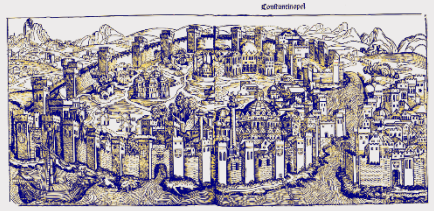
Byzantine Agreement

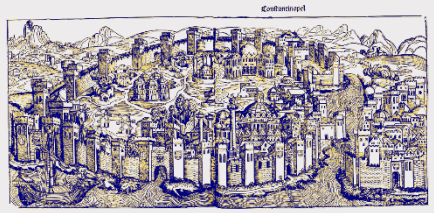
Jacob Gardner & Chuan Guo

What is agreement?



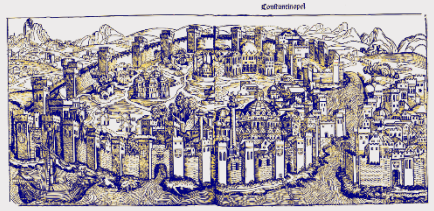
What is agreement?

Attack, or
Retreat?



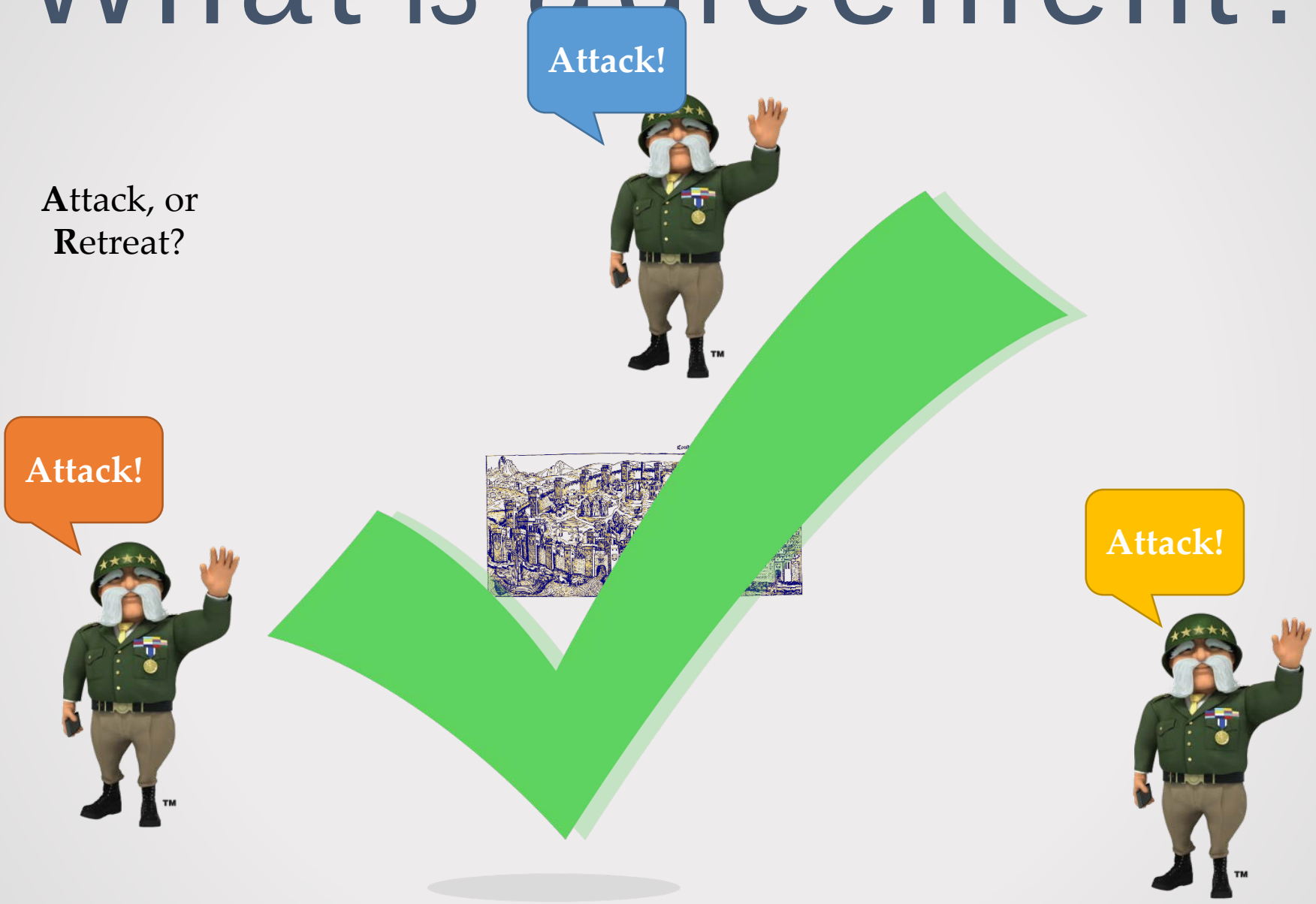
What is agreement?

Attack, or
Retreat?



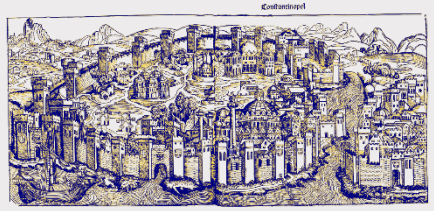
What is agreement?

Attack, or
Retreat?



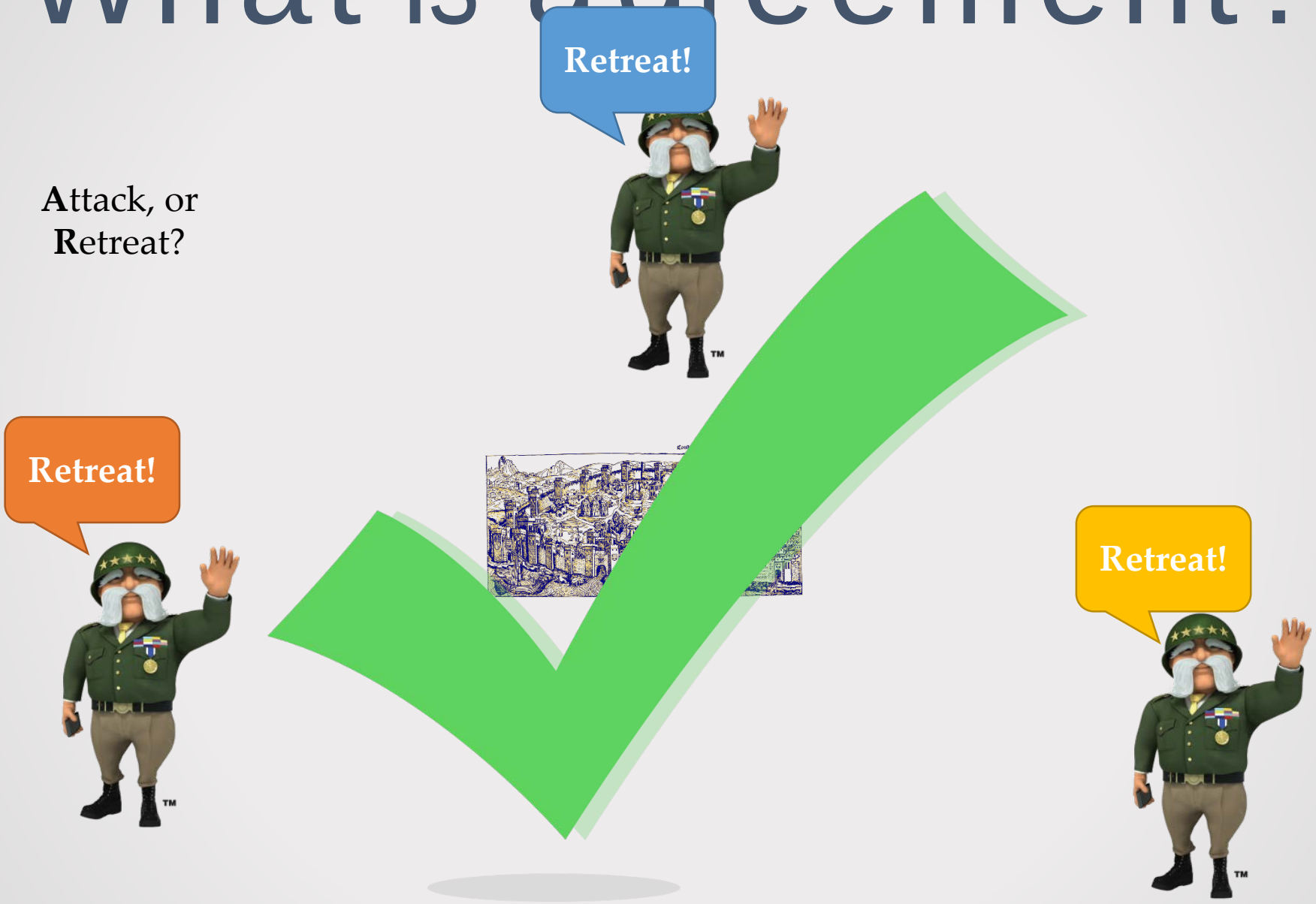
What is agreement?

Attack, or
Retreat?



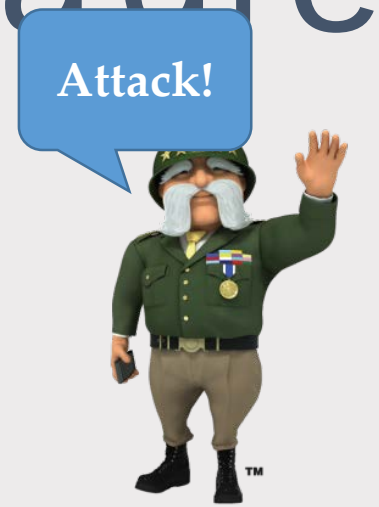
What is agreement?

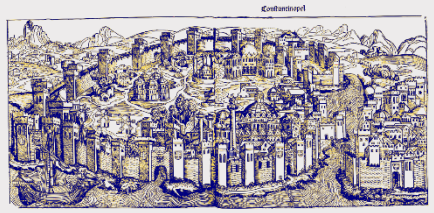
Attack, or
Retreat?



What is agreement?

Attack, or
Retreat?





What is agreement?

Attack, or
Retreat?

What is agreement?

All generals decide upon the same plan of action.

Caveat: This is **not** a paper about military strategy.

What is agreement?

All generals decide upon the same plan of action.

What basic tools do we need to achieve agreement?

Caveat: This is **not** a paper about military strategy.

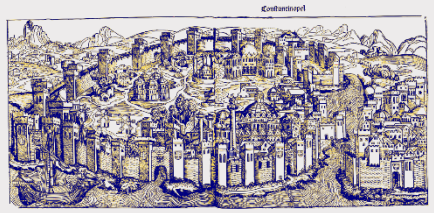
Assumptions (p 387)

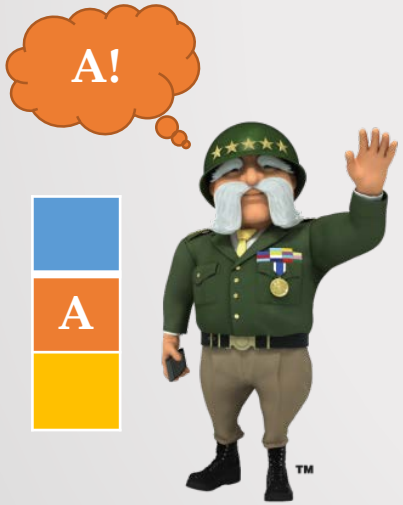
- All messages are delivered correctly.
- All recipients can verify who sent each message.
- The absence of a message can be detected.

"Oral Messages"

Agreement

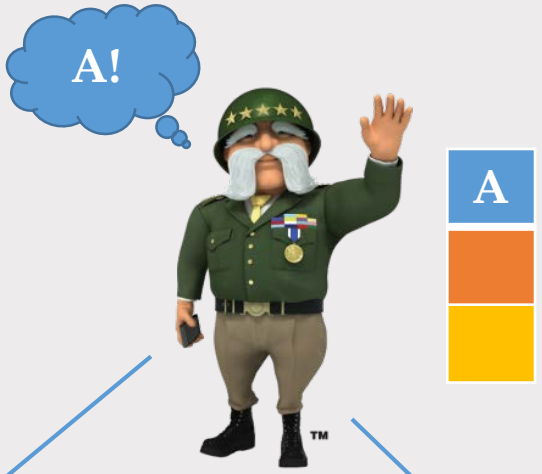
Attack, or
Retreat?

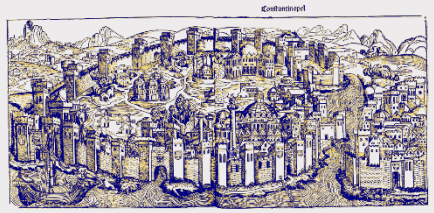


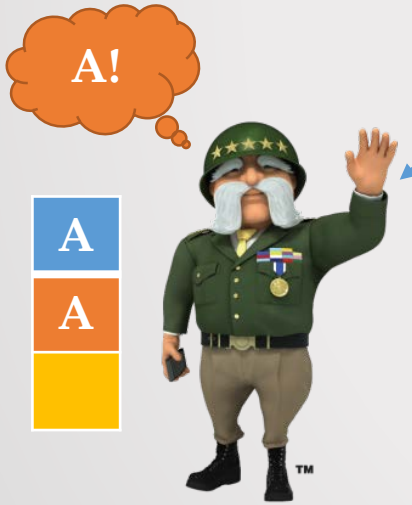


Agreement

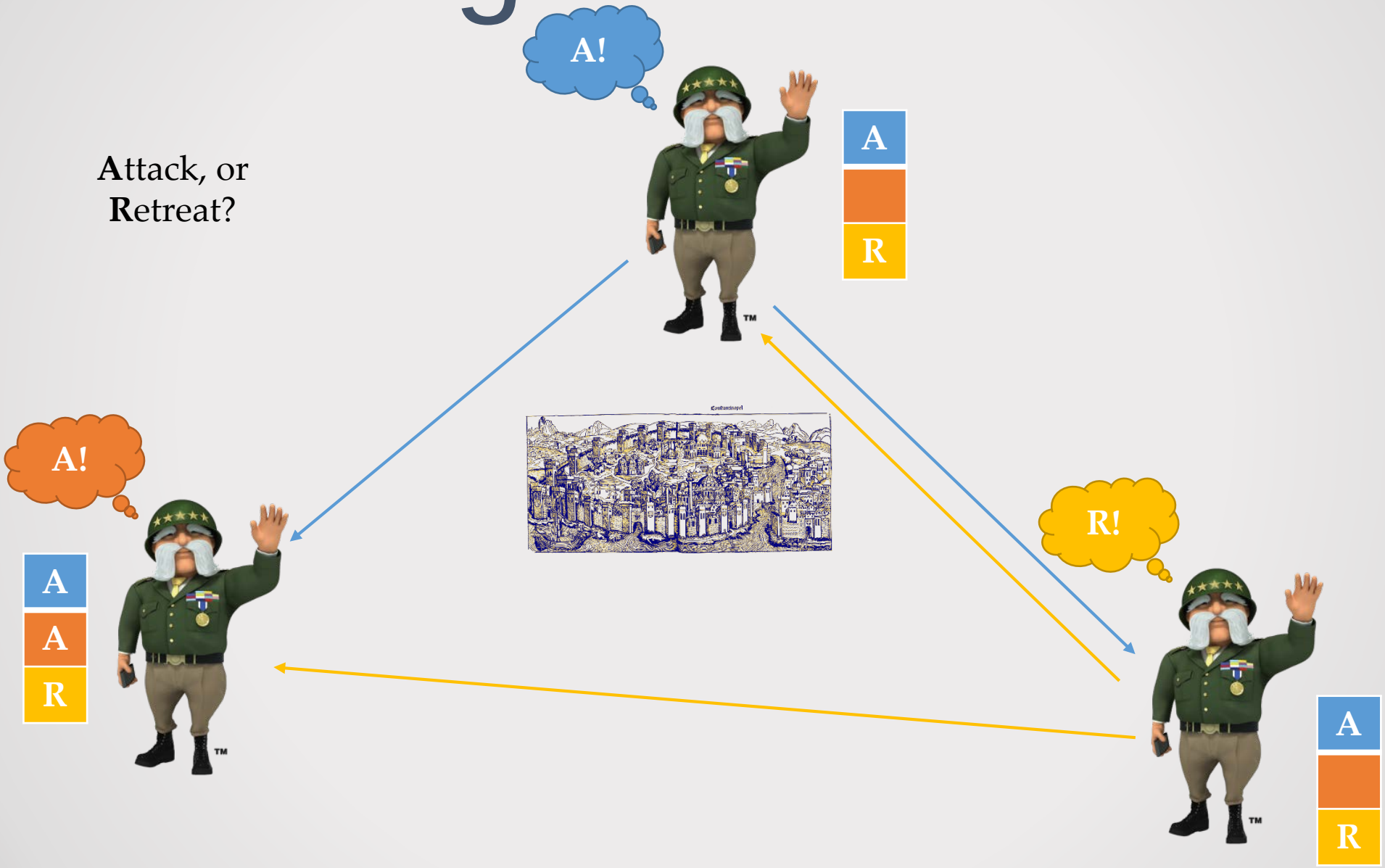
Attack, or
Retreat?



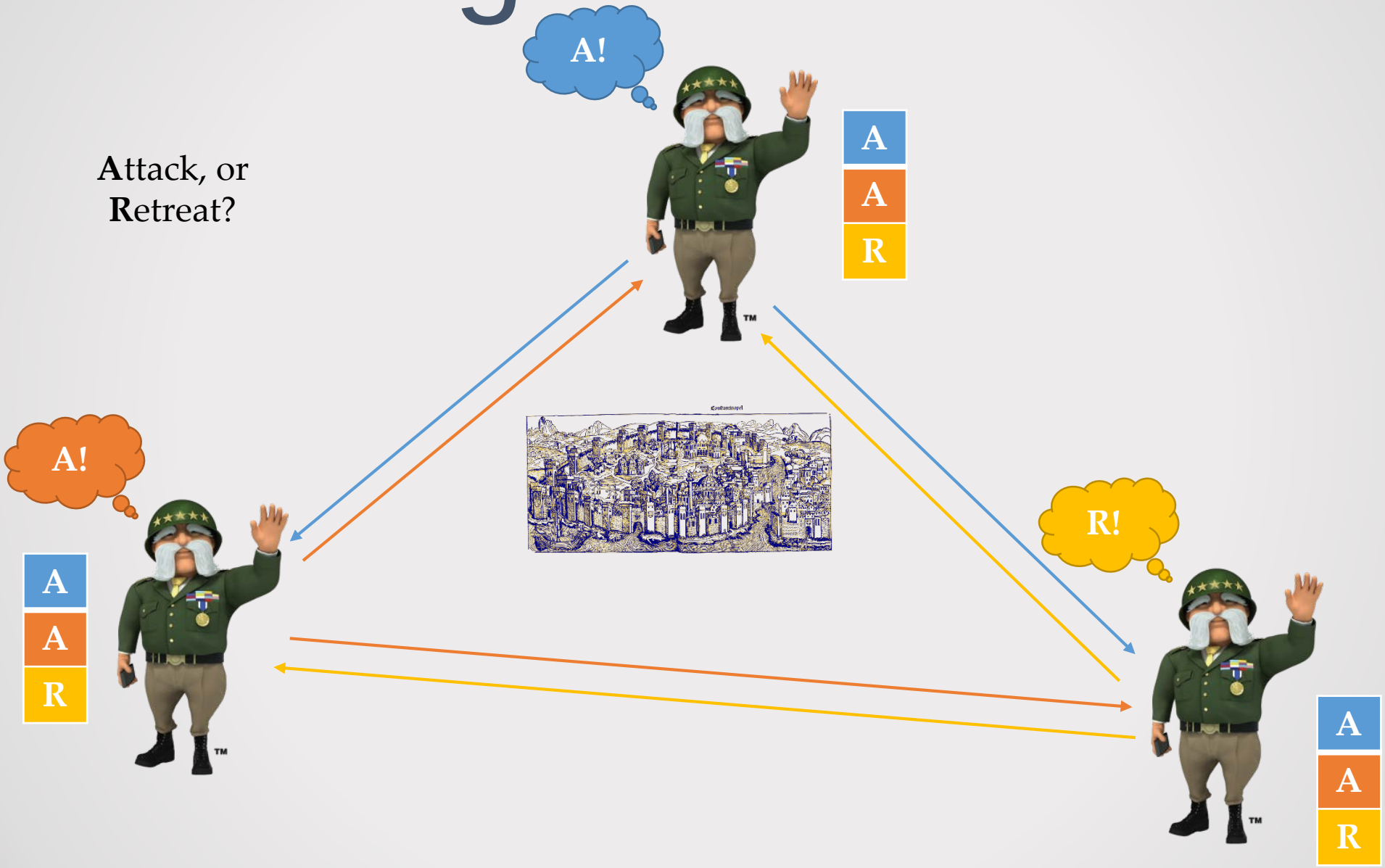




Agreement

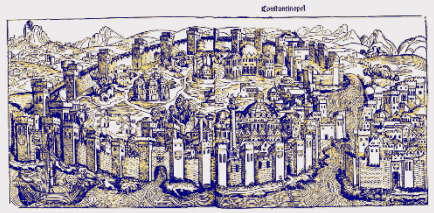


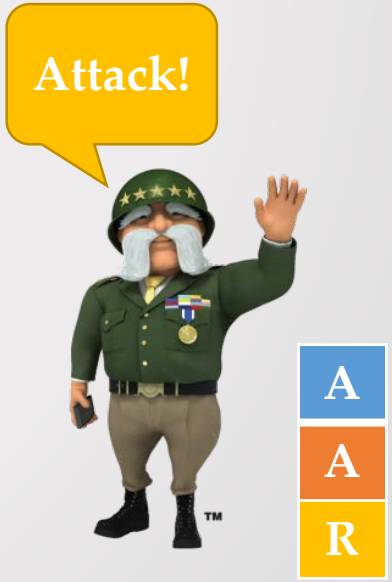
Agreement



Agreement

Attack, or
Retreat?





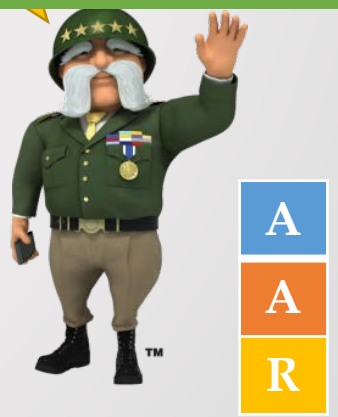
Agreement

Attack, or
Retreat?

