

14-11-2025

Planeación Proyecto Final

Análisis de Algoritmos



Equipo:

Eduardo Ramos Ochoa
Andres Santiago Aguirre Macias
Cesar Emmanuel Gómez Martínez

Códigos:

304489918
220293594
214843698

Profesor:

JORGE ERNESTO LOPEZ ARCE DELGADO

Modulo:

D01

Centro Universitario de Ciencias Exactas e Ingenierías

❖ ALGORITMO SELECCIONADO

- Hemos elegido emplear e incorporar el **algoritmo de PRIM**.

- **¿En qué parte del algoritmo base se integrará?**

La integración se hace como un modo adicional del sistema de cracking que se ha venido empleando a lo largo del semestre (con los enfoques de Fuerza Bruta, Divide y Vencerás, así como Programación Dinámica, luego entonces, en lugar de generar candidatos en orden lexicográfico o por prefijos (tal cual se he empleado anteriormente), con la integración del algoritmo de PRIM, lo que se hace en la integración es:

- a) Se construirá un grafo donde cada nodo equivale a una palabra del wordlist que se proporcione, esto, toda vez que se trabaja en un ambiente controlado.
- b) Se estiman las aristas entre nodos mediante distancias Levenshtein (se almacenan solo las k vecinas más cercanas para cada nodo—grafo k-NN).
- c) Se ejecuta **Prim** sobre ese grafo para obtener el MST.
- d) El **recorrido por el MST (BFS/DFS por componentes)** produce un **orden voraz** en el que se van probando candidatos (palabras del *wordlist*).

Esto sustituye el orden de prueba de candidatos del módulo base por `attack_voraz_prim(...)`.

- **Metodología de implementación**

A continuación, se procede a detallar de manera breve pero concisa, la forma a través de la cual, se planea llevar a cabo la implementación del algoritmo de PRIM en nuestro proyecto, a efecto de realizar dicho planteamiento de manera organizada, cada paso se numerará conforme a su correspondiente orden de realización previsto.

- a) Construcción del grafo en base al wordlist en un ambiente controlado.
- b) Aplicar PRIM para obtener el MST resultante.
- c) Generar el orden de prueba de contraseñas en función del MST.
- d) Integrar el modo PRIM en la GUI como una nueva alternativa de cracking.
- e) Pruebas y ajustes, ejecutando diferentes wordlists y contraseñas objetivo.
- f) Comparar resultados contra Fuerza Bruta y DyV, evaluando:
 - Tiempo real
 - Número de candidatos probados (contraseñas)

g) Documentación y preparación de la entrega final.

- Descripción breve de la GUI

La GUI para este proyecto, se actualizará agregando una nueva opción dentro de la lista de algoritmos con los que ya se contaba anteriormente (FB, DyV y Dinámica), dicha añadidura se denominará como **“Voraz (PRIM + MST)”**.

Cuando se compile el programa y, se haga clic en la opción antes referida, la GUI permitirá realizar las acciones siguientes:

- a) Introducir la contraseña objetivo o su hash
- b) Seleccionar el wordlist de prueba
- c) Visualizar el progreso mientras se ejecuta la búsqueda
- d) Mostrar en pantalla los resultados obtenidos (tiempo y número de pruebas)
- e) Presentar una ventana donde se muestre el MST dibujado, como evidencia del funcionamiento del algoritmo voraz

Gracias a lo anterior, el usuario podrá comparar directamente el rendimiento de PRIM frente a las demás estrategias analizadas en clase.

- **ETAPAS**

A continuación, se señalan las etapas que seguiremos para la incorporación correcta del algoritmo de PRIM en nuestro proyecto:

I. Análisis

- Revisión del código base existente.
- Determinar dónde se integrará PRIM y qué funciones deben modificarse.
- Verificar el wordlist y la manera de generar el grafo.

II. Codificación

- Programar la parte voraz con PRIM dentro del proyecto.
- Agregar la nueva opción en la GUI y los controles necesarios.
- Asegurar que el orden de prueba de contraseñas se base en el MST.

III. Pruebas

- Realizar pruebas con contraseñas reales en un ambiente controlado.
- Analizar tiempos y número de intentos requeridos.
- Corregir posibles errores o comportamientos inesperados.

IV. Integración final

- Comparar el comportamiento del algoritmo PRIM contra los otros métodos vistos en clase (FB, DyV y PD).
- Generar capturas y evidencias para el reporte final y la exposición.

- **¿Cómo validaremos el correcto funcionamiento del algoritmo voraz que hemos elegido?**

Para asegurarnos de que la implementación del algoritmo de PRIM funcione de forma correcta y eficiente, verificaremos lo siguiente:

- I. Que encuentre la contraseña correcta cuando esta se encuentre dentro del wordlist.
- II. Que el tiempo y el número de intentos se registren correctamente en la GUI.
- III. Que el recorrido del MST realmente se esté utilizando para el orden de búsqueda.
- IV. Que la visualización del MST se muestre sin errores dentro de la interfaz.
- V. Que al comparar con los otros enfoques, PRIM muestre mejoras en algunos casos, al encontrar antes la contraseña.
- VI. Se realizarán varias pruebas cambiando:
 - a) La contraseña objetivo
 - b) El tamaño del wordlist
 - c) Los parámetros de PRIM (vecinos k)

Una vez definido lo anterior, procedemos a formular nuestro..

❖ PLAN DE TRABAJO

A continuación, se presenta el esquema de plan de trabajo definido por nuestro equipo, cabe señalar que, tomando en consideración que, durante la semana pasamos la mayor parte del tiempo en el campus, la mayor parte de las veces, nuestras reuniones se llevarán a cabo de forma presencial.

| Tarea | Responsable | Fecha compromiso | Entregable |
|--|---------------------------------------|-------------------------|-------------------------------------|
| Actualización de la GUI para incluir opción PRIM y visualización MST | Eduardo Ramos Ochoa | 14–17 nov | Código Python actualizado (.py) |
| Implementación de Prim sobre wordlist y pruebas funcionales | César Emmanuel Gómez Martínez | 14–18 nov | Resultados funcionales y evidencias |
| Diseño del contenido visual para explicación (capturas, diagramas) | Andrés Santiago Aguirre Macías | 18–20 nov | Diapositivas/recursos del proyecto |
| Redacción de reporte final del proyecto | Eduardo Ramos Ochoa | 19–21 nov | PDF final con conclusiones |
| Validación, ensayo de exposición y revisión general | Todo el equipo | 21 nov | Entrega en Classroom |

Itinerario de reuniones y medios de realización.

| Actividad / Reunión | Fecha | Medio | Objetivo |
|----------------------------|--------------|--------------------------|--|
| Reunión 1 | 14 nov | Presencial | Revisión del código base y división de tareas |
| Reunión 2 | 16 nov | WhatsApp (video llamada) | Integración del algoritmo PRIM + pruebas iniciales |
| Reunión 3 | 19 nov | Presencial | Validación GUI + plan para reporte y exposición |
| Reunión 4 | 21 nov | Presencial | Ensayo final y revisión de entregables |