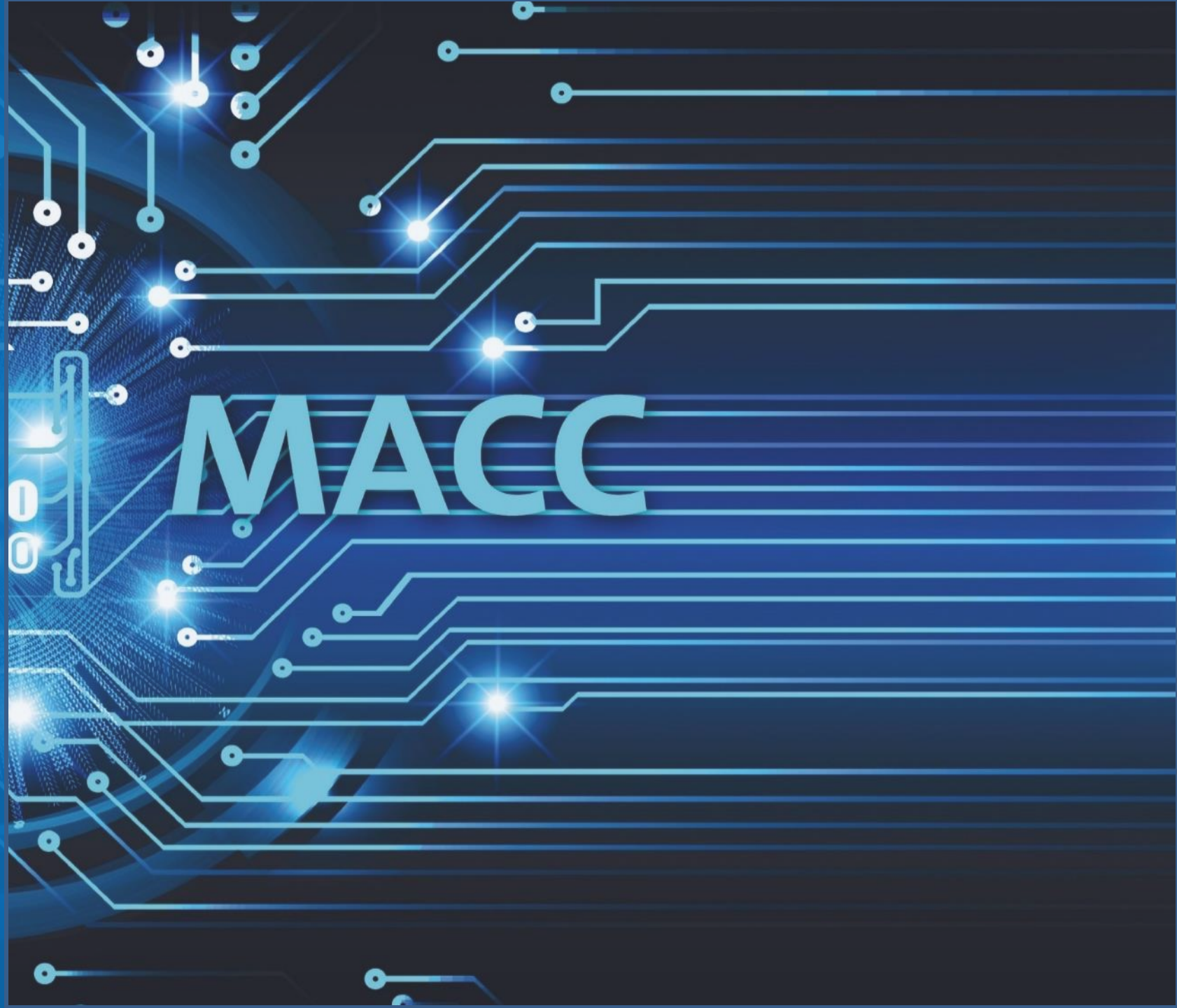
Universidad del Rosario

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Aplicaciones Blockchain II- Propiedades de seguridad

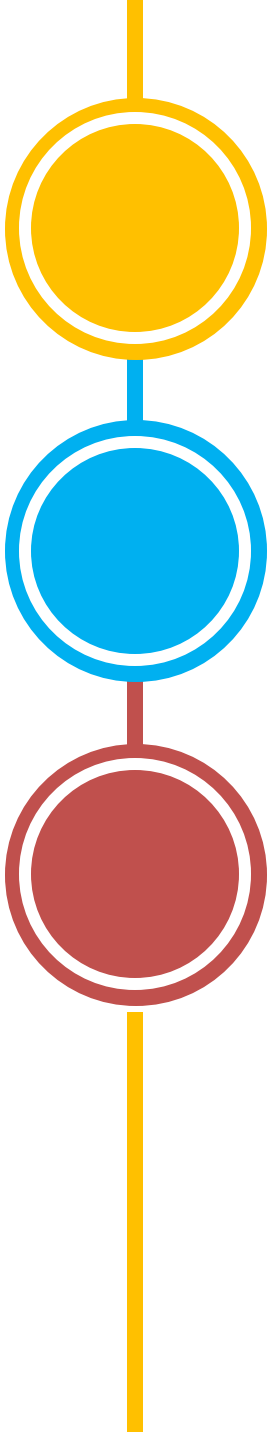


Carlos Castro
Valérie Gauthier
Martín Ochoa

Departamento MACC
Facultad de Economía
Universidad del Rosario

CONTENIDO

Aplicaciones Blockchain



1. Confidencialidad e Integridad

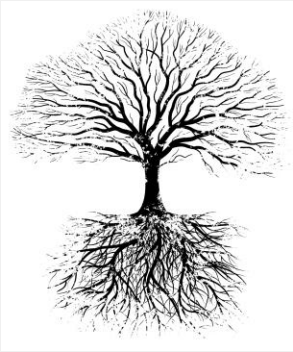
2. Disponibilidad

3. Consenso

1. Qué es ciberseguridad?

Tradicionalmente, las propiedades de ciberseguridad están definidas en terminos de Confidencialidad, Integridad y Disponibilidad de datos y recursos.

- Qué significan exactamente estos terminos?
- Como podemos demostrar que un sistemas es seguro?
- Existen definiciones rigurosas?

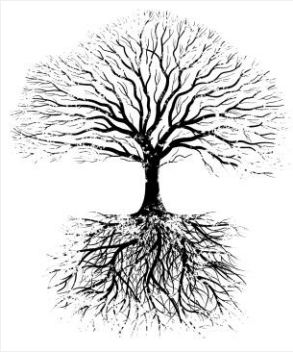


1. Confidencialidad

Definición informal:

"*is the property, that information is not made available or disclosed to unauthorized individuals, entities, or processes*"

(ISO27000).



