TECNOLÓGICA DEL ORIENTE
INSTITUCIÓN DE EDUCACIÓN SUPERIOR

INTERNET DE LAS COSAS

TÉCNICAS DE CIFRADO PARA DISPOSITIVOS IOT

TÉCNICAS DE CIFRADO PARA DISPOSITIVOS IOT

Las técnicas de autenticación en IoT deben abordar desafíos únicos derivados de la naturaleza autónoma de dispositivos, limitaciones de interfaz de usuario, y requerimientos de operación continua sin intervención humana frecuente. Los métodos tradicionales de autenticación basados en credenciales usuario/contraseña resultan inadecuados para dispositivos IoT que operan de forma autónoma, requiriendo técnicas especializadas como autenticación basada en certificados, fichas de hardware, y esquemas de autenticación mutua que verifican tanto la identidad del dispositivo como del servidor. La autenticación en IoT debe considerar también aspectos de escalabilidad para gestionar potencialmente millones de dispositivos, así como flexibilidad para acomodar dispositivos con diferentes capacidades computacionales y de conectividad.

Los esquemas de autenticación específicos para IoT incluyen técnicas como Motor de Composición de Identidad de Dispositivo que establece identidades criptográficas únicas basadas en características inmutables del hardware, autenticación basada en Funciones Físicas No Clonables que aprovechan variaciones microscópicas de fabricación para crear identificadores únicos, y protocolos de autenticación ligeros optimizados para dispositivos con severas limitaciones de recursos. Como detalla Ruiz-Velasco Sánchez y Bárcenas López (2025) en investigaciones sobre desarrollo de aplicaciones IoT seguras para dispositivos con recursos limitados, estas técnicas proporcionan niveles de seguridad comparables a métodos tradicionales mientras operan eficientemente en hardware embebido. La autenticación continua representa otro aspecto crítico donde dispositivos deben volver a autenticarse periódicamente sin interrumpir operaciones normales, requiriendo protocolos que balanceen seguridad con disponibilidad operacional. Los protocolos optimizados pueden completar autenticación mutua en menos de 100ms con dispositivos de 8 bits.

En aplicaciones de ingeniería de software, las técnicas de autenticación IoT se implementan frecuentemente mediante arquitecturas de Infraestructura de Clave Pública adaptadas para escala IoT. Un ejemplo específico se observa en sistemas de vehículos conectados donde cada vehículo posee un certificado digital único emitido por una autoridad certificadora del fabricante. Durante comunicaciones Vehículo a Infraestructura, el vehículo presenta su certificado junto con una firma digital de los datos transmitidos, permitiendo que la infraestructura verifique tanto la identidad del vehículo como la integridad de los datos recibidos. Este esquema incluye mecanismos de revocación distribuida para manejar certificados comprometidos y protocolos de renovación automática para mantener validez de certificados sin intervención del conductor.

Aplicación Práctica:

- Para implementar autenticación basada en certificados para IoT, se configura una Autoridad de Certificación usando OpenSSL: openssl req -new -x509 -keyout clave-ac.pem -out cert-ac.pem -days 365.
- Se generan certificados individuales para cada dispositivo: openssl genrsa -out clave-dispositivo1.pem 2048, openssl req -new -key clave-dispositivo1.pem -out

dispositivo1.csr, openssl x509 -req -in dispositivo1.csr -CA cert-ac.pem -CAkey clave-ac.pem -out cert-dispositivo1.pem.

- Se implementa verificación mutua TLS donde el dispositivo verifica el certificado del servidor y viceversa usando: contexto_ssl.check_hostname = False, contexto_ssl.verify_mode = ssl.CERT_REQUIRED, contexto_ssl.load_cert_chain('cert-dispositivo1.pem', 'clave-dispositivo1.pem').
- Se configura validación de Lista de Revocación de Certificados para manejar certificados comprometidos.

El resultado esperado es un sistema donde dispositivos y servidores se autentican mutuamente usando certificados X.509, proporcionando autenticación fuerte con capacidad de gestión centralizada de identidades y revocación granular de accesos comprometidos.

