**Examen Enero Erasmus 2018**

**1.Explica firma digital, como comprueba validez, aspectos que están protegidos. (1,5 puntos)**

### Escenario 4: firma digital con resumen seguro.

Alice quiere publicar un documento M de forma que cualquiera pueda verificar su procedencia.

1. Alice calcula un resumen de longitud fija del documento Resumen(M).
2. Alice encripta el resumen con su clave privada, lo adjunta a M y hace el resultado (M, {Resumen(M)}KApriv) público.
3. Bob obtiene el documento firmado, extrae M y computa Resumen(M).
4. Bob usa la clave pública de Alice para desencriptar {Resumen(M)} KApriv y lo compara con el resumen calculado por él. Si coincide, entonces la firma es válida.

La función de resumen debe ser segura frente al “ataque del cumpleaños”.

### Función de resumen seguro h = H(M):

1. Dado M, debe ser fácil calcular h.
2. Dado h, debe ser muy difícil calcular M.
3. Dado M, deber ser difícil encontrar otro M’, tal que H(M) = H(M’).

También llamada función de dispersión de un solo sentido.

**2.Needham-Schroeder, Kerberos. (2 puntos)**

## PROTOCOLO NEEDHAM-SCHROEDER

En los primeros sistemas distribuidos (1974-84) era difícil proteger los servidores. No había mecanismos de autenticación del origen de las peticiones y la criptografía pública no estaba disponible.

Needham y Schroeder desarrollaron un protocolo de autenticación y distribución de claves para uso en red local:

* Supuso un primer ejemplo del cuidado en el diseño de protocolos de seguridad.
* Introdujeron varias ideas de diseño.

La autenticación de clave secreta sigue el siguiente proceso:

1. A solicita una clave a S para comunicarse con B.
2. S devuelve un mensaje encriptado en la clave secreta de A, con una clave nueva KAB y un “ticket” encriptado en la clave secreta de B. La ocasión NA demuestra que el mensaje fue enviado en respuesta al anterior. A confía en que S envió el mensaje porque sólo S conoce la clave secreta de A.
3. A envía el ticket a B.
4. B desencripta el ticket y utiliza la nueva clave KAB para encriptar otra ocasión NB.
5. A demuestra a B que fue el emisor del mensaje anterior devolviendo una transformación acordada sobre NB.

NA es una ocasión: entero que se añaden a los mensajes para demostrar la frescura de la transacción. Son generador por el proceso emisor cuando se necesita.

## KERBEROS

Comunicación segura con servidores en una red local. Basado en Needham-Shroeder,

estandarizado e incluido en muchos sistemas operativos. El servidor Kerberos crea una clave secreta compartida para cada servidor solicitado y la envía encriptada al computador del usuario. El password del usuario es el secreto compartido inicial en Kerberos.

**3.Exigencias exclusión mutua. Prueba y justifica si los algoritmos (Servidor central, anillo, Ricart y Agrawala) las cumplen.**

EM1: Seguridad en todo momento, como máximo hay un solo proceso ejecutando la región crítica

EM2: Vitalidad a todo proceso que lo solicita se le concede la entrada/salida en la región crítica en algún momento:evita el abrazo mortal (deadlock) e inanición (starvation)

EM3: Ordenación la entrada en la región crítica debe concederse según la relación sucedió - antes

### **Algoritmo basado en servidor central**

El servidor central concede permisos en forma de testigo que concede acceso a la sección crítica. Al salir de dicha sección, el proceso devuelve el testigo.

Suponiendo que no hay caídas ni pérdidas de mensajes:

* Se cumplen EM1 y EM2.
* EM3 está asegurada en el orden de llegada de los mensajes al servidor.

Se necesitarían 2 mensajes para entrar a la sección crítica y 1 mensaje para salir de ella.

Este algoritmo puede producir cuello de botella en el servidor, caídas en el mismo, lo cual nos llevaría a elegir un nuevo servidor y no aseguramos la EM3 o una caída o fallo del proceso en la sección crítica.

### **Algoritmo basado en anillo**

La exclusión se logra por la obtención de un testigo.

