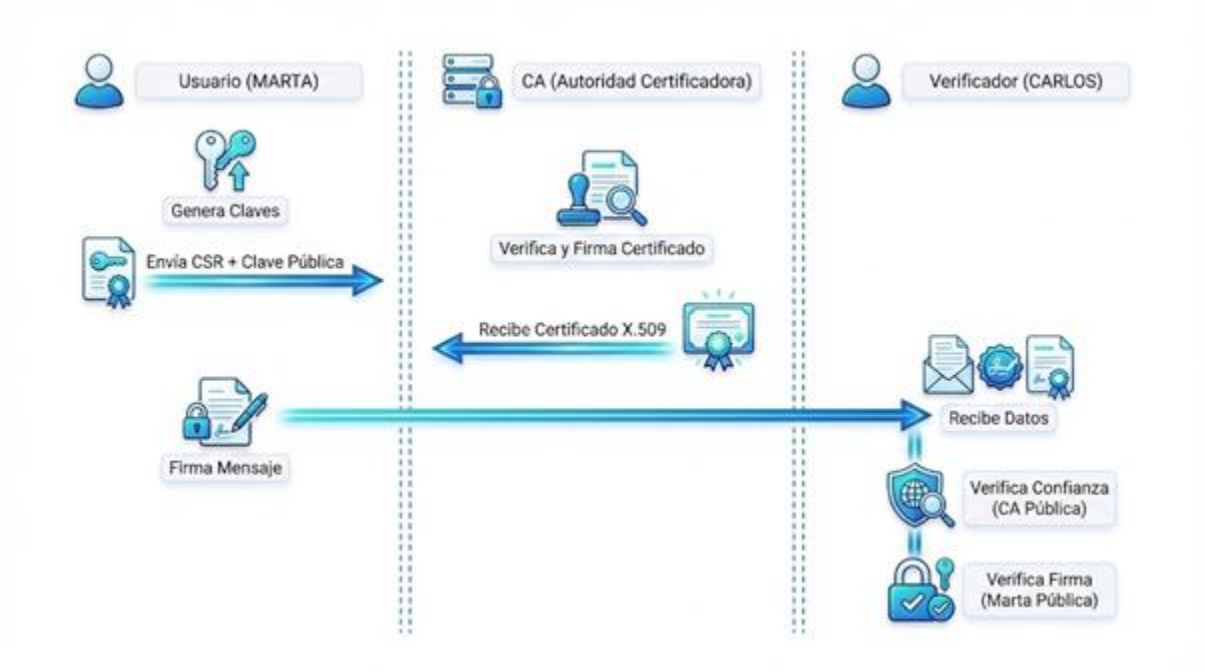
4.2 Protocolos de Validación y Revocación

La emisión es solo el principio; la validación continua es vital. Un certificado puede ser revocado antes de su expiración (por robo de clave privada, cambio de nombre de la empresa, etc.).

- **CRL (Certificate Revocation List):**
  - Es una lista firmada digitalmente por la CA que contiene los números de serie de los certificados que han sido revocados y la fecha de revocación.
  - *Mecanismo:* El verificador descarga la lista (indicada en la extensión CRL Distribution Point del certificado) y comprueba si el certificado está en ella.
  - *Limitaciones:* Problemas de latencia (la lista puede no estar actualizada al segundo) y tamaño (las listas pueden crecer mucho, consumiendo ancho de banda).
- **OCSP (Online Certificate Status Protocol - RFC 6960):**
  - Un protocolo de solicitud/respuesta en tiempo real. El verificador envía una consulta a un "OCSP Responder" preguntando por el estado de un certificado específico. El respondedor contesta: "Bueno", "Revocado" o "Desconocido", firmado digitalmente.
  - *OCSP Stapling (TLS Certificate Status Request):* Para mejorar la privacidad y el rendimiento, el servidor web (el titular del certificado) consulta periódicamente a la CA, obtiene la respuesta OCSP firmada y la "agrapa" (staples) al certificado durante el handshake TLS. Así, el cliente recibe el certificado y la prueba de su validez en un solo paso, sin tener que contactar a la CA directamente (preservando la privacidad de navegación del usuario).



5. Arquitectura y Diseño Esquemático del Funcionamiento



1. **Generación de Claves (Usuario/Marta):**
    - Marta (la entidad final) utiliza software local para generar un par de claves criptográficas: una Clave Privada (que guarda en secreto, idealmente en un token o HSM) y una Clave Pública.
2. **Solicitud de Certificado (CSR - Certificate Signing Request):**

- Marta genera una CSR (generalmente en formato PKCS#10). Este archivo contiene su Clave Pública y sus datos de identidad (Nombre: Marta, Org: UCE, País: EC).
- Marta firma digitalmente la CSR con su propia clave privada para probar la posesión de la misma (Proof of Possession). Envía la CSR a la Autoridad de Registro (RA).

