Criptografía:
Implmentación y Desarrollo

Alcibíades Bustillo-Zárate

Fecha de Inicio: 2 de abril
Fecha de Entrega: 4 de mayo

1 Introducción

En este proyecto, tendrán la oportunidad de implementar varios criptosistemas y algoritmos de utilidad. Este proyecto tiene como objetivo principal proporcionar a los estudiantes una comprensión práctica de los conceptos teóricos abordados durante el curso de este semestre.

2 Objetivos del Proyecto

Los objetivos del proyecto son los siguientes:

1. Implementar el cifrado afín.
2. Implementar un cifrado de flujo (stream cipher).
3. Generar claves mediante un registro de desplazamiento lineal con retroalimentación (Linear Feedback Shift Register, LFSR).
4. Implementar DES (Data Encryption Standard).
5. Implementar AES (Advanced Encryption Standard).
6. Implementar RSA junto con la generación de claves.
7. Implementar el algoritmo Square-and-Multiply para la exponenciación modular.
8. Implementar la prueba de primalidad de Fermat.
9. Implementar la prueba de primalidad de Miller-Rabin.
10. Implementar el protocolo de ElGamal basado en el intercambio de claves de Diffie-Hellman.
11. Implementar el intercambio de claves basado en curvas elípticas Diffie-Hellman.

3 Metodología

Los estudiantes trabajarán de manera individual para completar este proyecto. Cada pareja deberá seleccionar un conjunto de objetivos para implementar, asegurándose de cubrir una variedad de técnicas y algoritmos criptográficos. El código implementado debe estar debidamente documentado para facilitar su comprensión y revisión.

4 Calendario Tentativo

A continuación se presenta un calendario tentativo para el proyecto, que abarca un período de cinco semanas desde el 2 de abril hasta el 4 de mayo:

- **Semana 1 (2 de abril - 8 de abril):**
  - Los estudiantes escogen sus parejas y seleccionan los objetivos del proyecto.
  - Repaso sobre los algoritmos y criptosistemas seleccionados.
  - Configuración inicial del entorno de desarrollo (instalación de Python, configuración del repositorio de código, etc.).
- **Semana 2 (9 de abril - 15 de abril):**
  - Implementación del cifrado afín y documentación del código.
  - Implementación del cifrado de flujo (stream cipher) y documentación del código.
- **Semana 3 (16 de abril - 22 de abril):**
  - Implementación de la generación de claves mediante LFSR y documentación del código.
  - Implementación de DES y documentación del código.
- **Semana 4 (23 de abril - 29 de abril):**
  - Implementación de AES y documentación del código.
  - Implementación de RSA junto con la generación de claves y documentación del código.
- **Semana 5 (30 de abril - 4 de mayo):**
  - Implementación del algoritmo Square-and-Multiply para la exponenciación modular y documentación del código.
  - Implementación de la prueba de primalidad de Fermat y de Miller-Rabin, y documentación del código.
  - Implementación del protocolo de ElGamal basado en el intercambio de claves de Diffie-Hellman y documentación del código.

- Implementación del intercambio de claves basado en curvas elípticas Diffie-Hellman y documentación del código.
- Preparación y entrega del proyecto.

5 Entrega del Proyecto

La fecha de entrega del proyecto es el 4 de mayo. Los estudiantes deberán presentar su código implementado junto con un informe que incluya una descripción detallada de los algoritmos y criptosistemas implementados, así como ejemplos de su funcionamiento y resultados obtenidos.

6 Conclusiones

Este proyecto proporcionará a los estudiantes una valiosa experiencia práctica en la implementación de algoritmos criptográficos. Al completar este proyecto, los estudiantes habrán fortalecido su comprensión de los conceptos teóricos de la criptografía y habrán adquirido habilidades prácticas en el desarrollo de software seguro y protegido.

7 Criterios de Evaluación

1. Implementación de Criptosistemas:
   - El cifrado afín se implementa correctamente en Python.
   - Se implementa un cifrado de flujo (stream cipher) siguiendo las especificaciones.
   - La generación de claves mediante Linear Feedback Shift Register (LFSR) se realiza adecuadamente.
   - La implementación de DES (Data Encryption Standard) es correcta y funcional.
   - Se implementa AES (Advanced Encryption Standard) con éxito.
   - RSA junto con la generación de claves se implementa de manera correcta.
   - Se implementa el algoritmo Square-and-Multiply para la exponenciación modular.
   - La prueba de primalidad de Fermat se implementa correctamente.
   - La prueba de primalidad de Miller-Rabin se implementa adecuadamente.
   - El protocolo de ElGamal basado en el intercambio de claves de Diffie-Hellman se implementa con éxito.
   - El intercambio de claves basado en curvas elípticas Diffie-Hellman se implementa correctamente.
2. Documentación del Código:
   - El código implementado está debidamente documentado, incluyendo comentarios claros y explicativos.
   - Se proporciona una explicación clara de cada algoritmo y criptosistema implementado en el código.
3. Calidad y Funcionalidad del Código:
   - El código funciona según lo esperado para cada uno de los criptosistemas y algoritmos implementados.
   - Se manejan adecuadamente los errores y excepciones.
   - El código sigue las mejores prácticas de programación en Python.
4. Informe del Proyecto:
   - Se presenta un informe completo que describe detalladamente los algoritmos y criptosistemas implementados.
   - Se incluyen ejemplos de funcionamiento y resultados obtenidos para cada implementación.

- El informe presenta una organización clara y coherente.

5. Cumplimiento de Plazos:
    - El proyecto se entrega antes o en la fecha límite especificada.

8 Puntuación

- **Excelente (10):** Todos los criterios de evaluación se cumplen de manera excepcional y el proyecto demuestra un alto nivel de comprensión y habilidad en la implementación de los criptosistemas.
- **Bueno (7-9):** La mayoría de los criterios de evaluación se cumplen satisfactoriamente y el proyecto demuestra una comprensión sólida de los conceptos de criptografía y habilidades en programación.
- **Aceptable (4-6):** Algunos de los criterios de evaluación se cumplen, pero hay áreas que requieren mejoras en la implementación, documentación o calidad del código.
- **Insuficiente (0-3):** La mayoría de los criterios de evaluación no se cumplen y el proyecto muestra una comprensión limitada de los conceptos de criptografía o habilidades deficientes en programación.