Observation (p 384)

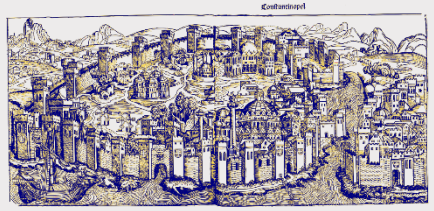
We can restrict our consideration to the problem of how a single general sends his value to the others.

v(1



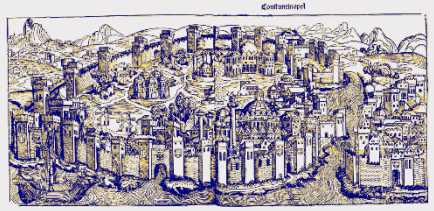
Reformulation

Attack, or
Retreat?



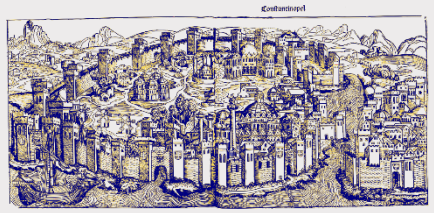
Reformulation

Attack, or
Retreat?

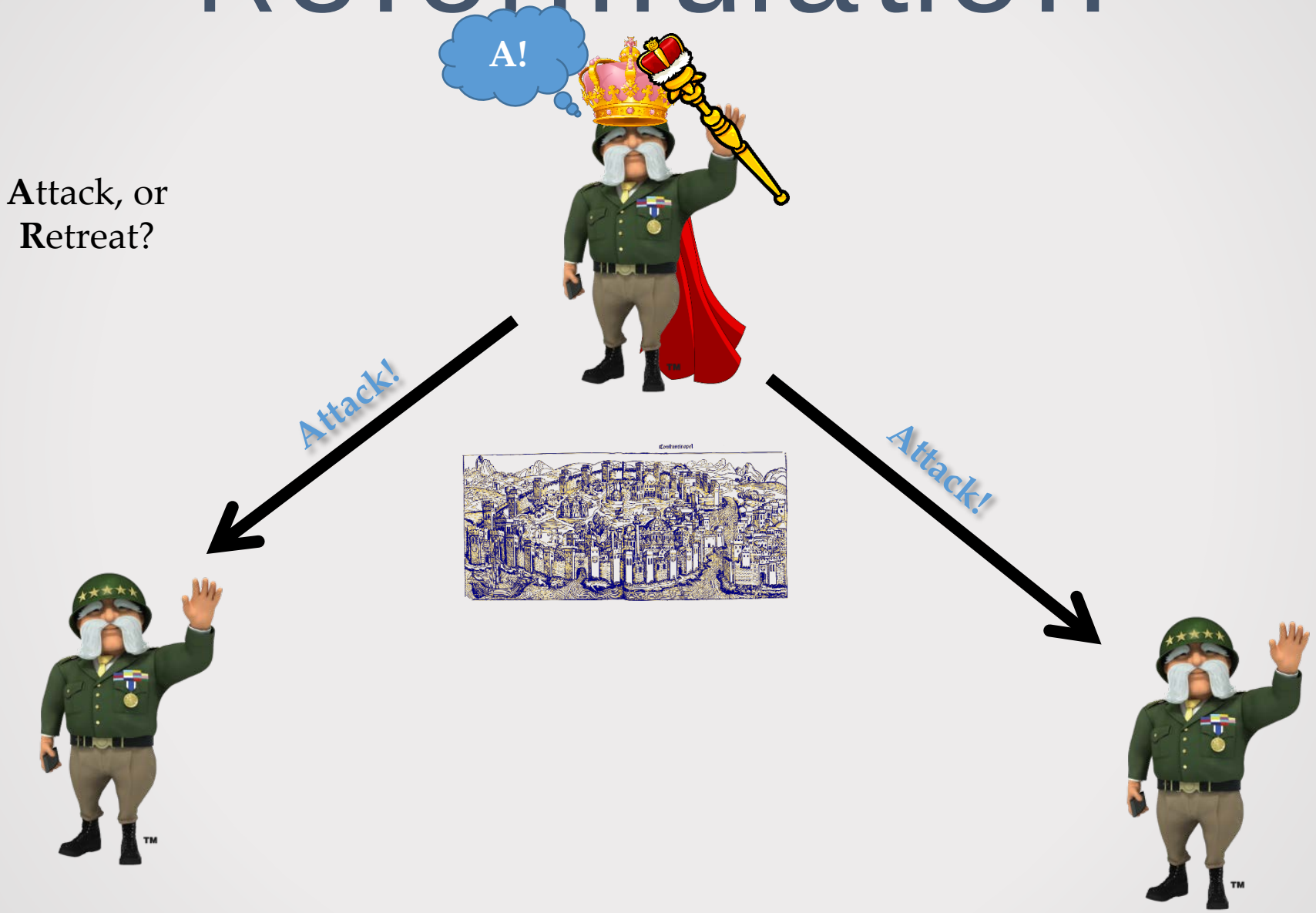


Reformulation

Attack, or
Retreat?

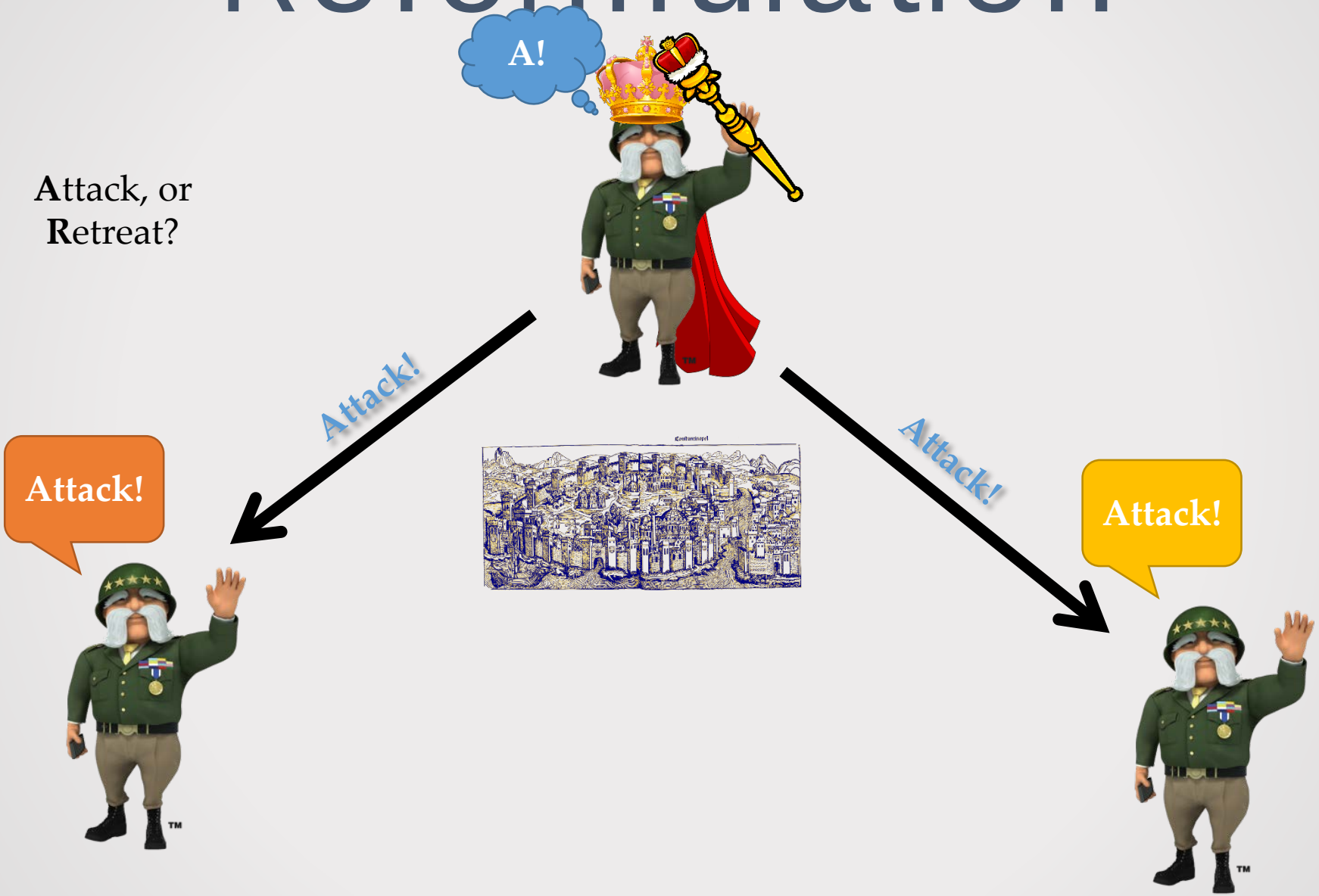


Reformulation



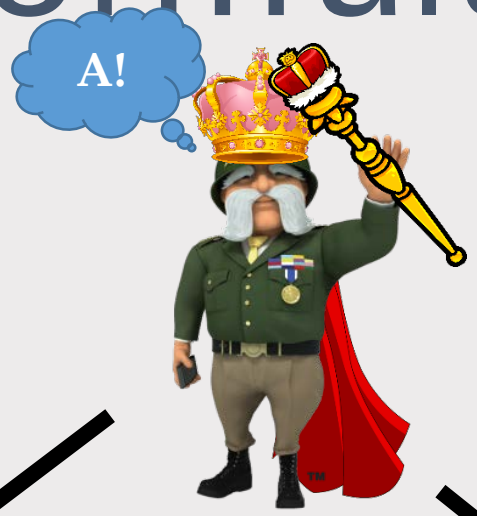
Reformulation

Attack, or
Retreat?

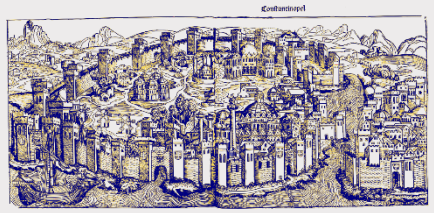


Reformulation

Attack, or
Retreat?



Attack!



Attack!

How does this differ from Paxos?

What is agreement?

All generals decide upon the same plan of action.

Caveat: This is **not** a paper about military strategy.

What is agreement?

Loyal

All generals decide upon the same plan of action.

Caveat: This is **not** a paper about military strategy.

What is agreement?

Loyal

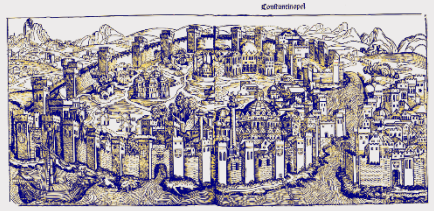
All generals decide upon the same plan of action.

Why?

Caveat: This is **not** a paper about military strategy.

What could go wrong?

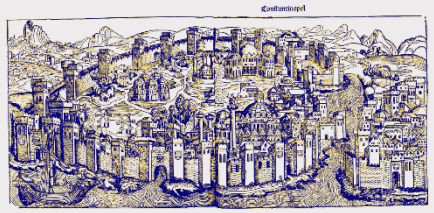
Attack, or
Retreat?



What could go wrong?

Attack, or
Retreat?

Attack!



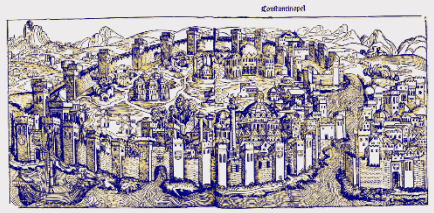
Retreat!

What could go wrong?

Attack, or
Retreat?

Attack!

Retreat!

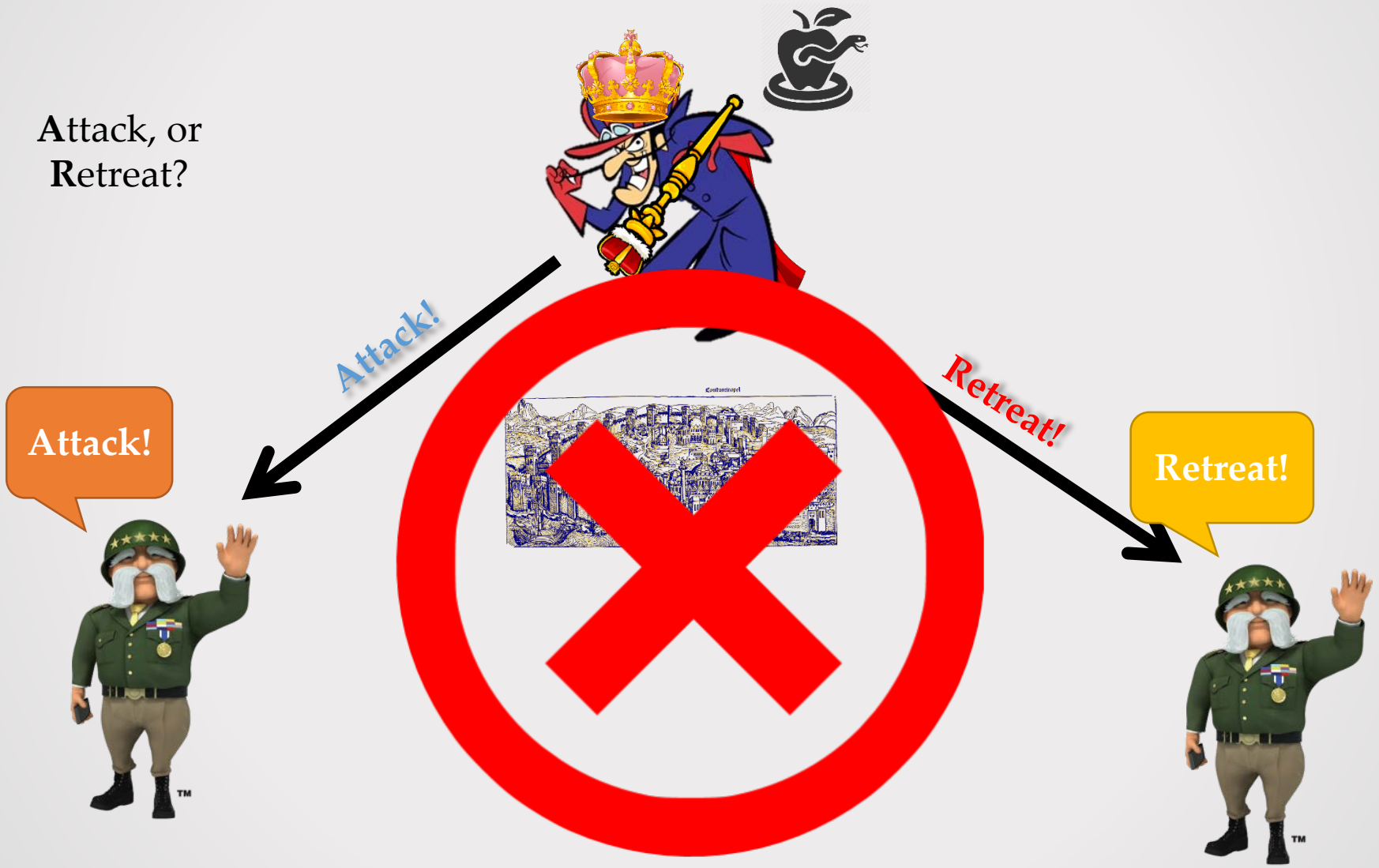


Attack!

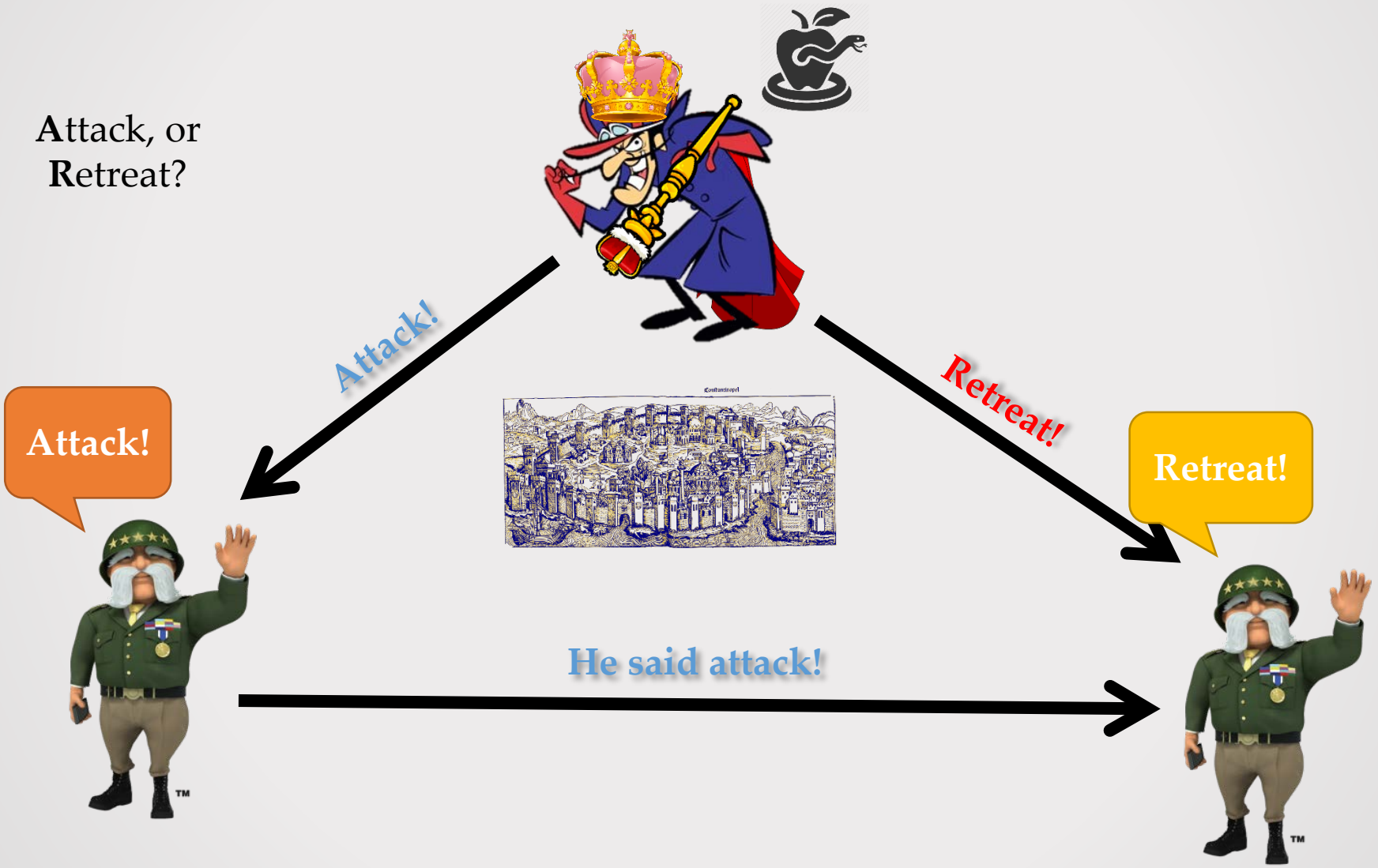
Retreat!

What could go wrong?

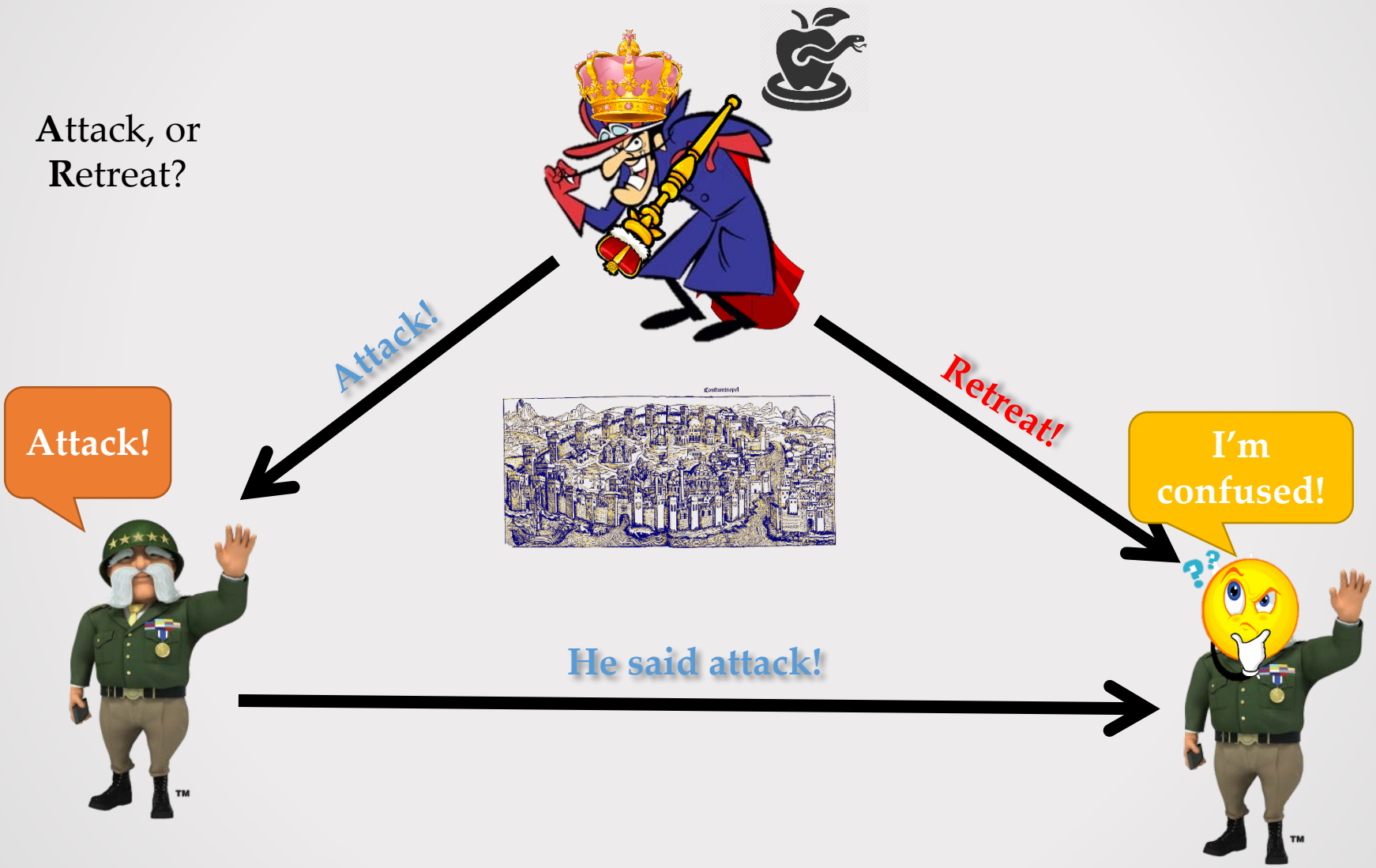
Attack, or
Retreat?



What could go wrong?

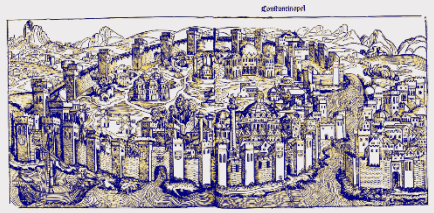


What could go wrong?



What could go wrong?