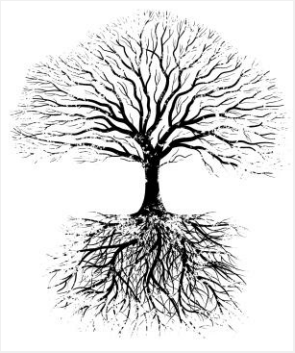
La habilidad de distinguir grupos de usuarios es crucial!

1. Integridad

Definición informal de integridad:

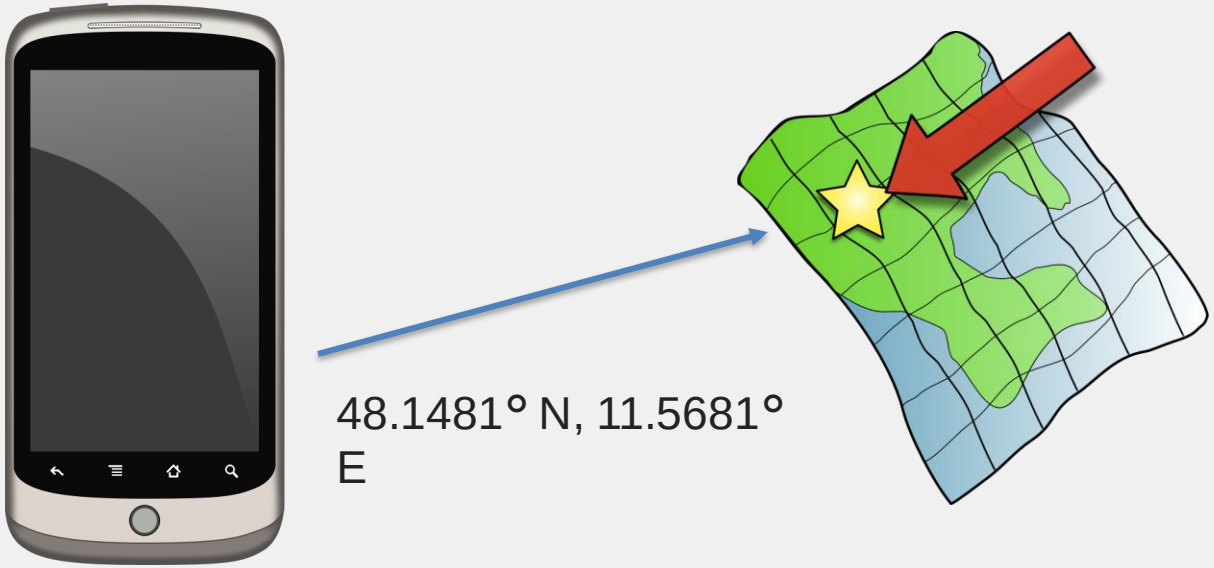
"Integrity involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle."

http://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA



En IoT también el estado del mundo fisico!

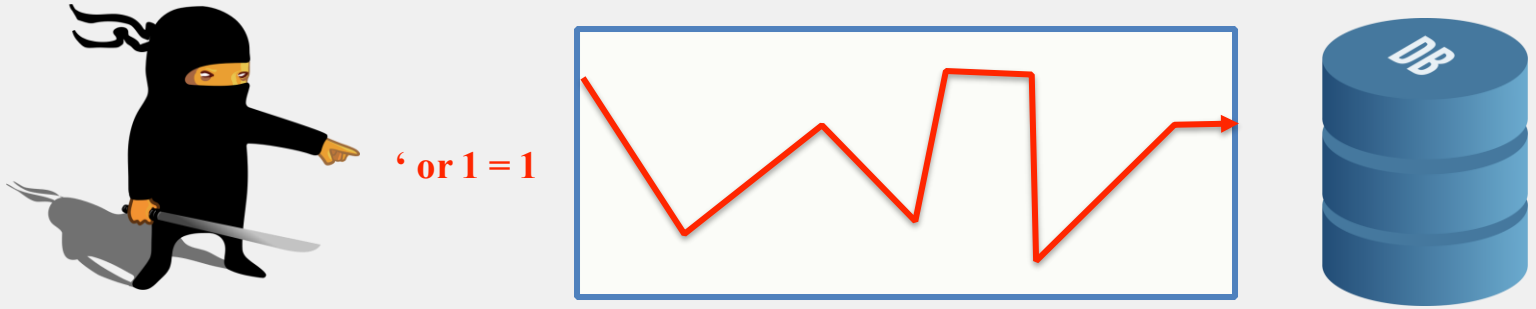
1. Flujos de información



Por ejemplo, una aplicación pide acceso a los datos de ubicación de un telefono para mostrar un mapa.
Qué sucede si la aplicación retransmite estos datos a terceros?

1. Flujos de información

- Si un atacante explota una vulnerabilidad puede violar la integridad de los datos:



- Y si logra exfiltrar datos, puede violar la confidencialidad de los mismos:

1. Limitaciones

En general, prohibir del todo los flujos es imposible.

Por ejemplo:

```
if (input == password) then
    access = 1
else
            access = 0
```

¡Es técnicamente un flujo!

1. Passwords

Idealmente el espacio de busqueda es muy grande, y la probabilidad del password de estar en algún lado es uniformemente distribuida.

1. Passwords

Por esto las recomendaciones de los passwords suelen ser:

•Al menos 8 caracteres:

-If only alphabetic: 26x26…x26 = 26^8 ca. 2^37

-If alpha-numeric: 36x36 …x36 = 36^8 ca. 2^41

-If alpha-numeric with Upper and Lower case: =62^8 ca. 2^47

-If alpha-numeric with special characters: = 95^8 ca. 2^52

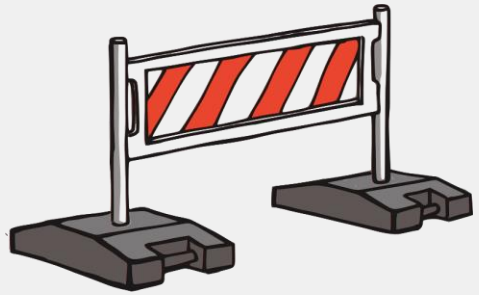
Sin embargo, es importante que los passwords sean generados aleatoriamente, de lo contrario:

Password123!

Puede ser fácil de adivinar.

1. Control de acceso

- ¿Como se garantiza la seguridad de los flujos de información?
- Historicamente la idea es bloquear el acceso a ciertos recursos por parte de ciertos usuarios.



1. Control de acceso

- Esto se puede representar con una matriz de Usuarios vs. Recursos y usando atributos de control de acceso.