3. **Verificación de Identidad (RA):**
- La RA recibe la solicitud y valida la identidad de Marta según las políticas establecidas (ej. verificando su documento de identidad físico o credenciales corporativas). Si es válido, la RA autoriza la emisión y remite la solicitud a la CA.
4. **Emisión del Certificado (CA):**
- La CA recibe los datos validados. Construye la estructura del certificado X.509 v3 (incluyendo extensiones como KeyUsage).
- La CA calcula el hash del certificado y lo cifra con la Clave Privada de la CA. Esta es la "Firma de la CA".
- El certificado firmado se devuelve a Marta y se publica en un repositorio (LDAP/HTTP).
5. **Uso: Firma Digital (Marta):**
- Marta quiere enviar un documento firmado ("Contrato 2025") a Carlos.
- Marta calcula el hash del documento (SHA-256).
- Marta cifra el hash con su Clave Privada. Esto genera la firma digital del documento.
- Marta envía el documento y la firma a Carlos. (También puede adjuntar su certificado público).
6. **Verificación (Carlos):**
- Carlos recibe el paquete. Primero necesita validar el certificado de Marta.
- **Validación de Confianza:** Carlos comprueba si el certificado de Marta fue emitido por una CA en la que él confía (su sistema operativo tiene la clave pública de la CA Raíz preinstalada). Verifica la firma de la CA en el certificado de Marta.
- **Validación de Estado:** Carlos consulta el estado de revocación (vía CRL u OCSP) para asegurar que el certificado no ha sido anulado.
- **Verificación de la Firma:** Si el certificado es válido, Carlos extrae la **Clave Pública de Marta**. Descifra el hash de la firma del documento.

- **Comprobación de Integridad:** Carlos calcula independientemente el hash del documento recibido. Compara el hash calculado con el hash descifrado. Si coinciden, la firma es válida: el documento es auténtico (viene de Marta) e íntegro (no ha sido modificado).

6. Escenarios de Uso Frecuente

A. Seguridad Web (HTTPS/TLS)

Es el uso más ubicuo. Permite la autenticación de servidores (y opcionalmente clientes) y el cifrado del tráfico web. El navegador verifica que el certificado del servidor (ej. banco.com) tenga un nombre de dominio que coincida en el campo SAN, que esté vigente y firmado por una CA raíz confiable (como DigiCert, Let's Encrypt o GlobalSign). Aquí se utiliza el modelo híbrido: PKI para el handshake y AES para el túnel de datos.

B. Identidad Digital y Facturación Electrónica

En Ecuador, el Servicio de Rentas Internas (SRI) utiliza PKI para garantizar la validez legal de los comprobantes electrónicos. Los contribuyentes obtienen certificados de CAs acreditadas (ej. Banco Central, Security Data). Al firmar una factura XML, el certificado garantiza la autoría (Autenticidad) y que los valores no han sido alterados (Integridad), otorgando validez legal plena (No Repudio) bajo la Ley de Comercio Electrónico.

C. Autenticación en Redes Corporativas (VPN y Wi-Fi)

Sustituye o complementa las contraseñas. Un dispositivo (laptop, smartphone) presenta un certificado cliente para conectarse a la VPN o al Wi-Fi empresarial (protocolo 802.1x EAP-TLS). Esto mitiga el riesgo de robo de credenciales, ya que el atacante necesitaría robar la clave privada instalada en el dispositivo, no solo una contraseña.

D. Seguridad en IoT (Internet de las Cosas)

Dispositivos industriales, medidores inteligentes y vehículos conectados utilizan certificados para autenticarse ante los servidores de control, evitando que dispositivos no

autorizados inyecten comandos falsos en redes críticas (como redes eléctricas o sistemas de control de tráfico).

7. Banco de 5 Preguntas

1. Verdadero / Falso

La PKI es un conjunto robusto de roles, políticas, hardware, software y procedimientos que establecen un ecosistema de confianza para la comunicación segura en redes.
Respuesta: Verdadero

2. ¿Qué algoritmo de función de hash se menciona en la presentación como esencial para asegurar que los datos no han sido alterados (integridad)?

A) RSA de 1024 bits
B) AES-256
C) SHA-256
D) TLS 1.3
Respuesta Correcta: C) SHA-256
Justificación: La presentación indica que SHA-256 produce un resumen único esencial para verificar la integridad de los datos.