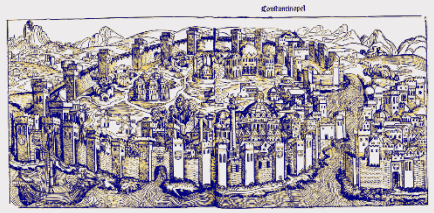
Attack, or
Retreat?



What could go wrong?

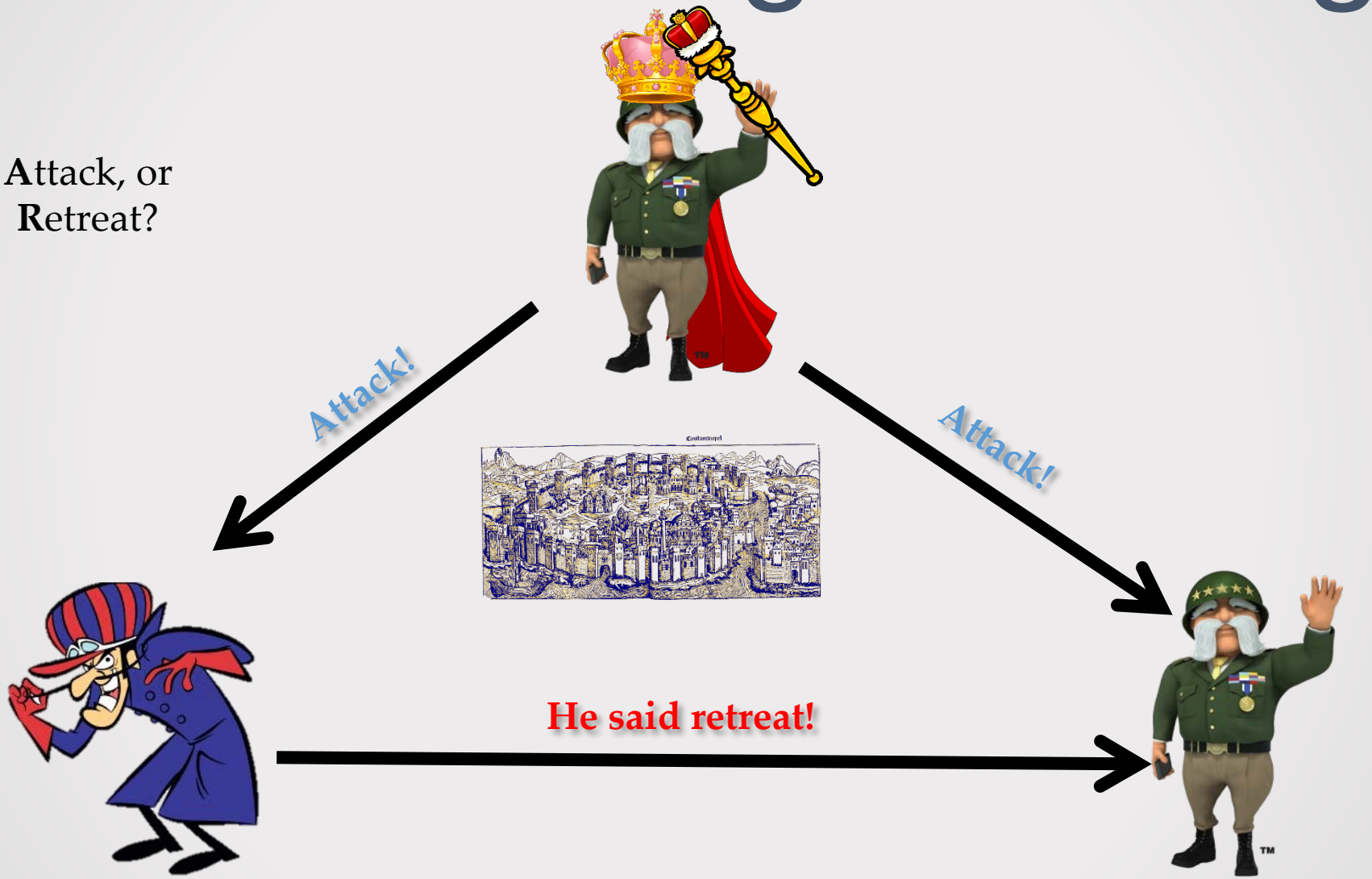
Attack, or
Retreat?

Attack!

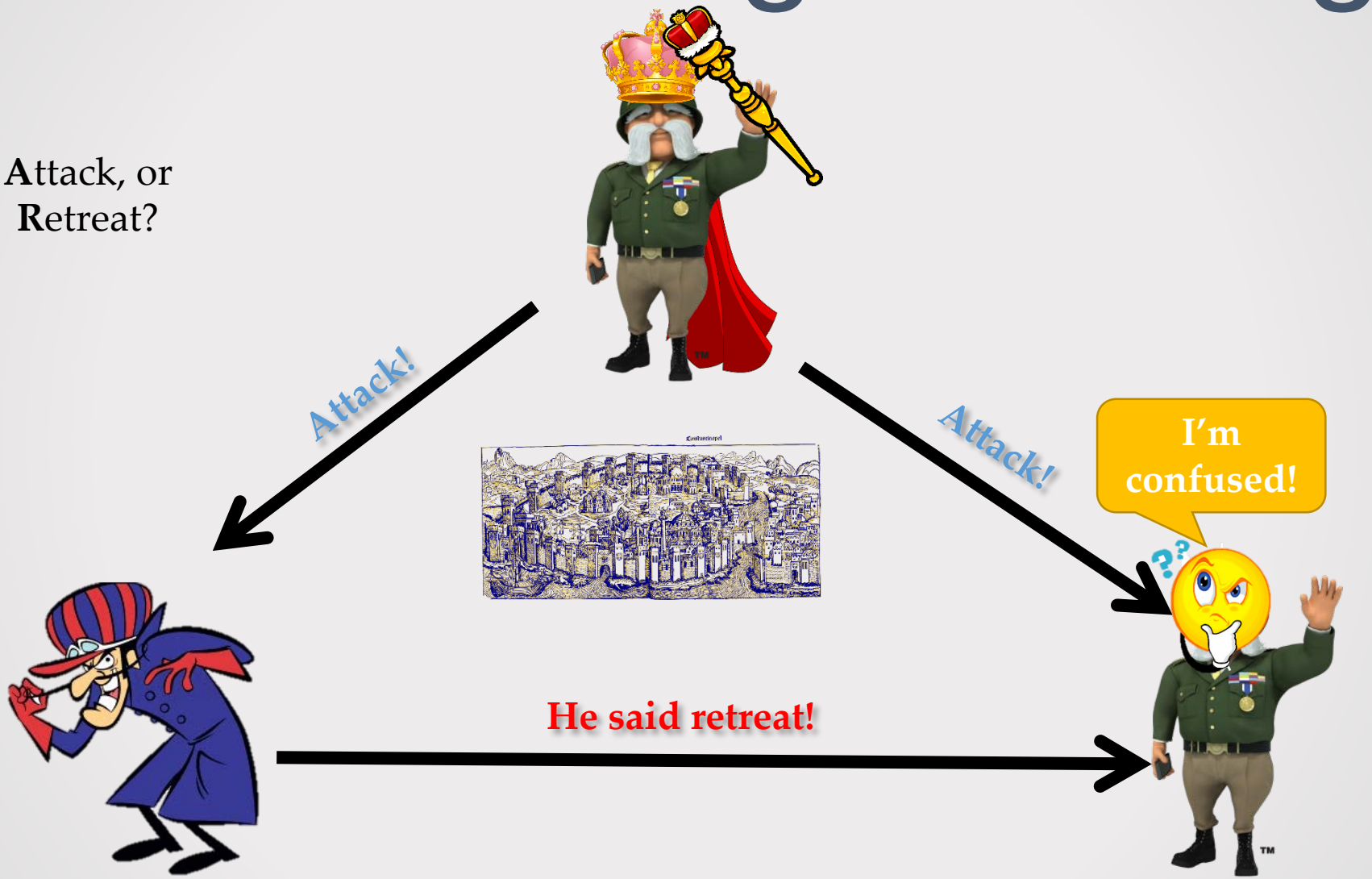


Attack!

What could go wrong?



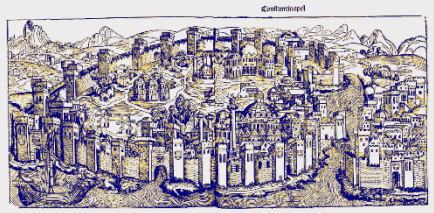
What could go wrong?



What could go wrong?

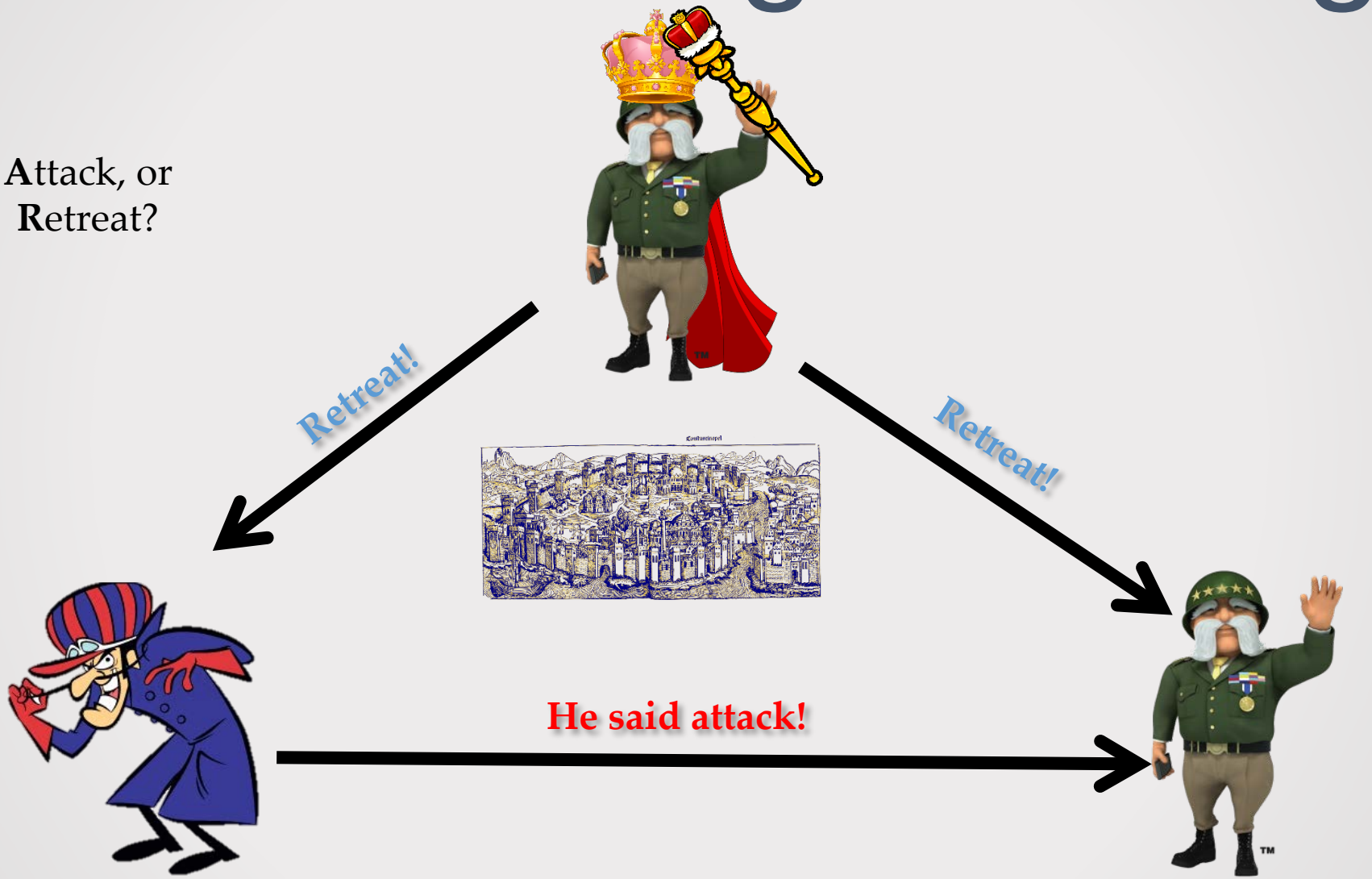
Attack, or
Retreat?

Retreat!

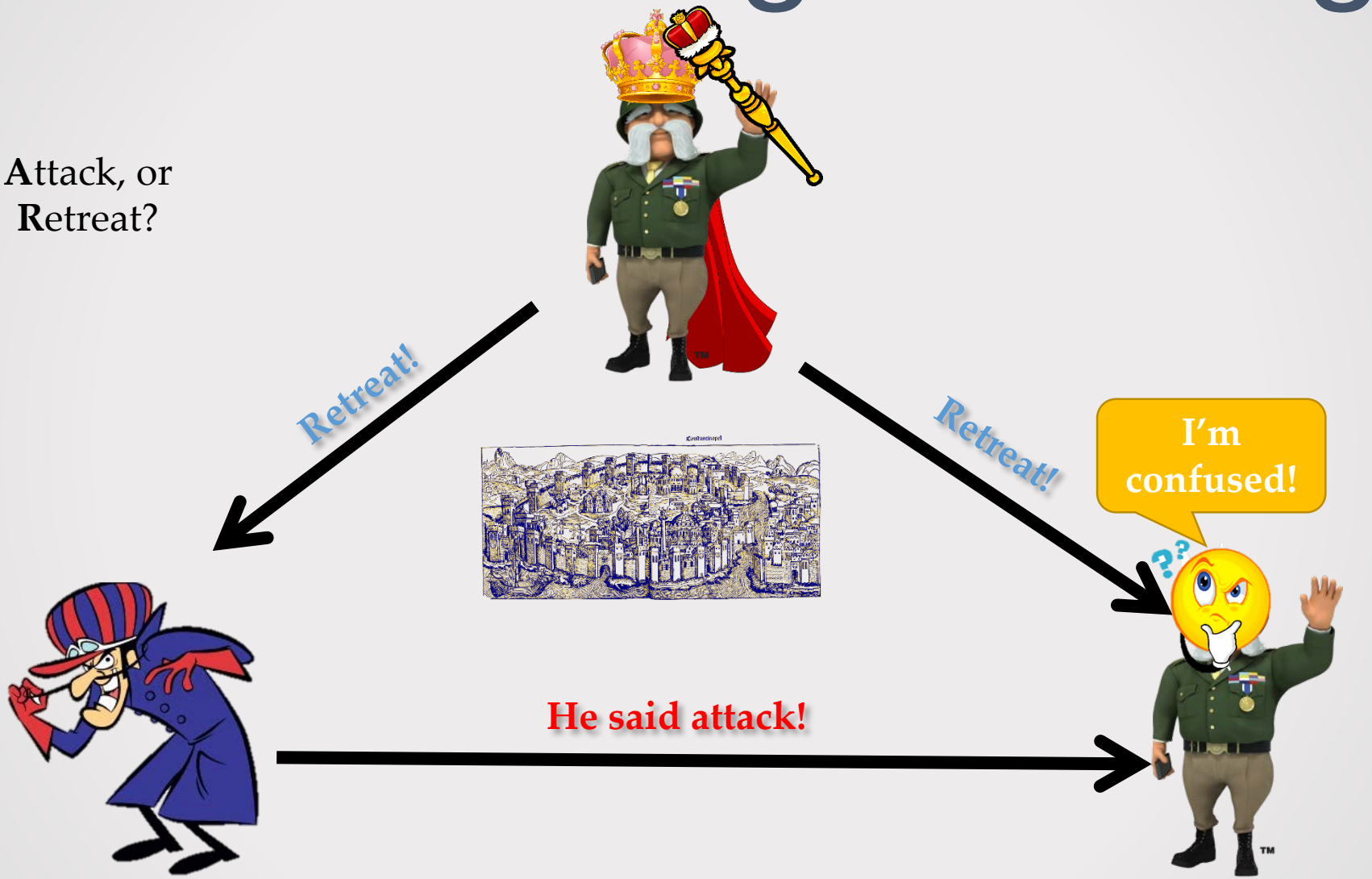


Retreat!

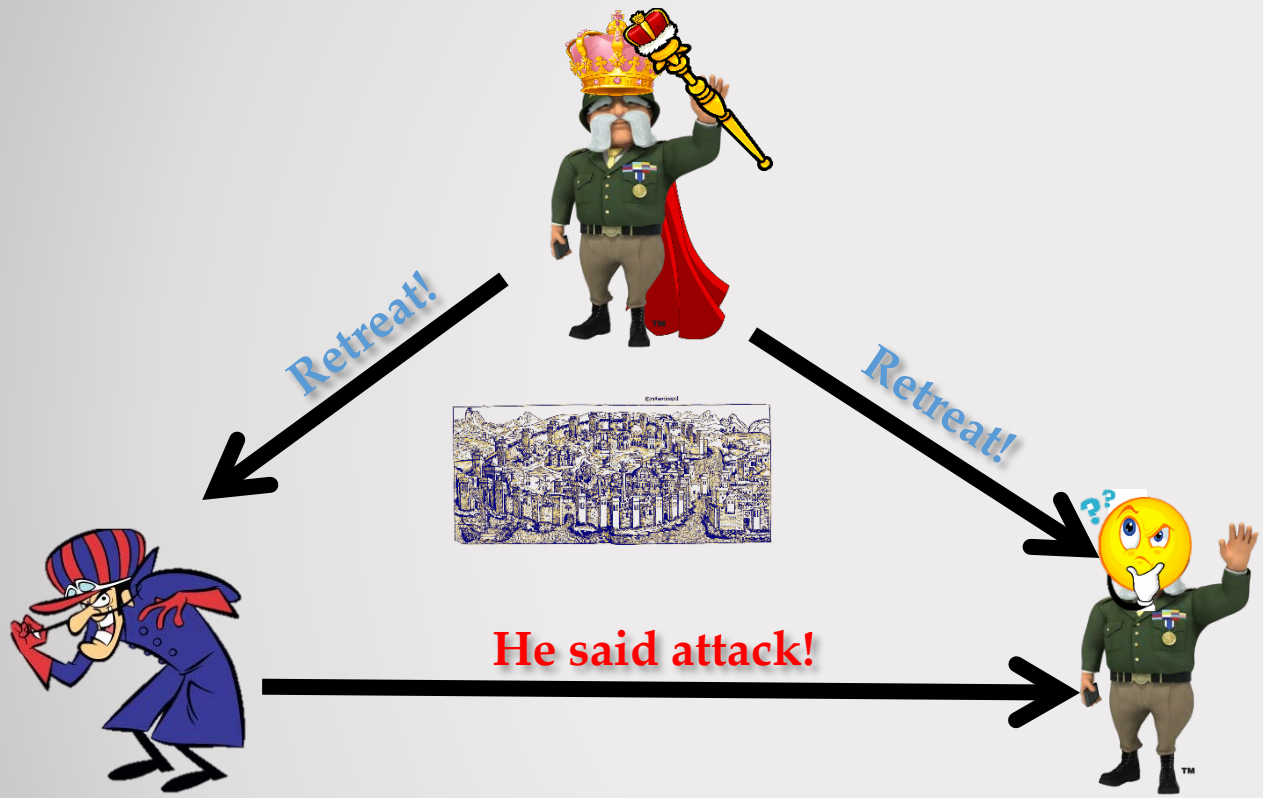
What could go wrong?



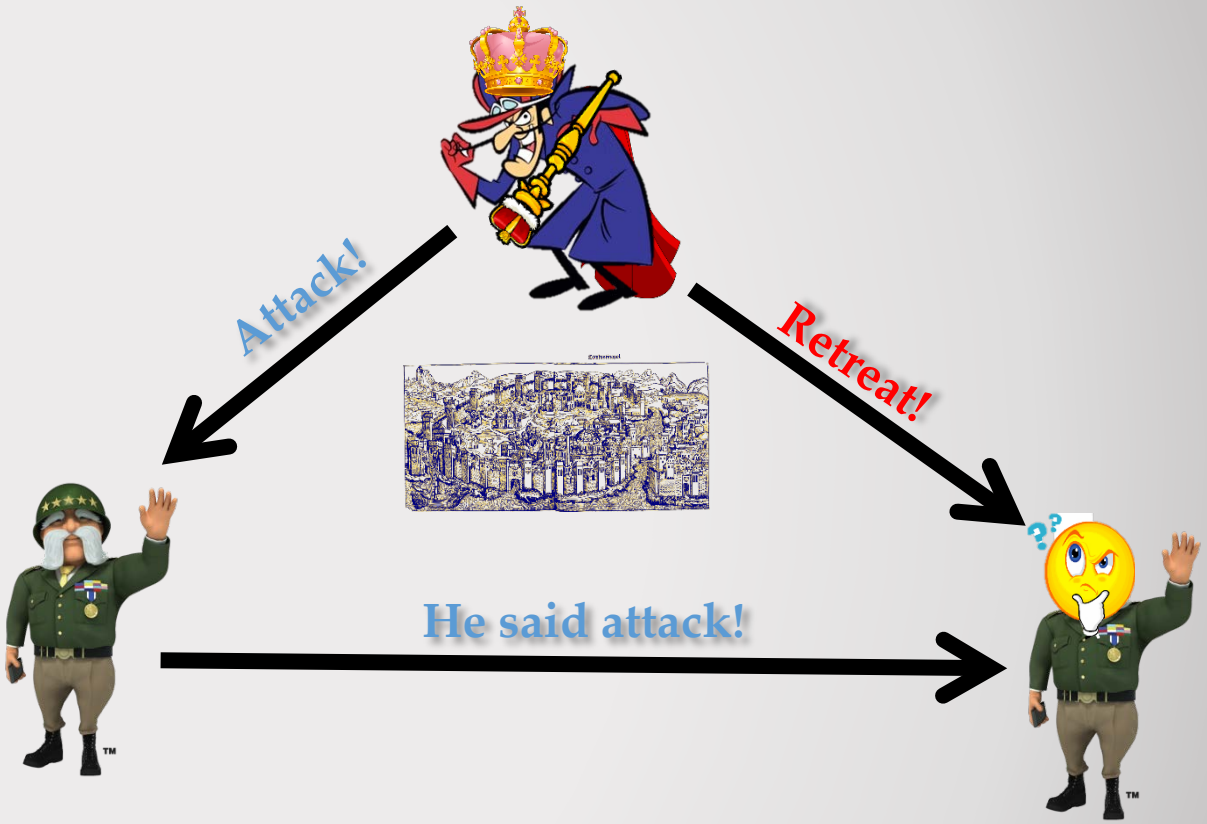
What could go wrong?

