Seguridad y Algo II

Algunos ejemplos...

Lic. Leandro Meiners

DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Agenda

- Quién soy y por qué un salpicón de seguridad...
- Seguridad y Complejidad Algorítmica
- Estructuras de Datos para Seguridad

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Seguridad de la Información: Conceptos Básicos

- Atributos que busca proveer la SI:
  - Integridad
  - Disponibilidad
  - Confidencialidad

- Denegación de Servicio:

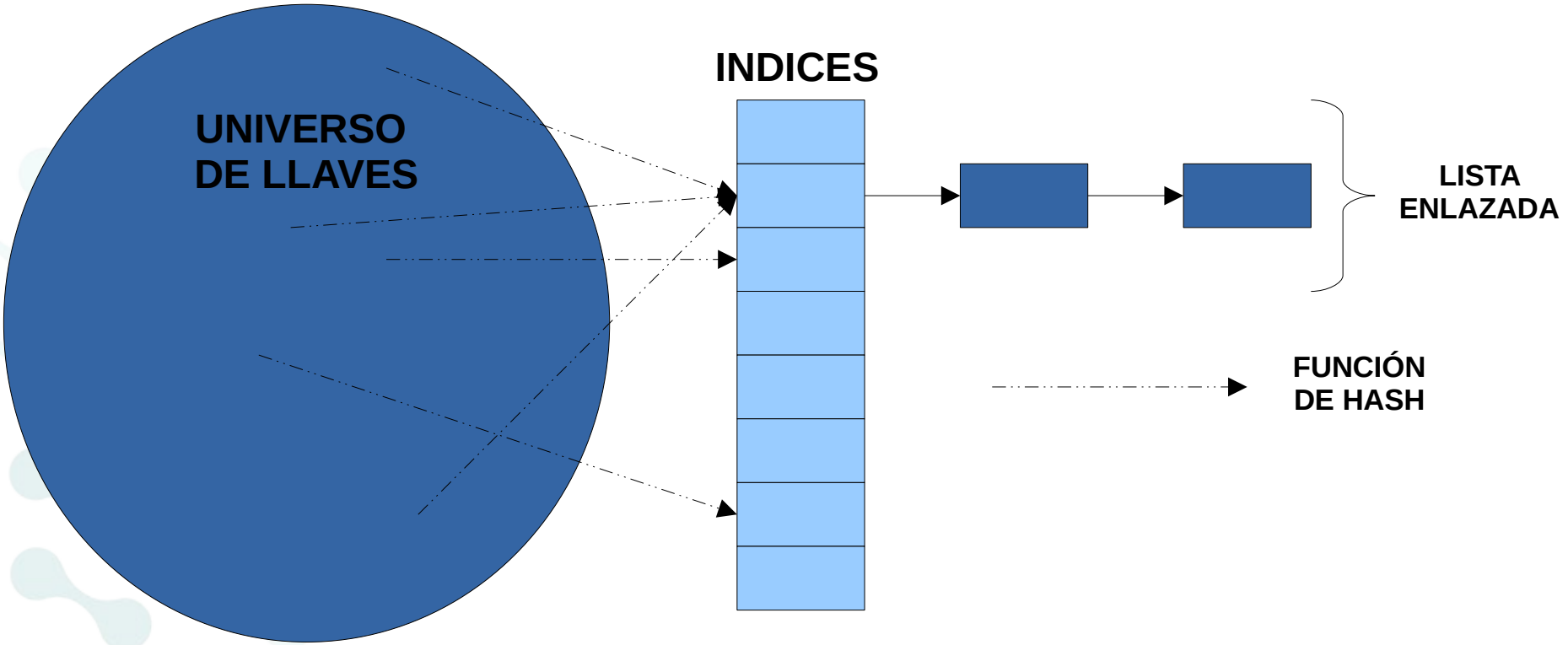
  “Ataque dónde lo que se busca es limitar la disponibilidad de un recurso”

Clases de Denegación de Servicio

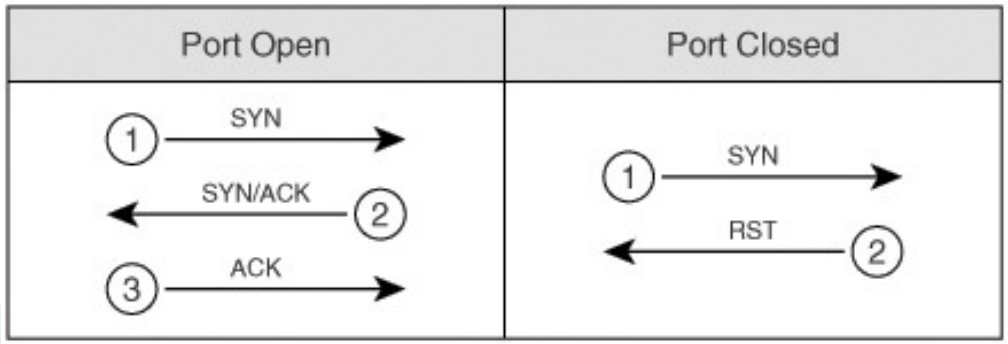
- Interrupción del servicio
  - Falla del recurso debido a una condición no prevista
  - Manipulación del “estado”
- Consumo desmedido de recursos:
  - Inundación
  - Amplificación
  - Distribuidos (DDoS)

