RETO 1 CODEFEST AD ASTRA 2024: Ciberseguridad en sistemas satelitales de observación de la Tierra

1. Introducción

En un mundo cada vez más dependiente de la tecnología, la seguridad de la información se ha vuelto crítica. Las misiones satelitales de observación de la Tierra son vitales para la recopilación de datos en diversos campos como la meteorología, la agricultura, la gestión de desastres y la seguridad nacional. Sin embargo, la seguridad de los datos transmitidos desde y hacia estos satélites es esencial para evitar interferencias malintencionadas que puedan poner en riesgo la integridad de la información y la misión en sí. Por lo tanto, se requieren sistemas de cifrado robustos y eficientes para garantizar la seguridad de la comunicación satelital.

2. Motivación

Las misiones satelitales de observación de la Tierra tienen una importancia crítica en la toma de decisiones en una amplia gama de campos, desde la agricultura hasta la gestión de desastres naturales. Sin embargo, la seguridad de los datos transmitidos entre los satélites y las estaciones terrestres es esencial para garantizar la integridad de la información y evitar interferencias maliciosas. El cifrado y descifrado de datos desempeñan un papel crucial en este sentido, ya que garantizan que la información transmitida no pueda ser interceptada o manipulada por terceros no autorizados.

3. Objetivo

El objetivo principal de este reto es promover el desarrollo y la mejora de técnicas de cifrado y descifrado para su implementación en misiones satelitales de observación de la Tierra. Al participar en este desafío, los participantes del CODEFEST AD ASTRA 2024 tendrán la oportunidad de:

- ***Ampliar conocimientos en criptografía:*** Los participantes podrán profundizar en los fundamentos teóricos de la criptografía y comprender cómo aplicar esos conceptos en el diseño de sistemas de cifrado eficientes y seguros.

- ***Aplicar conceptos teóricos en un entorno práctico:*** A través del desarrollo de soluciones criptográficas para la comunicación satelital, los participantes podrán aplicar sus conocimientos teóricos en un entorno práctico y realista.

- ***Desarrollar soluciones innovadoras:*** Se espera que los participantes desarrollen soluciones innovadoras que aborden los desafíos específicos relacionados con la seguridad de la comunicación satelital, como la mitigación de interferencias y la protección contra ataques de ingeniería inversa.

4. Descripción del reto

Cada equipo debe diseñar e implementar una solución para cifrado y descifrado de imágenes satelitales en lenguaje C/C++ con las siguientes funcionalidades:

- ***Desarrollo de un algoritmo de cifrado avanzado*****:** los participantes deberán diseñar e implementar una función de cifrado que cumpla con los requisitos de seguridad y rendimiento necesarios para la comunicación satelital. Esto incluye considerar la resistencia a ataques criptoanalíticos, la eficiencia computacional y la capacidad de implementación en sistemas embebidos con recursos limitados. La función debe usar como parámetros la ruta de la imagen de entrada, y la ruta de la imagen de salida.

- ***Desarrollo de un algoritmo de descifrado*****:** al tratarse de una solución simétrica, los participantes deberán diseñar e implementar también descifrado que cumpla con la función de descifrar los datos cifrados que envía el satélite hasta la estación terrena, **y verificar que la información recibida no haya sido objeto de modificación alguna**. La función debe usar como parámetros la ruta de la imagen de entrada, y la ruta de la imagen de salida.

- ***Uso de llaves dinámicas:*** simulando el escenario de comunicación entre el satélite y otro activo del segmento espacial o tierra, se debe diseñar una estrategia para generación de las llaves desacoplada, es decir la llave no debería ser siempre la misma y cada función (para cifrar y descifrar) debe usar la correspondiente llave usada en el algoritmo de cifrado. Recuerden que la generación de la llave no puede ser trivial (e.g., un hash del timestamp) porque esto haría que todo el proceso sea vulnerable. Esta es una de las partes más interesantes del reto !!!

5. TIPS y CONDICIONES

- **Cualquier comportamiento inapropiado (e.g., envío de malware) será informado a sus correspondientes universidades.**

- Revisen los estándares para cifrado en misiones espaciales

    - CCSDS 350.0-G-3 https://public.ccsds.org/Pubs/350x0g3.pdf (Sección 2 a 4.5)
    - CCSDS 350.1-G-3 https://public.ccsds.org/Pubs/350x1g3.pdf (Sección 2 a 3.4)
    - CCSDS 350.9-G-2 https://public.ccsds.org/Pubs/350x9g2.pdf (Sección 2 a 3.4)
    - CCSDS 352.0-B-2 https://public.ccsds.org/Pubs/352x0b2.pdf (Sección 2 a 3.4)

- Identifiquen librerías en código C /C++ que les permitan implementar el reto.

- Las operaciones de cifrado y descifrado deben ser optimizadas y consientes de la cantidad de memoria disponible; tengan en cuenta que el código está pensado para ser ejecutado en un sistema embebido con no más de 4 G de RAM.

- Dediquen tiempo al diseño de la solución; recuerden que el algoritmo AES necesita llaves y estas llaves deben ser dinámicas. Piensen muy bien en la estrategia de generación o de transmisión segura de la llave, el satélite y la estación terrena solo se conectan cuando están alineados.

- Dado que la evaluación es automatizada la interface de su solución debe seguir una plantilla que les será proporcionada.

- No olviden hacer pruebas y evaluar la calidad del código. La calidad interna del código la pueden evaluar usando las herramientas INFER (Facebook) y SonarQube.

- El código entregado debe tener "*inline comments",* es decir, comentarios en el código a nivel de función y bloques de sentencias. Adicionalmente, se recomienda el buen nombramiento de los identificadores (clases, variables, funciones) de acuerdo con las buenas prácticas recomendadas para el lenguaje C/C++, así como la buena *indentación* del código.

- Se debe entregar un documento (PDF) que describa la solución propuesta: (i) estrategia de cifrado, (ii) estrategia de descifrado, (iii) estrategia para uso de llaves dinámicas, (iii) estrategia para gestión de memoria en sistema embebido, (iv) librerías utilizadas, (iv) estrategia de verificación y validación usada para medir la calidad interna del código y la calidad de la solución. Mala redacción y falta de ortografía en el documento dará lugar a puntos negativos en la evaluación de la solución.

- En caso de usar código de terceros, se debe mencionar explícitamente la fuente y la licencia del código (e.g., eclipse license).

- Se les proporcionará un conjunto de imágenes que podrán usar para probar su solución.

- La solución (código y documentación) se debe alojar en un repositorio GitHub. El PDF de la documentación debe estar en la carpeta raíz del repositorio.

- El repositorio debe ser privado y se debe agregar a este los siguientes usuarios con permiso "read": **stevenllerenan**, **alejo940502**, **mlinarev**. **Esta condición es descalificatoria si no la cumplen.**

- El repositorio debe incluir un video de no más de 2 minutos donde aparezcan los 4 integrantes explicando el código y la solución.

- El repositorio debe contar con un archivo README explicando la organización de este.

- El enlace del repositorio se debe enviar a más tardar el día 2 de agosto a las 23:55 (Hora Colombia) a través del formulario que les será enviado**.**

- Todos los miembros de cada equipo deben diligenciar la encuesta de percepción del evento que les será enviada.

- Al momento de evaluar la solución, se tomará el último commit en el repo que se haya hecho dentro del rango de tiempo de entrega válido, es decir los commits

realizados después de las 23:55 (Hora Colombia) del 2 de agosto, no serán tomados en cuenta.

6. WORKFLOW DE EVALUACIÓN