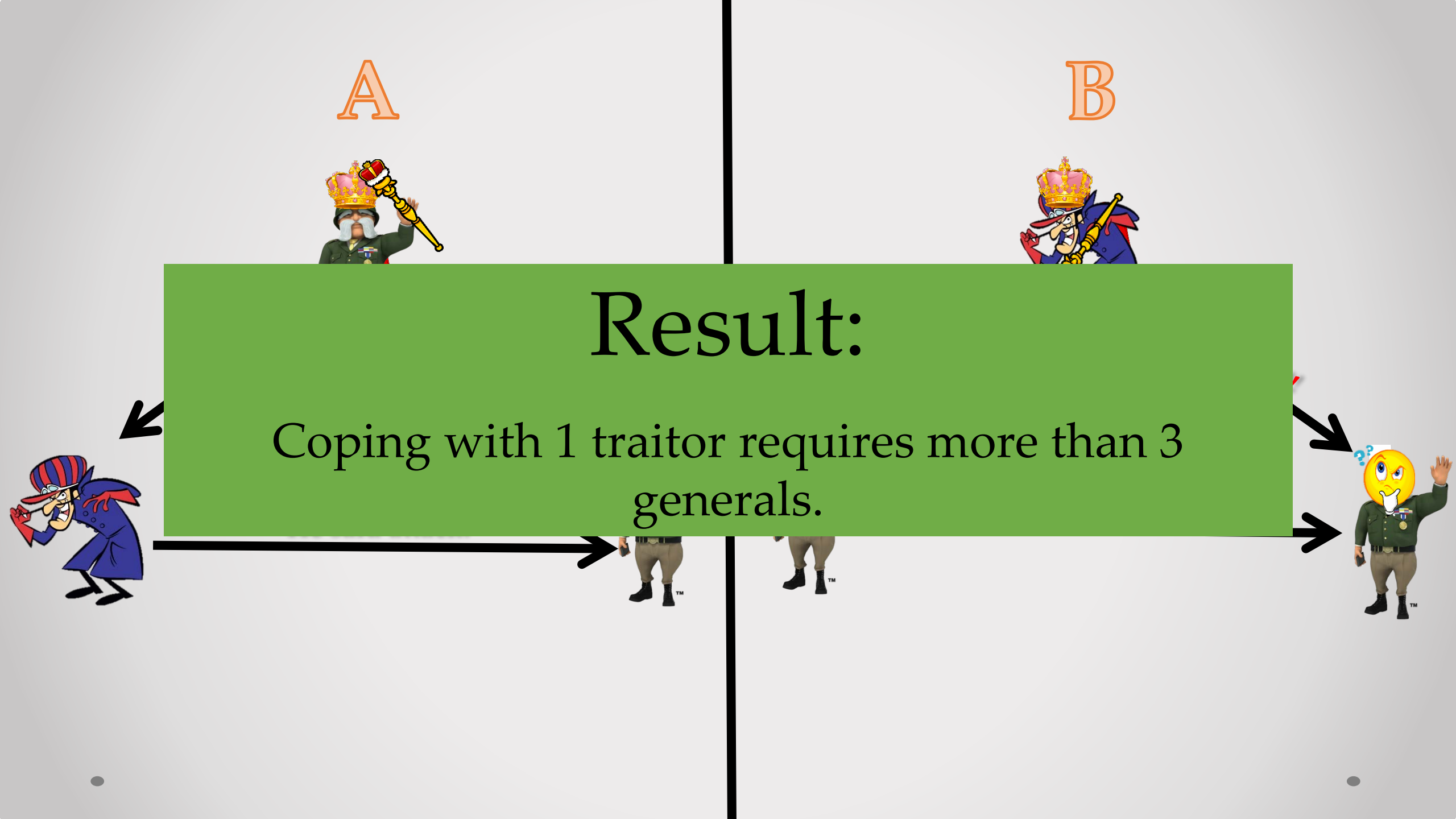


A


Retreat!
Retreat!
He said attack!

B


Attack!
Retreat!
He said attack!

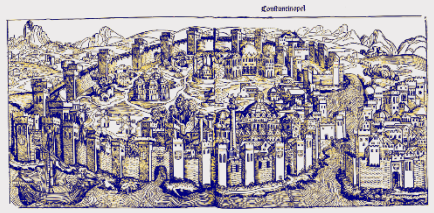


Result:

Coping with 1 traitor requires more than 3 generals.

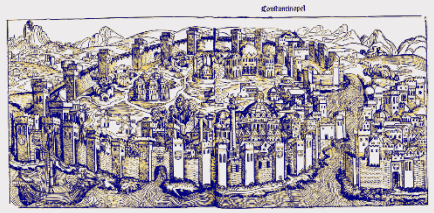
>3 Generals

Attack, or
Retreat?



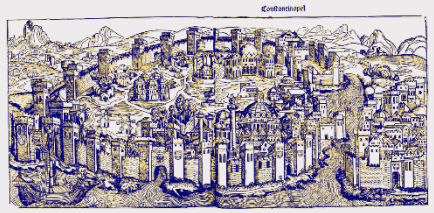
>3 Generals

Attack, or
Retreat?



>3 Generals

Attack, or
Retreat?



>3 Generals

Attack, or
Retreat?

>3 Generals

Attack, or
Retreat?

Result:

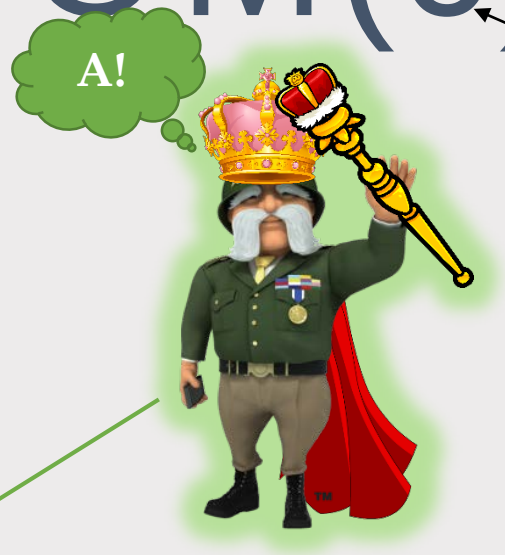
No solution with fewer than $3m + 1$ generals can cope with m traitors.

OM(0)

Number of traitors assumed

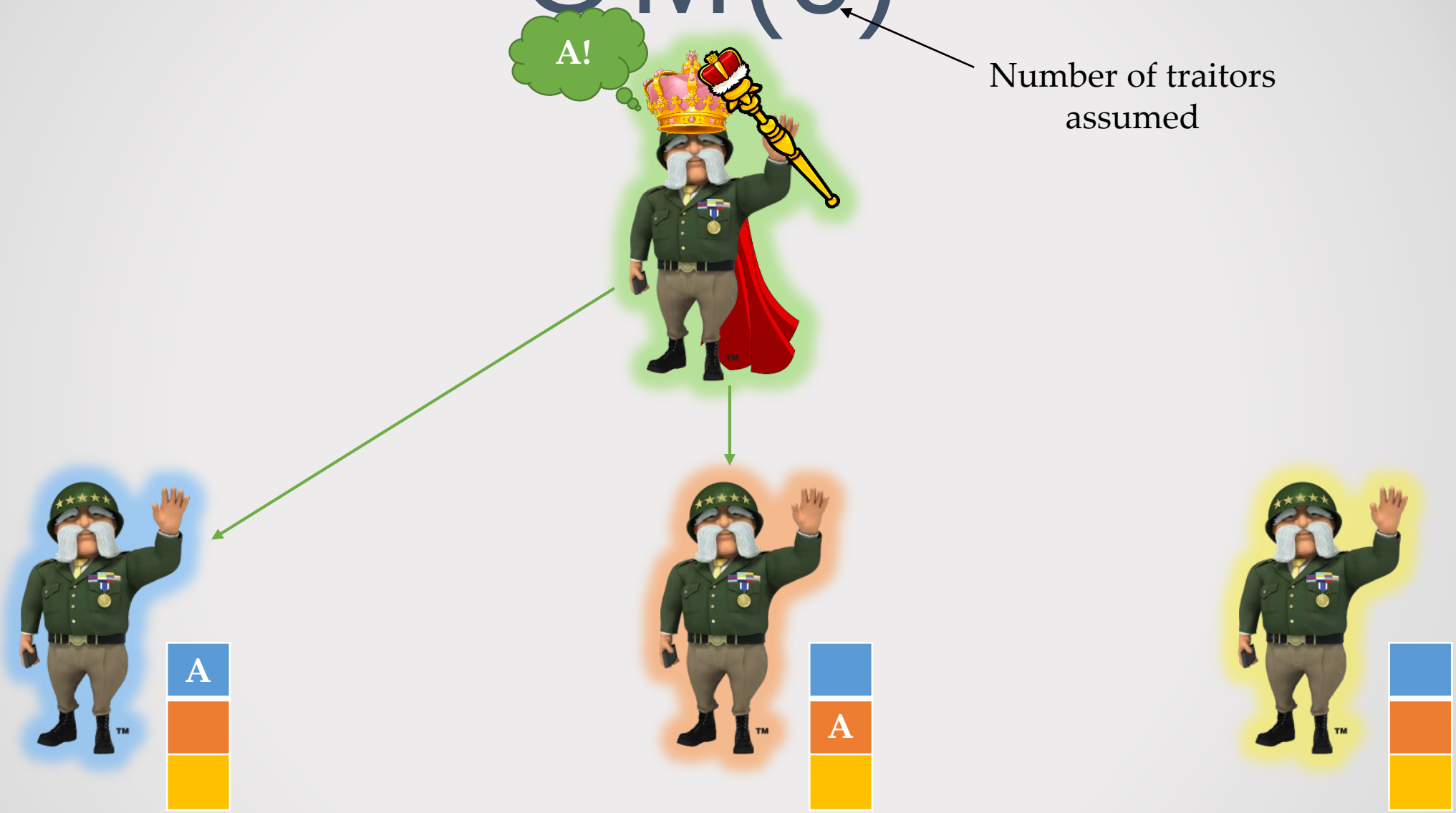
OM(0)

Number of traitors assumed

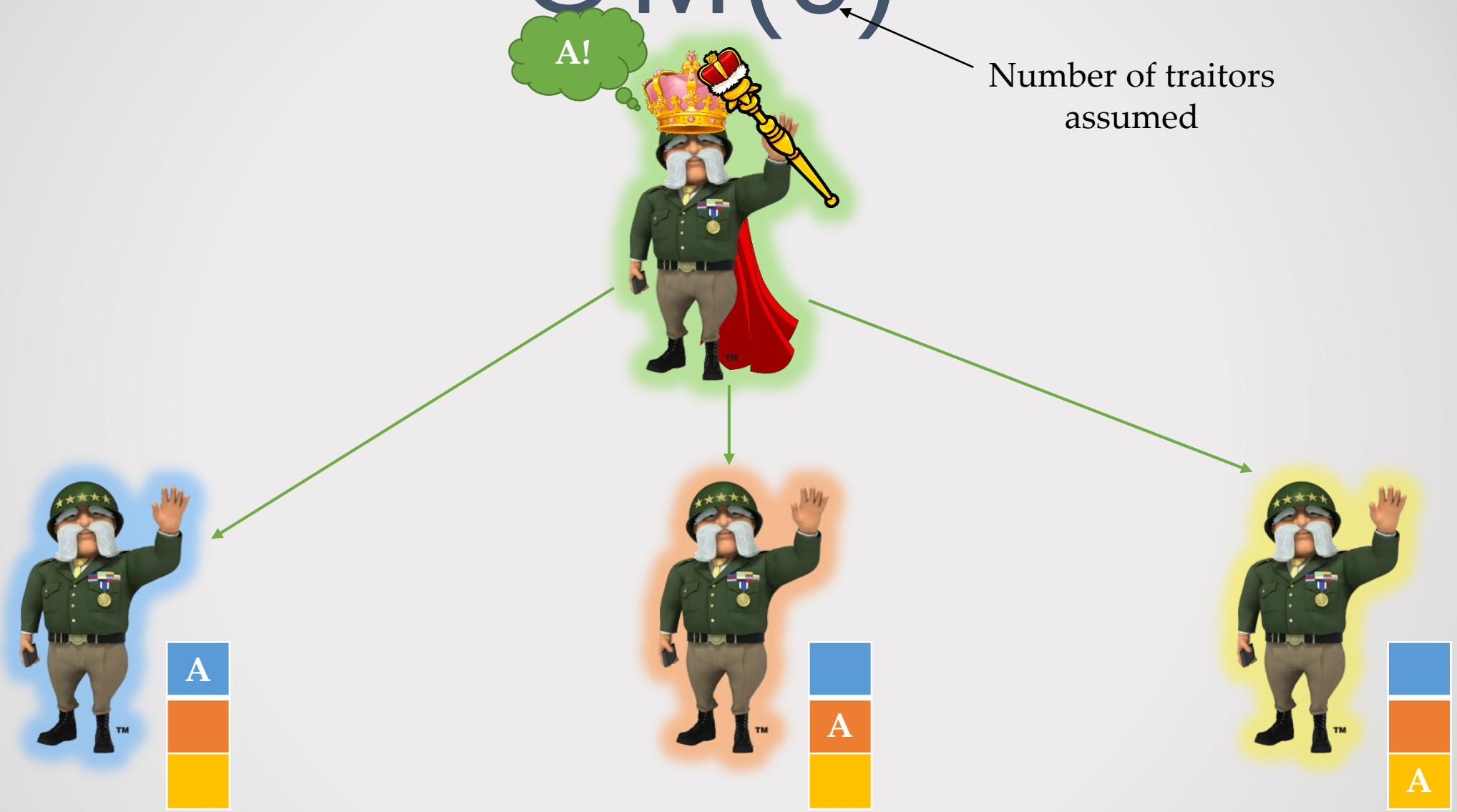


OM(0)

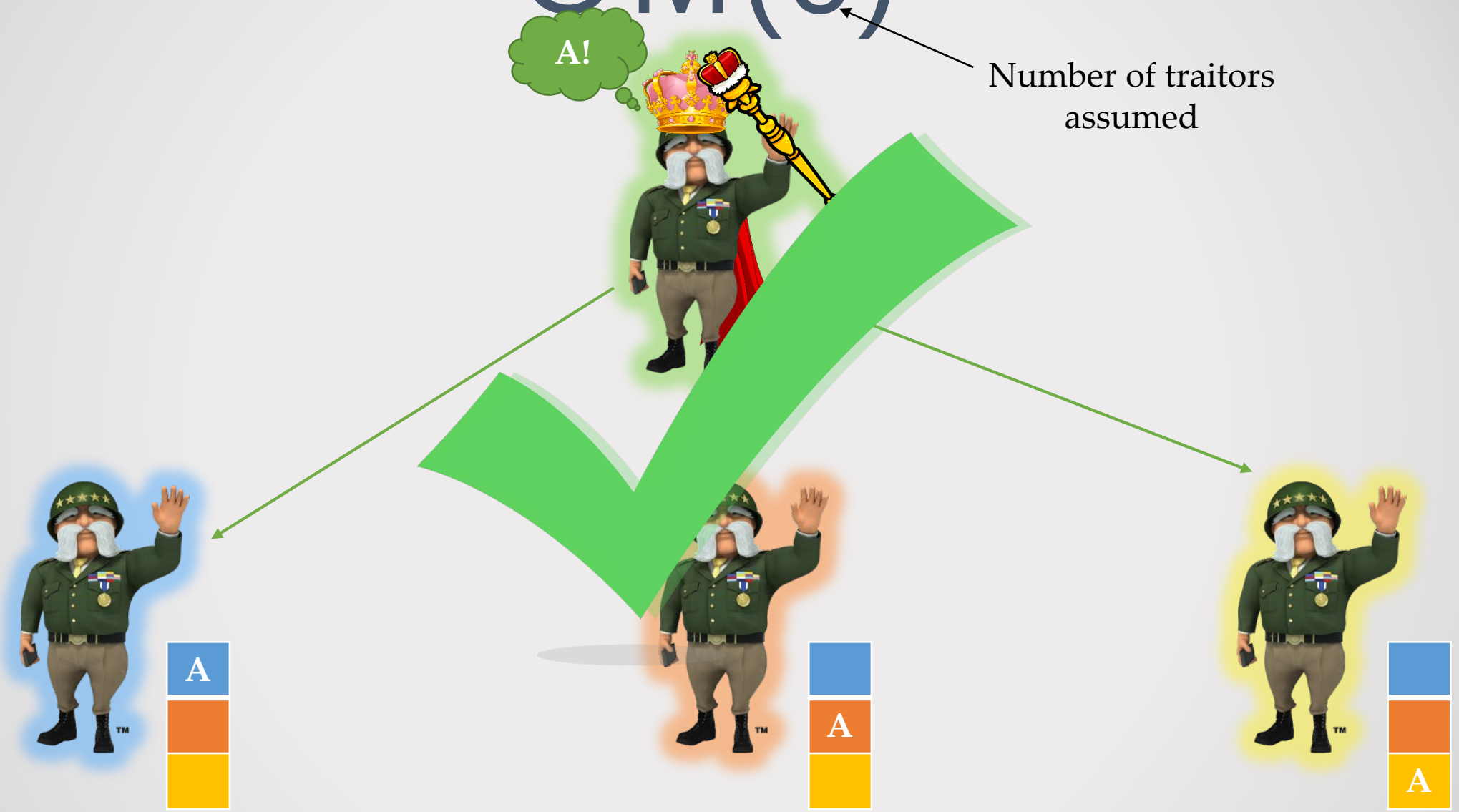
Number of traitors assumed



OM(0)

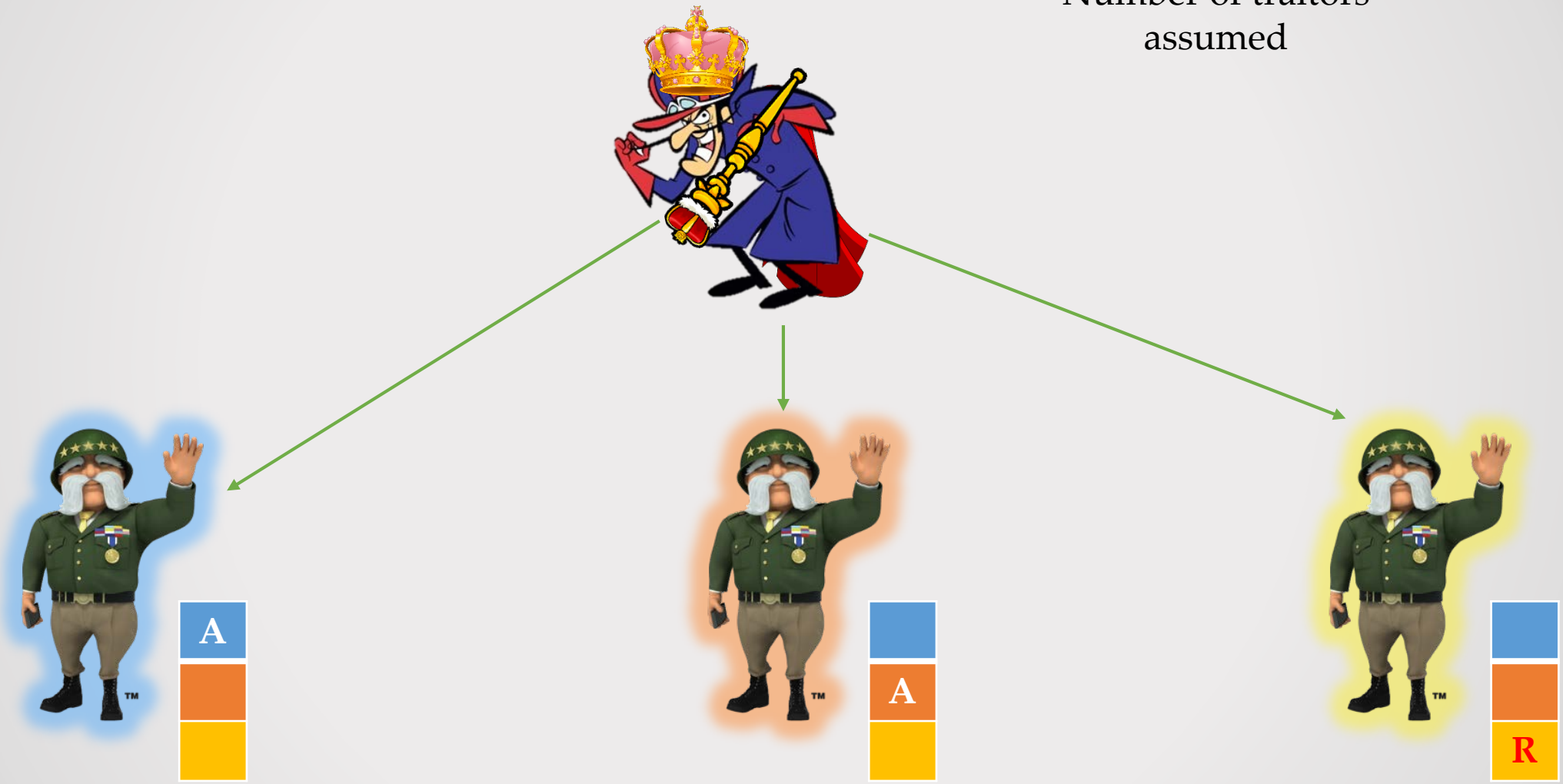


OM(0)



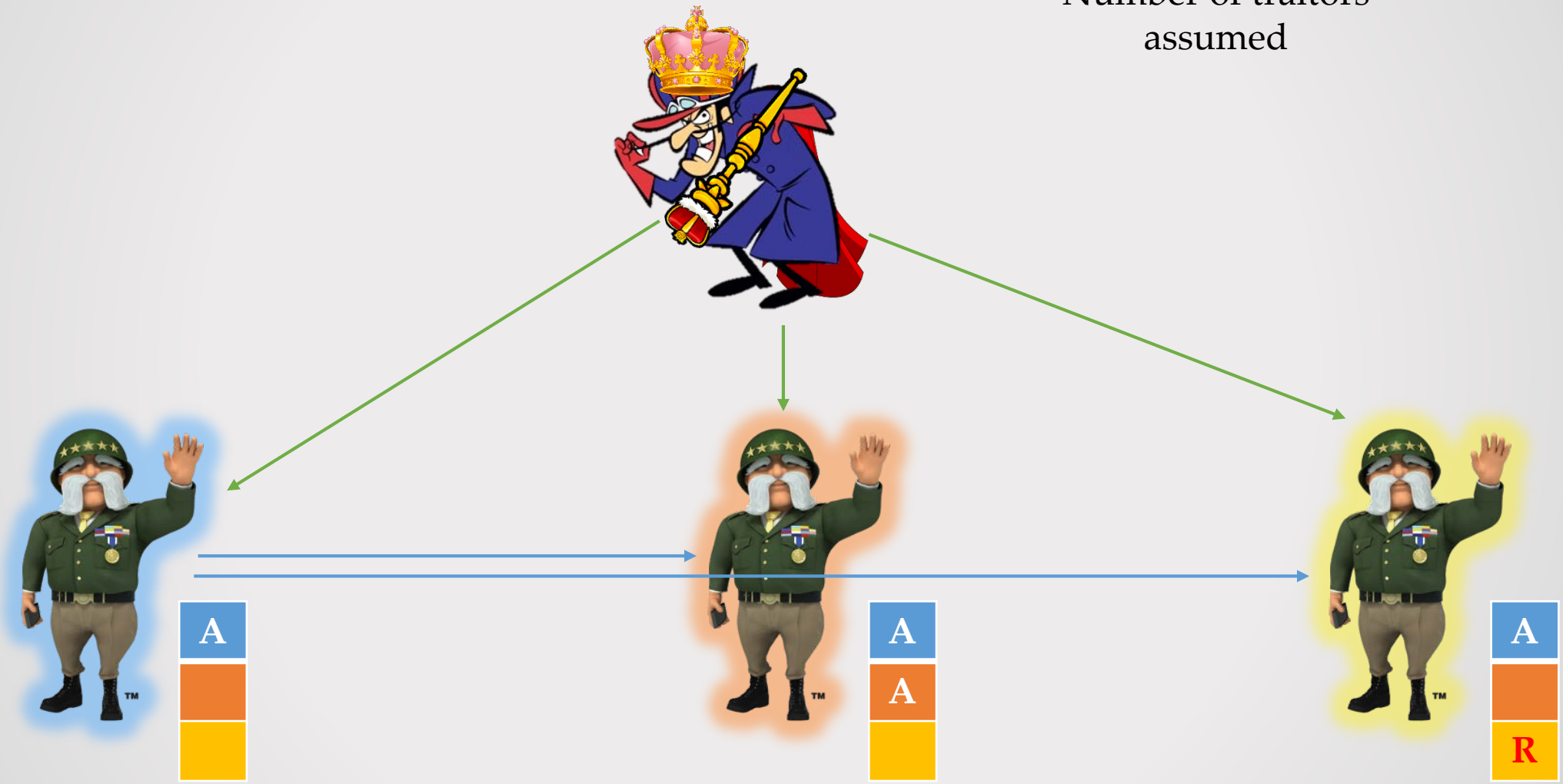
OM(1)