3. Opción múltiple (3 respuestas)

Selecciona los elementos que intervienen en un proceso típico de validación dentro de una PKI.
a) Certificado digital del usuario
b) Clave privada del verificador
c) Autoridad Certificadora (CA)
d) Lista de Revocación (CRL)
e) Algoritmo AES
Respuestas correctas: a), c), d)

4. Emparejamiento

Relaciona cada concepto con su definición correspondiente:

Concepto	Definición
1. CA (Autoridad de Certificación)	A. Entidad confiable que emite, valida y revoca certificados digitales dentro de una PKI (Infraestructura de Clave Pública).
2. Identidad Digital	B. Representación electrónica de un usuario, servidor o entidad asociada a una clave pública en un certificado.
3. RSA (Rivest-Shamir-Adleman)	C. Algoritmo de criptografía asimétrica utilizado comúnmente para intercambio de claves y firmas digitales en PKI.
4. HTTPS (Protocolo de Transferencia de Hipertexto Seguro)	D. Protocolo seguro que utiliza TLS (Seguridad de la Capa de Transporte) y certificados digitales para proteger la comunicación entre cliente y servidor.

Respuestas correctas:

1 → A

2 → B

3 → C

4 → D

5. Completar con una palabra

El estándar utilizado para definir la estructura de los certificados digitales en una PKI se denomina _____.

Respuesta: X.509

8. Análisis Técnico de la Implementación (Python)

El desarrollo práctico adjunto al proyecto utiliza la librería `cryptography` de Python, que es el estándar de facto actual para operaciones criptográficas seguras, sustituyendo a librerías obsoletas como `PyCrypto`. Este análisis disecciona la lógica de los componentes de software utilizados, explicando cómo se traducen los conceptos teóricos de PKI a objetos programáticos sin recurrir a diagramas de flujo.

8.1 Gestión de Primitivas Criptográficas ("Hazmat")

El código interactúa con la capa "Hazardous Materials" (`hazmat`) de la librería. Se seleccionan explícitamente algoritmos robustos:

- **Generación de Claves:** Se utiliza el objeto `rsa` para generar pares de claves. El script define un `public_exponent=65537` (Fermat F4), que es el estándar de la industria por su eficiencia en la verificación, y un tamaño de clave de 2048 bits, alineado con las recomendaciones NIST actuales.
- **Algoritmo de Hash:** Se instancia `hashes.SHA256()` para todas las operaciones de resumen, garantizando resistencia a colisiones en la firma.

8.2 Construcción de Objetos X.509 (El Certificado)

El script no manipula bytes crudos manualmente, sino que utiliza el patrón de diseño "Builder" a través de la clase `x509.CertificateBuilder`.

- **Identidad y Sujeto:** Se construyen objetos `x509.Name` que encapsulan los atributos de identidad (País, Organización, Nombre Común).
- **Diferenciación CA vs. Usuario Final:**
    - **Certificado CA:** El código crea un certificado "auto-firmado". Técnicamente, esto se logra asignando el mismo objeto `x509.Name` tanto al campo `issuer_name` como al `subject_name`, y firmando el certificado con la propia clave privada que se está certificando.
    - **Certificado de Usuario (Marta):** Aquí se observa la cadena de confianza. El `Builder` recibe el `subject_name` de Marta, pero el `issuer_name` se extrae del certificado de la CA. Crucialmente, el método `.sign()` utiliza la **clave privada de la CA**, no la de Marta.
- **Extensiones:** El código añade extensiones críticas mediante `add_extension`. Aunque no se detalla en profundidad, el uso de estas extensiones es lo que convierte un archivo plano en un certificado X.509 v3 válido.

8.3 Distinción Técnica en Esquemas de Padding

Un aspecto técnico notable del código es el uso diferenciado de esquemas de relleno (padding) según el propósito, demostrando buenas prácticas de ingeniería de seguridad:

1. **Para la Estructura del Certificado (PKCS#1 v1.5):** Al firmar los certificados X.509, el código utiliza `padding.PKCS1v15()`.
    - *Razón Técnica:* Aunque es un esquema más antiguo, es el estándar obligatorio para la compatibilidad de certificados X.509. La mayoría de los sistemas operativos no reconocerían un certificado firmado con PSS.
2. **Para la Firma de Mensajes (PSS):** Al firmar el documento ("Contrato"), el código cambia a `padding.PSS()`.
    - *Razón Técnica:* El Probabilistic Signature Scheme (PSS) introduce aleatoriedad en la firma. Esto hace que firmar el mismo documento dos veces genere dos firmas binarias diferentes (aunque ambas válidas), eliminando vectores de ataque deterministas que podrían afectar a PKCS#1 v1.5.

