# MISIÓN 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 el uso de llaves dinámicas en un ambiente desconectado.

- ***Usar un modelo de ingeniería:*** Los estudiantes desarrollaran una solución de cifrado que podrá ser probada en un modelo de ingeniería que emula la OBC de un CubeSat.

## 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 de forma **dinámica y desacoplada**, es decir la llave no debería ser siempre la misma y cada función (para cifrar y descifrar) debe usar la correspondiente llave pero sin que esta se encuentre en un archivo local. 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 !!!

- ***Sistema embebido:*** para la implementación de la solución cada equipo contará con un kit de desarrollo jetson nano, un teclado, un mouse, y una pantalla. La jetson nano ya viene configurada con las herramientas básicas de desarrollo (e.g., compilador C++), el sistema operativo Ubuntu. Su solución debe ser entregada como código fuente y programa compilado dentro de la tarjeta. Para tal efecto en eel dispositivo encontrará un directorio ***Home/codefest<n>/Codefest2024/Reto1***. **Su solución debe estar ahí al momento de finalizar el reto 1; el archivo main compilado debe estar ahí y los fuentes, no incluya más carpetas dentro de este directorio**. Por otro lado no olvide que su solución debe estar compilada con CMAKE. El equipo organizado les envió con antelación un video explicando el proceso de compilación.

## 5. TIPS y CONDICIONES

- **NO SAQUE LA MICRO SD DE LA JETSON NANO. EL SISTEMA OPERATIVO DE LA JETSON SE ENCUENTRA AHÍ.**

- **NO DESCONECTE LA FUENTE DE PODER SALVO QUE EL SISTEMA ESTÉ APAGADO. AL TERMINAR EL RETO APAGUE LA MAQUINA**

- **EVITE MANIPULAR LA TARJETA, SUS PUERTOS Y PINES CON LAS MANOS. LA ESTÁTICA PUEDE DAÑAR DE FORMA PERMANTENTE LOS COMPONENTES**

- **NO TOQUE EL DISIPADOR DE LA JETSON NANO**

- **Revisen qué librerías y compilador de C/C++ está disponible en la tarjeta. La jetson nano no tiene conexión WIFI.**

- **Desarrollen en sus equipos personales, y luego de que estén seguros que el código funciona, copien a través de USB el código a la jetson nano.**

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

- **El passwd para ingreso al equipo es *adastra* y el usuario es del estilo *Codefest<n>***

- 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. Este límite de consumo de RAM por parte de su solución es obligatorio, **por lo tanto las soluciones que consuman más de 4GB de RAM o no se logren ejecutar en la Jetson nano serán descalificadas.**

- 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. Las llaves no se pueden compartir a través de archivos que residen en la jetson nano; hagan de cuenta que el método para cifrar y el método para descifrar se ejecutan en dos máquinas diferentes y no existe comunicación entre estas.

- 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. Estas imágenes las encontrarán en la carpeta ***Home/codefest<n>/Codefest2024/Reto1/Test_Images***

- La solución (código y documentación) se debe alojar también 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 contar con un archivo README explicando la organización de este.

- El enlace del repositorio se debe enviar a más tardar el día 28 de septiembre a las 05:55 (Hora Colombia) a través del formulario que les será enviado**.**

## 6. CRITERIOS DE EVALUACIÓN

- El equipo envía el formulario dado con el repositorio con el código de la solución,y el equipo evaluador tiene acceso al repositorio
- El repo tiene el código, el documento de diseño, y el archivo readme
- El repo tiene una licencia opensource permisiva
- El código fuente está en la tarjeta jetson nano en la carpeta indicada
- El código está compilado en la tarjeta jetson nano en la carpeta indicada
- La solución no consume más de 4Gs de RAM ni al cifrar ni al descifrar
- El método de cifrado genera una imagen diferente a la original
- Luego de cifrar y descifrar no se pierde calidad en las imágenes de resultado, y estas son iguales a las imágenes originales (es decir, antes de cifrar)
- La solución implementa una estrategia para manejo de llaves dinámicas
- La estrategia para manejo de llaves dinámicas sigue la indicación de que el cifrado y descifrado se ejecutarán en máquinas diferentes
- El código fuente tiene comentarios (“inline comments”)
- El repositorio incluye el resultado de correr Facebook infer o SonarQube sobre el código fuente