Number of traitors assumed



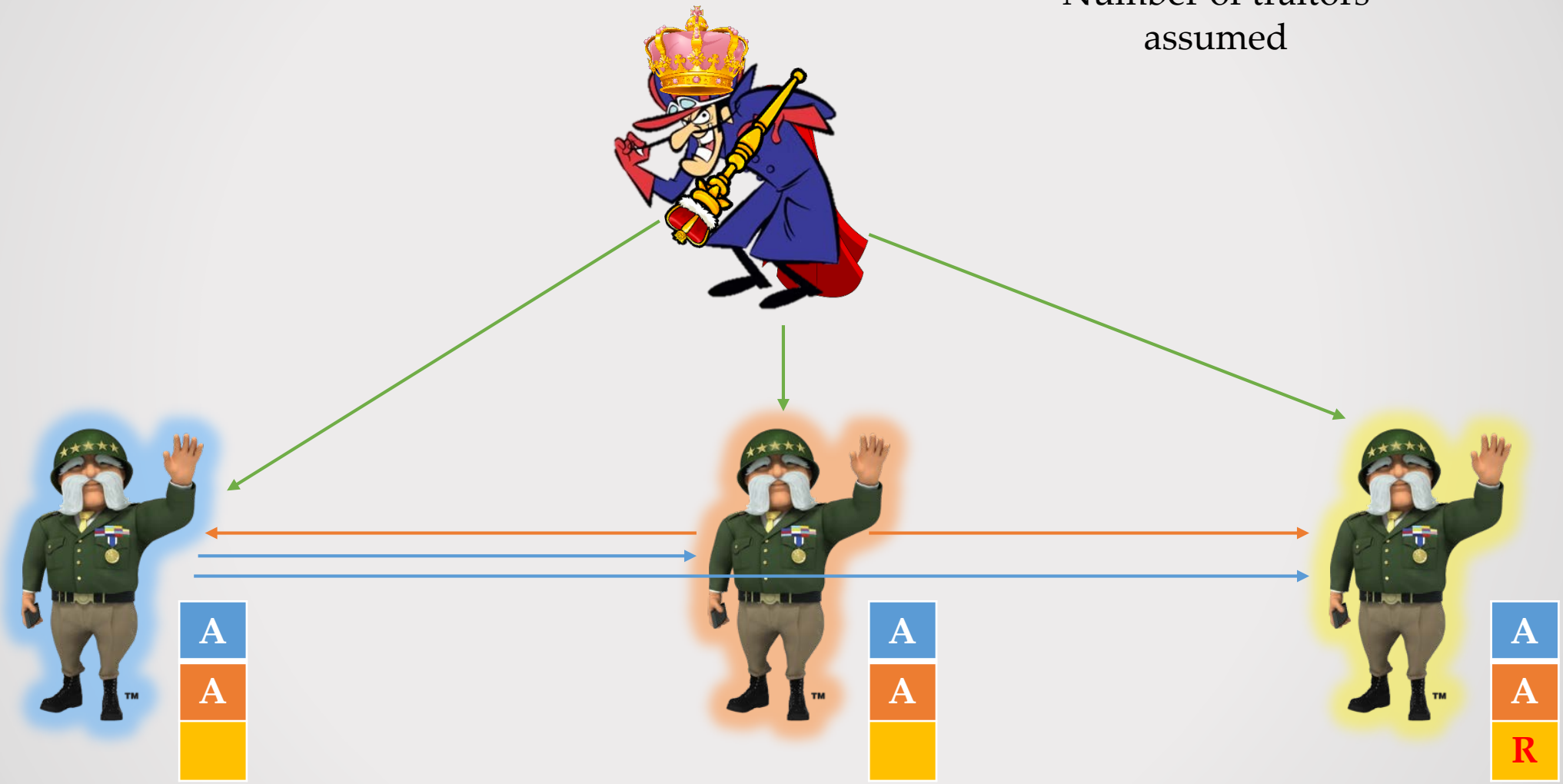
OM(1)

Number of traitors assumed



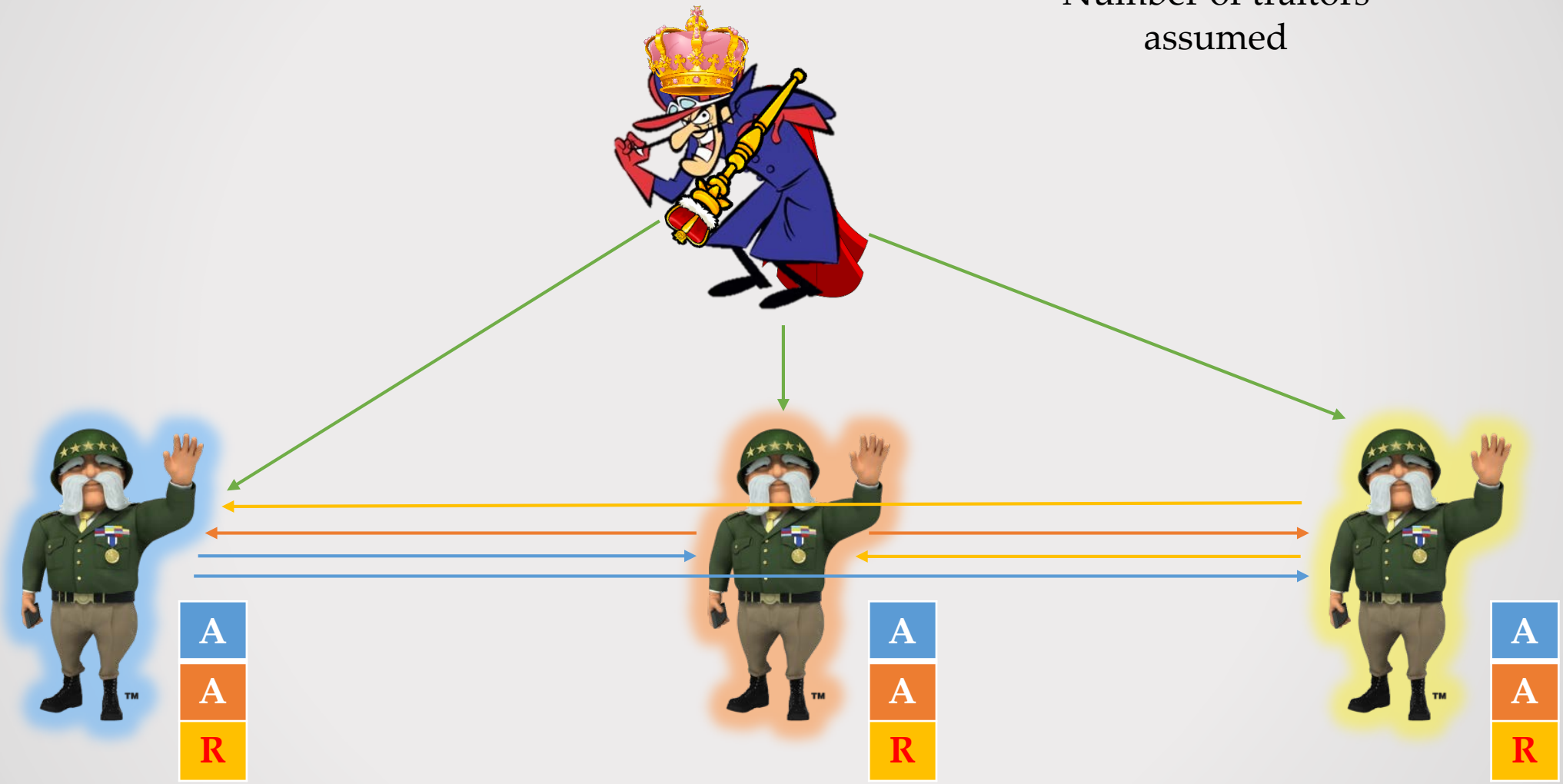
OM(1)

Number of traitors assumed

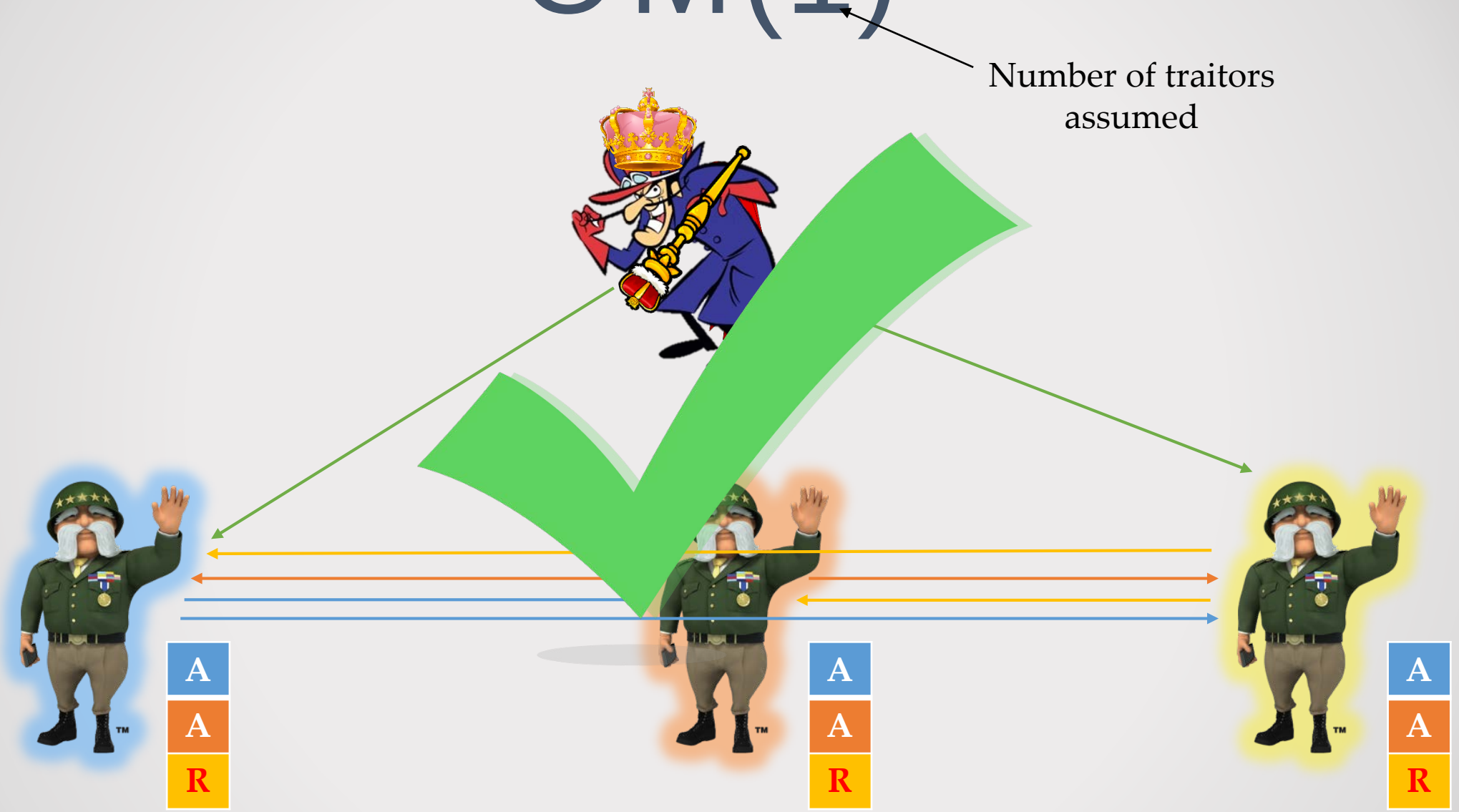


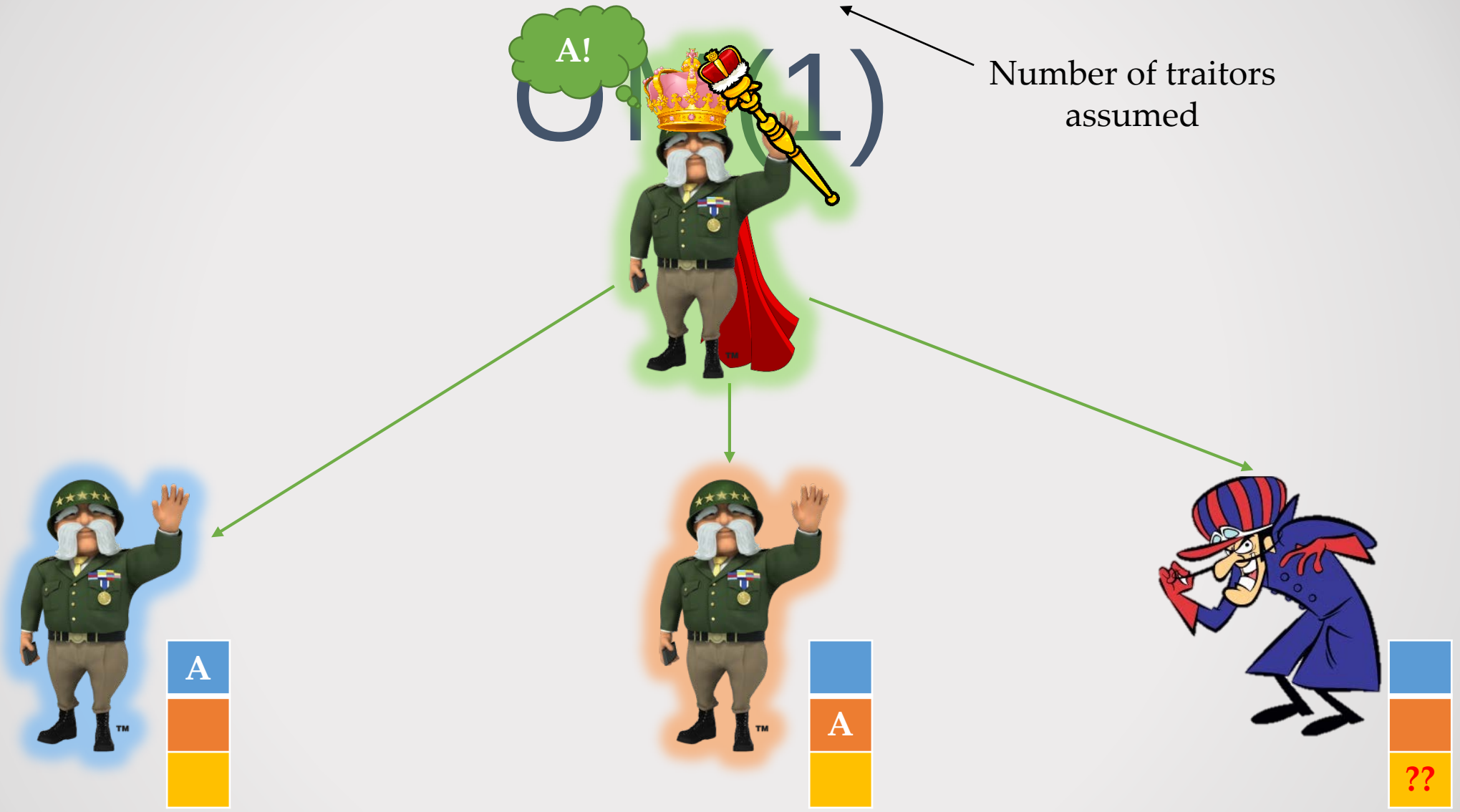
OM(1)

Number of traitors assumed



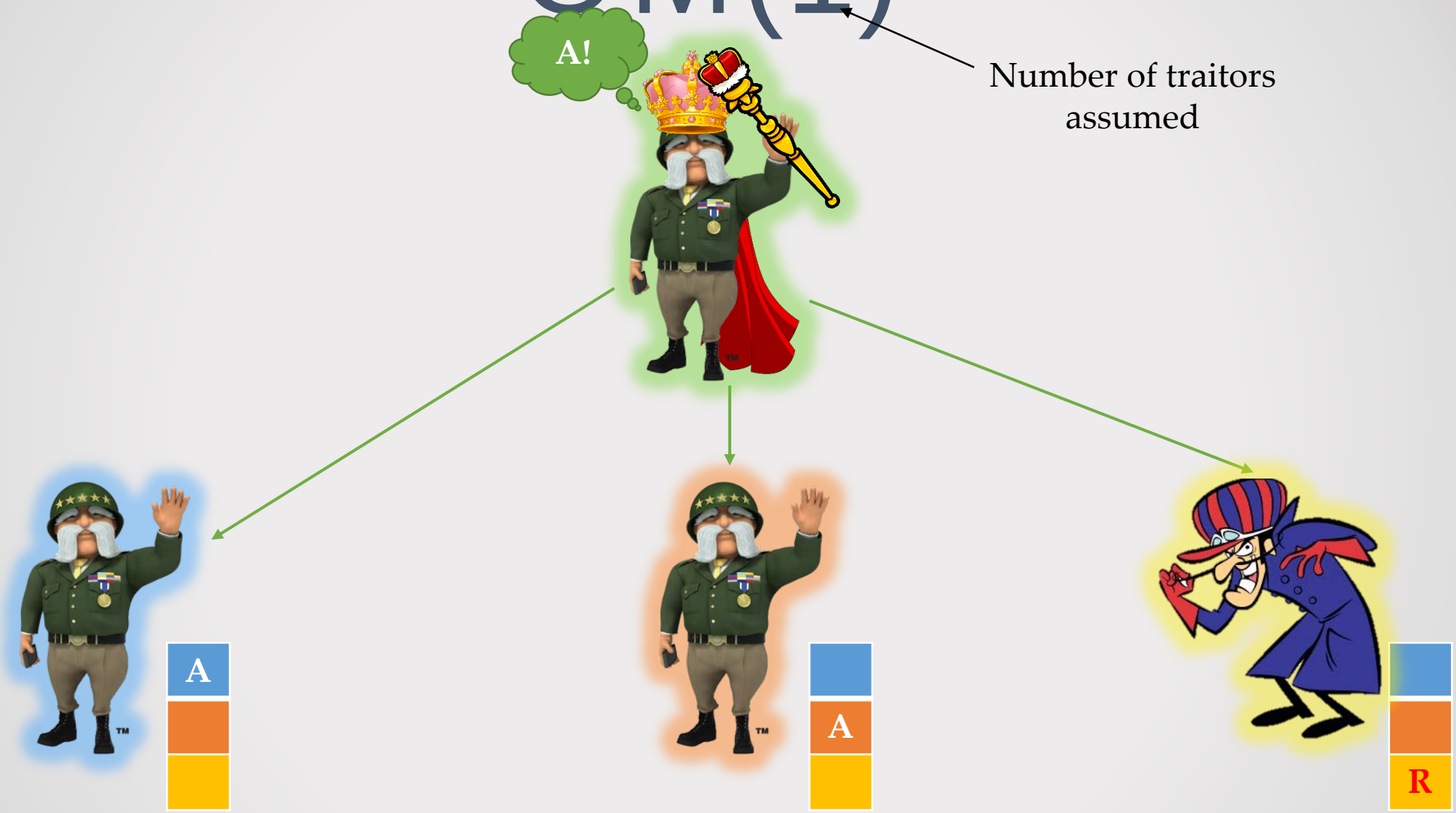
OM(1)



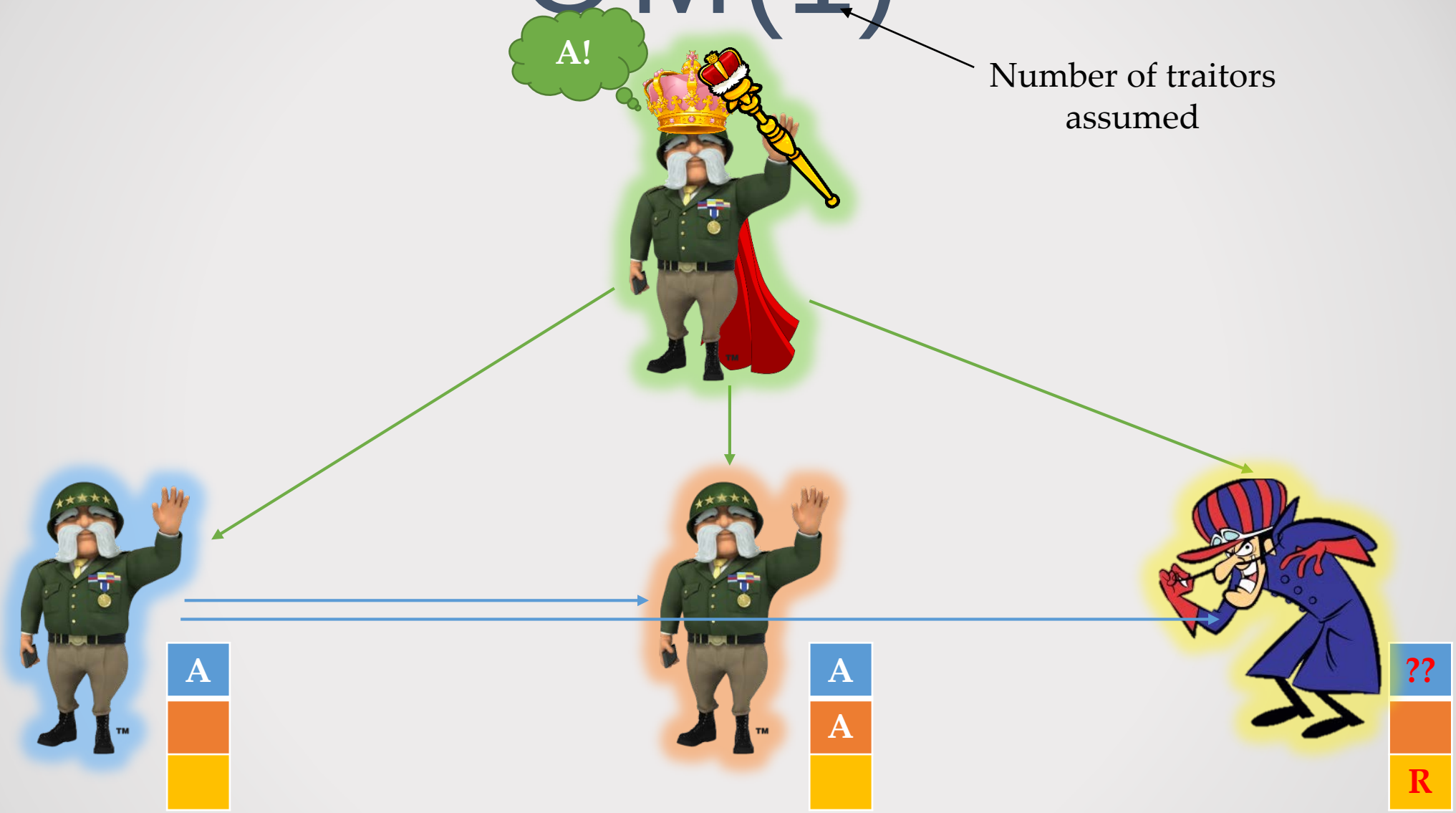

OM(1)
A!
Number of traitors assumed
A
A
??