Tabla/Mapa de Hash

- Implementación de un diccionario



TCP/IP Port Scanning



- Consiste en enviar un paquete SYN a un rango de direcciones IP (1 o más) evaluando qué puertos tiene abierto en un subrango de puertos (1-65535).

Ejemplo: Bro’s Detector de Scanning de Puertos

- Detección de un Scaneo de puertos: para cada IP origen Bro necesita saber cuántos puertos fueron “tanteados”
- Utiliza una Tabla de Hash de <IP SRC, puerto DST>
- Función de hash utilizada: SRC IP XOR Puerto DST, entonces en x86:
  - Primeros 16 bits provienen de los últimos 16 bits del IP
  - Segundos 16 bits hash de primeros 16 bits de la IP XOR puerto DST

TRIVIAL GENERAR COLISIONES

Resultados del Ataque: DoS de CPU & Pérdida de Paquetes

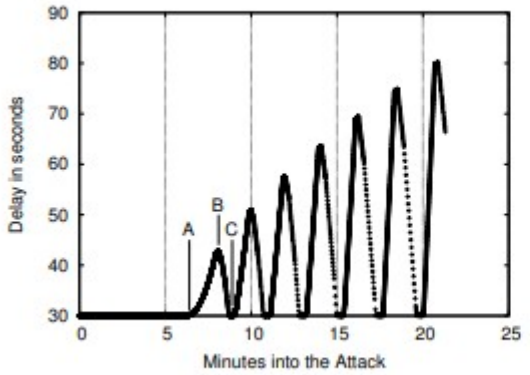


Figure 3: Packet processing latency, 16kb/s.

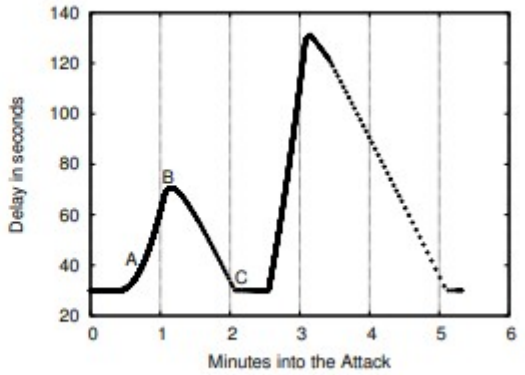


Figure 5: Packet processing latency, 64kb/s.

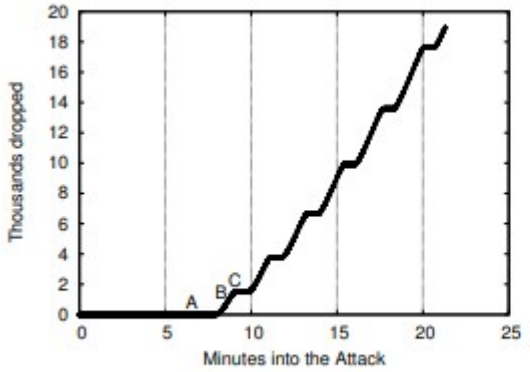


Figure 4: Cumulative dropped packets, 16kb/s.

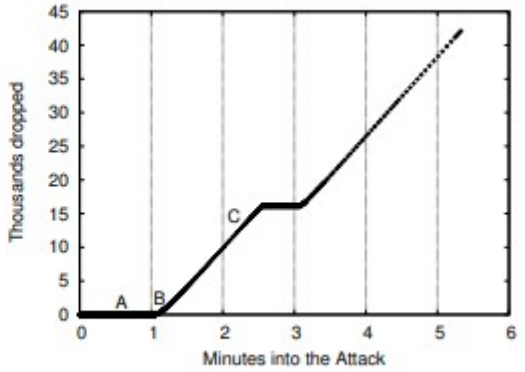


Figure 6: Cumulative dropped packets, 64kb/s.