Anillo lógico: se creada dando a cada proceso la dirección de su vecino. El testigo siempre circula por el anillo. Cuando un proceso lo recibe:

* Si quiere entrar a SC lo retiene.
* Si no quiere entrar a SC lo envía a su vecino.
* Al salir de SC lo envía a su vecino.

Con este algoritmo se verifican EM1 y EM2 pero no aseguramos EM3. La obtención del recurso requiere entre 1 y (n-1) mensajes.

Se presentan algunos problemas:

* Carga de la red aunque ningún proceso quiera entrar a SC.
* Si un proceso cae necesita reconfiguración y si además tenía el testigo se debe regenerar el mismo.
* Para asegurarnos de que un proceso ha caído, es necesario varios testigos.
* Desconexión o ruptura de la red.

**4.Protocolos de actualización de réplicas de ficheros.**

Replicación de solo lectura: se aplica a ficheros inmutables

Escribir en todos-Leer de cualquiera: las escrituras son costosas, problemas si un servidor cae

Protocolo basado en la disponibilidad de copias: problemas de inconsistencias en caso de partición de la red

Protocolo de copia primaria: se lee de cualquiera, se escribe en una problemas si cae el servidor primario

Protocolo basado en quorum: los clientes requieren un quorum de Nr servidores para leer y Nw para escribir, Nr + Nw > N (N = número de réplicas)

**Examen enero 2016**

**2- Analogias y diferencias entre los algoritmos de sincronización de relojes Cristian y NTP. Razona la respuesta, explica los aspectos clave de cada uno y contextualiza los escenarios de caso de uso. [1,5 puntos]**

**Método de Cristian (para sincronizar relojes externamente):**

Un servidor de tiempo S recibe señales UTC.

* El proceso p solicita el tiempo en un mensaje m y recibe t en Mt de S.
* P establece su tiempo a t + Tround/2 (Tround es el tiempo de ida y vuelta).
* Precisión: más o menos será (Tround/2 - min). El min será el mínimo estimado de transmisión.
  + El momento más temprano en que S pone t en Mt es min después de que P enviara Mr.
  + El momento más tardío es min antes de que Mt llegue a p.
  + El tiempo de S cuando Mt llega está en el rango (t + min, t + Tround - min).

## Protocolo de tiempo de red (NTP)

Se trata de un servicio de tiempo para internet. Sincroniza a los clientes con UTC.

Servicio fiable, redundante, reconfigurables si alguno cae, escalables, con autenticación de las fuentes de tiempo.(se puede pensar como una forma de árbol).

* Los servidores primarios están conectados a fuentes UTC.
* Los servidores secundarios sincronizados a los primarios.
* [Subred de sincronización] y en el nivel más bajo de servidores están los PC.

### Sincronización de servidores en NTP.

La subred de sincronización se puede reconfigurar si se produce un fallo:

* un primario que pierde su conexión con UTC puede pasar a secundario.
* Un secundario que pierde a su primario puede seleccionar a otro primario.

Los modos de sincronización son:

* Multidifusión (multicast)
  + En LAN de alta velocidad. Un servidor reparte el tiempo al resto que establecen su tiempo asumiendo un retraso de transmisión.
* Llamada a procedimiento.
  + Similar al de Cristian. El servidor acepta peticiones. Precisión más alta.
* Simétrica.
  + Pares de servidores se intercambian mensajes conteniendo información de tiempo.
  + Usado en los casos en que se necesita muy alta precisión (primeros niveles).

### Intercambio de mensajes entre pares de servidores NTP.

Todos los modos usan UDP. Cada mensaje lleva marcas de tiempo de los eventos recientes (del evento anterior y del tiempo local de envío del mensaje actual).

El receptor anota el tiempo local cuando recibe T.

El protocolo UDP carece de seguridad, y puede haber retraso entre la llegada de un mensaje y el envío del siguiente, o pérdida de mensajes.

### Precisión de NTP.

Para cada par de mensajes entre servidores, NTP estima una compensación entre los dos relojes (deriva) y un retardo d (medida de precisión. Tiempo total de transmisión para los dos mensajes t y t’).

Los servidores NTP mantienen pares del tipo <O , d> estimando la fiabilidad de las variaciones y permitiendo cambiar el propio par.

Los algoritmos como Cristian o NTP sincronizan los relojes a pesar de sus derivas y el retardo de los mensajes.

**3- Explica las tres razones principales por las que proteger una red WIFI y las medidas que debemos aplicar para evitar su ataque o reducir el peligro. [1,5 puntos]**

1. El tráfico de red puede ser capturado y examinado

2. Los recursos de red están expuestos a usuarios desconocidos directamente por la vulnerabilidad del canal de transmisión

3. Uso de la conexión para asuntos ilegales o para delinquir

Contramedidas:

- Cambiar las opciones por defecto de routers y webs de configuración (no usando información personal en el SSID)

- Actualizar firmware y hardware  objetivo WPA o WPA2

- Apagar el AP cuando no se usa (o con algún timer)

- Filtrado de MAC y número de clientes simultáneos

- Bajar al mínimo útil la potencia de transmisión de AP

- Encriptación WPA o WPA2 (con claves largas, no en diccionario y cambio periódico)

- Encriptar los volúmenes – particiones y ficheros del sistema

- Incorporar siempre antivirus, firewalls de dos direcciones (ver COMODO  Windows; iceFloor  Mac) y software anti-intrusión

**4- Define los siguientes conceptos [0,5 puntos cada uno]**

a) Algoritmo criptográfico simétrico;

b) Ataque de cumpleaños;

c) Corte inconsistente;

d) Sistema distribuido asíncrono;

e) Tiempo UTC;

f) Función de resumen seguro;

g) Reloj de Lamport;

h) Sincronización interior de relojes.

**5- Dentro de las prácticas no guiadas realizadas este curso, de sockets, RMI y servicios WEB, explica esquemáticamente (con un gráfico y breve explicación del funcionamiento) la comunicación e interacción del controlador con los servicios web y componentes RMI [1 punto] ¿Qué ventajas e inconvenientes relacionados con los conceptos de SD has encontrado en el uso de cada una de las dos propuestas? [0,5 puntos]**

**Examen enero 2012**

**1.- Modos de Sincronización y algoritmos**

No existe un reloj universal, pero podemos hacer algunas aproximaciones con relojes lógicos, estados globales o relojes vectoriales.

Para la sincronización de relojes, se puede realizar mediante el tiempo universal coordinado (UTC). Es un estándar para la comprobación y sincronización del tiempo, que se difunde por radio en tierra y mediante satélites. Se basa en el tiempo atómico y es calibrado eventualmente con el tiempo astronómico. Dos tipos de sincronización:

- Externa: un reloj se conecta a una fuente UTC exacta.

- Interna: dos relojes se envían mensajes para su sincronización. Dos relojes que están sincronizados internamente no significa que lo estén externamente, puesto que pueden derivar juntos.

Podemos decir que un SD es síncrono cuando:

- Se conoce tiempo máximo y mínimo de ejecución de cada paso en cada proceso.

- Se conoce tiempo máximo y mínimo para la recepción de mensajes.

- Se conocen los tiempos de deriva de los relojes implicados.

Los algoritmos de sincronización serían:

**Primera aproximación (sincronización interna):** un proceso p le envía un mensaje m a p2 con tiempo t. P2 actualizaría su reloj con tiempo t2 a t + Ttransmisión del mensaje.

**Método de Cristian (sincronización externa):** con conexión a servidor UTC S. Un proceso p solicita a S una sincronización mediante un mensaje m. S contesta y p pondrá su tiempo a t + Tround/2 (Tround es el tiempo de ida y vuelta). La precisión será de Tround/2 – min.

**Algoritmo de Berkeley (sincronización interna):** un computador actúa de maestro, y recoge los datos de reloj de todos los computadores del sistema (esclavos). Asumiendo los tiempos de ida y vuelta, realiza el promedio de los tiempos recibidos incluyéndose, enviando el tiempo calculado al resto de computadores. Si el maestro falla podremos elegir otro.

Otra forma de sincronización sería NTP (protocolo de tiempo de red). Estándar para el establecimiento del tiempo para internet. Se puede estudiar como un tipo árbol, donde los niveles primarios estarían conectados a fuentes UTC, los secundarios a los niveles primarios y la subred de sincronización sería el nivel más bajo, siendo los PC. Es tolerante a fallos. Si un primario pierde la conexión con UTC pasa a secundario, si un secundario pierde a su primario puede escoger otro. Entre pares de servidores NTP se utiliza la sincronización por mensajes en modo UDP. Los métodos de sincronización son:

- Multicast. En redes LAN de alta velocidad. Un servidor reparte el tiempo a los demás.

- Por llamada a procedimiento. Parecido al método de Cristian pero más preciso.

- Simétrica. Para niveles superiores, con alta precisión.

Otros métodos de sincronización serían Lamport (mediante sincronización por eventos y no por relojes) y los relojes lógicos; relojes vectoriales; estados globales, con el algoritmo de Chandy y Lamport.

**2.- Explicar por que el escenario 2 ("el del servidor") no es apto para el negocio electrónico/transacciones en un negocio electrónico**

En el caso del escenario 2, el servidor Sara es quien recoge todos los datos de credenciales de todos los implicados en la comunicación. Esto es bueno cuando hay un número de participantes concretos, pero no para un tipo de comercio en creciente evolución, como es el comercio electrónico. Debido a las constantes altas y posibles bajas de usuarios en el servidor, sería fácil una posible caída del mismo, o incluso una sobrecarga en las peticiones al mismo, por lo que este modelo sería no apto para este tipo de comercio.

**4.- ¿El algoritmo TEA se podría lleva a otra arquitectura hardware?**

El algoritmo TEA utiliza operaciones que dependen de la arquitectura como es el desplazamiento lógico. No será lo mismo en 32 que en 64 bits. Si sigues desarrollándolo por ahí, como comenta Lady Sun, sacas más problemas de arquitectura.  
**5.- Definir entre los algoritmos simétricos y asimétricos criptográficos cuáles son mejor o peor**

En los algoritmos criptográficos podemos encontrar:

- Algoritmos simétricos (con clave secreta). La forma usual de ataque es la fuerza bruta.

- Algoritmos asimétricos (con clave pública). Se basa en el uso de funciones de puerta falsa.

Simétricos:

- TEA: triple de veloz que el DES. Muy efectivo a pesar de su simplicidad.

- DES: su modelo original no era muy veloz. Debido a su carga de código, se implementó en VLSI.

- Triple-DES: ejecuta DES tres veces con 2 claves distintas.

- IDEA: muy parecido al TEA pero no tan veloz.

- AES.

Asimétricos:

- RSA: es el más común y el más utilizado. Para encriptar con RSA el texto es dividido en bloques.

- Curvas elípticas: nuevo modelo. Mucho más eficiente. Claves más cortas y más veloz.

Los algoritmos asimétricos son 1000 veces más lentos y no son prácticos para encriptaciones masivas. Sin embargo, sus propiedades los hacen idóneos para distribución de claves y para autenticación. **6.- Definir los pasos de Chandy- Lamport, cuando se acaba y finalidad**

El algoritmo de Chandy y Lamport se usa para determinar los estados globales de sistemas distribuidos. Registra un conjunto de estados de procesos de forma que el estado global sea consistente. Se registra el estado de cada proceso localmente.

Los pasos a seguir son:

- Recepción de marcador para el proceso P.

Si (p no ha registrado su estado). Lo registra. Registra el estado de C como el conjunto vacío. Activa el registro de mensajes.

Si (p ha registrado su estado). P registra el estado de C como el conjunto de mensajes recibidos desde que C guardó su estado.

- Envío de marcador para el proceso P.

Después de que P haya registrado su estado para cada canal de salida C, P envía un mensaje de marcador sobre C.

El algoritmo termina si se cumplen todas las restricciones de conectividad e inexistencia de fallo en la comunicación.

**Otro examen**

**2- Desarrolla el ataque del cumpleaños indicando en que se basa, en que contexto se puede dar y qué aspectos son determinantes para protegernos de ese ataque.**

El ataque del cumpleaños puede darse en el escenario 4: firma digital con resumen seguro. Se puede dar en el caso de que la función resumen no sea segura.

El ataque del cumpleaños se basa en que es más probable encontrar un par idéntico entre un conjunto que la probabilidad de encontrar una pareja para un individuo dado.

El ataque del cumpleaños se produce cuando.

1. Alice prepara 2 versiones M y M’ de un contrato para Bob.

2. Alice fabrica varias versiones sutilmente diferentes de M y M’.

3. Alice envía M a Bob, quien lo firma digitalmente usando su clave privada.

4. Cuando lo devuelve, Alice sustituye M por M’, pero manteniendo la firma de Bob sobre M.

Para protegernos de esta paradoja, es necesario tener funciones de resumen seguras y funcionar con firmas digitales conocidas, de forma que sean todavía más seguras. Si encontramos firmas desconocidas en el proceso, es posible que nuestro resumen o mensaje haya sido interceptado.  
  
**3-Desarrolla el algoritmo de acceso a sección crítica llamado "Ricart-Agrawala"  
¿se cumplen en todos los casos las tres exigencias de exclusión mutua? Razona la respuesta. Además indica y justifica el número de mensajes que se necesitan en su funcionamiento.**

Dicho algoritmo es un algoritmo basado en relojes lógicos. Se trata de que cuando un proceso quiera entrar en la sección crítica compartida, preguntará a todos los demás si puede hacerlo. Cuando obtenga respuesta de TODOS, podrá entrar. La comunicación se realiza mediante mensajes en forma de tuplas.

El número de mensajes necesario para su funcionamiento depende de las infraestructuras:

- Si no se permite envío multicast se necesitará 2(n-1) mensajes.

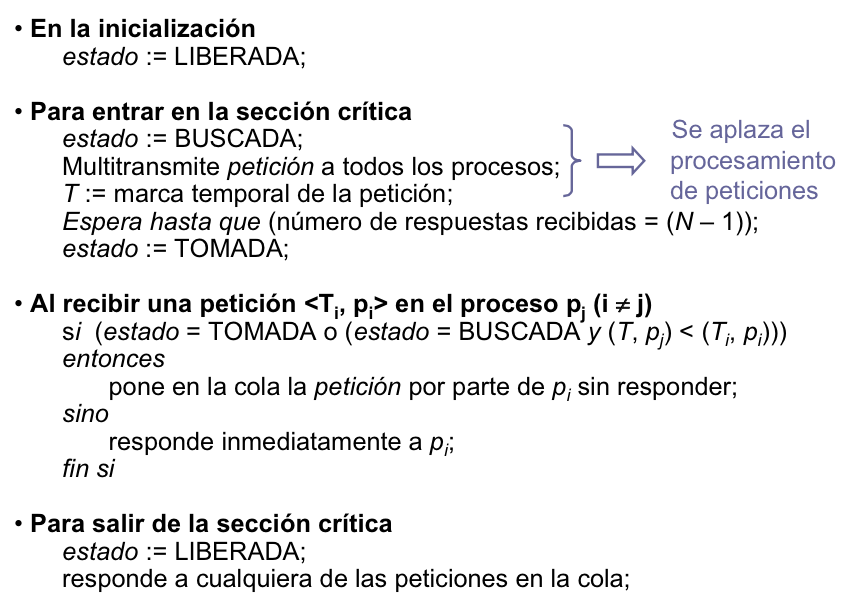
- Si se permite envío multicast serán necesarios n mensajes.

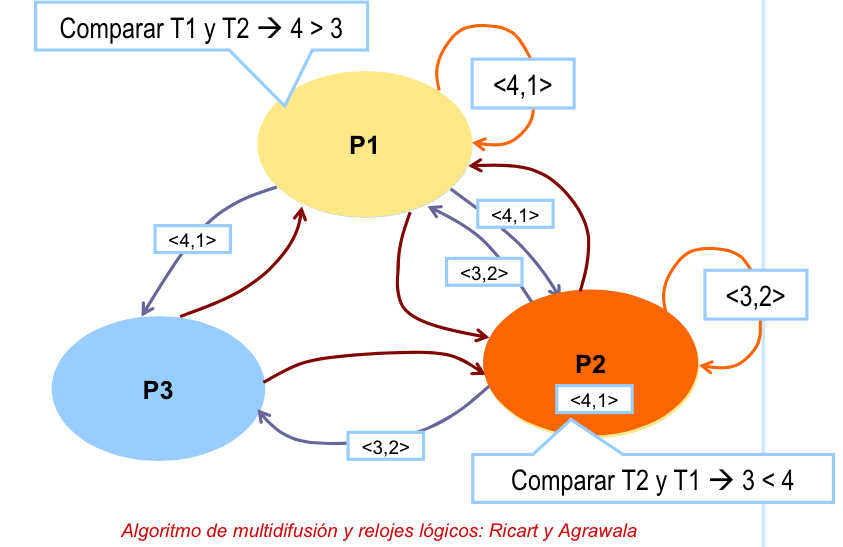
- El algoritmo fue refinado para usar n mensajes en envío no multicast.

Se cumplirían con este algoritmo las exigencias EM1 y EM2 (seguridad y vitalidad), pero no aseguraríamos EM3 (ordenación), debido a que este método puede hacer que el servidor sufra caídas. Además dicho método tendría igual o peor congestión en el servidor que el método de servidor central, y sería más costoso.

La idea básica es que cuando un proceso quiere entrar en la SC les pregunta a los demás si puede entrar. Cuando todos los demás le contesten entra.

El acceso se realiza mediante testigo, de manera que cada proceso guarda el estado en relación a la SC: liberada, buscada o tomada. Los mensajes son tuplas: Tupla<Ti,Pi,SCi>.





El número de mensajes necesarios para obtener el recurso varía según las características del servidor:

* Sin soporte multicast: 2(n-1).
* Con soporte multicast: n.
* El algoritmo fue refinado hasta n mensajes sin soporte multicast por Raynal en 1988.

Los problemas que presenta el algoritmo son:

* Más costoso que el del servidor central.
* El fallo de cualquier proceso bloquea el sistema.
* Los procesos implicados reciben y procesan cada solicitud: igual o peor congestión que el servidor central.

En resumen, ninguno de los algoritmos vistos puede tratar el problema de caídas. En el algoritmo de servidor es el que tiene menor número de mensajes, pero supone cuello de botella.

En conclusión, es preferible que el servidor que gestiona el recurso implemente también la exclusión mutua.

**4. Justifica la existencia o no de un reloj universal de referencia. Razona también la respuesta en el contexto de los SD.**

No existe un reloj universal de referencia. El tiempo es relativo. En relación con los SD, el tiempo es una de las problemáticas más usuales, ya que no existe dicho reloj. Podríamos realizar algunas aproximaciones que resultarían muy precisas, utilizando relojes lógicos, relojes vectoriales o algoritmos para determinar el estado global de SD en cada momento.

Cada computador que conforma el SD tendría un reloj local, que sería el utilizado por cada uno de los procesos que se ejecutasen en dicho computador. De manera que aunque nosotros sincronizáramos todos los relojes al mismo tiempo, dicho reloj global variaría significativamente con el tiempo.

Es cierto que hay aproximaciones muy válidas para marcar los eventos en cada computador, como por ejemplo utilizando estándares de sincronización como NTP (protocolo de tiempo de red) o ajustándonos al UTC( coordinación de tiempo universal), relojes con gran precisión y baja tasa de deriva.

Por la diferencia existente entre los relojes de un SD tenemos dos términos:

- Sesgo: diferencia de tiempo entre dos relojes en un instante determinado.

- Tasa de deriva: diferencia por unidad de tiempo que difiere el reloj de cada computador del reloj perfecto.

**5. Desarrolla el concepto de reloj vectorial y compáralo con el reloj lógico de Lamport, indicando las características principales, reglas de fijación de marcas temporales y álgebras de comparación del orden de sucesión de los eventos en cada caso.**

Los relojes lógicos de Lamport son contadores software monocrecientes. No debemos confundirlos nunca con los relojes físicos. Todos los procesos de un sistema tienen un reloj lógico.

Mattern y Fidge crean los relojes vectoriales para arreglar las deficiencias de los relojes lógicos de Lamport. Un reloj vectorial es un array de N enteros que utilizan los procesos para llevar a cabo su ejecución. Cada uno de los procesos utiliza este reloj para establecer marcas de sus eventos locales.

**7. Explica en qué consisten los cifradores de bloques, tipos y para qué se utilizan. Pon un ejemplo de cada uno, funcionamiento y debilidades**.

La mayoría de cifradores de bloques se basan en bloques de 64 bits. La debilidad de un cifrador simple es que los patrones repetidos pueden ser detectados. La conexión debe ser fiable, no se pueden perder bloques.

Los tipos y sus ejemplos son:

- Simétricos. El TEA es el más rápido a pesar de ser muy simple.

- Asimétricos. El RSA es el más utilizado actualmente. Para la encriptación necesita dividir el texto en varios bloques.

- De resumen seguro. SHA. Basado en MD4 pero más seguro.

Los algoritmos asimétricos con 1000 veces más lentos que los simétricos y no son adecuados para trabajar con gran carga. A pesar de ello, son útiles para la distribución de claves y autenticación. En los algoritmos de resumen seguro se pueden producir ataques de cumpleaños, por lo que los algoritmos de resumen deben trabajar sobre firmas digitales conocidas y resúmenes seguros.

**8. Describe las 3 exigencias de los algoritmos de exclusión mutua distribuida. Además razona si se cumplen en anillo y Ricart-Agrawala.**

EM1. Seguridad. Sólo un proceso en ejecución en la zona de sección crítica compartida del sistema distribuido.

EM2. Vitalidad. Si un proceso solicita entrar a la sección crítica, se le debe conceder el permiso en algún momento de su ejecución.

EM3. Ordenación. Los procesos deberán entrar en la sección crítica según el orden parcial de “suceder antes”.

En el algoritmo de anillo, el testigo pasa de computador en computador, conociendo cada uno de éstos solamente la dirección de su vecino. Se cumplirán la EM1 y EM2, ya que si un computador tiene el testigo será el que entre en SC si lo requiere y en algún momento se otorgará permiso para que un proceso que lo ha solicitado entre en la SC. La EM3 no siempre se cumplirá, debido a que al conocer solamente la dirección del vecino no podemos asegurar que las peticiones se atiendan en el orden correcto.

En el algoritmo de Ricart-Agrawala cumpliremos las mismas reglas, ya que al preguntar a todos los demás nos aseguramos que no hay nadie ocupando la SC y además siempre entraremos en la misma si lo hemos pedido. El único problema es que no podemos asegurar la EM3, debido a que si se produce fallo en el servidor o cuello de botella podemos perder el orden.

**9. Indica y justifica el número de mensajes necesarios para la elección de coordinador mediante el algoritmo “Bully”.**

El número de mensajes para elegir coordinador:

- En el caso mejor: se da cuenta el segundo más alto (n-2) mensajes.

- En el caso peor: se da cuenta el más bajo (n2)

**AMPLIACION DE Bully:**

Para este algoritmo de elección, todos los miembros del grupo deben conocer las identidades y direcciones de los demás miembros. Se supone una comunicación fiable.

El algoritmo selecciona al miembro superviviente con mayor identificador. Los procesos pueden caer durante la elección. Hay 3 tipos de mensajes:

* mensaje de elección: para anunciar una elección.
* Mensaje de respuesta a un mensaje de elección.
* Mensaje de coordinador: anuncia identidad de nuevo coordinador.

El número de mensajes para elegir coordinador:

* En el caso mejor: se da cuenta el segundo más alto. (n-2) mensajes.
* En el caso peor: se da cuenta el más bajo. 0(n2).

Un proceso inicia una elección al darse cuenta de que el coordinador ha caído:

* Envía un mensaje de elección a los procesos con identificador mayor que el suyo.
* Espera algún mensaje de respuesta. Si el temporizador vence, el proceso actual se erige como coordinador y envía mensaje de coordinador a todos los procesos con identificadores más bajos.
* Si recibe respuesta, espera mensaje de coordinador. Si vence el temporizador, lanza una nueva elección.

Si un proceso recibe un mensaje de coordinador, guarda el identificador y trata a ese proceso como nuevo coordinador.

Si un proceso recibe un mensaje de elección, contesta con un mensaje de respuesta y lanza una elección.

Cuando un proceso se reinicia, lanza una elección a menos que sea el de identificador más alto, en cuyo caso se convertiría en el nuevo coordinador.

**DEFINICIONES**

**Algoritmo criptográfico asimétrico.++:**  también llamada **criptografía de clave pública** es el método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona que ha enviado el mensaje. Una clave es *pública* y se puede entregar a cualquier persona, la otra clave es *privada* y el propietario debe guardarla de modo que nadie tenga acceso a ella. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

**Algoritmo criptográfico simétrico**:  es un método criptográfico en el cual se usa una misma clave para cifrar y descifrar mensajes en el emisor y el receptor. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez que ambas partes tienen acceso a esta clave, el remitente cifra un mensaje usando la clave, lo envía al destinatario, y éste lo descifra con la misma clave.

**Ataque de cumpleaños.++:** Ataca a la función de resumen y se basa en la probabilidad de encontrar un par idéntico en un conjunto es mucho mayor que la de encontrar la pareja para un individuo dado.

**Tasa de Deriva.++**: diferencia por unidad de tiempo que difiere un reloj de un computador del reloj perfecto.

**Función de Resumen Seguro.++:**  son algoritmos que consiguen crear a partir de una entrada (ya sea un texto, una contraseña o un archivo, por ejemplo) una salida alfanumérica de longitud normalmente fija que representa un resumen de toda la información que se le ha dado.

## Sistema distribuido síncrono.++++: un sistema distribuido en general suele ser asíncrono, debido a que es complicado sincronizar todos los relojes. Imposible detectar fallos. Se dice que un SD es síncrono si están definidos los límites siguientes:

Tiempo máximo y mínimo para ejecutar cada paso de un proceso.

Tiempo máximo y mínimo de recepción de un mensaje.

Los límites de deriva de cada reloj local donde se ejecuta cada proceso son conocidos.

**Sistema distribuido asíncrono** es aquel en que no existen limitaciones sobre: La velocidad de procesamiento, los retardos de transmisión de mensaje y las tasas de deriva de reloj.

**Reloj correcto.++:** se dice de aquel reloj H del cual conocemos su tasa de deriva

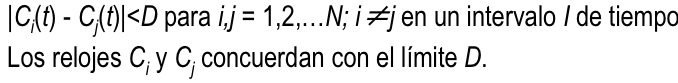
**Reloj de Lamport o lógico.++++:** Un reloj lógico es un contador software monótono creciente. Son contadores que se actualizan de acuerdo con la relación “suceder antes” entre eventos.

**Tiempo UTC.++**: estándar internacional de establecimiento y mantenimiento del tiempo transcurrido. Se basa en el tiempo atómico y ocasionalmente ajustado al tiempo astronómico. La señal se difunde mediante estaciones de radio por tierra y mediante satélites. Las computadoras pueden sincronizar sus relojes mediante receptores adecuados.

**Algoritmo de Berkeley:** Se utiliza para sincronizar relojes internamente.

* + Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos.
  + El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos.
  + Promedia los resultados incluyéndose y eliminando cualquier valor que no sea consistente.
  + Envía la magnitud de ajuste de cada reloj, que puede ser positivo o negativo.

Si el maestro falla, se puede elegir a un nuevo maestro.

**Sincronización interna de relojes++:** cualquier par de computadores están sincronizados si sus relojes cumplen: 

**Corte inconsistente:**

**Corte consistente:** un corte tal que para todos cada evento que contiene, también contiene todos los sucesos que sucedieron antes que él.

**Reloj universal:UTC**

**Firma digital:** consiste en crear un hash de la información enviada y cifrarlo con nuestra clave privada para que cualquiera con nuestra clave pública pueda ver el hash real y verificar que el contenido del archivo es el que hemos mandado nosotros.

**Reloj vectorial:** es un array de N enteros que utilizan los procesos para llevar a cabo su ejecución. Cada uno de los procesos utiliza este reloj para establecer marcas de sus eventos locales. Mejoran los de Lamport, ya que determinan si dos eventos están ordenados por la relación “suceder antes” o son concurrentes, comparando los vectores de marcas.

**Sesgo:** diferencia de tiempo entre dos relojes en un instante determinado.

**Criptografia:** Arte y técnica de escribir con procedimientos o claves secretas o de un modo enigmático, de tal forma que lo escrito solamente sea inteligible para quien sepa descifrarlo.

**Confusion**

**Difusion**

**Corte de la ejecución del sistema**

**Certificado:** sentencia firmada por un principal que sirve de credencial y/o autenticación. Un certificado necesita:

* + Un formato estándar acordado.
  + Acuerdo sobre la forma en que se construyen las cadenas de certificados.
  + Fechas de expiración, de forma que pueda ser revocado.