



- **SEGURIDAD** es una medida de confianza de que se preservará la integridad de un sistema y sus datos
- **PROTECCIÓN** es el conjunto de mecanismos que controlan el acceso de los procesos y usuarios a los recursos definidos por los sistemas informáticos.

## PROTECCIÓN: CONTENIDO

- Discutir los objetivos y principios de la protección en un sistema de computación moderno
- Explicar como los dominios de protección combinados con las matrices de acceso son usados para especificar como puede un proceso acceder a los recursos
- Examinar los sistemas de protección basados en capacidades y lenguajes

## OBJETIVOS DE LA PROTECCIÓN

- El SO consiste de una colección de objetos, hardware o software.
- Cada objeto tiene un único nombre y puede ser accedido por un conjunto de operaciones bien definidas.
- El problema de protección – asegura que cada objeto es accedido correctamente y solo por aquellos procesos que les está permitido hacerlo.

## PRINCIPIOS DE PROTECCIÓN

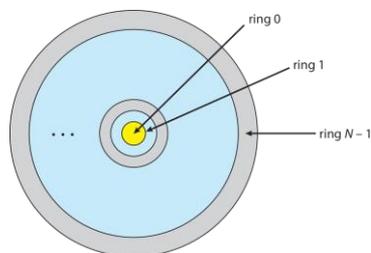
- Principio guía – **principio del menor privilegio**
  - Programas, usuarios y sistemas debería obtener suficientes privilegios para realizar sus tareas
- Considerar el **aspecto de granularidad**
  - Baja (gruesa) granularidad
  - Fina granularidad
- Dominio puede ser usuario, proceso, procedimiento
- **Seguimiento de auditoría**
- Ningún principio es una panacea para las vulnerabilidades de seguridad: se necesita una defensa en profundidad (*defense in depth*)

## PROTECCIÓN EN ANILLOS

### SEPARACIÓN DE PRIVILEGIOS

- Kernel – mayores privilegios
  - **Hypervisors** introducen la necesidad de un nuevo anillo
  - ARMv7 procesadores agregan un anillo **TrustZone(TZ)** para proteger las funciones de criptografía utilizando una nueva instrucción o llamada **Secure Monitor Call (SMC)**
- Implementación ➡ anillos concéntricos

$\left\{ \begin{array}{l} \text{Dado } D_i \text{ y } D_j \text{ sean dos dominios de anillos} \\ \text{Si } j < i \Rightarrow D_i \subseteq D_j \end{array} \right.$

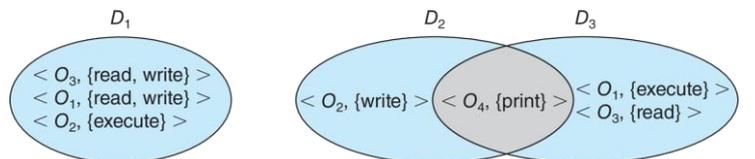


## DOMINIOS DE PROTECCIÓN

- Los anillos de protección separan las funciones en dominios y las ordenan jerárquicamente. Generalización sin jerarquía.
- Principio **necesidad de saber (need-to-know)**
  - **Política: necesidad de saber**
  - **Mecanismo: menor privilegio**

## ESTRUCTURA DE DOMINIOS

- Dominio es conjunto de derechos de acceso
- Derecho de Acceso =  $\langle \text{nombre del objeto, conjunto de derechos} \rangle$  donde el *conjunto de derechos* es un subconjunto de todas las operaciones válidas que pueden ser realizadas por el objeto.
- Los dominios pueden ser:
  - Usuario
  - Proceso
  - Procedimiento



## IMPLEMENTACIÓN DE DOMINIOS (UNIX)

- UNIX

- Dominio = es asociado con un usuario
- Conmutación de dominios realizado vía sistema de archivos
  - Cada archivo está asociado con una identificación de usuarios y un bit de dominio (setuid bit)
  - Cuando el archivo está ejecutando y el setuid = on, entonces la identificación de usuario es pasada al dueño del archivo en ejecución. Cuando se completa la ejecución la identificación de usuario es retornada a su original.