OM(1)

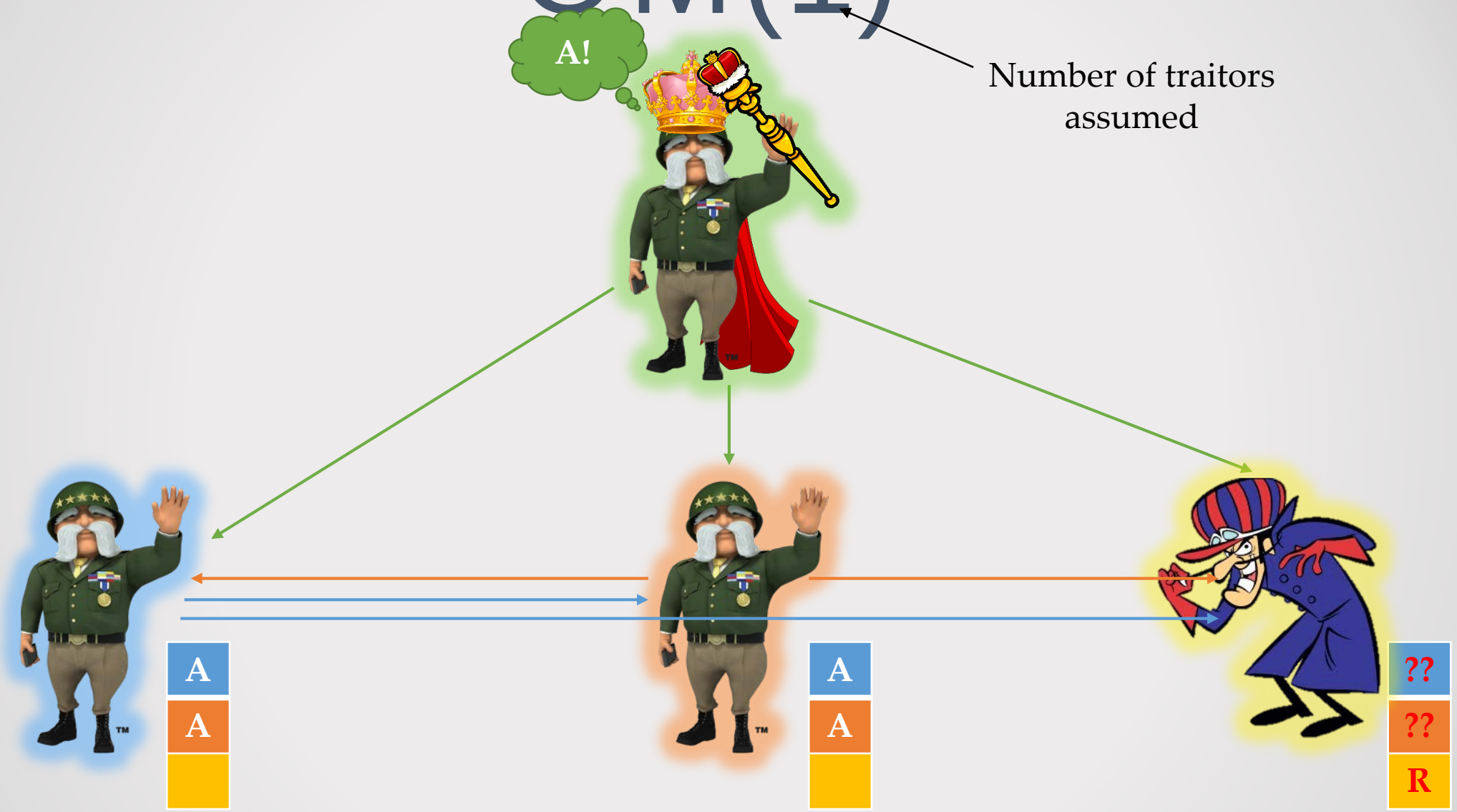
Number of traitors assumed



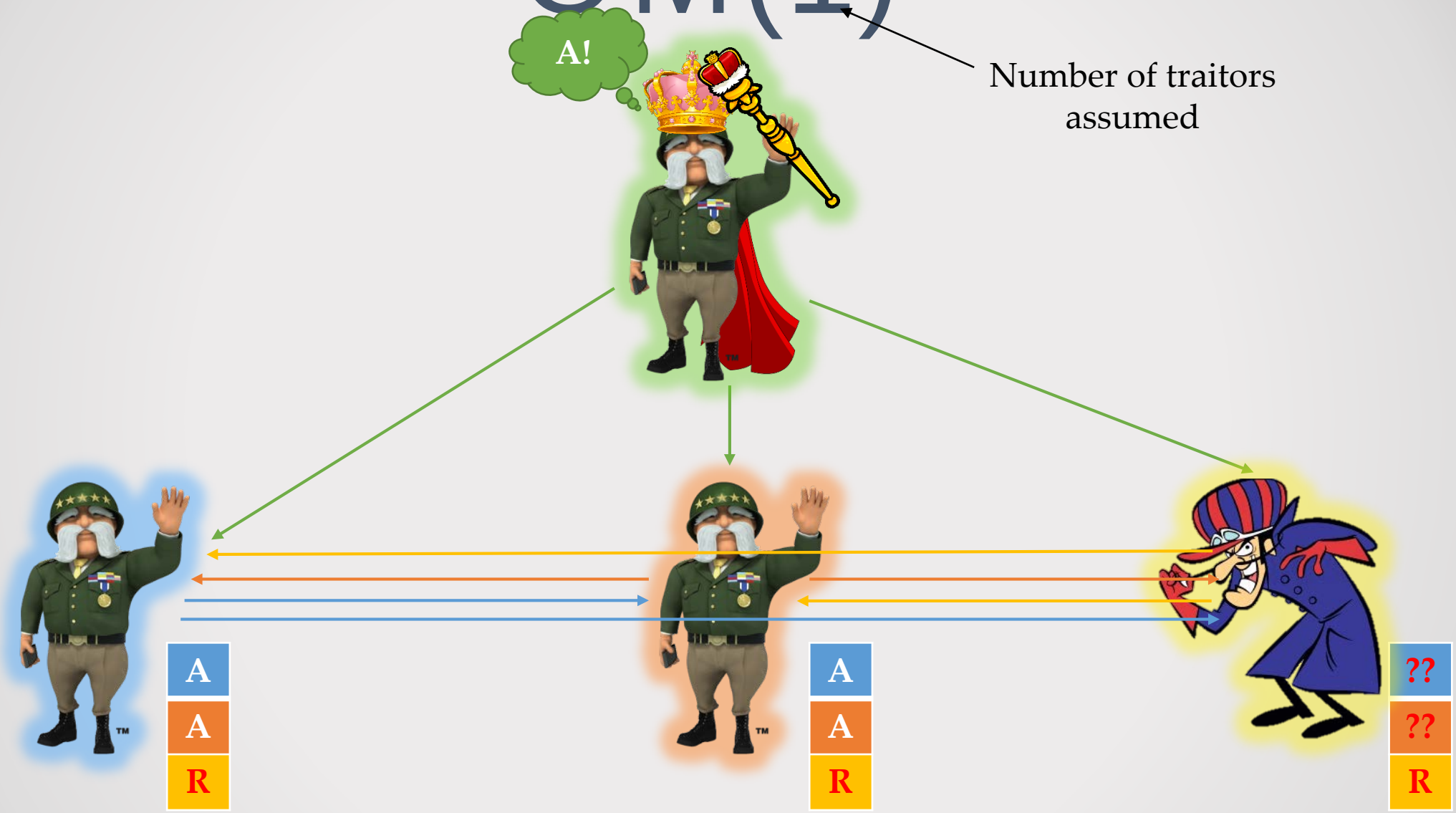
OM(1)



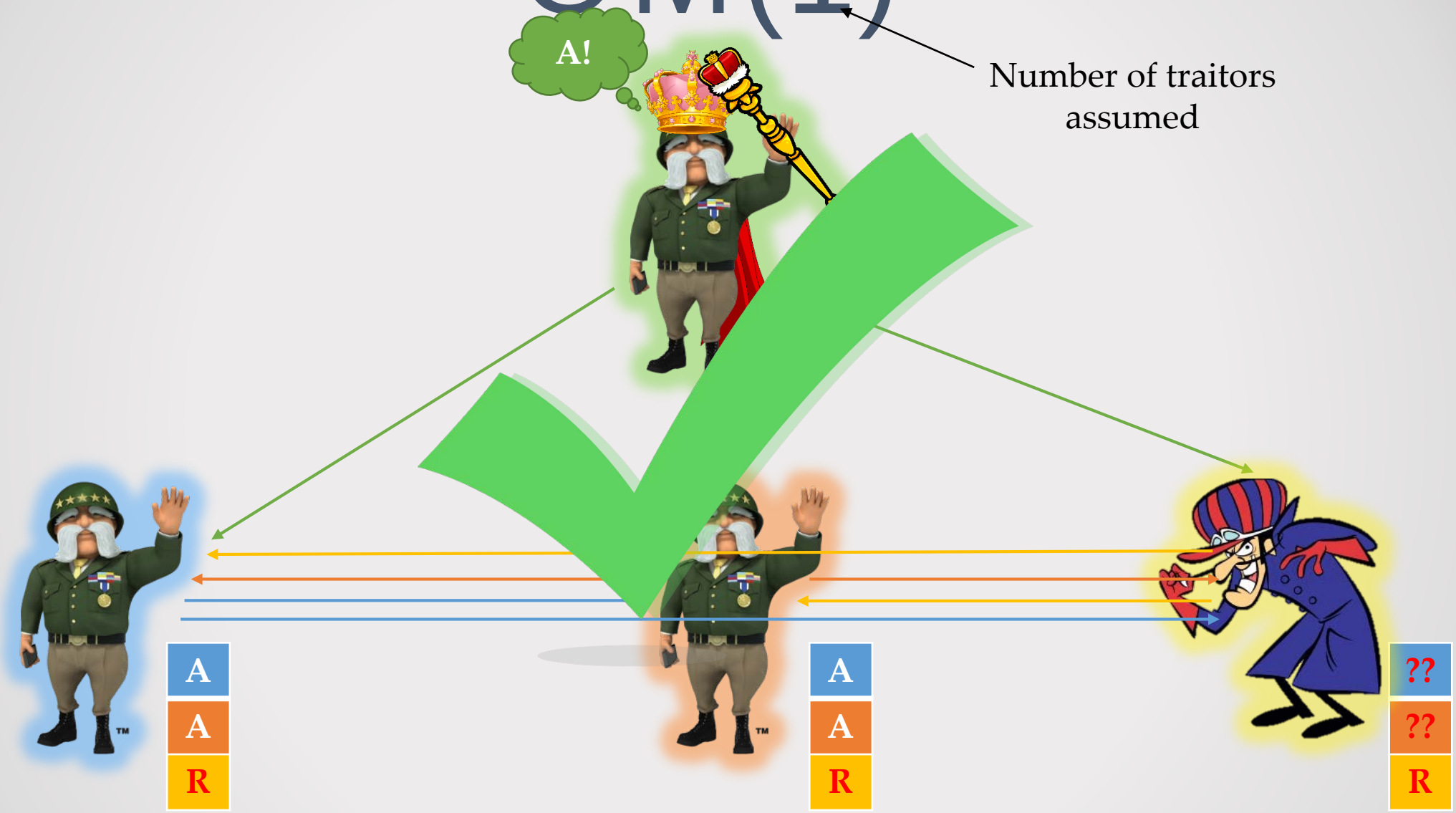
OM(1)



OM(1)

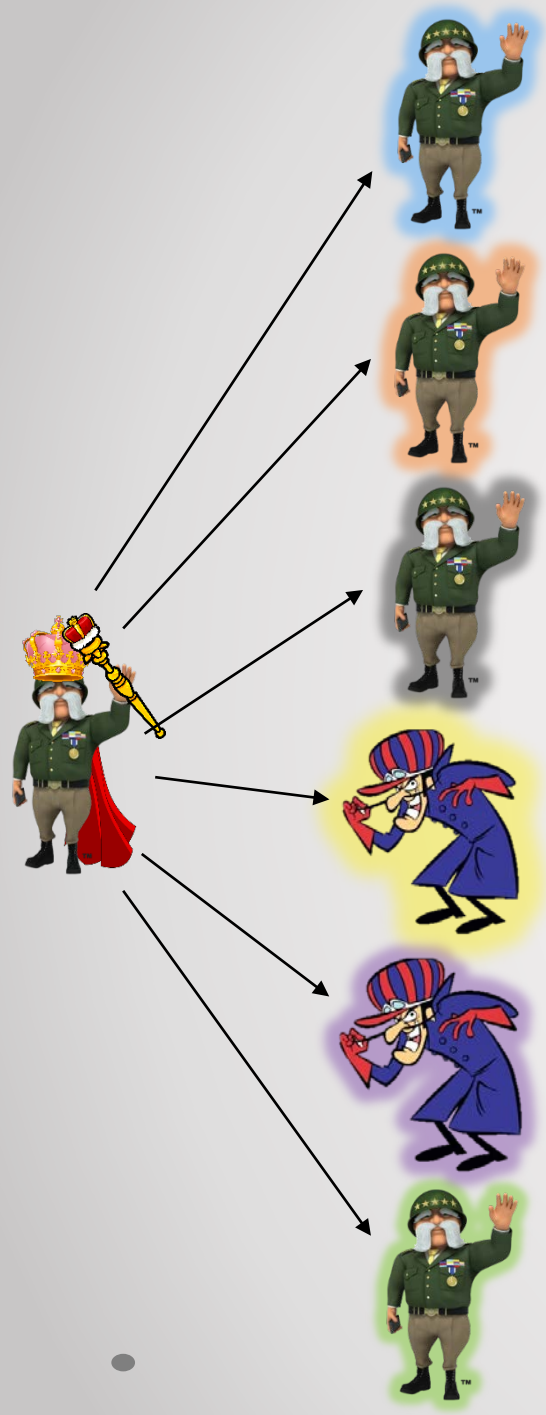


OM(1)



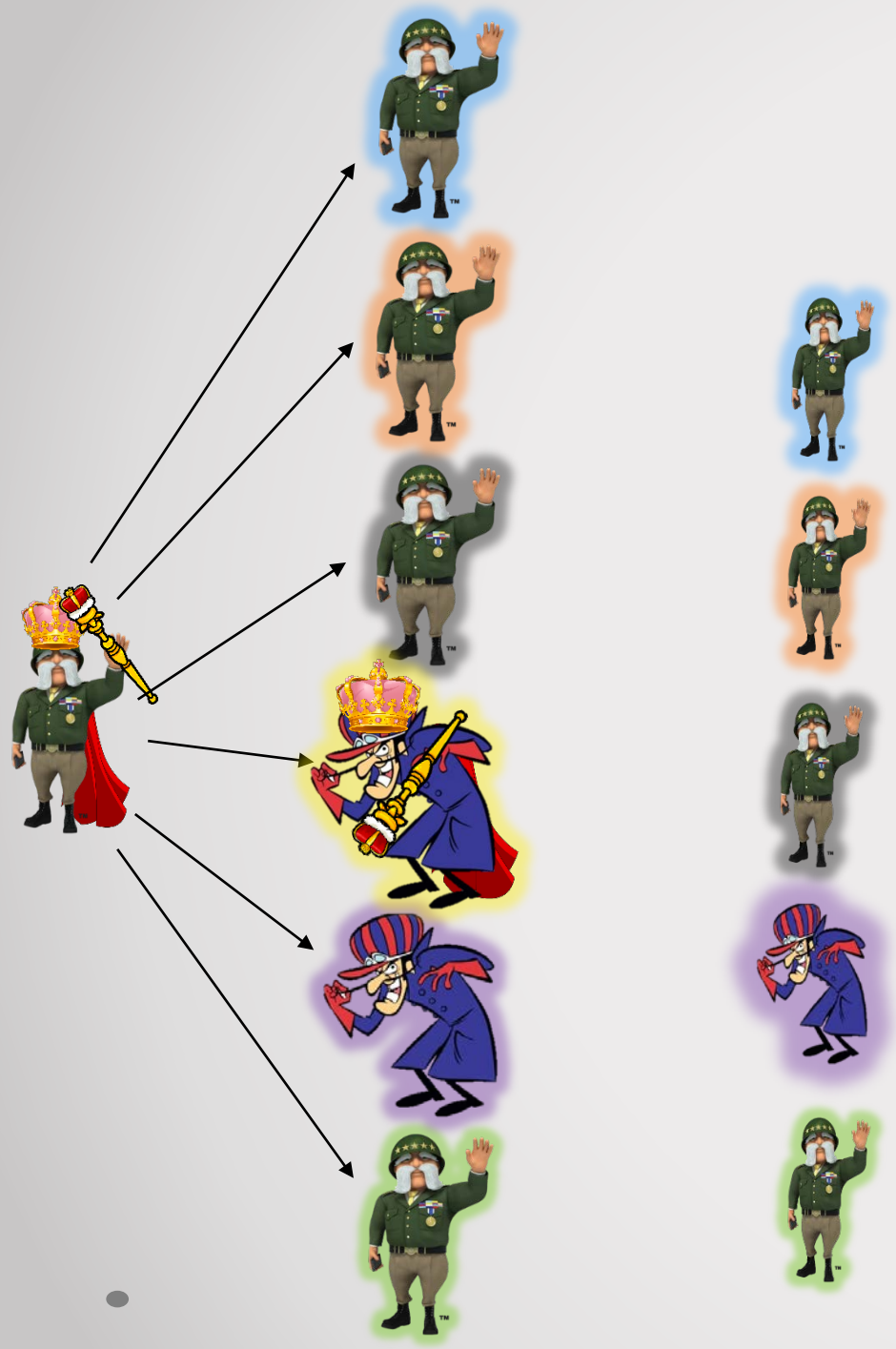
OM(2)

Number of traitors assumed



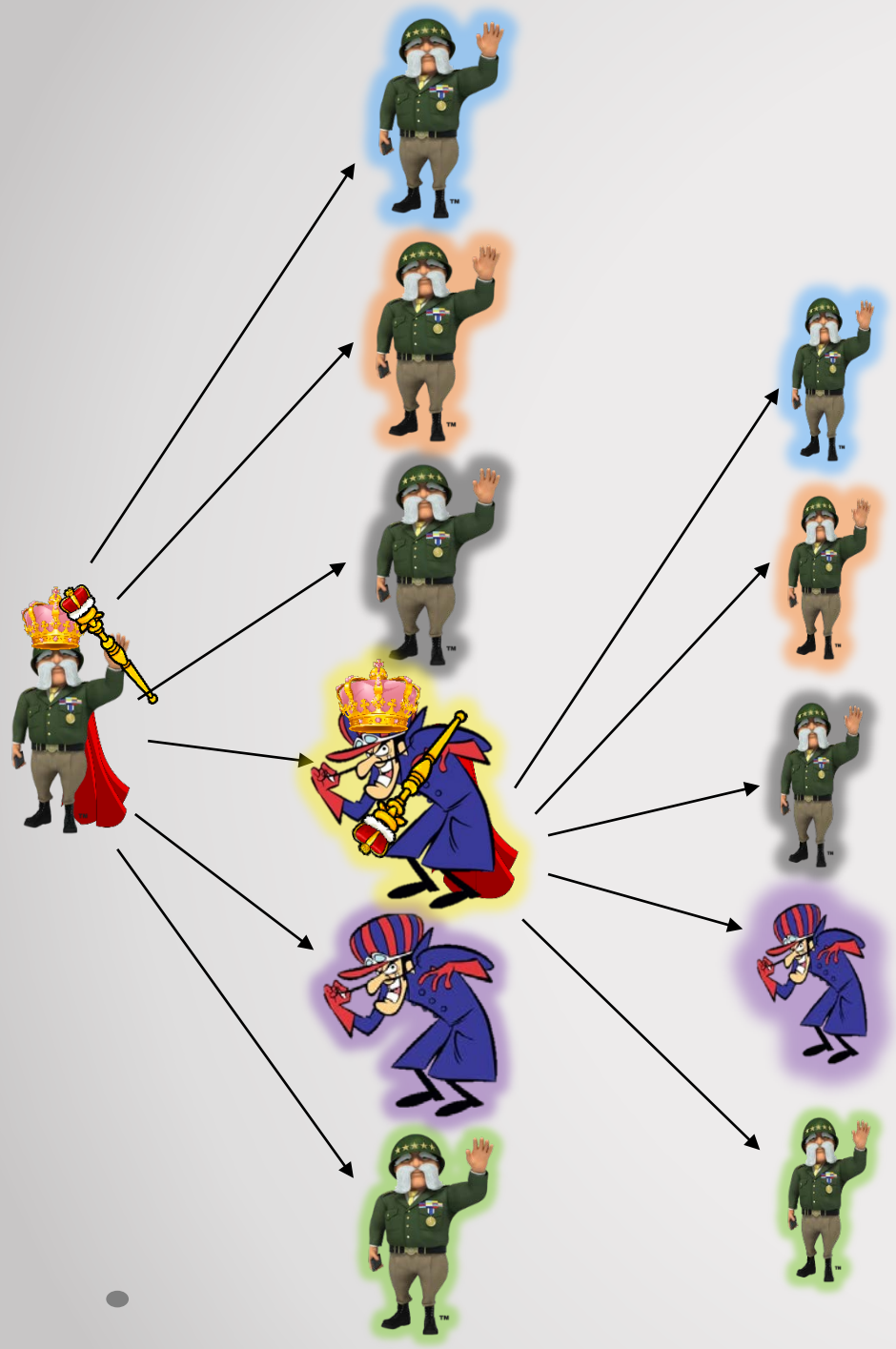
OM(2)

Number of traitors assumed



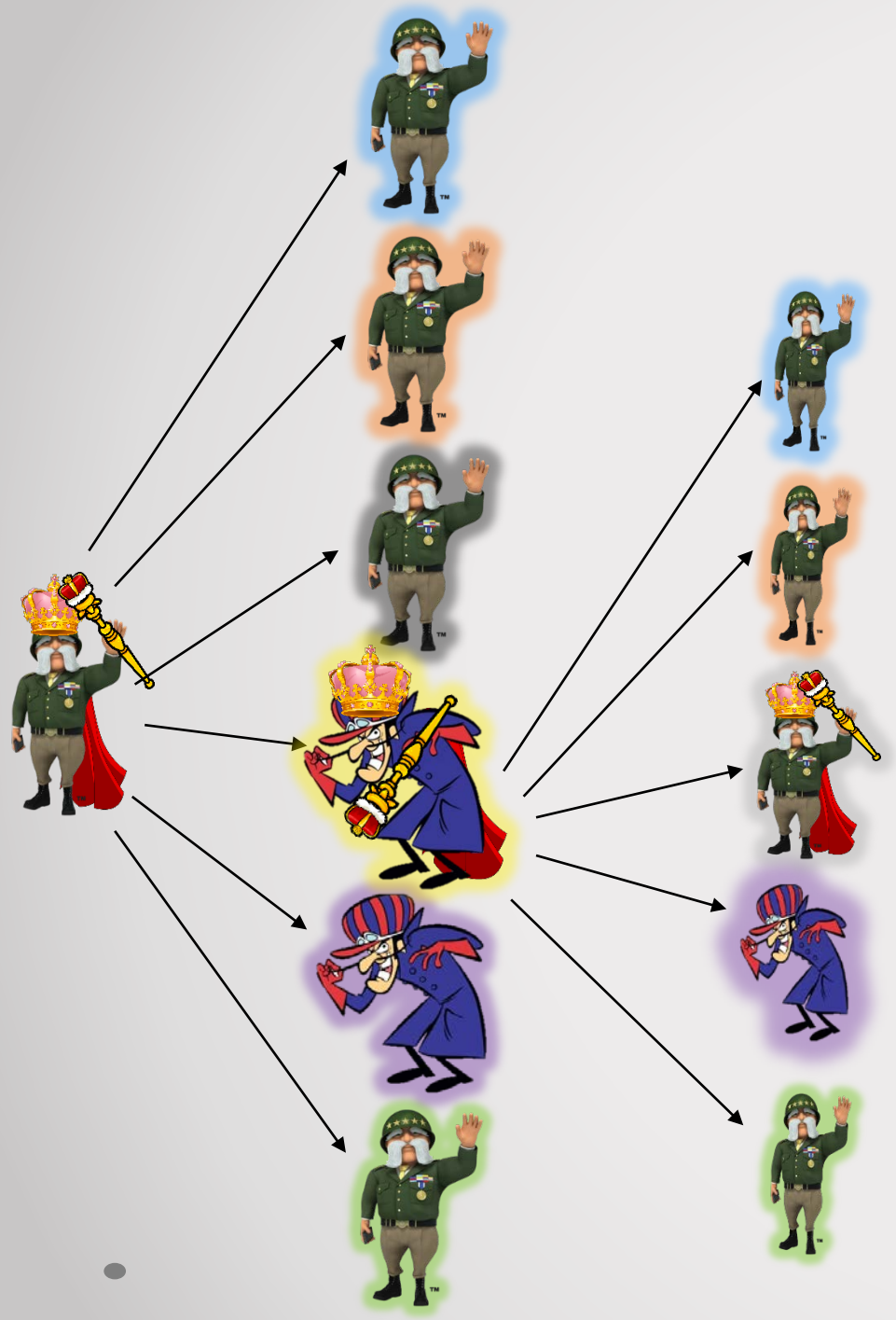
OM(2)

Number of traitors assumed



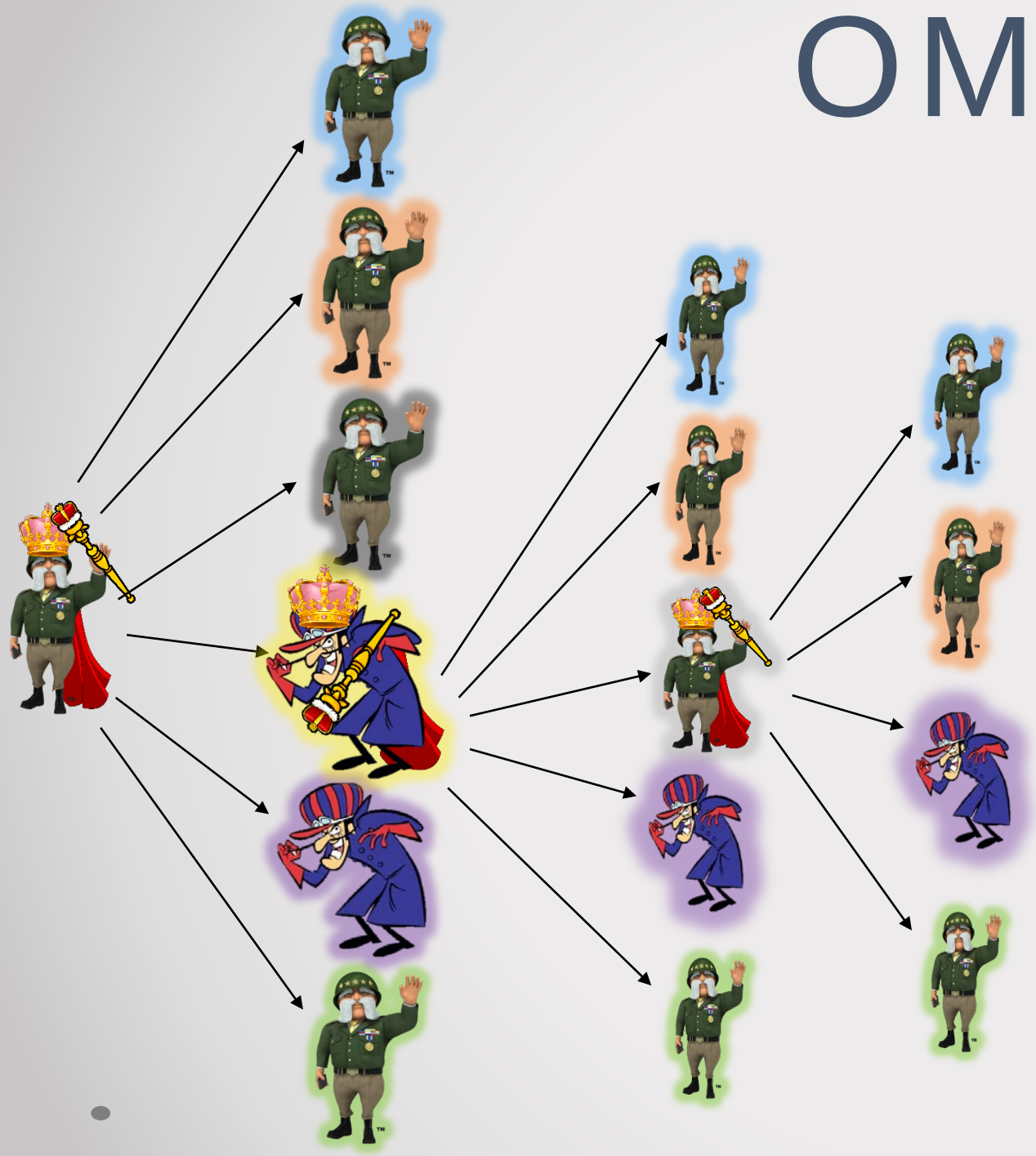
OM(2)

Number of traitors assumed



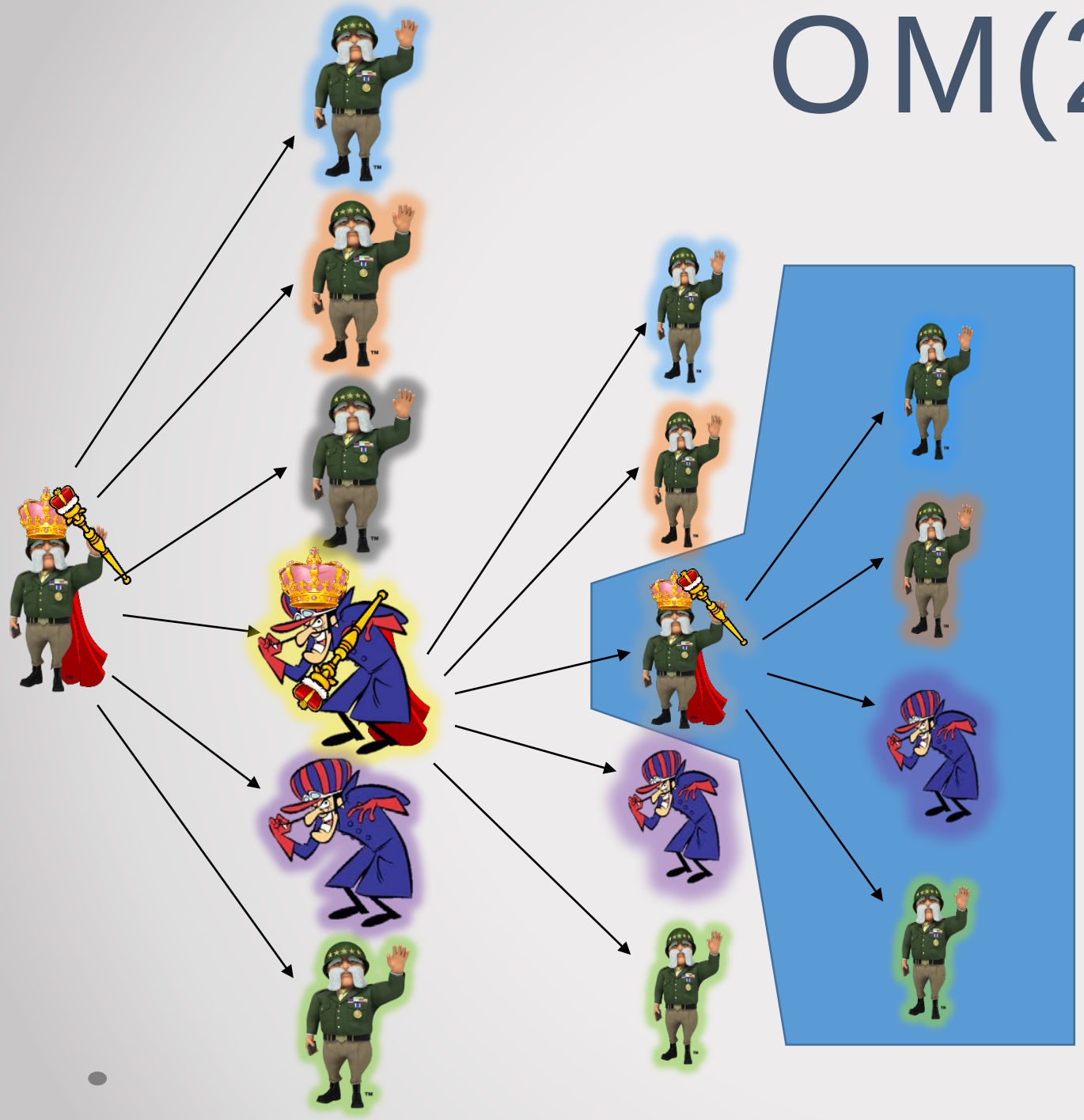
OM(2)

Number of traitors assumed



OM(2)

Number of traitors assumed



What are some drawbacks of this?

Assumptions (p 387)

- All messages are delivered correctly.
- All recipients can verify who sent each message.
- The absence of a message can be detected.

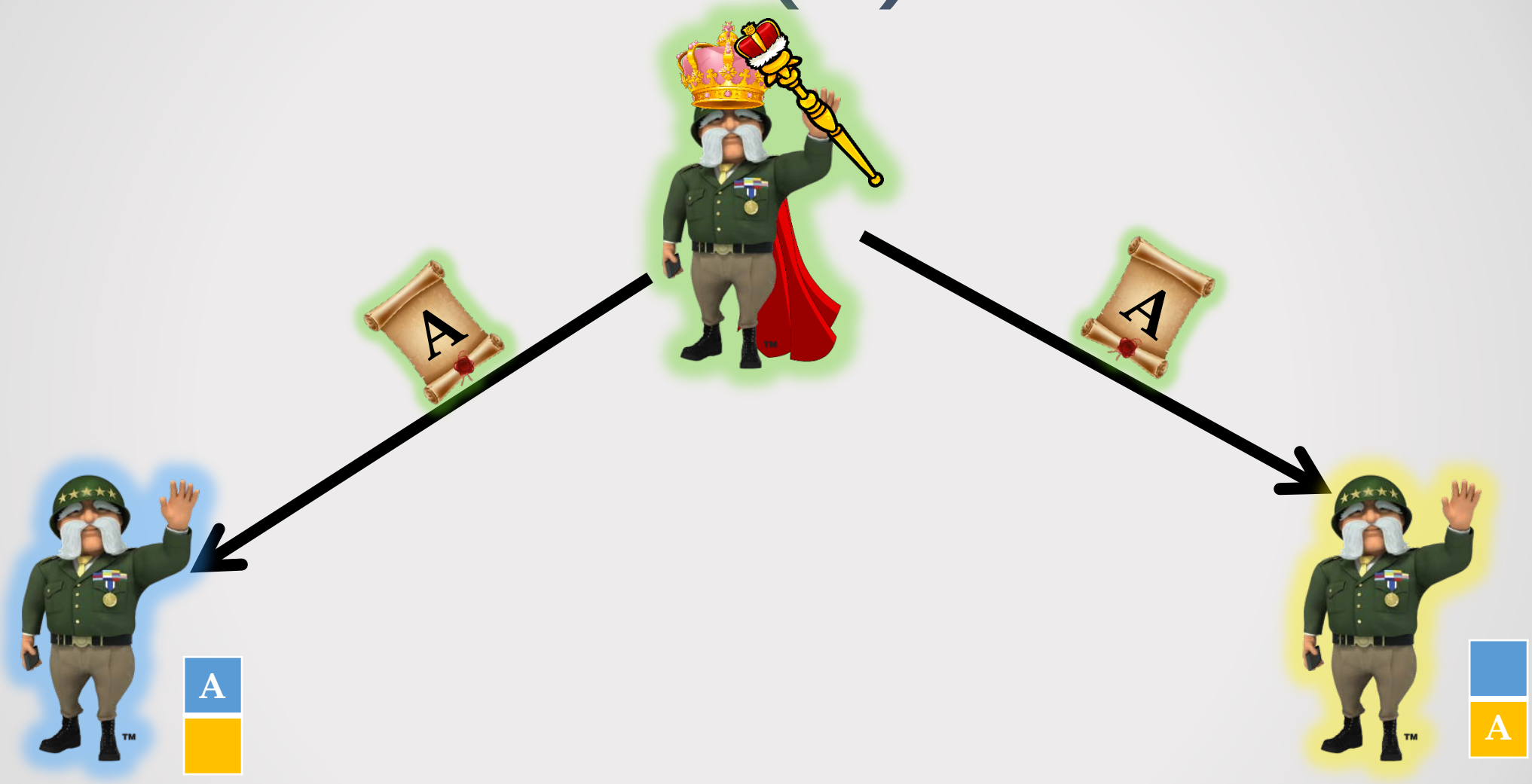
"Oral Messages"

Assumptions (p 391)

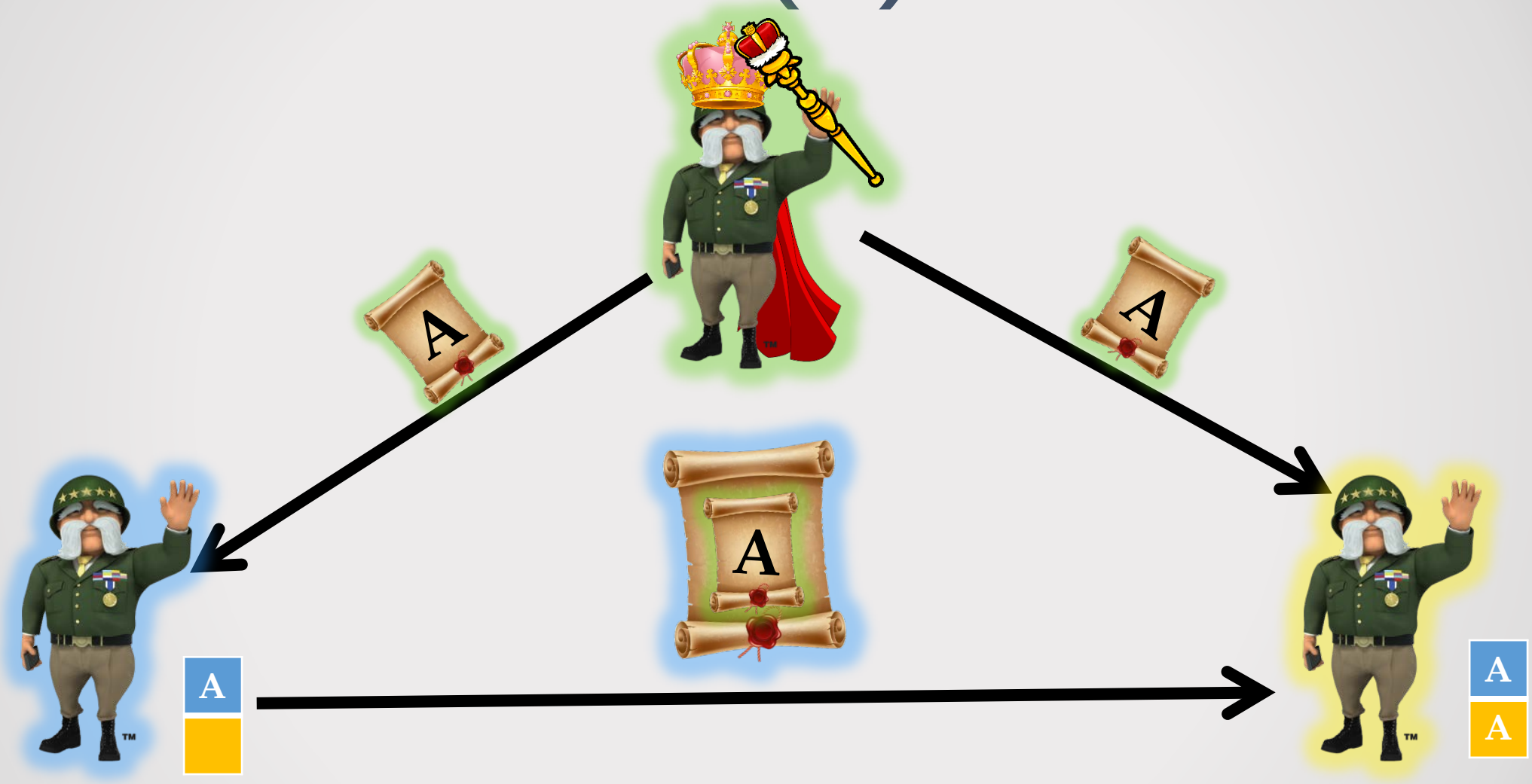
- All messages are delivered correctly.
- All recipients can verify who sent each message.
- The absence of a message can be detected.
- A loyal general's signature cannot be forged, alterations can be detected, and anyone can verify authenticity of signatures.

"Signed Messages"

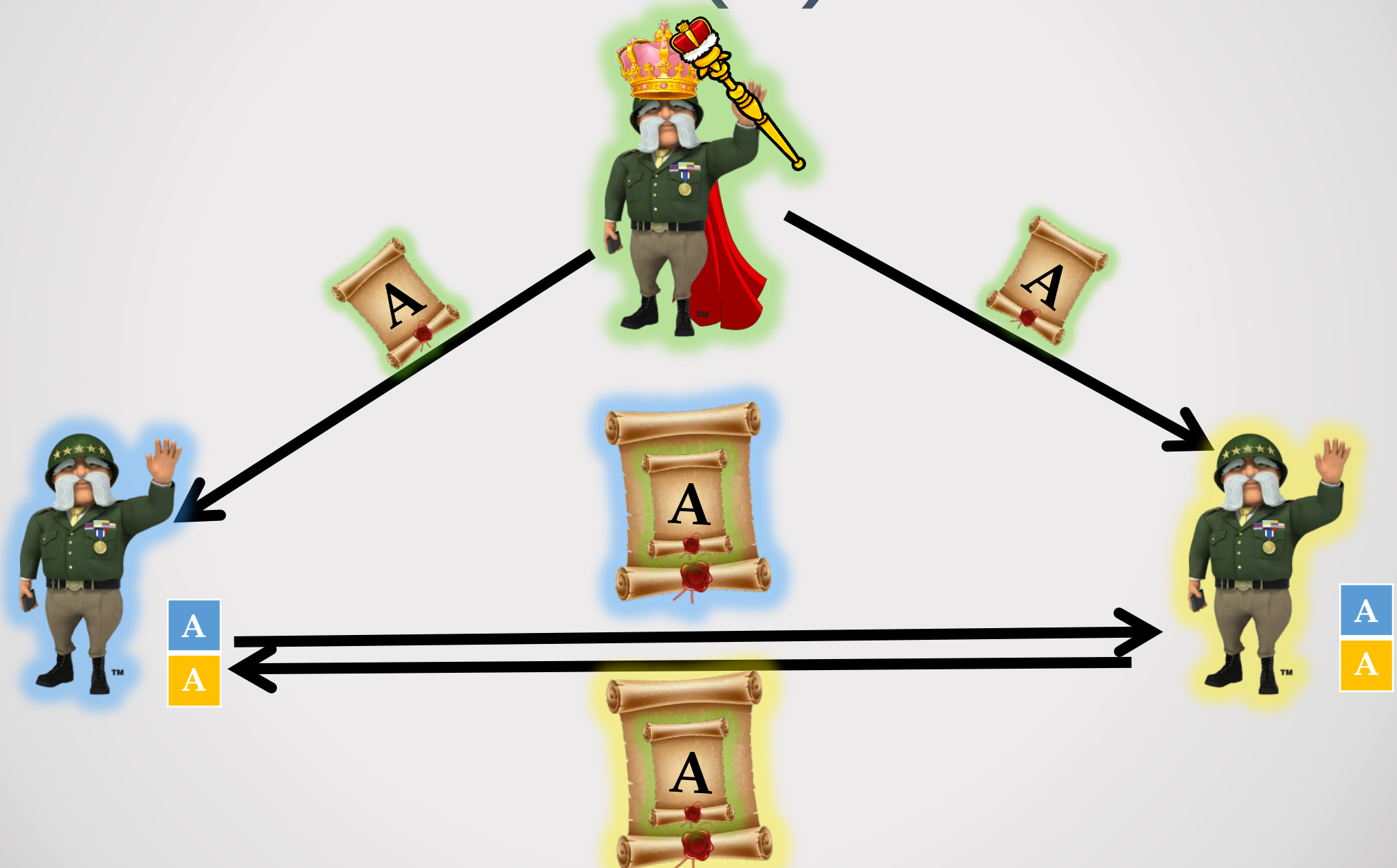
SM(0)