Gestión de claves en redes IoT

La gestión de claves criptográficas en redes IoT presenta complejidades únicas derivadas de la escala masiva de dispositivos, distribución geográfica extensa, y limitaciones operacionales que dificultan la intervención humana directa. Los sistemas de gestión de claves IoT deben abordar el ciclo de vida completo de material criptográfico incluyendo generación segura, distribución inicial, rotación periódica, y revocación de claves comprometidas, todo mientras mantienen operación continua de dispositivos críticos. La complejidad se amplifica por la heterogeneidad de dispositivos IoT que pueden tener diferentes capacidades criptográficas, protocolos de comunicación, y requerimientos de seguridad, requiriendo sistemas de gestión de claves flexibles que puedan adaptarse a esta diversidad tecnológica.

Las arquitecturas de gestión de claves para IoT incluyen enfoques centralizados donde un Servicio de Gestión de Claves central gestiona todas las claves, sistemas distribuidos que delegan responsabilidades de gestión a nodos intermedios, y esquemas híbridos que combinan elementos de ambos enfoques según las características específicas de la red IoT. Según establece Domínguez Mínguez (2021) en investigaciones sobre implementación segura de sistemas criptográficos distribuidos, los desafíos técnicos específicos incluyen arranque seguro de dispositivos que se conectan por primera vez sin claves preexistentes, gestión de claves en dispositivos con conectividad intermitente, y sincronización de rotación de claves a través de dispositivos distribuidos sin crear ventanas de vulnerabilidad. Estas consideraciones requieren protocolos especializados que puedan operar eficientemente en las condiciones operacionales únicas de redes IoT. Los sistemas avanzados pueden gestionar hasta 1 millón de dispositivos con rotación automática de claves cada 30 días manteniendo disponibilidad del 99.9%.

Para ampliar el concepto sobre la ciberseguridad, le invitamos a ver el siguiente vídeo.

IoT CS LAC. (2023, 27 de agosto). IoT y la Ciberseguridad. [Vídeo] YouTube. https://youtu.be/yvv9APAei8Q

En el contexto de ingeniería de software, la gestión de claves IoT frecuentemente requiere integración con sistemas de gestión de dispositivos existentes y plataformas de orquestación en la nube. Un ejemplo práctico se observa en sistemas de ciudades

inteligentes donde miles de sensores ambientales distribuidos por la ciudad requieren rotación periódica de claves de cifrado sin interrumpir el monitoreo continuo de calidad del aire. El sistema utiliza una arquitectura jerárquica donde pasarelas regionales actúan como Centros de Distribución de Claves que reciben claves maestras del Servicio de Gestión de Claves central y derivan claves específicas para dispositivos en su área de cobertura. Esta arquitectura permite la rotación de claves por sectores geográficos, reduciendo el impacto de claves comprometidas y facilitando operaciones de mantenimiento localizadas.

Aplicación Práctica:

- Para implementar gestión de claves jerárquica en IoT, se configura un Servicio de Gestión de Claves central usando HashiCorp Vault: vault secrets enable -path=claves-iot kv-v2, vault kv put claves-iot/clave-maestra value=<clave_generada>.
- Se implementa derivación de claves usando HKDF: clave_derivada = HKDF(clave_maestra, salt=id_dispositivo, info="cifrado_sensor", length=32). Se configura rotación automática mediante trabajos cron que ejecuten: nueva_clave_maestra = os.urandom(32), vault kv put claves-iot/clave-maestra-v2 value=nueva_clave_maestra, seguido de notificación a pasarelas regionales.
- Se implementa protocolo de transición donde dispositivos mantienen tanto claves antiguas como nuevas durante un período de gracia: if falla_descifrado_con_clave_actual: intentar_clave_anterior().

El resultado esperado es un sistema donde las claves se rotan automáticamente cada 30 días, los dispositivos mantienen operación continua durante transiciones, y las claves comprometidas pueden ser revocadas segmentadamente por región geográfica sin afectar dispositivos en otras áreas.