Más información:
https://www.usenix.org/legacy/events/sec03/tech/full_papers/crosby/crosby.pdf

Soluciones

- Algoritmos de tiempo constante
- Dificultad de predecir colisiones
- Detección del ataque

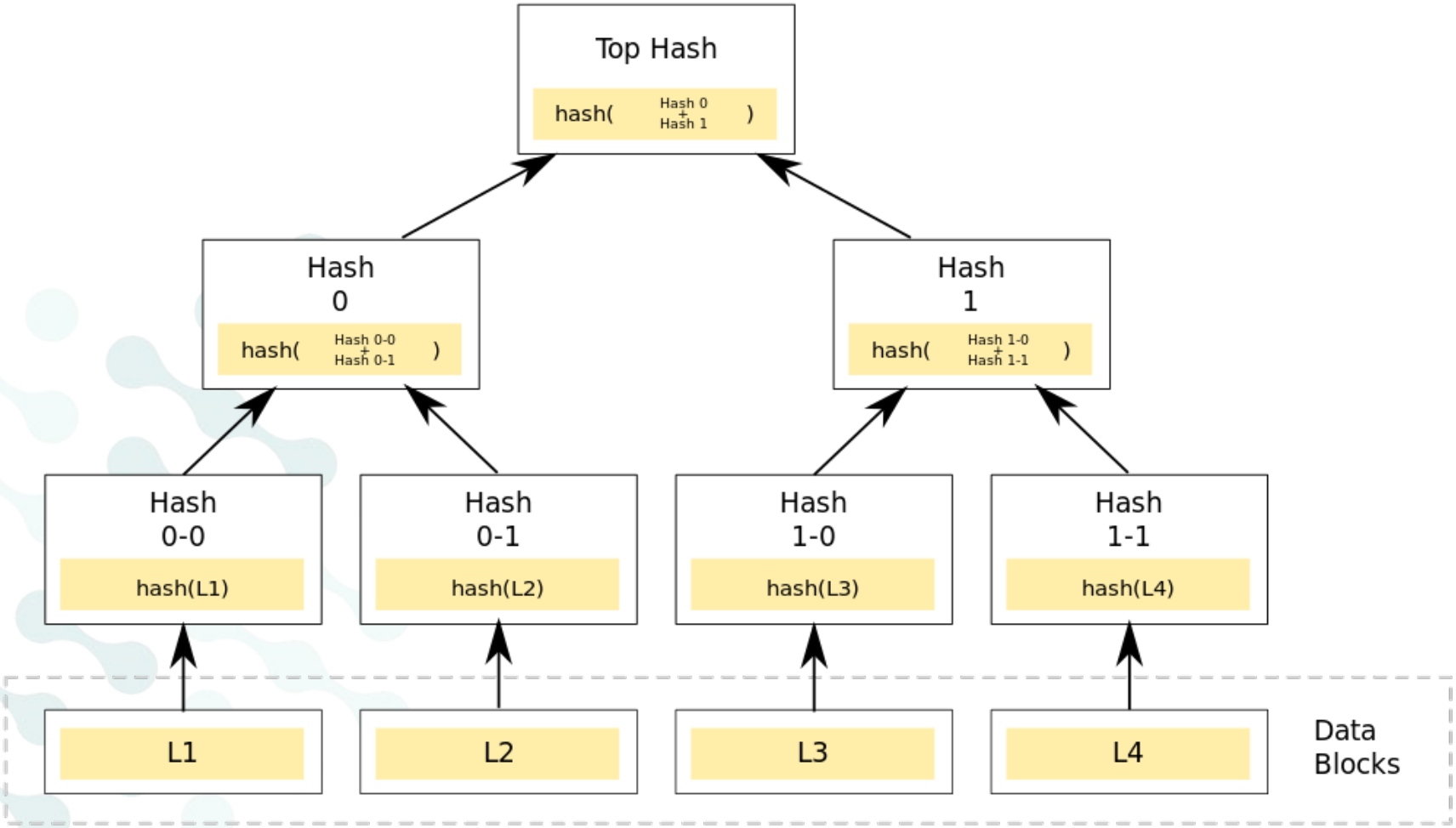
¿En qué contexto les parece que debemos tomar esos recaudos?

Funciones de Hash Criptográficamente Seguras

- Determinista
- Computacionalmente eficiente de calcular
- No reversible / Resistencia al cálculo de pre-imagen:
  - Dado el hash h, debe ser computacionalmente difícil encontrar m / h = hash(m)
- Resistencia al cálculo de segunda pre-imagen:
  - Dado m1, debe ser computacionalmente difícil encontrar m2 / hash(m1) = hash(m2)
- Resistencia a colisiones:
  - Debe ser computacionalmente difícil encontrar m1 y m2 / hash(m1) = hash (m2)

"One-way"