	File 1	File 2	Process 1	Process 2
User A	read, write, own	read	read, write, execute, own	write
User B	append	read, own	read	read, write, execute, own

1. Control de acceso basado en roles

- Se puede generalizar usando roles:

	File 1	File 2	Process 1	Process 2
Role A	read, write, own	read	read, write, execute, own	write
Role B	append	read, own	read	read, write, execute, own

	Role
User 1	A,B
User 2	A
User 3	B

1. Control de acceso basado en roles

- Ejemplo: el acceso a archivos en Linux:

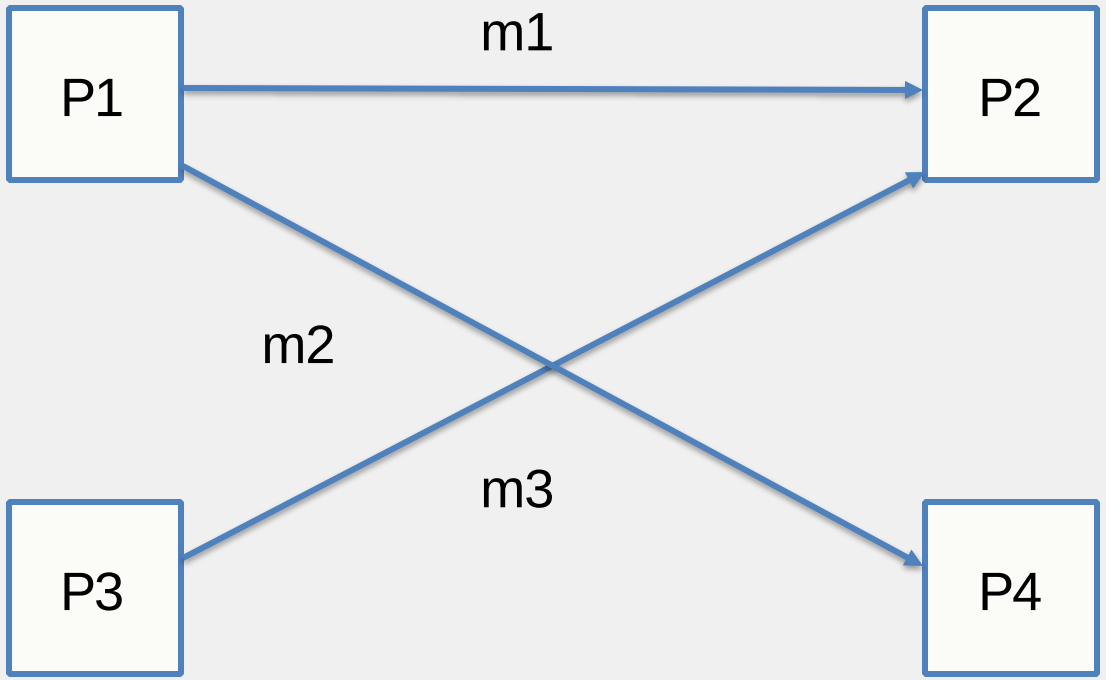
```
-rw-r--r--   1 martin_ochoa  staff      774 Aug 18  2016 Makefile
drwxr-xr-x  15 martin_ochoa  staff      510 Aug 18  2016 fig
-rw-r--r--   1 martin_ochoa  staff    58786 Aug 18  2016 generated_references.bib
-rwxr-xr-x   1 martin_ochoa  staff     1570 Aug 18  2016 getbib.py
-rwxr-xr-x   1 martin_ochoa  staff    39642 Aug 18  2016 iosart1c.cls
```

- r: read
- w: write
- x: execute
- d: directory
- Fist set of permissions: owner
- Second set: group (RBAC)
- Third groups: others

2. Sistemas distribuidos

- Qué son los sistemas distribuidos?
  - Colección de procesos autonomos que se comunican para lograr un objetivo computacional común.
  - Ejemplos:
    - Una red (Internet, LAN)
    - Juegos en línea multi-usuario.
    - Sistemas de control industrial.
    - Multi-threading y virtualización.
- Porqué son relevantes?
  - Cada vez más populares (multi-core computing, Blockchain)
  - En el futuro aún más: Internet of Things

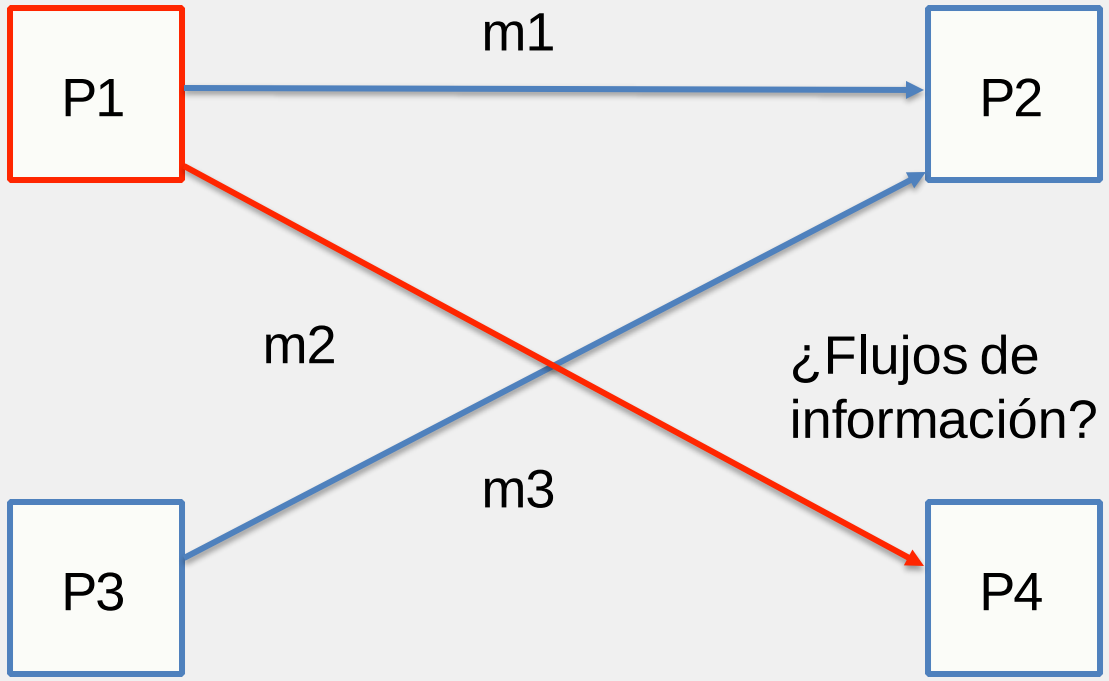
2. Sistemas distribuidos



- Cada proceso tiene un estado local.