Seguridad en la transmisión de datos IoT

La seguridad en transmisión de datos IoT requiere protocolos especializados que protejan confidencialidad, integridad, y autenticidad de información mientras operan eficientemente en redes con características únicas como alta latencia, conectividad intermitente, y ancho de banda limitado. Los protocolos de seguridad tradicionales como TLS/SSL, aunque ampliamente utilizados, requieren adaptaciones específicas para contextos IoT incluyendo protocolos de enlace optimizados, reanudación de sesión mejorada, y conjuntos de cifrado seleccionados para eficiencia energética. Los protocolos emergentes como DTLS (TLS de Datagramas) para comunicaciones UDP y protocolos de seguridad específicos para IoT como Seguridad de Objetos para Entornos RESTful Restringidos proporcionan alternativas optimizadas que mantienen garantías de seguridad mientras reducen sobrecarga computacional y de red.

La implementación de seguridad extremo a extremo en transmisiones IoT presenta desafíos adicionales cuando datos transitan por múltiples intermediarios incluyendo pasarelas, proxy, y sistemas de agregación que pueden requerir acceso a información específica para funciones de enrutamiento o procesamiento. Como documenta Rosa (2021) en investigaciones sobre protección de datos en sistemas distribuidos y transformación digital, las técnicas avanzadas incluyen cifrado por capas donde diferentes elementos de datos se cifran con claves específicas según los

requerimientos de acceso de cada intermediario, y esquemas de cifrado homomórfico que permiten computación sobre datos cifrados sin requerir descifrado intermedio. Estas técnicas permiten arquitecturas de seguridad sofisticadas que balancean protección de datos con funcionalidad operacional requerida por sistemas IoT complejos. Los protocolos optimizados pueden reducir la sobrecarga de seguridad en un 40% comparado con TLS tradicional manteniendo el mismo nivel de protección.

En aplicaciones de ingeniería de software, la seguridad de transmisión IoT frecuentemente requiere arquitecturas híbridas que combinan múltiples protocolos según las características específicas de cada segmento de comunicación. Un ejemplo específico se presenta en sistemas de telemetría médica donde dispositivos portátiles transmiten datos biométricos sensibles a través de múltiples redes incluyendo Bluetooth al teléfono inteligente del paciente, WiFi/celular a sistemas en la nube, y APIs REST a sistemas hospitalarios. Cada segmento utiliza protocolos de seguridad apropiados: Bluetooth LE con emparejamiento autenticado y AES-128, TLS 1.3 para comunicaciones en la nube con secreto perfecto hacia adelante, y TLS mutuo con certificados cliente para APIs médicas, garantizando protección de extremo a extremo adaptada a las capacidades y requerimientos de cada segmento de red.

Aplicación Práctica:

- Para implementar seguridad de transmisión optimizada para IoT, se configura DTLS para comunicaciones UDP: contexto_dtls = ssl.create_default_context(ssl.Purpose.SERVER_AUTH), contexto_dtls.check_hostname = False, contexto_dtls.verify_mode = ssl.CERT_REQUIRED. Se implementa reanudación de sesión para reducir protocolos de enlace: contexto_dtls.set_session_cache_mode(ssl.SESS_CACHE_CLIENT), sesion_almacenada = contexto_dtls.session.
- Se configuran conjuntos de cifrado optimizados para IoT: contexto_dtls.set_ciphers('ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-CCM').
- Se implementa Seguridad de Capa de Aplicación usando COSE (Firma y Cifrado de Objetos CBOR): cabeceras_protegidas = {1: -7}, mensaje_cose = cose.enc0.encode_and_encrypt(texto_plano, clave, cabeceras_protegidas). Se configura monitoreo de conexiones para detectar anomalías: if intentos_conexion > umbral: activar_alerta_seguridad().

El resultado esperado es un sistema que mantiene <200ms de latencia adicional por protocolo de enlace DTLS, utiliza 40% menos ancho de banda que TLS tradicional debido a cabeceras optimizadas, y proporciona secreto hacia adelante con rotación de claves de sesión cada 1 hora para mitigar impacto de compromiso de claves.

Protección de la privacidad del usuario en IoT

La protección de privacidad del usuario en ecosistemas IoT trasciende técnicas tradicionales de anonimización debido a la riqueza y granularidad de datos que estos sistemas recolectan continuamente, incluyendo patrones de comportamiento detallados, ubicación geográfica precisa, y preferencias personales inferidas de interacciones con dispositivos inteligentes. Las técnicas de protección de privacidad para IoT deben considerar no solo la protección de datos individuales, sino también la prevención de inferencias no autorizadas que pueden derivarse de análisis de

correlación y aprendizaje automático aplicado a conjuntos de datos agregados. Los enfoques incluyen privacidad diferencial que añade ruido estadísticamente controlado a conjuntos de datos para prevenir re-identificación, técnicas de k-anonimato adaptadas para datos temporales IoT, y cifrado homomórfico que permite análisis sobre datos cifrados sin revelar información individual.

La implementación de privacidad por diseño en sistemas IoT requiere consideración integral desde las fases iniciales de desarrollo, incorporando principios como minimización de datos donde solo se recolecta información estrictamente necesaria para funcionalidad específica, procesamiento local que mantiene datos sensibles en dispositivos locales cuando es posible, y controles granulares de consentimiento que permiten especificar exactamente qué información se comparte y con qué propósitos. Según analiza Llaneza González (2018) en estudios sobre responsabilidad legal en sistemas IoT, las técnicas avanzadas incluyen aprendizaje federado que permite entrenar modelos de aprendizaje automático sin centralizar datos sin procesar, y computación segura multipartita que facilita análisis colaborativo entre múltiples organizaciones sin revelar datos individuales. Estos enfoques son particularmente relevantes en contextos donde regulaciones como GDPR requieren controles estrictos sobre procesamiento de datos personales. Las implementaciones pueden lograr niveles de privacidad equivalentes a una anonimización completa mientras preservan 85% de la utilidad estadística de los datos.

En el contexto de ingeniería de software, la protección de privacidad IoT frecuentemente requiere arquitecturas que balancean la utilidad de datos con protección individual. Un ejemplo específico se observa en sistemas de hogar inteligente donde múltiples dispositivos recolectan información detallada sobre actividades familiares incluyendo patrones de ocupación, preferencias de temperatura, y rutinas de entretenimiento. El sistema implementa técnicas de análisis que preservan privacidad donde datos sin procesar permanecen cifrados localmente en un concentrador doméstico, mientras que algoritmos de aprendizaje federado permiten mejoras del sistema basadas en patrones agregados sin exponer información específica de hogares individuales. Los usuarios mantienen control granular sobre qué tipos de datos contribuyen a análisis colaborativos y pueden revocar consentimiento selectivamente.

Aplicación Práctica:

- Para implementar protección de privacidad con privacidad diferencial en IoT, se configura un sistema que añada ruido calibrado a datos antes de análisis: epsilon = 0.1 (presupuesto de privacidad), sensibilidad = cambio_maximo_posible_en_resultado_consulta, escala_ruido = sensibilidad / epsilon, resultado_ruidoso = resultado_verdadero + numpy.random.laplace(0, escala_ruido).
- Se implementa privacidad diferencial local donde cada dispositivo añade ruido antes de transmisión: epsilon_local = 0.05, ruido_local = numpy.random.laplace(0, 1/epsilon_local), valor_privado = lectura_sensor + ruido_local. Se configura k-anonimato temporal para datos de ubicación: ubicaciones_grupo = agrupar_ubicaciones_cercanas(radio=100m), ubicacion_anonimizada = calcular_centroide(ubicaciones_grupo) if len(ubicaciones_grupo) >= k else suprimir_ubicacion().

- Se implementan controles de consentimiento granular: consentimiento_usuario = {ubicacion: True, temperatura: False, ocupacion: True}, datos_filtrados = aplicar_filtro_consentimiento(datos_sensor, consentimiento_usuario).

El resultado esperado es un sistema que mantiene utilidad estadística de datos agregados mientras garantiza que información individual no puede ser inferida con confianza superior al 90%, cumpliendo con requerimientos regulatorios mientras preserva funcionalidad para análisis legítimos y mejora del sistema.

El estudio profundo de protocolos y seguridad en IoT revela la complejidad intrínseca de desarrollar sistemas conectados que balanceen eficiencia operacional con robustez de seguridad. Los protocolos especializados como MQTT y CoAP han emergido como soluciones técnicas maduras que abordan los requerimientos únicos de dispositivos con recursos limitados, mientras que la seguridad en IoT requiere enfoques holísticos que consideren vulnerabilidades desde el hardware hasta las aplicaciones. La comprensión profunda de estas tecnologías resulta fundamental para ingenieros de software que desarrollan soluciones IoT en un panorama donde la conectividad ubicua coexiste con amenazas de seguridad sofisticadas y regulaciones de privacidad cada vez más estrictas.

En el contexto laboral contemporáneo, los profesionales en ingeniería de software enfrentan la responsabilidad de diseñar sistemas IoT que no solo cumplan requerimientos funcionales, sino que también garanticen protección adecuada de datos de usuarios y resistencia ante ataques cibernéticos. Las técnicas de cifrado y gestión de claves estudiadas proporcionan las herramientas técnicas necesarias, pero su implementación efectiva requiere comprensión integral de las limitaciones y capacidades específicas de dispositivos IoT. La capacidad de seleccionar protocolos apropiados, implementar medidas de seguridad eficientes, y diseñar arquitecturas resilientes determina directamente el éxito y sostenibilidad de proyectos IoT en entornos empresariales e industriales.

La evolución continua del ecosistema IoT hacia mayor integración con inteligencia artificial, computación perimetral, y sistemas autónomos amplifica tanto las oportunidades como los desafíos asociados con protocolos y seguridad. Los profesionales que dominan estos fundamentos técnicos estarán posicionados para liderar el desarrollo de la próxima generación de sistemas IoT que no solo aprovechan las capacidades de conectividad ubicua, sino que también establecen nuevos estándares de seguridad y privacidad. Esta competencia técnica se traduce en ventajas competitivas significativas en un mercado laboral donde la experiencia en IoT seguro representa una diferenciación profesional crítica para roles de liderazgo técnico en transformación digital empresarial.

Bibliografía

- Domínguez Mínguez, T. (2020). Desarrollo de aplicaciones IoT en la nube para Arduino y ESP8266 (1.ª ed.). Marcombo. https://elibro.net/es/ereader/tecnologicadeloriente/280029?page=1
- Domínguez Mínguez, T. (2021). Google Assistant: desarrollo de aplicaciones IoT para Arduino y ESP8266 (1.ª ed.). Marcombo. https://elibro.net/es/ereader/tecnologicadeloriente/281469?page=1

- Llaneza González, P. (2018). Seguridad y responsabilidad en la internet de las cosas (IoT). LA LEY Soluciones Legales S.A. https://elibro.net/es/ereader/tecnologicadeloriente/58379?page=1
- Rosa, J. M. D. L. (2021). De la micro a la nanoelectrónica: impulsando la transformación digital (1.ª ed.). Los libros de la Catarata. https://elibro.net/es/lc/tecnologicadeloriente/titulos/233414
- Ruiz-Velasco Sánchez, E., & Bárcenas López, J. (2025). Educatrónica tecnología de inteligencia artificial (robótica, programación e internet de las cosas) (1.ª ed.). Newton Edición y Tecnología Educativa. https://elibro.net/es/ereader/tecnologicadeloriente/280532?page=1
- IoT CS LAC. (2023, 27 de agosto). Protegiendo la privacidad en la era del IoT: La importancia del cifrado de datos [Vídeo]. YouTube. https://youtu.be/iyvNMTdn_qo
- IoT CS LAC. (2023, 27 de agosto). IoT y la ciberseguridad [Vídeo]. YouTube. https://youtu.be/yvv9APAei8Q