Merkle Trees

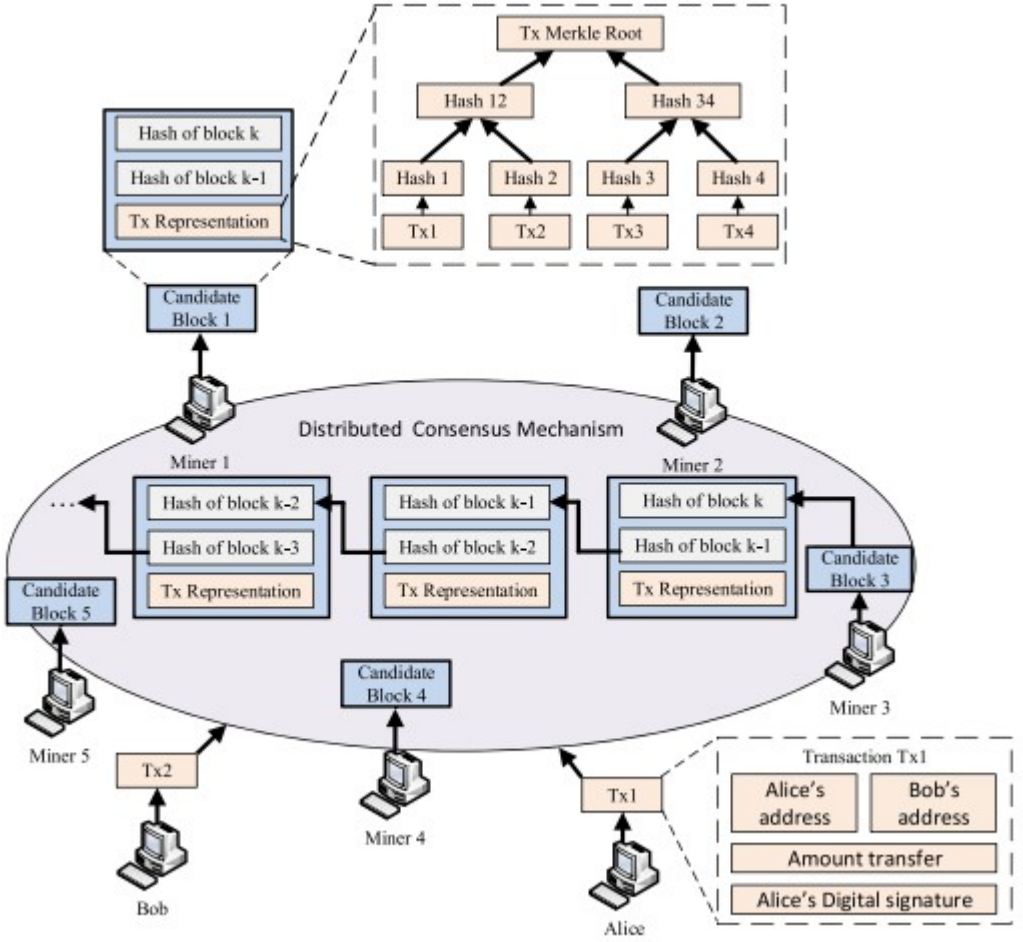


https://en.wikipedia.org/wiki/Merkle_tree#/media/File:Hash_Tree.svg

Usos en Sistemas Distribuidos

- Verificación de bloques de datos intercambiado entre pares
  - Ejemplos:
    - BitCoin:
      - Las transacciones de bitcoin se agrupan en bloques que forman un Merkle Tree
      - Cada bloque es validado por PoW
      - Cambiar un bloque implica cambiar todo una rama
    - GIT:
      - Hash del objecto que es el commit
      - Dificulta “inventar” un commit espurio → implica cambiar ramas
      - Identificadores únicos

Red BlockChain



https://ieeexplore.ieee.org/document/8746079

Almacenamiento de Contraseñas

- ¿Cómo se imaginan que se almacenan las contraseñas?
  - Implementación “inocente pajarito”: texto plano
  - 1era Mejora: Función de hash
    - Problemas:
      - ¿usuarios que repiten la contraseña?
      - Diccionarios de hashes para contraseñas comunes
      - Rainbow tables
  - 2da Mejora: Uso de salts
    - Problemas:
      - Ataques de fuerza bruta
  - 3ra Mejora: Complejidad incremental...

Propiedades de las funciones para almacenar contraseñas

- Resistente al cálculo de la pre-imagen
  - Si de h(pass), puedo calcular p / h(p) = h(pass)
- Uso de un salt para:
  - Evitar colisiones ante la misma contraseña
  - “Agrandar” el espacio de búsqueda
    - Permite mitigar/dificultar ataques con tablas pre-computadas
- Complejidad variable:
  - Trade-off entre tiempo de cálculo y seguridad (para prevenir ataques de fuerza bruta a medida que el poder de cómputo aumenta)

Take-Aways…

- En seguridad no sólo es importante la complejidad algorítmica, sino que también utilizamos el concepto de “computacionalmente intratable” para “garantizar” un nivel de seguridad… ¡siempre es una carrera!

¡Gracias!

¿Preguntas?

lmeiners@dc.uba.ar

DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA