Informe de Proyecto

Trabajo Fin de Máster

Plataforma OTA segura y escalable para dispositivos IoT con criptografía ligera y post-cuántica

Alumno:	Álvaro Díez-Andino Herrera
Director:	Iñaki Garitano
Tutor:	Haritz Saiz
Empresa:	Ikerlan S. Coop.
Máster:	Máster en análisis de datos, ciberseguridad y computación en la nube
Universidad:	Mondragón Unibertsitatea

Curso 2025-2026

1 Título del Proyecto

Plataforma OTA segura y escalable para dispositivos IoT con criptografía ligera y post-cuántica

2 Área de conocimiento

El proyecto se enmarca en las siguientes áreas de conocimiento:

- Ciberseguridad y criptografía
- Internet of Things (IoT)
- Sistemas embebidos
- Arquitecturas de software distribuidas
- DevOps y gestión del ciclo de vida del software

3 Sector industrial

El proyecto se desarrolla en **Ikerlan S. Coop.**, centro tecnológico perteneciente al sector de la **investigación y desarrollo tecnológico**, con aplicación directa en los sectores de:

- Industria 4.0 y manufactura avanzada
- Internet of Things (IoT)
- Infraestructuras críticas
- Dispositivos embebidos y sistemas ciber-físicos

4 Equipo del Proyecto

- **Director:** Iñaki Garitano
- **Tutor:** Haritz Saiz
- **Alumno:** Álvaro Díez-Andino Herrera

5 Descripción del Proyecto

El proyecto se desarrolla en Ikerlan S. Coop., específicamente en el grupo SDZ (Sistemas Distribuidos) formado por 60 expertos analistas especializados en IoT, ciberseguridad y arquitecturas distribuidas.

El proyecto aborda el desarrollo de una plataforma integral para la gestión segura de actualizaciones OTA (Over-The-Air) en dispositivos IoT. La solución implementada integra algoritmos de criptografía ligera (LWC) y resistente a ataques cuánticos (PQ) para garantizar la seguridad de las actualizaciones en dispositivos con recursos limitados.

La plataforma desarrollada permite cifrar y firmar paquetes de actualización utilizando algoritmos como ASCON (LWC) y algoritmos post-cuánticos estandarizados por el NIST, proporcionando confidencialidad, integridad y autenticidad. Se ha desarrollado una interfaz web completa que abstrae la complejidad criptográfica, facilitando la gestión de claves, la creación de paquetes de actualización y el despliegue mediante estrategias de rollout controladas.

El proyecto incluye modificaciones significativas en herramientas de código abierto (SWUpdate y SWUGenerator) para extender su soporte criptográfico, así como la integración con sistemas de orquestación de workflows (WFX) y gestión de identidades (Lamassu IoT PKI).

6 Competencias adquiridas

Durante el desarrollo del TFM, he adquirido y demostrado las siguientes competencias del máster:

6.1 M2N111 - Gestión del ciclo de vida del software

Diseño y automatización de procesos de gestión de pruebas, cambios, despliegue y actualizaciones de soluciones empresariales optimizando el ciclo de vida del software.

La gestión completa del ciclo de vida (pruebas automatizadas, pipelines CI/CD, etc.) queda pendiente de implementación en fases futuras del proyecto.

6.2 M2N110 - Arquitecturas escalables y flexibles

Definición, diseño e implementación de arquitecturas escalables, flexibles y resistentes que aborden problemas existentes y aceleren el despliegue de aplicaciones.

Aplicación en el proyecto:

- Diseño de una arquitectura basada en microservicios que separa responsabilidades (KMS, Updates, WFX)
- Implementación de una capa de abstracción que permite integrar múltiples algoritmos criptográficos
- Desarrollo de un modelo de datos escalable para gestión de flotas de dispositivos
- Creación de APIs RESTful que facilitan la integración con otros sistemas

Nota: La orquestación con Kubernetes para despliegue en entornos de producción queda como trabajo futuro para completar la arquitectura cloud-native.

6.3 M2N206 - Investigación e innovación

Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.

