UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

LABORATORIO ELECTRÓNICA DIGITAL III
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
LAB01 – PROGRAMACIÓN EN LENGUAJE C (FUNCIÓN ML-KEM)

OBJETIVOS

- Aplicar los conceptos de programación en lenguaje C en la implementación de aplicaciones reales.
- Implementar la función hash ML-KEM del estándar de criptografía NIST-FIPS-203.
- Comprender y utilizar documentación técnica como estándares internacionales.

DESCRIPCIÓN

El estándar NIST FIPS 203 es una especificación técnica desarrollada por el Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés) de los Estados Unidos, que establece los requisitos para los algoritmos de encapsulación de clave basados en módulo de retícula (ML-KEM). Este estándar detalla los procesos para la generación de claves, encapsulación y desencapsulación de las mismas, proporcionando conjuntos de parámetros específicos para asegurar su implementación segura. La creación de este estándar es un paso crítico hacia la estandarización de tecnologías de cifrado que puedan resistir potenciales amenazas futuras, incluidos los ataques de computadoras cuánticas.

Los algoritmos de encapsulación de clave, como los especificados en el estándar NIST FIPS 203, desempeñan un papel crucial en la seguridad de las comunicaciones cifradas al facilitar un intercambio seguro de claves criptográficas entre las partes. Estos algoritmos permiten que una parte genere un par de claves, una pública y una privada, y luego use la clave pública para "encapsular" una clave de sesión o clave simétrica. La parte receptora, que posee la clave privada correspondiente, puede entonces "desencapsular" esta clave de sesión y utilizarla para descifrar los mensajes o datos transmitidos. Este proceso asegura que solo las partes que poseen las claves privadas adecuadas puedan acceder al contenido cifrado, proporcionando una capa robusta de seguridad para las comunicaciones digitales. Este mecanismo es fundamental para establecer canales de comunicación seguros en internet, protegiendo la transmisión de información sensible contra interceptaciones no autorizadas.

La necesidad de este estándar surge del creciente consenso en la comunidad científica y tecnológica sobre la eventual amenaza que las computadoras cuánticas representarán para la criptografía actual. Los algoritmos de cifrado clásicos, como RSA y ECC, que se basan en la dificultad de factores primos grandes y en la criptografía de curva elíptica, respectivamente, podrían ser fácilmente vulnerables ante un ataque cuántico efectivo. El desarrollo de estándares como el FIPS 203 es esencial para adelantarse a estas vulnerabilidades, garantizando la seguridad de la información digital en una era post-cuántica mediante el uso de criptografía que se espera sea resistente a los poderes de cómputo de las futuras computadoras cuánticas.

Las posibles aplicaciones de los estándares definidos por el NIST FIPS 203 son vastas y fundamentales para la seguridad de las comunicaciones digitales y el almacenamiento de datos en el futuro. Desde la protección de la infraestructura crítica y las comunicaciones gubernamentales hasta la seguridad en transacciones financieras y el intercambio de información sensible en el sector privado, este estándar tiene el potencial de asegurar la integridad y confidencialidad de los datos contra adversarios con capacidades de cómputo avanzadas. Además, su implementación podría ser un componente crucial en el desarrollo de tecnologías seguras de blockchain, sistemas de identificación digital y otros servicios en línea que requieren una robusta protección criptográfica.

PROCEDIMIENTO

1. Estudie el estándar NIST FIPS 203, enfocándose en los algoritmos de encapsulación de clave basados en módulo de retícula (ML-KEM) y comprenda sus procesos para la generación de claves, encapsulación y desencapsulación, así como los conjuntos de parámetros asociados.

2. Diseñe un diagrama de flujo o escriba el pseudocódigo para el proceso de encapsulación y desencapsulación de claves según se especifica en el estándar NIST FIPS 203.

3. Desarrolle en lenguaje C un programa que implemente el algoritmo ML-KEM del estándar NIST FIPS 203. Este programa deberá ser capaz de generar un par de claves (pública y privada), encapsular un mensaje (o clave de sesión) utilizando la clave pública, y luego desencapsular el mensaje utilizando la clave privada. El programa recibirá como parámetros en consola los datos necesarios para la generación de claves y la encapsulación/desencapsulación. El resultado de la operación (ya sea la clave de sesión generada durante la encapsulación o el mensaje desencapsulado) se imprimirá en consola. El programa únicamente deberá utilizar librerías estándar de C y deberá funcionar en Linux sin modificaciones.

4. Evalúe el desempeño del programa en su computadora personal para diferentes tamaños de mensajes y diferentes configuraciones de parámetros de seguridad (por ejemplo, diferentes tamaños de claves).

5. Implemente el código del algoritmo ML-KEM en una Raspberry Pi Pico y mida el tiempo de ejecución para el proceso de encapsulación y desencapsulación, utilizando diferentes configuraciones de parámetros de seguridad. Asegúrese de que el microcontrolador (MCU) funcione a su máxima frecuencia posible.

6. Compare el tiempo de ejecución en el microcontrolador con el tiempo de ejecución en su computadora personal.

7. Elabore un reporte que incluya el procedimiento de compilación y ejecución del programa, una explicación del diagrama de flujo o pseudocódigo desarrollado en el paso 2, los resultados de las mediciones de desempeño y una discusión sobre estos resultados, junto con las conclusiones pertinentes.

8. Suba el reporte y el código fuente en un solo archivo comprimido a la plataforma Moodle del laboratorio antes de la media noche del viernes 8 de marzo, asegurándose de nombrar el archivo de acuerdo con las instrucciones dadas, incluyendo el número de la práctica y los apellidos de los integrantes del grupo.

EVALUACIÓN

1. Funcionamiento (40%) – Se verificará el correcto funcionamiento del código y se evaluará el tiempo de ejecución. Un programa cuyos resultados sean erróneos tiene nota cero en la rúbrica de funcionamiento.
2. Sustentación (30%) – preguntas sobre el funcionamiento del código y dominio del estándar y en particular de la función ML-KEM definida en el estándar.
3. Estructura, documentación y organización del código C (20%) – la correcta estructuración del código en funciones y librerías, la correcta documentación de las funciones utilizando Doxygen y el uso de tabulaciones serán tenidos en cuenta en este ítem de calificación.
4. Reporte (10%) – La calidad del reporte y en particular de las conclusiones será vital para una buena calificación en este ítem. Utilice la plantilla de latex dispuesta para los informes de laboratorio.