2. Sistemas distribuidos

¿Control de acceso?



2. Sistemas distribuidos: seguridad?

- Todo lo que hemos discutido hasta ahora.

Además:

- Concurrencia.
  - Race conditions.
  - Deadlocks.
  - Secure time.

- Tolerancia a fallos.
  - Denial of Service.

2. Sistemas distribuidos: seguridad?

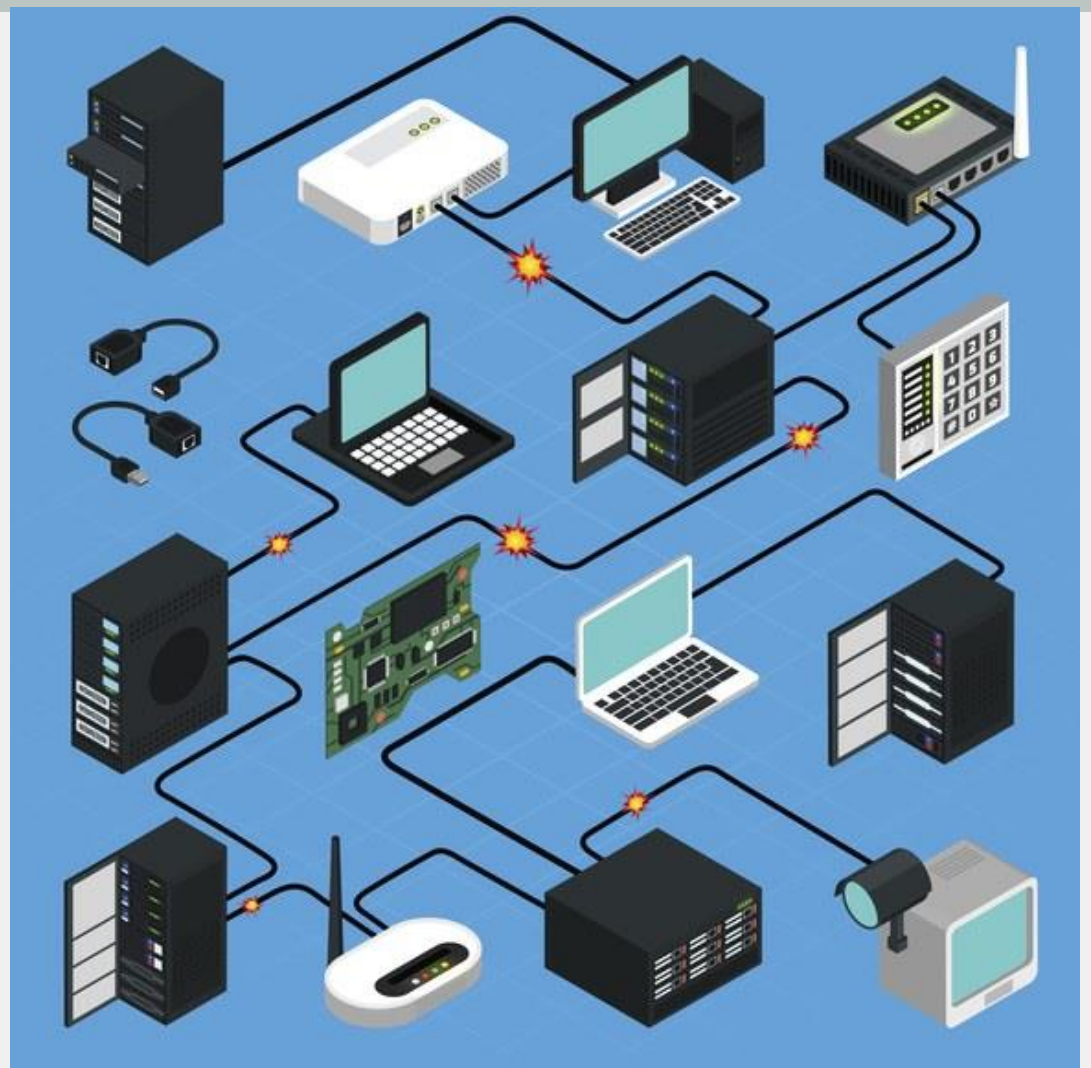
- La concurrencia es una fuente de problemas en sistemas de computación.
  - Es dificil prever todas las interacciones del sistema.
  - *En sistemas operativos*: interacciones no deterministas dadas por el scheduler.
  - *En redes:* interacciones no esperadas dadas por los nodos autonomos.
  - *En sistemas críticos*: sensores y eventos del ambiente (velocidad, presión, sensores que fallan).

2. Concurrencia y disponibilidad: DoS

- Components of a system might be forced to fail by malicious adversaries with different purposes:
  - Take a service down
  - Make a certain machine restart to temporarily impersonate it.

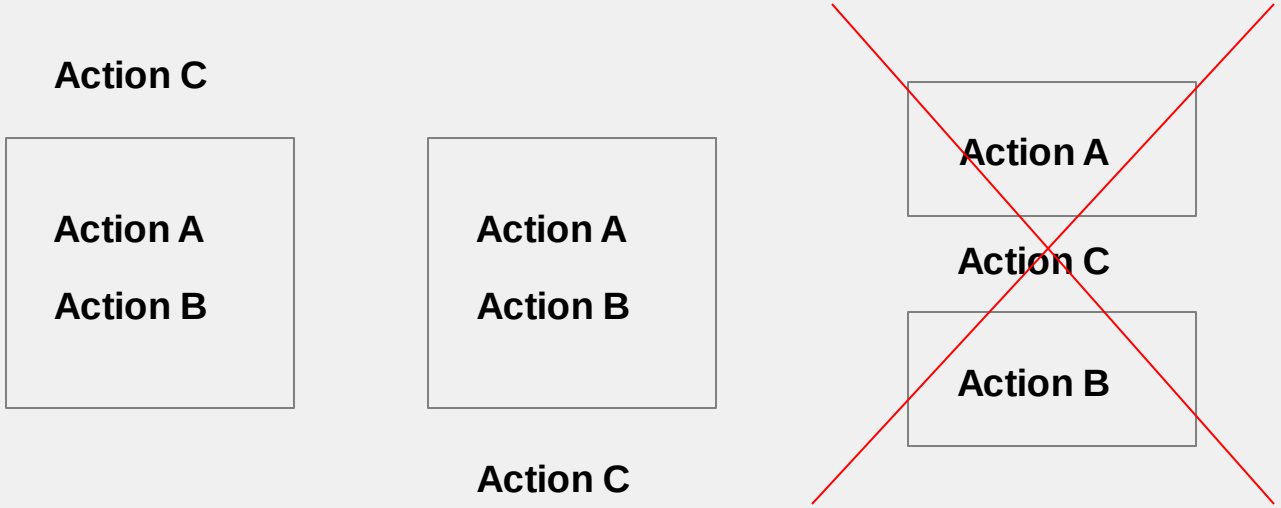
- Nowadays even harder with Distributed Denial of Service attacks (DDoS)
  - A botnet of infected machines can be used to flood a system (resource exhaustion).
  - How do you know which machines to block and which not to?
  - Need for extreme replication!

2. Concurrencia y disponibilidad: DoS



Mirai botnet believed to be responsible of several spectacular DDoS attacks in late 2016 (attacking popular DNS service).

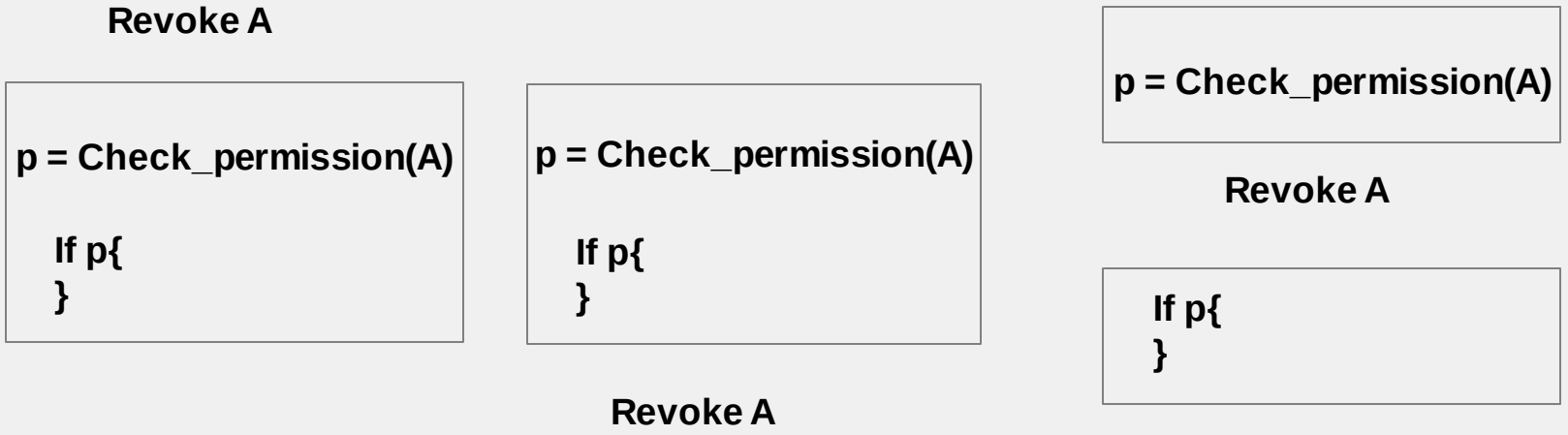
2. Concurrencia y seguridad



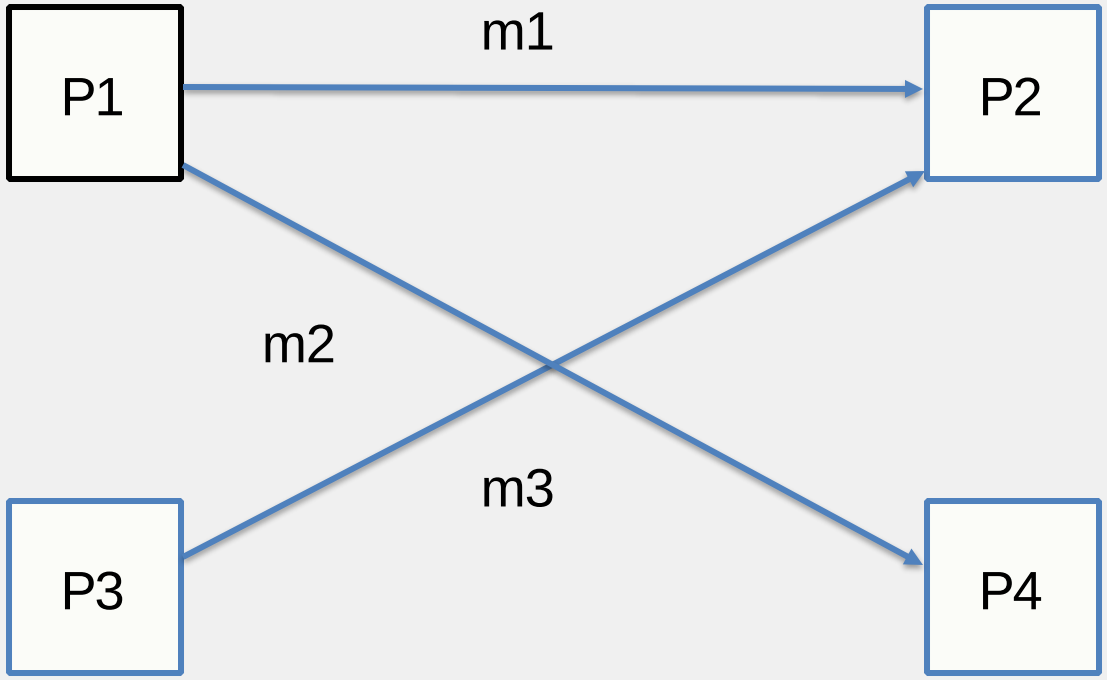
Un problema común de la concurrencia: violaciones a la atomicidad.

2. Concurrencia y seguridad: TOCTOU

Time of check, Time of Use:



2. Concurrencia y disponibilidad



- Cambios en el estado local deben ser propagados a todo el sistema.

2. Propagación del estado

- Números de tarjetas de crédito son robados a diario.

- Los bancos mantienen una lista de los números robados. Una lista global contiene millones de números.

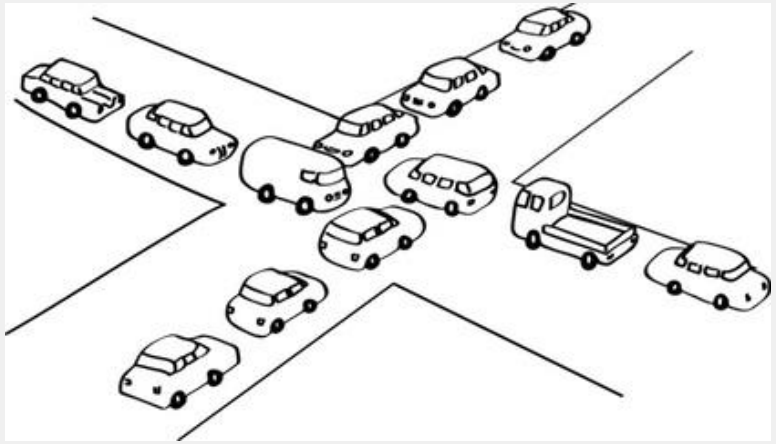
Por motivos de costo y funcionalidad, esta lista no se propaga instantaneamente todo el tiempo.