8.4 Lógica de Verificación de Confianza

La función de verificación no es un simple "if". El código implementa la validación criptográfica real:

- **Extracción de Clave Pública:** El script extrae la clave pública de la CA (`ca_cert.public_key()`) y la utiliza para verificar la firma incrustada en el certificado de Marta.
- **Validación de Integridad (TBS):** La función `verify` toma los bytes crudos del certificado de Marta ("To Be Signed" bytes) y comprueba matemáticamente que coinciden con la firma, garantizando que ni un solo bit de la identidad de Marta ha sido alterado desde que la CA lo emitió.

Ejecución

```
Par de claves RSA generado correctamente.
Certificado X.509 autofirmado para 'Autoridad Certificadora Central' creado correctamente.
Par de claves RSA generado correctamente.
Certificado X.509 para 'Marta' emitido por 'Autoridad Certificadora Central' creado correctamente.

Mensaje firmado por Marta.

=== carlos verifica el certificado de Marta con la CA ===
✓ Certificado del usuario verificado con la CA.

=== carlos verifica la firma del mensaje original ===
✓ FIRMA VERIFICADA: el mensaje es auténtico y no ha sido alterado.

=== carlos verifica la firma con un mensaje alterado ===
✗ ERROR: la firma NO es válida.
Detalles del error:

Proceso completo de PKI ejecutado correctamente (Marta → carlos con CA).
```

9. Conclusiones y Perspectivas Futuras

La Infraestructura de Clave Pública (PKI) constituye la columna vertebral invisible pero indispensable de la seguridad digital moderna. A través de este análisis técnico, se ha demostrado que la PKI no es un monolito tecnológico, sino una orquestación compleja y delicada de primitivas criptográficas (RSA, ECC, SHA-2, AES), protocolos de comunicación estándar (X.509, TLS, OCSP) y, crucialmente, marcos de gobernanza y políticas (CP/CPS).

Las correcciones y expansiones presentadas en este informe subrayan tres conclusiones fundamentales:

1. **La distinción es vital:** Diferenciar entre la matemática (criptografía) y la gestión (infraestructura) es esencial para comprender los puntos de falla. La mayoría de los incidentes de seguridad en PKI no se deben a la ruptura del algoritmo RSA, sino a fallas en los procesos de validación de identidad por parte de las CAs o a la mala gestión de claves privadas por los usuarios.
2. **La naturaleza híbrida de la seguridad:** La PKI no opera en el vacío. Su integración con algoritmos simétricos como AES en el protocolo TLS demuestra que la eficiencia y la seguridad deben coexistir. La PKI proporciona la confianza inicial, mientras que la criptografía simétrica soporta la carga de trabajo.
3. **Evolución continua:** La infraestructura es dinámica. La transición de listas estáticas (CRL) a validación en tiempo real (OCSP), la adopción de curvas elípticas (ECC) y la

evolución de los campos de certificados (como la obligatoriedad del SAN) reflejan una adaptación constante a nuevas amenazas y requisitos de rendimiento.

Mirando hacia el futuro, la PKI enfrenta su mayor desafío con la llegada de la computación cuántica, que amenaza con romper los algoritmos asimétricos actuales (RSA, ECC). La preparación para la Criptografía Post-Cuántica (PQC), liderada por el NIST y reflejada en las actualizaciones de estándares como la SP 800-78-5, será el próximo gran salto evolutivo de esta infraestructura, asegurando que la confianza digital perdure en las próximas décadas.

10. Bibliografía

[1] A. Albarqi, E. Alzaid, F. Al-Ghamdi, S. Asiri, y J. Kar, "Public Key Infrastructure: A Survey," *Journal of Information Security*, vol. 6, no. 1, pp. 31–37, 2015. DOI**:** 10.4236/jis.2015.61004.

[2] H. Ferraiolo y A. Regenscheid, "Cryptographic Algorithms and Key Sizes for Personal Identity Verification," NIST Special Publication 800-78-5, Jul. 2024. DOI: 10.6028/NIST.SP.800-78-5.

[3] ITU-T Recommendation X.509, "Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks," Oct. 2019. URL: https://handle.itu.int/11.1002/1000/14033.

[4] H. Hadan, N. Serrano, y L. J. Camp, "A holistic analysis of web-based public key infrastructure failures," *Journal of Cybersecurity*, vol. 7, no. 1, 2021. DOI: 10.1093/cybsec/tyab025.

[5] W. Stallings, "Digital signature algorithms," *Cryptologia*, vol. 37, no. 4, pp. 311–327, 2013. DOI: 10.1080/01611194.2013.797044.