## MATRIZ DE ACCESO

- Vista de la protección como una matriz (*matriz de acceso*).
- Las filas representan dominios.
- Las columnas representan objetos.
- $Acceso(i, j)$  es el conjunto de operaciones que un proceso ejecutando en Dominio<sub>i</sub> puede invocar sobre un Objeto<sub>j</sub>.

object \ domain	$F_1$	$F_2$	$F_3$	printer
$D_1$	read		read	
$D_2$				print
$D_3$		read	execute	
$D_4$	read write		read write	

## USO DE LA MATRIZ DE ACCESO

- Si un proceso en Dominio  $D_i$  trata de hacer “op” sobre el objeto  $O_j$ , entonces “op” debe estar en la matriz de acceso
- Puede ser expandido a protección dinámica
  - Agregar operaciones, borrar derechos de acceso
  - Derechos de acceso especiales:
    - *dueño de  $O_i$*
    - *copiar op desde  $O_i$  a  $O_j$*
    - *control –  $D_i$  puede modificar los derechos de acceso de  $D_j$*
    - *transferencia – conmutar del  $D_i$  a  $D_j$*

object domain	$F_1$	$F_2$	$F_3$	laser printer	$D_1$	$D_2$	$D_3$	$D_4$
$D_1$	read		read			switch		
$D_2$				print			switch	switch
$D_3$		read	execute					
$D_4$	read write		read write		switch			

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## USO DE MATRIZ DE ACCESO

- **La Matriz de Acceso:** su diseño separa mecanismos de políticas
  - Mecanismo
    - El SO provee matriz de acceso + reglas
    - La matriz es manipulada solamente por agentes autorizados y las reglas son estrictamente forzadas
  - Políticas
    - El usuario dicta la política
    - Quién puede acceder a que objeto y de que modo

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## IMPLEMENTACIÓN DE LA MATRIZ DE ACCESO

- **TABLA GLOBAL.** Consiste de un conjunto de triples <dominio, objeto, derechos>.
- Cada columna = **LISTA DE CONTROL DE ACCESO** por un objeto  
Define quien puede realizar que operación.

Dominio 1 = Read, Write

Dominio 2 = Read

Dominio 3 = Read

⋮

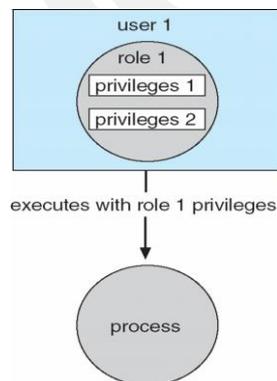
- Cada fila = **LISTA DE CAPACIDADES** (como una clave)  
Para cada dominio que operaciones están permitidas sobre que objetos.
  - Objeto 1 – Read
  - Objeto 4 – Read, Write, Execute
  - Objeto 5 – Read, Write, Delete, Copy

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## CONTROL DE ACCESOS

- La protección puede ser aplicada a recursos físicos
- Solaris 10
- Solaris 10 provee **control de accesos basado en roles (RBAC)** para implementar privilegios
  - Un privilegio es un derecho a ejecutar llamadas a sistema o usar una opción dentro de una llamada a sistema
  - Puede ser asignado a procesos
  - Los roles asignados a usuarios garantizan accesos a privilegios y programas



KMC © 2019

## REVOCACIÓN DE DERECHOS DE ACCESO

- **Lista de Accesos** – Borra derechos de acceso de la lista de accesos
  - Simple
  - Inmediato
- **Lista de Capacidades** – Requiere un esquema para localizar capacidades antes que puedan ser revocadas
  - Readquisición
  - Punteros hacia atrás
  - Indirección
  - Claves

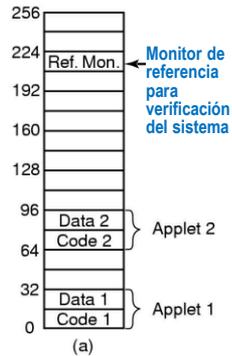
## PROTECCIÓN BASADA EN LENGUAJES

- La especificación de protección en lenguajes de programación permite una descripción en alto nivel de políticas para la alocaación y uso de recursos.
- La implementación del lenguaje puede forzar software para protección cuando la verificación automática soportada por hardware no está disponible.
- Especificación de protección interpretada para generar llamadas donde sea que la protección era llevada a cabo por el hardware y el SO.

## CÓDIGO MÓVIL - CAJAS DE ARENA

Dirección

virtual en MB



(a)

```
MOV R1, S1
SHR #24, S1
CMP S1, S2
TRAPNE
JMP (R1)
```

(b)

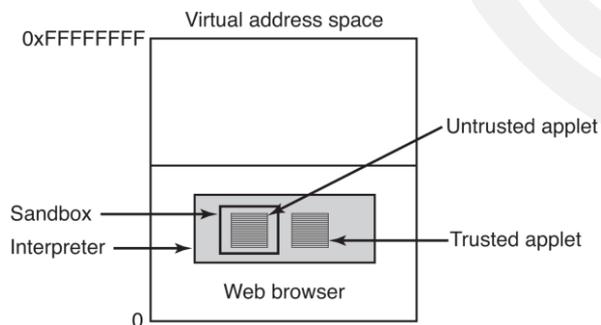
(a) Memoria dividida en cajas de arena de 1-MB

(b) Una forma de verificar la validez de una instrucción

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## CÓDIGO MÓVIL



Los applets pueden ser interpretados por el browser de Web

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## SEGURIDAD: CONTENIDO

- Discutir amenazas y ataques a la seguridad.
- Explicar los fundamentos de la encriptación, autenticación, y hashing.
- Examinar los usos de la criptografía en computación.
- Describir varias contramedidas a ataques a la seguridad.

## EL PROBLEMA DE SEGURIDAD

- La seguridad debe considerar el ambiente externo del sistema y proteger los recursos del sistema
- Los intrusos (crackers) intentan romper la seguridad
- **Una amenaza** es potencialmente una violación a la seguridad
- **Un Ataque** es un intento de romper la seguridad
- Un ataque puede ser accidental o malicioso
- Es más fácil proteger contra un uso accidental que contra uno malicioso

Meta	Amenaza
Confidencialidad	Revelación de los datos
Integridad	Corrupción de los datos
Disponibilidad	Denegación de servicio

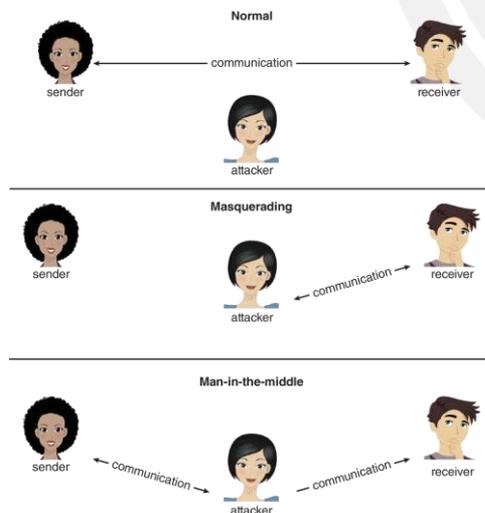
## VIOLACIONES DE SEGURIDAD

- Categorías
  - Fallo de confidencialidad
  - Fallo de integridad
  - Fallo de disponibilidad
  - Robo de servicio
  - Negación de Servicio (Denial of service)
- Métodos
  - Mascarada (brecha de autenticación)
  - Ataque Replay
    - Modificación de Mensajes
  - Ataque Hombre-en-el-Medio
  - Sesión de toma de control

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## ATAQUES COMUNES A LA SEGURIDAD



KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## NIVELES DE MEDIDAS DE SEGURIDAD

- La seguridad debe existir en cuatro niveles para ser efectiva:
  - Física
  - Sistema Operativo
  - Red
  - Aplicaciones
- La seguridad es tan débil como el eslabón más débil de la cadena
- Humana
  - Evite *ingeniería social, phishing, dumpster diving*

## MODELO DE SEGURIDAD DE CUATRO CAPAS



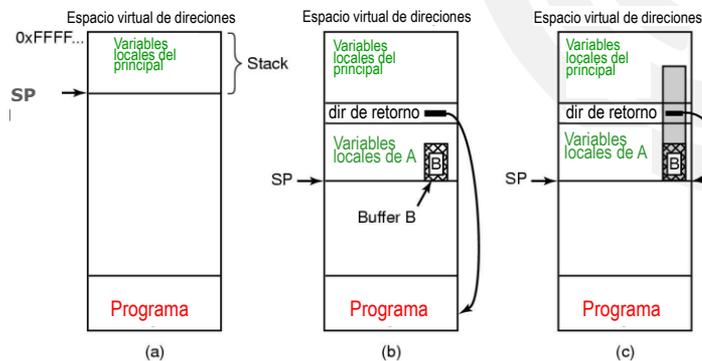
## PROGRAMAS PELIGROSOS

- **Caballo de Troya**
  - Segmento de código que se usa dentro de su ambiente
  - Explota mecanismos que permiten programas escritos por usuarios ser ejecutados por otros usuarios
  - [Spyware](#), [pop-up de ventanas en navegadores](#), [canales encubiertos](#)
- **Puerta Trampa**
  - Identificador de usuario específico y contraseña que saltea los procedimientos de seguridad normales
  - Pueden ser incluidas en un compilador
- **Bomba Lógica**
  - Programa que inicia un incidente bajo ciertas circunstancias
- **Rebalse de Stack y Buffer**
  - Explota un “bug” en un programa (rebalse en el stack o buffers de memoria)

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## REBALSE DE BUFFER



- Situación cuando el programa principal esta corriendo
- Luego del llamado al programa A
- El rebalse de buffer mostrado en gris

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

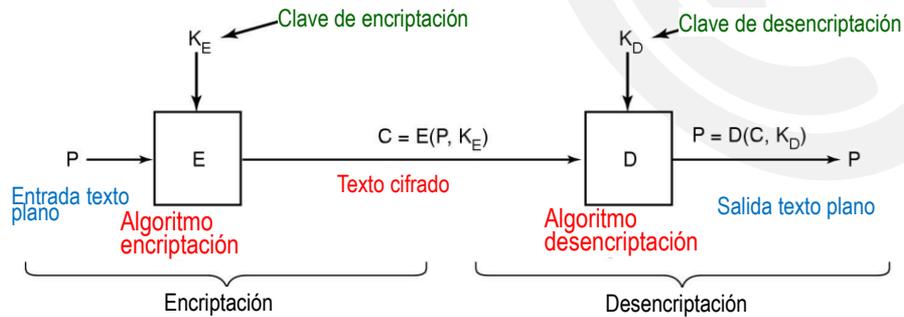
## AMENAZAS AL SISTEMA Y RED

- **Gusanos (Worms)** – usa mecanismos de [spawn](#); es un programa standalone
- **Worm Internet**
  - Explota características de red de UNIX (acceso remoto) y bugs en los programas *finger* y *sendmail*
  - Programa [Grappling hook](#) levanta el programa principal del gusano
- **Barrido de Pórticos**
  - Intento automatizado de conectar un rango de pórticos con una o un rango de direcciones IP
- **Negación de Servicio**
  - Sobrecarga la computadora blanco evitando que haga algún trabajo útil
  - Negación de servicios distribuido (Distributed denial-of-service (DDOS)) proviene de múltiples sitios a la vez

## CRIPTOGRAFÍA COMO HERRAMIENTA DE SEGURIDAD

- Herramienta de seguridad ampliamente disponible
  - La fuente y el destino de los mensajes no puede ser confiable sin la criptografía
  - Medio para limitar potenciales emisores (*sources*) y/o receptores (*destinations*) de los *mensajes*
- Basada en el secreto ([keys](#))

## BASES DE LA CRIPTOGRAFÍA



Relación entre el texto plano y el texto cifrado

## CRIPTOGRAFÍA CON CLAVE SECRETA

- Sustitución Monoalfabética
  - ▶ cada letra es reemplazada por otra letra diferente
- Clave de encriptación dada,
  - ▶ fácil de obtener la clave de desencriptación
- Clave criptográfica secreta llamada clave criptográfica simétrica

## CRIPTOGRAFÍA CON CLAVE PÚBLICA

- Todos los usuarios toman un par de claves: una clave pública y una clave privada
  - ▶ publica la clave pública
  - ▶ no publica la privada
- La clave pública es la clave de encriptación (depende.....)
  - ▶ La clave privada es la clave de desencriptación

## AUTENTICACIÓN DE USUARIO

- Es crucial para identificar correctamente al usuario, dado que el sistema de protección depende del user ID
- La identidad del usuario es frecuentemente establecida por contraseñas, pueden ser consideradas casos especiales de claves o capacidades
  - También puede incluirse algo útil y/o algún atributo del usuario
- Las contraseñas deben permanecer secretas
  - Cambios frecuentes de contraseñas
  - Uso de contraseñas no adivinables
  - Registro de todos los intentos de acceso inválidos
- Las contraseñas pueden ser encriptadas o permitir que se usen una sola vez

## AUTENTICACIÓN USANDO CONTRASEÑAS

Bobbie, 4238, e(Dog4238)
Tony, 2918, e(6%%TaeFF2918)
Laura, 6902, e(Shakespeare6902)
Mark, 1694, e(XaB@Bwcz1694)
Deborah, 1092, e(LordByron,1092)

Salt

Contraseña

El uso del **salt** para derrotar la precomputación de las contraseñas encriptadas.

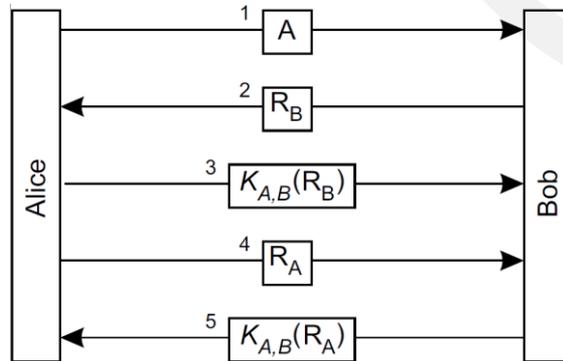
## AUTENTICACIÓN EMISORES

### Componentes del Algoritmo

- Un conjunto  $K$  de claves
- Un conjunto  $M$  de mensajes
- Un conjunto  $A$  de autenticadores
- Una función  $S : K \rightarrow (M \rightarrow A)$ 
  - Donde, para cada  $k \in K$ ,  $S(k)$  es una función para generar autenticadores desde los mensajes
  - $S$  y  $S(k)$  para cualquier  $k$  deben ser funciones eficientemente computables
- Una función  $V : K \rightarrow (M \times A \rightarrow \{\text{true}, \text{false}\})$ . Donde, para cada  $k \in K$ ,  $V(k)$  es una función de verificación de autenticadores en mensajes
  - $V$  y  $V(k)$  para cualquier  $k$  deben ser funciones eficientemente computables

## AUTENTICACIÓN EMISORES BASADA EN CLAVE SECRETA COMPARTIDA

- CINCO MENSAJES PARA EL PROTOCOLO

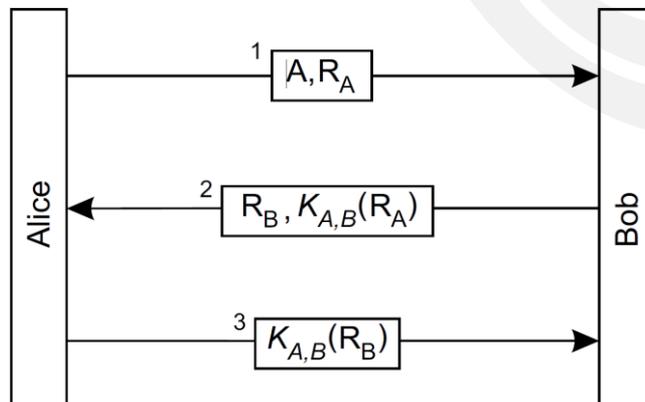


KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## AUTENTICACIÓN EMISORES BASADA EN CLAVE SECRETA COMPARTIDA

- TRES MENSAJES PARA EL PROTOCOLO



KMC © 2019

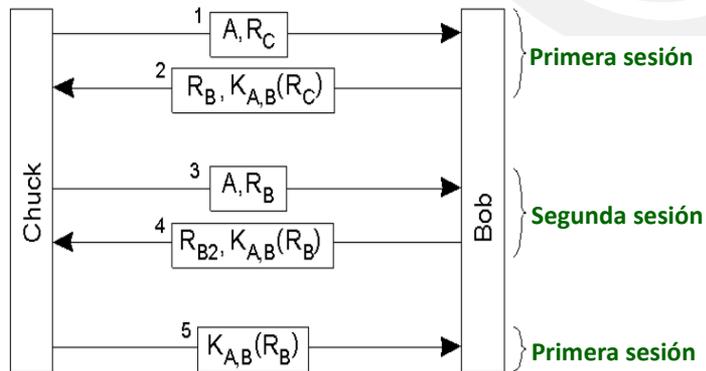
SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

# AUTENTICACIÓN EMISORES BASADA EN CLAVE SECRETA COMPARTIDA

## PROTOCOLO DE TRES MENSAJES



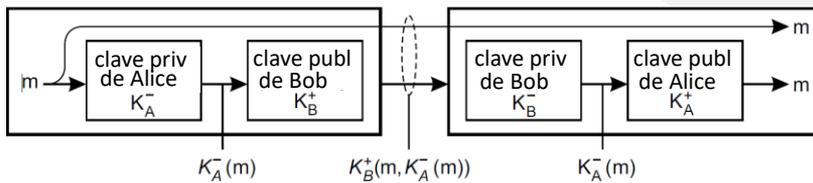
PROBLEMA: ataque por reflejo.



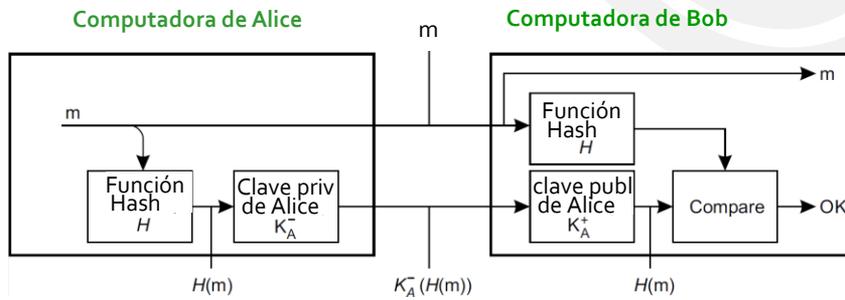
# FIRMA DIGITAL – CRIPTOGRAFÍA CON CLAVE PÚBLICA

Computadora de Alice

Computadora de Bob



## FIRMA DIGITAL – UTILIZACIÓN DE DIGESTO



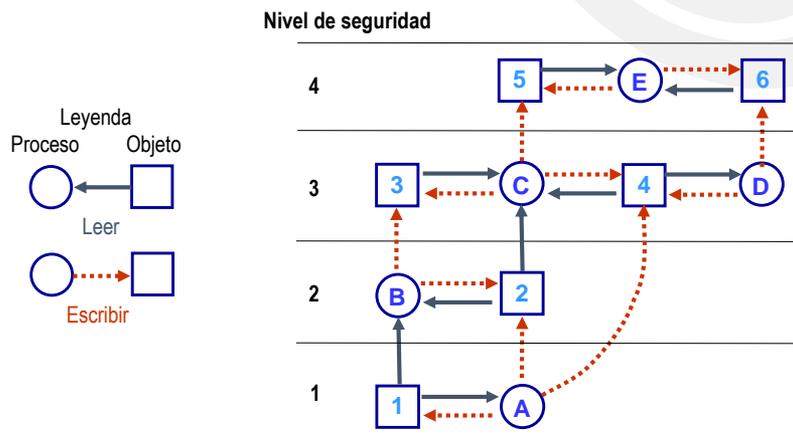
KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## SEGURIDAD MULTINIVEL – BELL-LA PADULA

### Modelo de Confidencialidad

Un proceso puede leer para abajo y escribir para arriba



KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## SEGURIDAD MULTINIVEL

### El Modelo Biba

Principios para garantizar la integridad de los datos

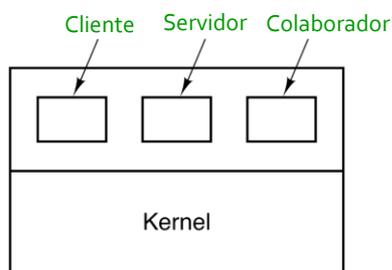
- Principio simple de integridad

El proceso puede escribir solamente objetos en su nivel de seguridad o inferior

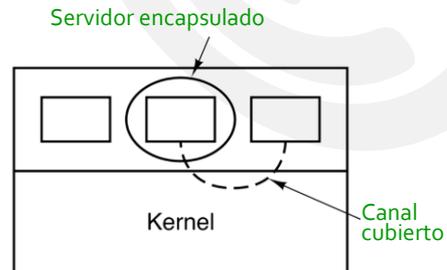
- La propiedad de integridad

El proceso puede leer solamente objetos en su nivel de seguridad o más alto

## CANALES ENCUBIERTOS



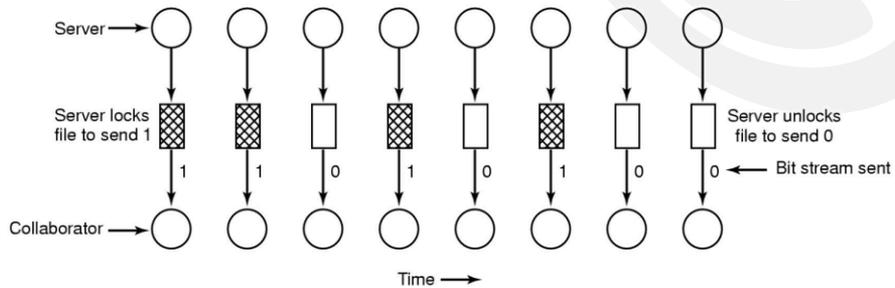
Procesos cliente, servidor y colaborador



El servidor encapsulado puede aún fugar datos a un colaborador via canales cubiertos

## CANALES ENCUBIERTOS

Un canal cubierto usando bloqueo de archivos (locking)



KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## CANALES CUBIERTOS

- Los cuadros parecen los mismos
- El cuadro de la derecha tiene el texto de 5 piezas de Shakespeare
  - encriptadas, insertadas en los bits de bajo orden de los valores de color



Zebras



Hamlet, Macbeth, Julius Caesar  
Merchant of Venice, King Lear

KMC © 2019

SISTEMAS OPERATIVOS – PROTECCIÓN Y SEGURIDAD

## ESTEGANOGRAFÍA

Esta demostración puede encontrarse en:

[www.cs.vu.nl/~ast/](http://www.cs.vu.nl/~ast/)

Haga click sobre el encabezamiento STEGANOGRAPHY DEMO luego siga las instrucciones en la página para descargar la imagen y las herramientas de esteganografía necesarias para extraer las piezas.

### Bibliografía:

- Silberschatz, A., Gagne G., y Galvin, P.B.; "*Operating System Concepts*", 7<sup>ma</sup> Edición 2009, 9<sup>na</sup> Edición 2012, 10<sup>ma</sup> Edición 2018.
- Tanenbaum, A.; "*Modern Operating Systems*", Addison-Wesley, 3<sup>ra</sup>. Edición 2008, 4<sup>ta</sup>. Edición 2014.