Aplicación en el proyecto:

- Investigación sobre algoritmos de criptografía ligera y su aplicabilidad en dispositivos IoT

- Estudio de algoritmos post-cuánticos y su integración en cadenas de actualización
- Evaluación comparativa de rendimiento entre algoritmos tradicionales y ligeros
- Contribución original mediante la extensión de herramientas open-source existentes

7 Objetivos del Proyecto

Los objetivos principales del proyecto son:

1. Desarrollar una plataforma segura y escalable para la distribución de actualizaciones OTA en dispositivos IoT que garantice confidencialidad, integridad y autenticidad mediante cifrado y firma digital.
2. Integrar algoritmos de criptografía ligera (ASCON) en toda la cadena de actualización, demostrando su viabilidad en dispositivos con recursos limitados y validando las ventajas en términos de rendimiento y consumo de recursos.
3. Incorporar soporte para algoritmos post-cuánticos en los mecanismos de firma digital, preparando la infraestructura para amenazas futuras y garantizando la relevancia a largo plazo de la solución.
4. Diseñar una interfaz de usuario intuitiva que abstraiga la complejidad criptográfica y de gestión de actualizaciones, facilitando su adopción por organizaciones sin equipos especializados en seguridad.

8 Pliego de condiciones

Este apartado detalla los recursos, requisitos y condiciones específicas del proyecto, siguiendo las directrices de la norma UNE 157001 de AENOR y adaptado a las características de este TFM.

8.1 Descripción del producto desarrollado

La plataforma desarrollada constituye un sistema distribuido de gestión de actualizaciones OTA seguras compuesto por:

- **Módulo de gestión centralizado (Updates):** Servicio backend implementado en Go que orquesta la creación, firma, cifrado y distribución de paquetes de actualización mediante API REST.
- **Interfaz web de administración:** Aplicación frontend que permite gestionar flotas de dispositivos, crear campañas de actualización y monitorizar despliegues.
- **Motor de flujos de trabajo (WFX):** Sistema de orquestación de estados para gestión del ciclo de vida de actualizaciones con workflows directos, por fases y personalizados.
- **Agente de actualización en dispositivo:** Cliente basado en SWUpdate extendido con soporte para ASCON (LWC) y MLDSA (PQC) para dispositivos Linux embebido.

8.2 Recursos de hardware y software utilizados

Hardware de desarrollo:

- Estación de trabajo: Intel Core i7, 32 GB RAM, 512 GB SSD
- Dispositivos de prueba: Raspberry Pi 3, Arduino UNO R4
- Servidor de desarrollo para despliegue de la plataforma

Software de desarrollo:

- **Sistema operativo:** Linux Ubuntu 22.04 LTS
- **Lenguajes y runtimes:** Go 1.21+, Python 3.10+, Node.js 18 LTS, GCC/G++ 11+
- **Herramientas de contenedores:** Docker 24.0+, Kubernetes 1.27+, kubectl, helm
- **Librerías criptográficas:** OpenSSL 3.5+ (compilado manualmente para soporte PQC), ASCON 1.3.0 (implementación oficial en C), liboqs
- **Herramientas embebidas:** Yocto Project 4.0+, U-Boot, toolchains ARM/ARM64
- **Frameworks del proyecto:** SWUpdate (modificado), WFX, Lamassu IoT PKI, SWUGenerator (extendido)

8.3 Requisitos funcionales

- Soporte para múltiples algoritmos de cifrado simétrico: AES (CBC/CTR/GCM en 128/192/256 bits) y ASCON
- Soporte para firma digital tradicional (RSA, ECDSA) y post-cuántica (MLDSA)
- Gestión de claves criptográficas mediante integración con Lamassu KMS
- Creación automatizada de paquetes de actualización en formato SWU
- Distribución de actualizaciones mediante workflows configurables (directo, por fases, personalizado)
- Despliegue gradual con estrategias de rollout (porcentaje/cantidad fija)
- Progresión entre batches en modo automático o manual
- Monitorización en tiempo real del estado de las actualizaciones
- Interfaz web completa para gestión del sistema
- Integración con Lamassu DMS para organización de dispositivos

8.4 Requisitos no funcionales

- **Seguridad:**
  - Cumplimiento de estándares NIST para algoritmos criptográficos
  - Autenticación mutua TLS (mTLS) entre dispositivos y backend
  - Almacenamiento seguro de claves mediante HSM o PKCS#11
  - Cumplimiento de Cyber Resilience Act (CRA) e IEC 62443-4-2
- **Escalabilidad:** Capacidad para gestionar miles de dispositivos mediante arquitectura cloud-native en Kubernetes
- **Rendimiento:** ASCON 2-5× más rápido que AES en dispositivos sin aceleración hardware
- **Usabilidad:** Interfaz intuitiva que abstrae complejidad criptográfica
- **Trazabilidad:** Registro auditable completo del ciclo de vida de cada actualización
- **Confiabilidad:** Estrategia A/B con rollback automático ante fallos

8.5 Especificaciones técnicas de dispositivos objetivo

Los dispositivos IoT que ejecutan el agente de actualización requieren:
Hardware mínimo:

- Procesador: ARM Cortex-A7 o superior
- RAM: 128 MB (recomendado 256 MB)
- Almacenamiento: 500 MB mínimo para esquema A/B
- Conectividad: Ethernet, Wi-Fi o módulo celular

Software:

- Linux embebido (kernel 5.10 LTS+) generado con Yocto/Buildroot
- U-Boot 2022.01+ con soporte para arranque condicional
- Estructura de particiones A/B (dual-copy)
- SWUpdate modificado con ASCON y MLDSA
- OpenSSL 3.5+ en dispositivo

8.6 Condiciones específicas del desarrollo

- **Modificación de código open-source:** Extensión de SWUpdate y SWUGenerator con nuevos algoritmos criptográficos manteniendo compatibilidad con versiones estándar
- **Integración con infraestructura Ikerlan:** Uso de Lamassu PKI para gestión de identidades y claves
- **Compatibilidad multiplataforma:** Solución debe funcionar en arquitecturas ARM, ARM64, x86 y RISC-V
- **Metodología ágil:** Desarrollo iterativo con entregas incrementales y revisiones periódicas
- **Control de versiones:** Gestión mediante Git con repositorios diferenciados para componentes públicos y privados
- **Documentación técnica:** Generación de documentación de API y guías de implementación

8.7 Condiciones de seguridad aplicadas

- Todas las claves privadas de firma almacenadas en HSM o mediante PKCS#11
- Rotación de claves conforme a política de seguridad (12 meses o tras compromiso)
- Autenticación de dispositivos mediante certificados X.509 de Lamassu PKI
- Cifrado en tránsito (TLS 1.3) y en reposo para paquetes de actualización
- Auditoría completa de operaciones sensibles (firma, despliegue, fallos)

8.8 Pruebas y validación realizadas

- **Benchmarks de rendimiento:** Comparativa ASCON vs AES en Raspberry Pi 3 y Arduino UNO R4
- **Pruebas de integración:** Validación del flujo completo de actualización OTA
- **Pruebas de seguridad:** Verificación de firmas digitales y cifrado de paquetes
- **Pruebas de rollback:** Validación de recuperación automática ante actualizaciones fallidas
- **Pruebas de escalabilidad:** Simulación de despliegues con múltiples dispositivos

9 Tareas desarrolladas

9.1 Investigación y análisis (Semanas 1-4)

- Estudio del estado del arte en criptografía ligera y post-cuántica
- Análisis de algoritmos candidatos: AES, ASCON, algoritmos PQ del NIST
- Evaluación de herramientas existentes: SWUpdate, WFX, Lamassu
- Diseño de la arquitectura de la solución

Departamentos implicados: Área de ciberseguridad, Área de IoT

9.2 Modificación de SWUpdate (Semanas 5-8)

- Análisis del código fuente de SWUpdate
- Implementación de selector de algoritmos de cifrado simétrico
- Integración de ASCON en implementación C nativa
- Soporte para autodetección de algoritmo con OpenSSL
- Extensión de metadatos para incluir información del algoritmo
- Habilitación de firma post-cuántica mediante menuconfig
- Actualización a OpenSSL 3.5+ con compilación manual

Departamentos implicados: Área de sistemas embebidos, Área de ciberseguridad

9.3 Desarrollo de SWUGenerator extendido (Semanas 9-12)

- Evaluación de librería Python de ASCON y detección de problemas de rendimiento
- Integración de código C de ASCON para optimización
- Implementación de firma directa con acceso a clave privada
- Implementación de firma CMS con PKCS#11
- Desarrollo de interfaz CLI para generación de paquetes
- Pruebas de compatibilidad con SWUpdate modificado

Departamentos implicados: Área de desarrollo de software

9.4 Desarrollo del módulo Updates (Semanas 13-16)

- Diseño del modelo de datos (Update-Pack, Launch)
- Implementación de API REST en Go
- Integración con Lamassu KMS para operaciones criptográficas
- Integración con Lamassu DMS para gestión de dispositivos
- Integración con WFX para creación y gestión de Jobs
- Lógica de rollout gradual (porcentaje/fixed)
- Sistema de monitorización del estado de actualizaciones