  - Si el monto de una compra es pequeño, no se revisa.
  - Si la tarjeta es local, revise lista local.
  - Si el monto es alto, revise con entidad, por ejemplo, VISA.
  - Si el monto es aún más alto, revise con banco emisor.

2. Deadlocks

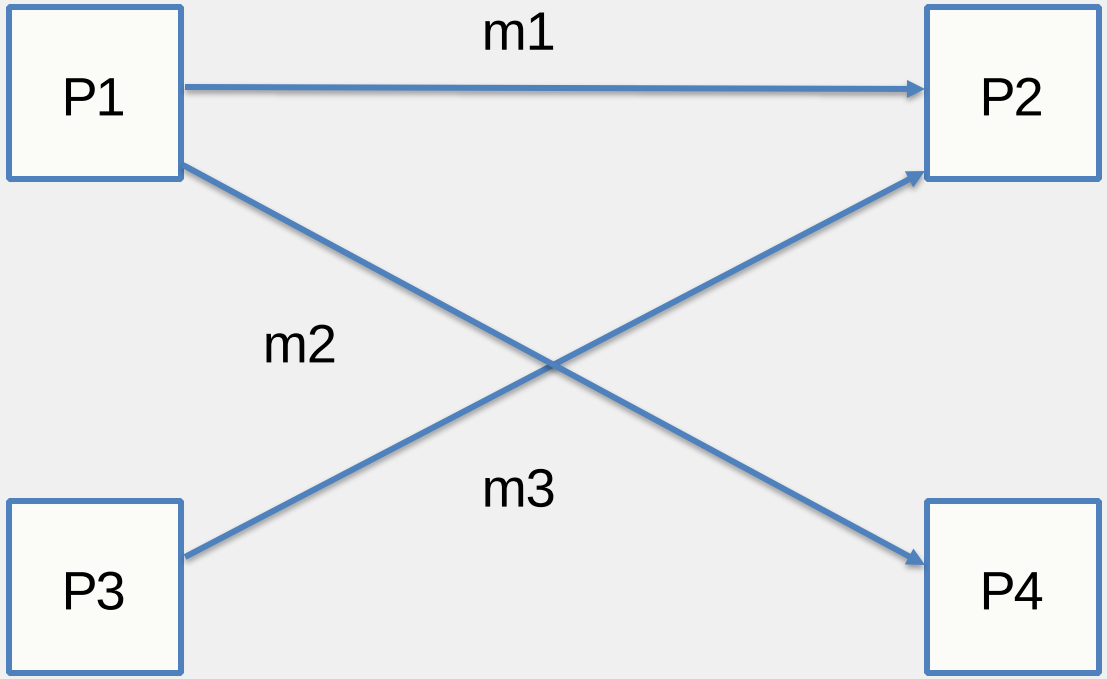
- Para prevenir actualizaciones indeseadas, es popular usar bloqueos a las transacciones (por ejemplo, como en git o svn, o bases de datos).

- Efectuar bloqueos sin causar problemas es complejo!



https://classic.csunplugged.org/routing-and-deadlock/

2. Deadlocks



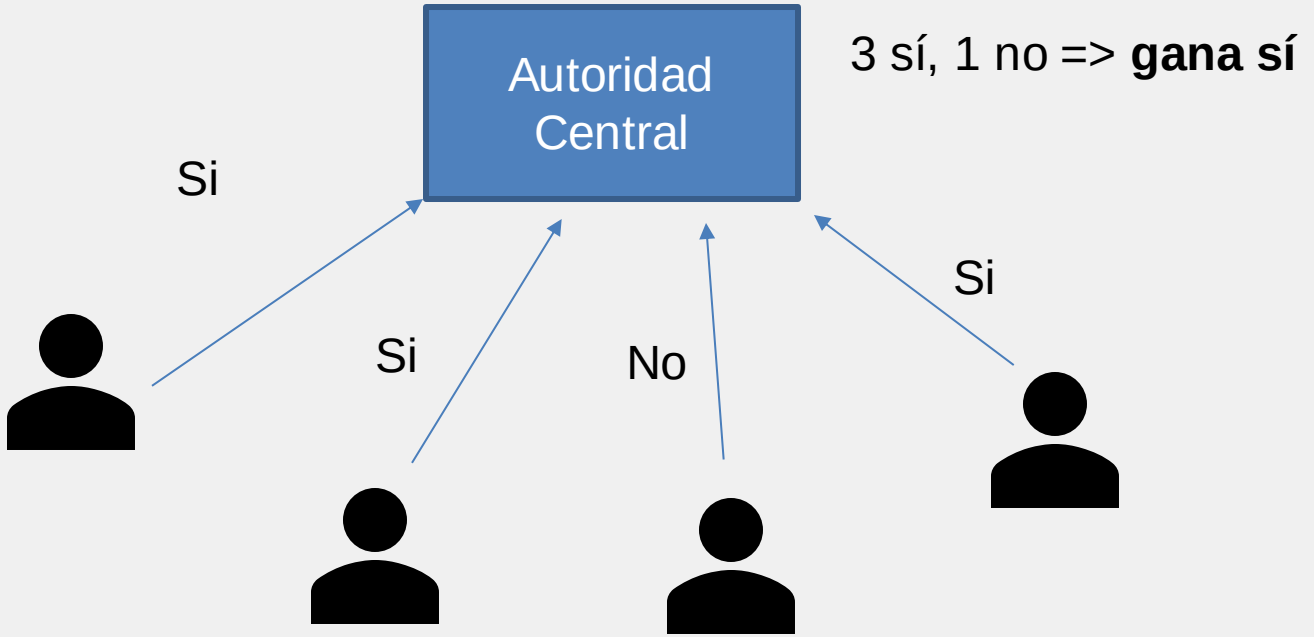
Si es posible llegar a un deadlock, esto afecta la disponibilidad del sistema.