Departamentos implicados: Área de desarrollo de software, Área de IoT

9.5 Desarrollo de la interfaz web (Semanas 17-20)

- Diseño de la experiencia de usuario (UX)
- Implementación en React de las vistas principales
- Módulo de gestión de claves (KMS)
- Módulo de creación de actualizaciones
- Módulo de gestión de lanzamientos
- Visualización temporal del estado de dispositivos
- Dashboard con estadísticas agregadas

Departamentos implicados: Área de desarrollo de software

9.6 Validación y demostración (Semanas 21-24)

- Configuración de entorno de demostración
- Desarrollo del caso de uso (sistema de control de tanques)
- Pruebas de actualización en dispositivos reales
- Validación de cifrado con ASCON vs AES
- Validación de firma post-cuántica
- Documentación técnica y memoria del TFM

Departamentos implicados: Área de IoT, Área de sistemas embebidos

10 Valoración de competencias adquiridas

El desarrollo del TFM ha permitido aplicar y ampliar significativamente los conocimientos adquiridos durante el máster:

10.1 Ciberseguridad

Los conceptos teóricos de criptografía simétrica y asimétrica estudiados en las asignaturas del máster se han puesto en práctica mediante la implementación de sistemas reales de cifrado y firma. La incorporación de algoritmos post-cuánticos ha requerido investigación adicional más allá del temario, proporcionando una visión prospectiva del campo.

10.2 Arquitecturas de software

Los principios de diseño de arquitecturas escalables y microservicios se han aplicado directamente en la descomposición de responsabilidades entre los diferentes módulos (KMS, Updates, WFX). La experiencia práctica ha reforzado conceptos como separación de concerns, diseño de APIs y patrones de integración.

10.3 Investigación e innovación

La necesidad de evaluar algoritmos emergentes (ASCON) y tecnologías anticipatorias (PQ) ha desarrollado competencias de investigación, análisis crítico de literatura científica y evaluación empírica de soluciones. La contribución a proyectos open-source mediante modificaciones de SWUpdate representa una aportación tangible a la comunidad.

11 Problemas planteados y resolución

11.1 Problema 1: Rendimiento de ASCON en Python

Planteamiento: La librería Python de ASCON presentaba un rendimiento inaceptable, tardando segundos en cifrar pocos megabytes, lo que imposibilitaba su uso para paquetes de firmware de tamaño real.

Resolución: Se optó por integrar directamente la implementación en C de ASCON, obteniendo una mejora de rendimiento de varios órdenes de magnitud. Esto requirió desarrollar wrappers de Python para invocar el código C nativo.

11.2 Problema 2: Limitaciones arquitecturales de SWUpdate

Planteamiento: SWUpdate solo soportaba AES-CBC sin mecanismo de selección de algoritmo, requiriendo modificaciones profundas en el código fuente.

Resolución: Se implementó un sistema de selección de algoritmo mediante parámetros CLI y autodetección con OpenSSL. Se modificó el parser del descriptor sw-description para incluir información del algoritmo. Se estableció una convención de nomenclatura de archivos de claves para facilitar la selección automática.

11.3 Problema 3: Dependencia de OpenSSL 3.5+

Planteamiento: El soporte post-cuántico requiere OpenSSL 3.5 o superior, versión no disponible en los repositorios oficiales de las distribuciones Linux actuales.

Resolución: Compilación e instalación manual de OpenSSL desde el repositorio oficial. Documentación detallada del proceso para facilitar la reproducibilidad del entorno de desarrollo.

11.4 Problema 4: Integración de metadatos en WFX Jobs

Planteamiento: WFX no proporciona un mecanismo estándar para transmitir información sobre algoritmos de cifrado a los dispositivos.

Resolución: Utilización del campo "definition"(diseñado para usos personalizados) en los Jobs de WFX para inyectar metadatos JSON con información del algoritmo, IV y URI de descarga. Modificación del agente SWUpdate para parsear estos metadatos y configurar el proceso de descifrado dinámicamente.

11.5 Problema 5: Gestión de claves en Lamassu

Planteamiento: Lamassu originalmente solo gestionaba claves asimétricas (PKI), sin soporte para claves simétricas requeridas por el cifrado.

Resolución: Extensión del módulo KMS de Lamassu para soportar generación, importación y almacenamiento de claves simétricas. Implementación de operaciones de cifrado/descifrado y generación de MAC directamente en el KMS.

12 Conclusiones y resultados obtenidos

El proyecto ha alcanzado satisfactoriamente todos los objetivos planteados, resultando en una plataforma funcional y completa para la gestión segura de actualizaciones OTA en dispositivos IoT.

Se ha demostrado empíricamente que la criptografía ligera, específicamente ASCON, merece la pena en dispositivos con recursos limitados. Los resultados muestran un menor consumo de recursos computacionales y de memoria comparado con AES, sin comprometer la seguridad. Esta validación es especialmente relevante para el ecosistema IoT, donde millones de dispositivos operan con restricciones estrictas de hardware.

La integración de algoritmos post-cuánticos constituye una preparación estratégica para amenazas futuras. Aunque las computadoras cuánticas no son una amenaza inmediata, la incorporación de estos algoritmos es un cambio que merece la pena de cara al futuro, considerando los largos ciclos de vida de los dispositivos IoT (10-20 años) y la posibilidad de ataques "harvest now, decrypt later".

La plataforma desarrollada permite distribuir actualizaciones de manera segura mediante cifrado y firma digital, proporcionando confidencialidad, integridad y autenticidad. La interfaz web abstrae la complejidad técnica, facilitando que organizaciones sin equipos especializados puedan implementar actualizaciones OTA seguras.

Los resultados técnicos incluyen:

- Extensión exitosa de SWUpdate y SWUGenerator con nuevos algoritmos criptográficos
- Desarrollo de un servicio Updates completo con API REST
- Implementación de una interfaz web intuitiva y funcional
- Integración coherente con WFX y Lamassu
- Demostración práctica mediante caso de uso real (sistema de control de tanques)

13 Nuevos productos y desarrollos derivados

El TFM ha generado las bases para varios desarrollos y oportunidades de negocio:

13.1 Producto: Plataforma OTA Segura como Servicio

La plataforma desarrollada puede comercializarse como solución SaaS (Software as a Service) para empresas que necesiten gestionar actualizaciones de dispositivos IoT. El valor diferencial reside en el soporte nativo de criptografía ligera y post-cuántica, características únicas en el mercado actual.

13.2 Proyectos de I+D derivados

El proyecto abre líneas de investigación en:

- Optimizaciones hardware-específicas de algoritmos LWC en microcontroladores
- Certificación formal de implementaciones criptográficas
- Mecanismos de recuperación ante actualizaciones fallidas
- Análisis de métricas de despliegue para optimización de estrategias de rollout

14 Aportaciones en materia de aprendizaje

El desarrollo del TFM en entorno empresarial ha supuesto aportaciones significativas más allá del aprendizaje académico:

14.1 Experiencia en entorno profesional real

Trabajar en Ikerlan ha proporcionado exposición a metodologías de trabajo profesionales, estándares de calidad de código, procesos de revisión y documentación técnica. Esta experiencia complementa la formación teórica del máster con competencias prácticas directamente aplicables en el mercado laboral.

14.2 Competencias de investigación aplicada

El proyecto ha desarrollado la capacidad de identificar, evaluar e integrar tecnologías emergentes (ASCON, algoritmos PQ) en soluciones prácticas. Esta competencia de investigación aplicada, que conecta la literatura científica con la implementación real, es fundamental para roles de I+D.

14.3 Visión integral de proyectos tecnológicos

La necesidad de abordar aspectos diversos (criptografía, sistemas embebidos, desarrollo web, arquitectura de software, UX) ha proporcionado una visión holística de proyectos tecnológicos complejos. Esta perspectiva transversal es especialmente valiosa para roles de arquitectura de soluciones o gestión técnica de proyectos.

15 Evaluación y sugerencias de mejora

15.1 Evaluación de las prácticas

Aspectos positivos:

- Libertad para investigar y proponer soluciones innovadoras
- Apoyo constante del tutor y equipo de Ikerlan
- Proyecto con aplicabilidad real y potencial comercial
- Integración en equipos multidisciplinares (ciberseguridad, IoT, desarrollo)
- Oportunidad de trabajar con tecnologías emergentes (PQ)

16 Firmas y fecha

Director del Proyecto	**Tutor del Proyecto**
Iñaki Garitano	Haritz Saiz
Fecha: _____	Fecha: _____

Alumno
Álvaro Díez-Andino Herrera
Fecha: _____