2. Importancia de la disponibilidad

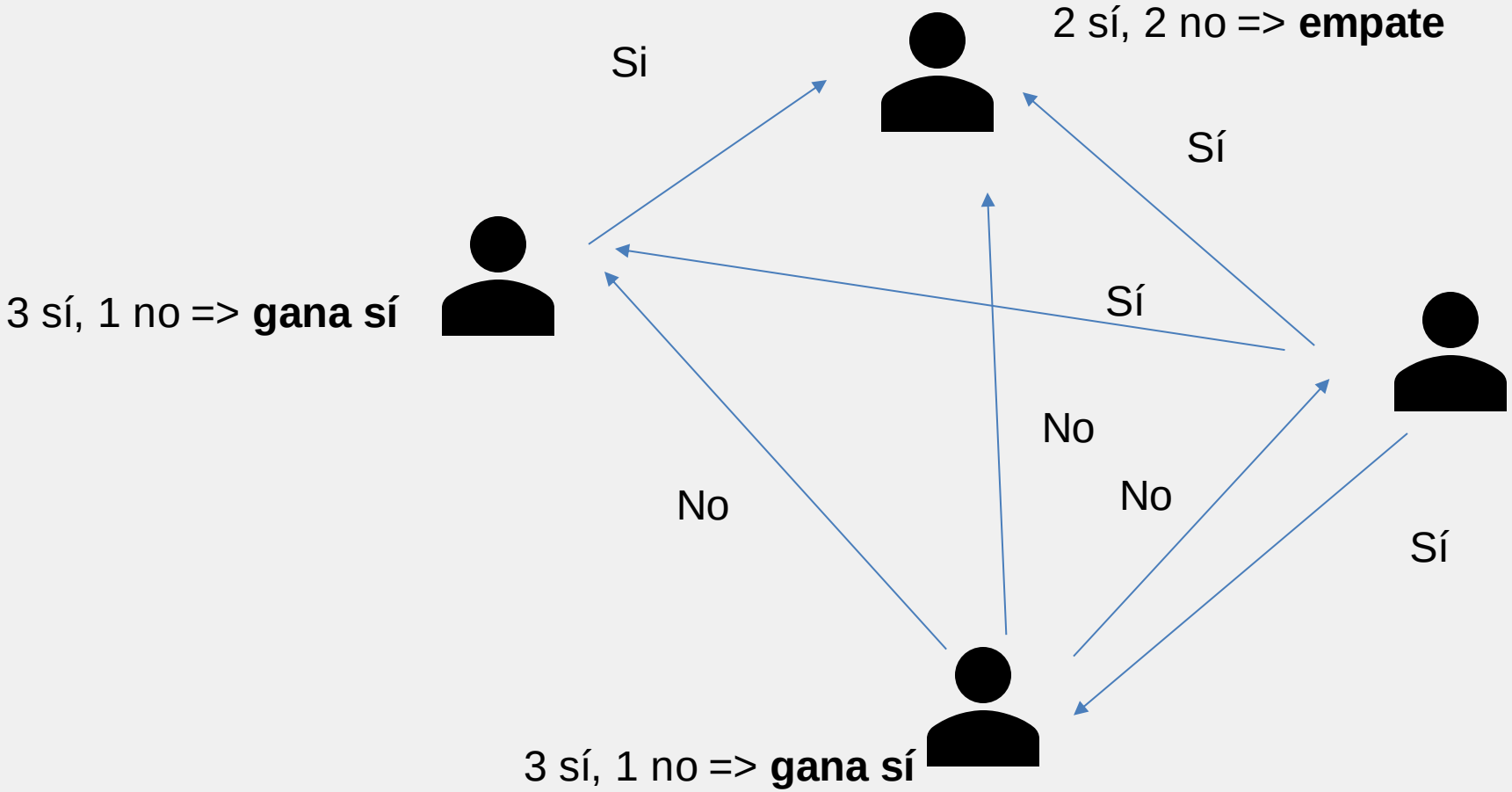
- La tolerancia a los errores es fundamental para la ciberseguridad.

- Investigación enfocada sobre todo en confidencialidad e integridad, sin embargo disponibilidad es más improtante en la práctica.
  - Cualquier servicio (bancos, social networks, tiendas) -> Disponibilidad es fundamental para la operación.
  - Cada minuto de Twitter caído cuesta 25MM USD:
    - http://www.cnet.com/news/the-cost-of-twitter-downtime/

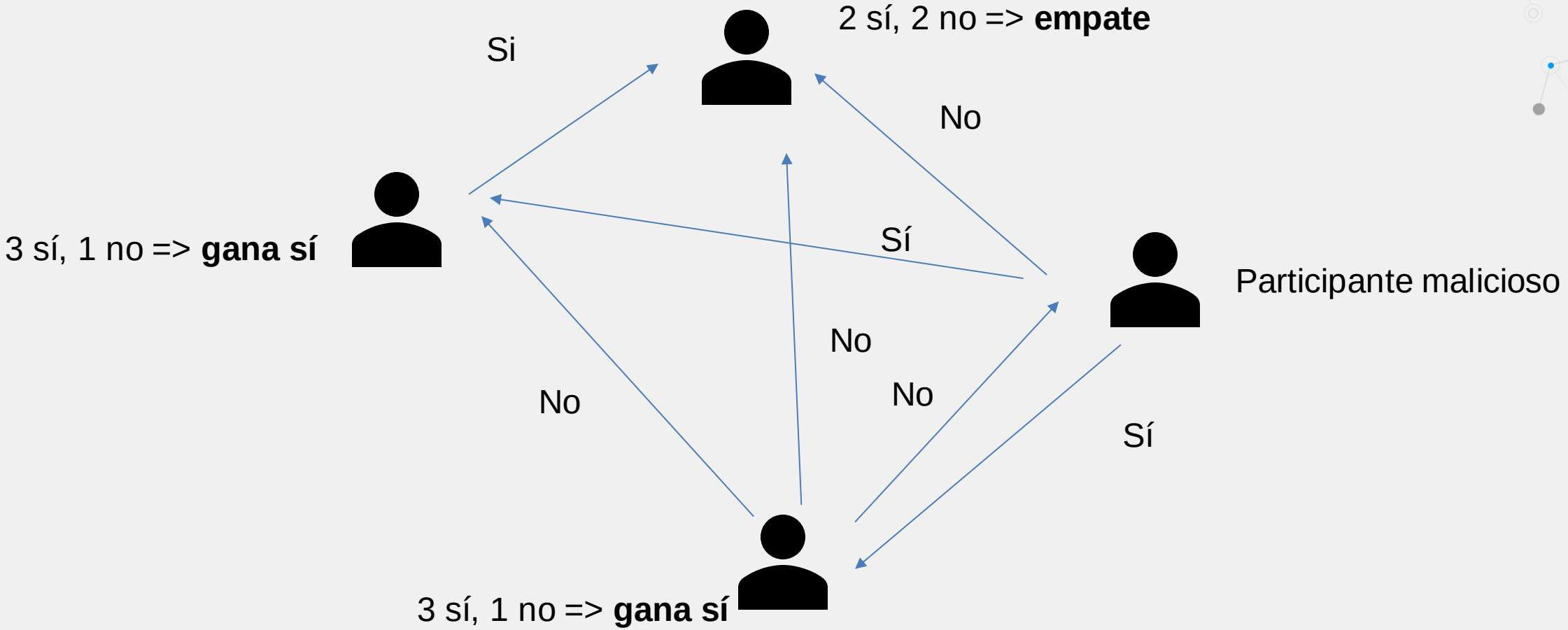
3. Consenso



3. Consenso


2 sí, 2 no => **empate**
Si
Sí
3 sí, 1 no => **gana sí**
Sí
No
No
No
Sí
3 sí, 1 no => **gana sí**

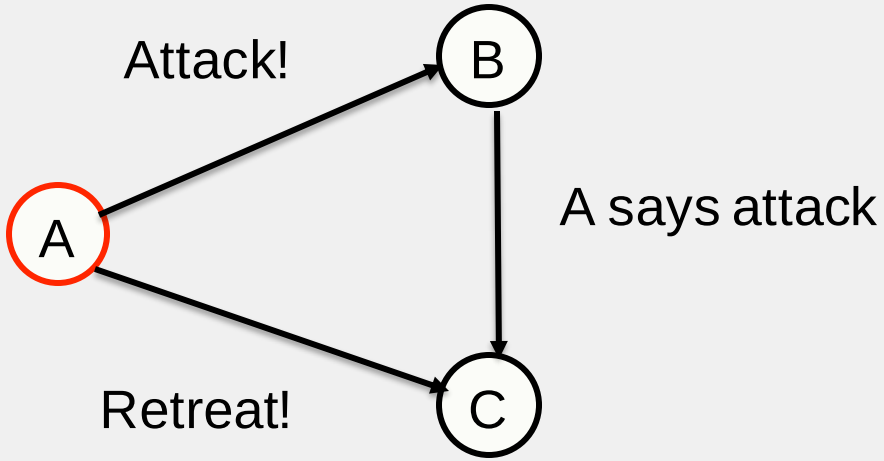
3. Consenso


2 sí, 2 no => **empate**
Si
No
3 sí, 1 no => **gana sí**
Sí
Participante malicioso
No
No
No
Sí
3 sí, 1 no => **gana sí**

3. Consenso

- Asuma que hay n generales defendiendo Bizancio, de los cuales t son traidores.
  - Todos se comunican entre sí, los traidores quieren confundir a los generales.

- Cúal es el número máximo de traidores que se pueden tolerar?



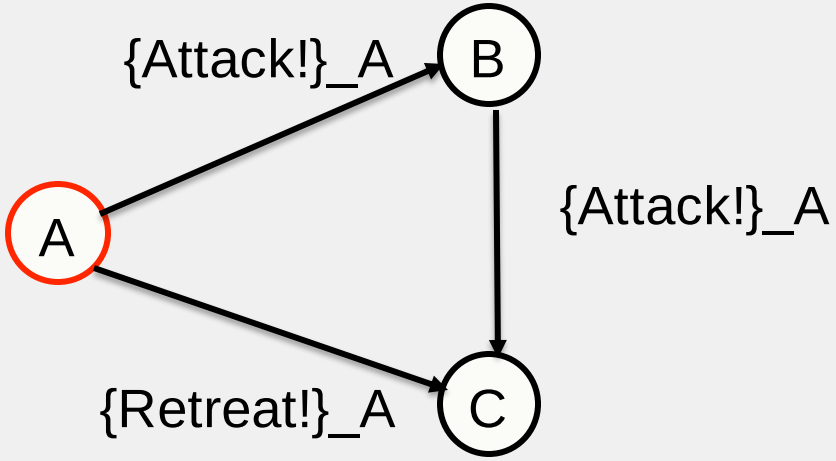
$n >= 3t + 1$ (Lamport et al.)

3. Consenso

- Asuma que hay n generales defendiendo Bizancio, de los cuales t son traidores.
  - Todos se comunican entre sí, los traidores quieren confundir a los generales.

- Cúal es el número máximo de traidores que se pueden tolerar?

3. Consenso



Firmas ayudan a detectar traidores

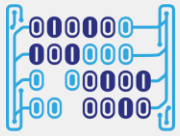
$n >= 3t + 1$ (Lamport et al.)

Resumen

- Existen 3 propiedades generales de seguridad: Confidencialidad, Integridad y Disponibilidad.
- Para proteger la integridad de un estado distribuido es importante tener algoritmos de consenso y propagación del estado.
- Estos conceptos son fundamentales para entender propiedades de seguridad de Blockchain.

Gracias

Universidad del Rosario



MACC
Matemáticas Aplicadas y Ciencias de la Computación

References

- M. Bishop, *Introduction to Computer Security, Addison Wesley*, 2005
- Goguen, J.A., Meseguer, J.: *Security policy and security models*. In: In Proceedings of Symposium on Secrecy and Privacy, pp. 11-20, 1982
- Rushby, J.: *Noninterference, transitivity, and channel-control security policies*. Technical report SRI CSL 92-02, December 1992
- R. Anderson, *Security Engineering*, 2nd edition, 2008
- Lamport, Leslie, Robert Shostak, and Marshall Pease. "*The Byzantine generals problem*." ACM Transactions on Programming Languages and Systems (TOPLAS) 4.3 (1982): 382-401.

Ejercicios

1. Clasifique los siguientes casos como una violación de integridad, confidencialidad, disponibilidad o una mezcla de las anteriores.

a) Juan se copia la tarea del curso de Blockchain de María.
b) Pablo apaga el celular de Carlos remotamente usando wi-fi.
c) Ana cambia el valor del cheque de Juan de 1B a 100B.
d) Sandra falsifica la firma de Roberto en un documento.
e) Juan descubre el número de tarjeta de crédito de Pedro y llama al banco para que cancelen la tarjeta.
f) Enrique adivina el password de Twitter de Julia y twittea en su nombre.

Ejercicios

2. De ejemplos de situaciones donde:

a) Un ataque a la confidencialidad da pie a un ataque a la integridad y viceversa.

b) Un ataque a la confidencialidad da pie a un ataque a la disponibilidad y viceversa.

c) Un ataque a la integridad da pie a un ataque a la disponibilidad y viceversa.

Ejercicios

3. El aforismo "seguridad por obscuridad" sugiere que esconder información garantiza un cierto nivel de seguridad. De un ejemplo en el cual esconder información no mejora apreciablemetne la seguridad de un sistema. De un ejemplo de una situación donde si ayuda.