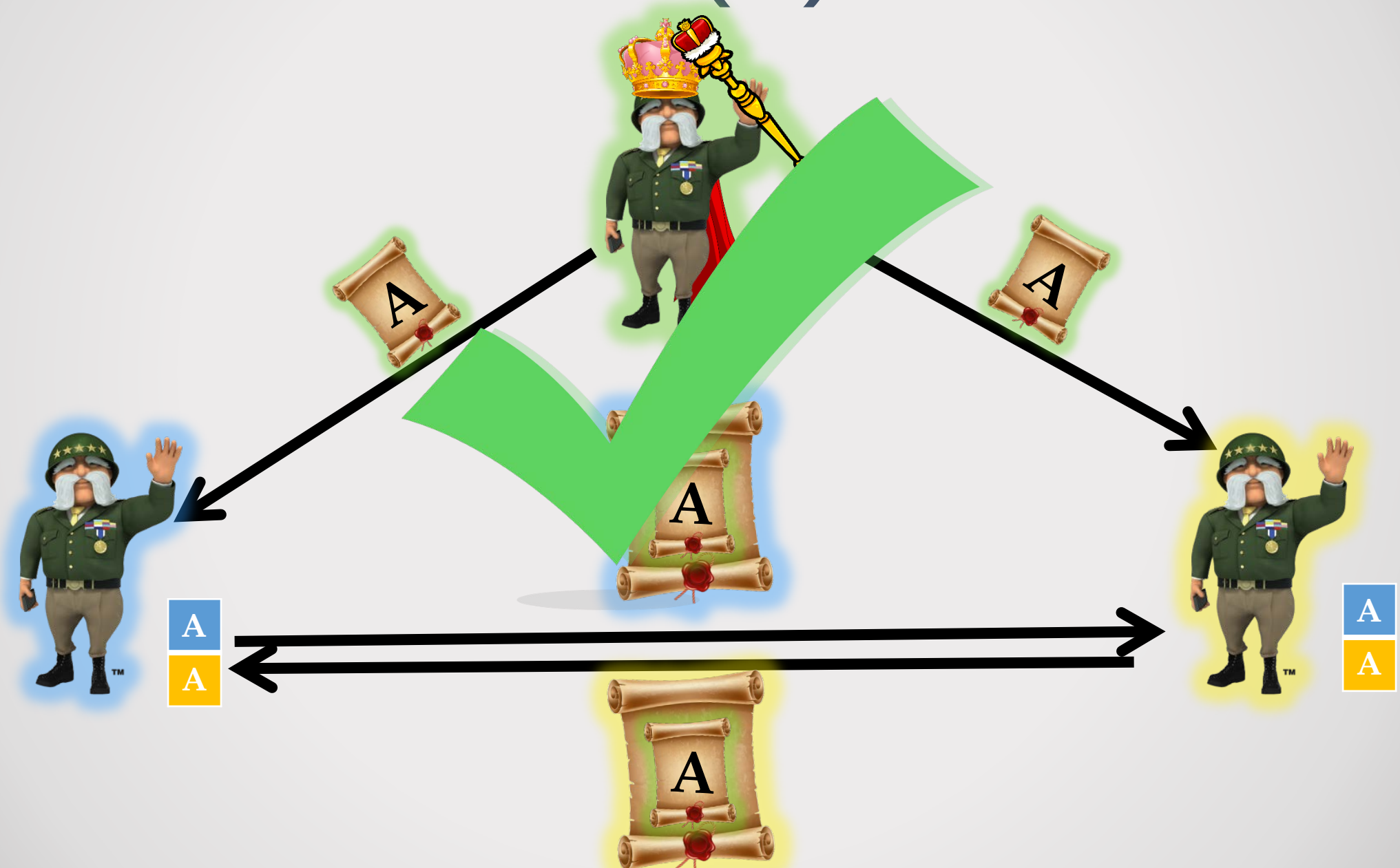
SM(0)



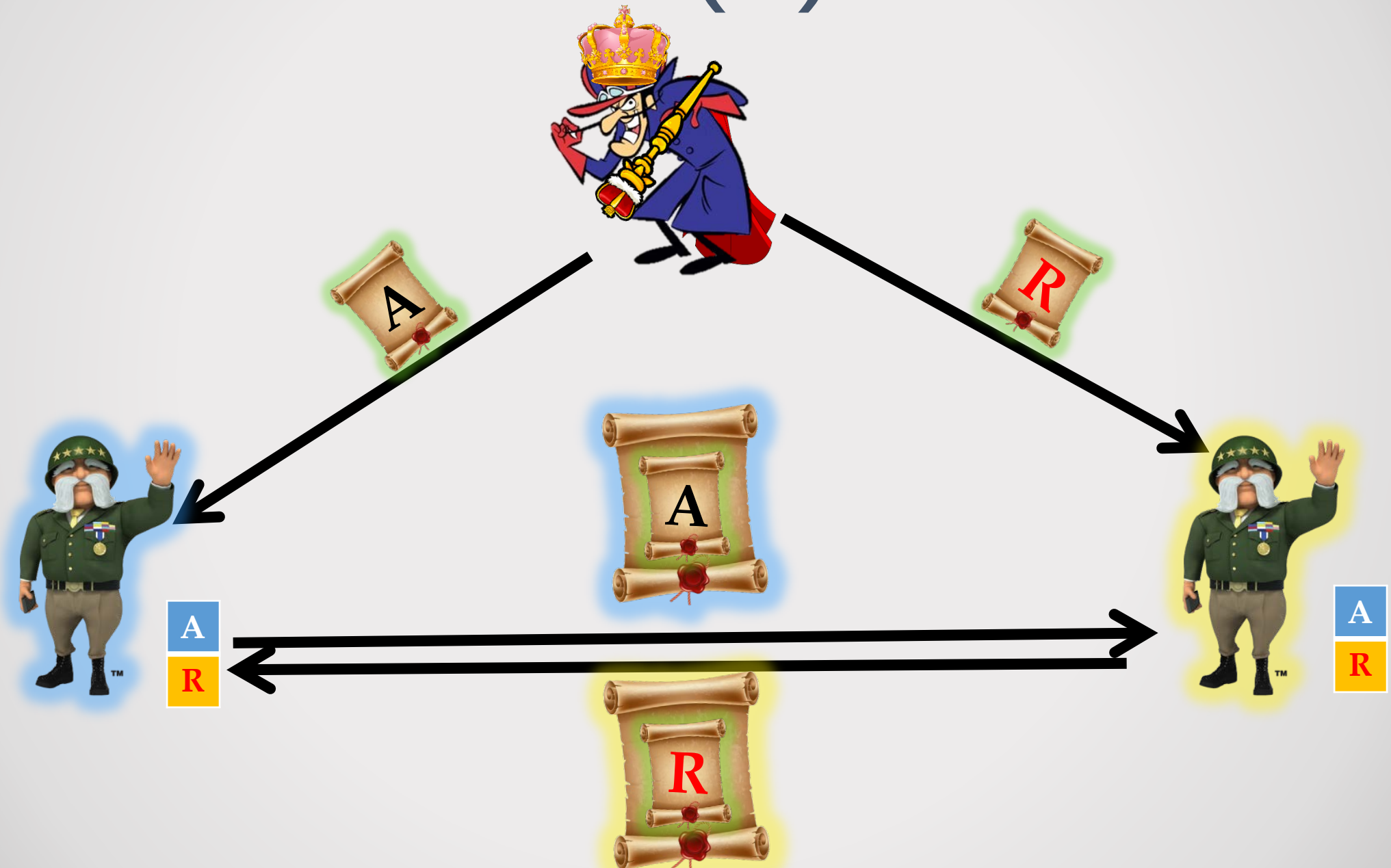
SM(0)



SM(0)

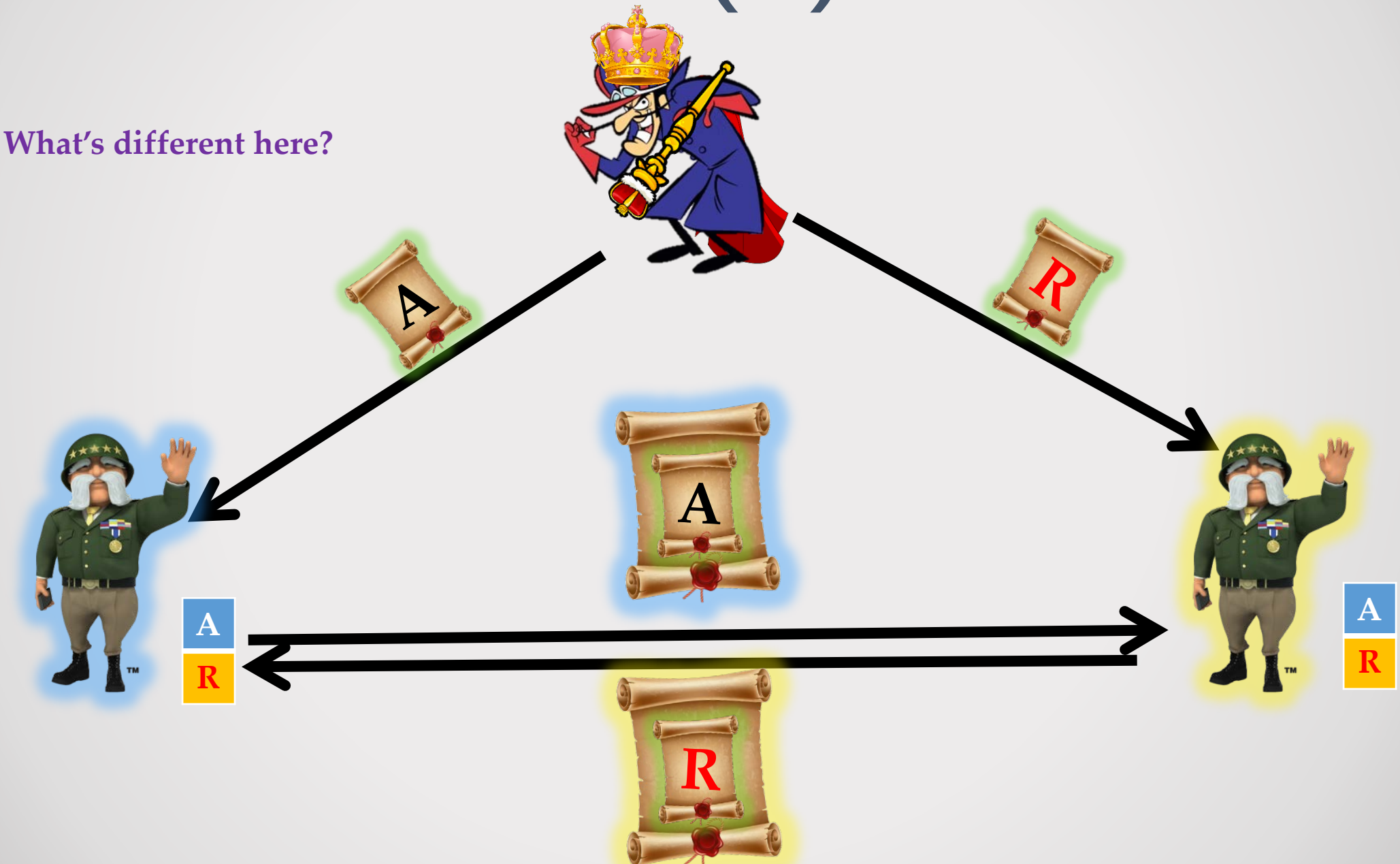


SM(1)



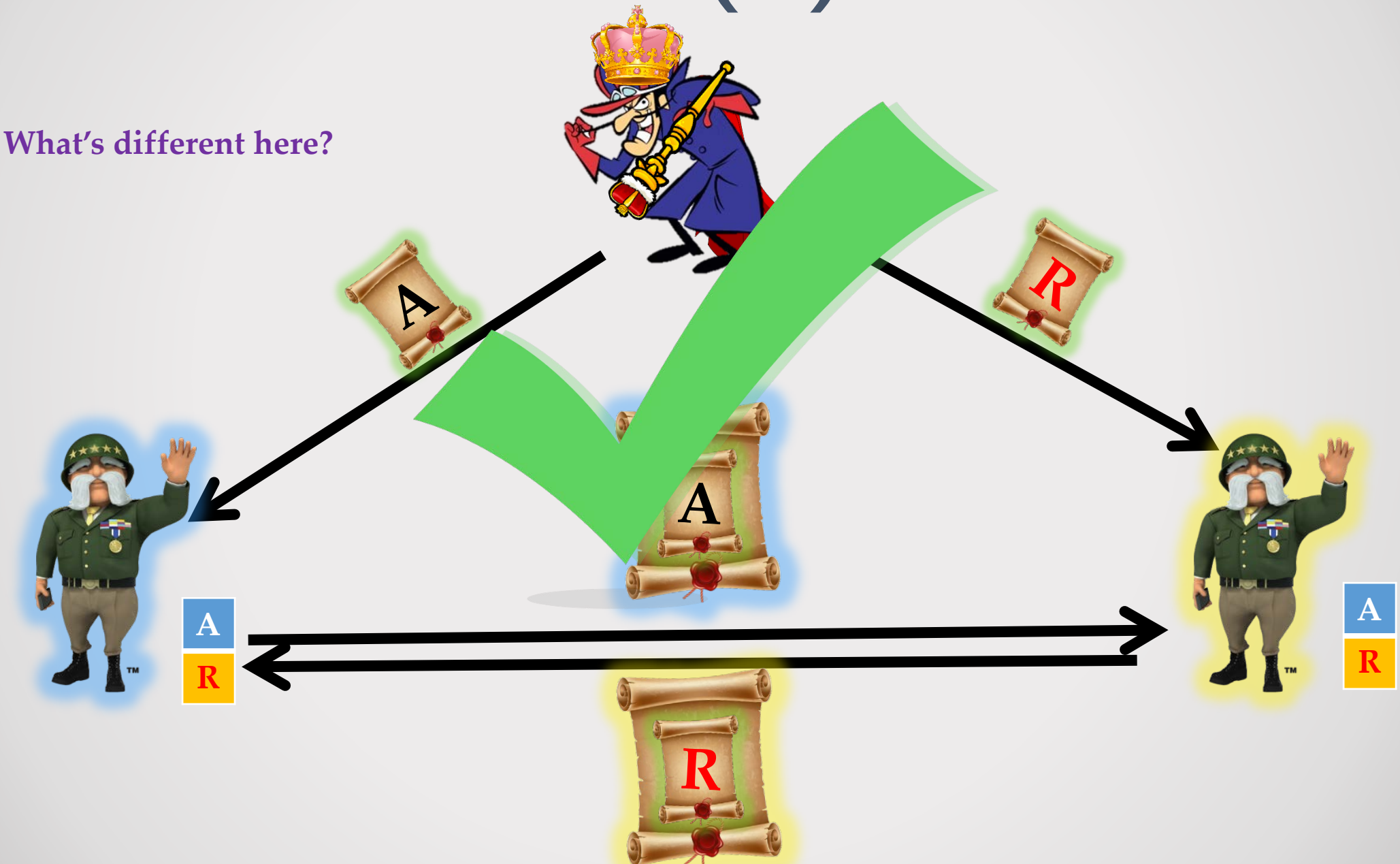
SM(1)

What's different here?



SM(1)

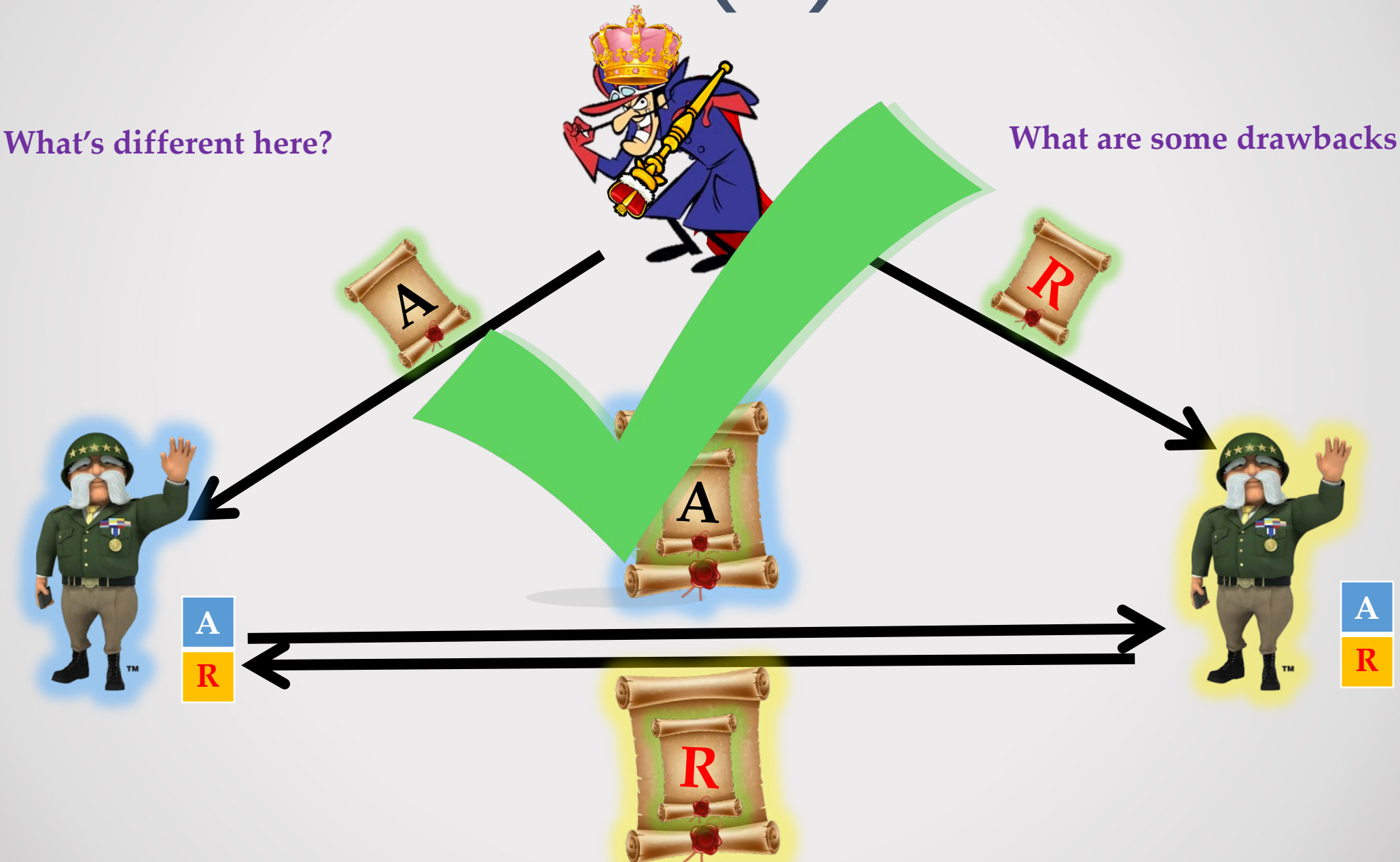
What's different here?



SM(1)

What's different here?

What are some drawbacks of this?



Relation to State Machine Replication

- All messages are delivered correctly.

- All recipients can verify who sent each message.

Relation to State Machine Replication

- All messages are delivered correctly.

Communication lines can fail, and we treat each failure as a node failure.

- All recipients can verify who sent each message.

Relation to State Machine Replication

- All messages are delivered correctly.

Communication lines can fail, and we treat each failure as a node failure.

- All recipients can verify who sent each message.

Messages need to be sent directly rather than using packet switching. Alternatively, we can use digital signatures or message authentication codes (MACs) to ensure authenticity.

Relation to State Machine Replication

- The absence of a message can be detected.

- A loyal general's signature cannot be forged, alternations can be detected, and anyone can verify authenticity of signatures.

Relation to State Machine Replication

- The absence of a message can be detected.

Use a synchronized clock and set timeout for incoming messages.

- A loyal general's signature cannot be forged, alternations can be detected, and anyone can verify authenticity of signatures.

Relation to State Machine Replication

- The absence of a message can be detected.

Use a synchronized clock and set timeout for incoming messages.

- A loyal general's signature cannot be forged, alternations can be detected, and anyone can verify authenticity of signatures.

Use digital signatures (e.g. RSA, DSA).

Question: Why not use MACs?

Efficiency Issues

- OM(m) is impractical since the message complexity is $O(n^m)$ for n nodes
- SM(m) has message complexity $O(n^2)$, but each node needs to send and verify $O(n)$ signatures for *every* request
- Time measurements for a 64-byte message with 1024-bit key
  - Signature generation: 43ms
  - Signature verification: 0.6ms
- Very high response time!

Practical Byzantine Fault Tolerance

- Miguel Castro and Barbara Liskov, OSDI 1999

- Semi-synchronous
  - Operates asynchronously until a *view change* (to be defined later) occurs

- Requires $n = 3m + 1$ nodes to tolerate m failures

- 3% slower than non-tolerant implementation of a network file system

Practical Byzantine Fault Tolerance

Basic Idea:

- Commander sends $v(i)$ to lieutenant i
- Every lieutenant sends commander's order to every other lieutenant
- Lieutenant i waits for at least $2m - 1$ messages of $v(i)$ from different lieutenants before committing

Practical Byzantine Fault Tolerance

Why does this work?

- Suppose lieutenant i commits $v(i)$ and lieutenant j commits $v(j)$
- Since at most m generals are Byzantine, at least $m + 1$ loyal generals (including i himself) have sent $v(i)$
- Same argument for $v(j)$
- There are only $2m + 1$ loyal generals, so some loyal general must have sent both $v(i)$ and $v(j)$. Hence $v(i) = v(j)$

Question: Liveness?

Practical Byzantine Fault Tolerance

To ensure liveness

- Use a local timer to check for timeouts
- Cycle between generals (i.e. nodes) to operate as commander (i.e. primary node), called *views*
- Initiate a synchronous *view change* consensus protocol if timeout occurs
- Full protocol involves many stages, too complicated to explain

Efficiency

- Asynchronous part does not require signatures
  - Use MACs to ensure authenticity of message
  - MACs can be computed 3 orders of magnitude faster than digital signatures!

- Synchronous view change protocol requires digital signatures
  - Happens rarely in realistic settings
  - Faulty node can only cause view change if it is the primary node

Follow-ups

Many papers built on top of PBFT

- 2000: Castro and Liskov – Proactive Recovery in a Byzantine-fault-tolerant System
- 2005: Abd-El-Malek et al. – Fault-scalable Byzantine Fault Tolerant Services.
- 2009: Kotla et al. – Zyzzyva: Speculative Byzantine Fault Tolerance
- 2009: Clement et al. – Making Byzantine Fault Tolerant Systems Tolerate Byzantine Faults
- 2010: Guerraoui et al. – The Next 700 BFT Protocols
- 2013: Aublin et al. – RBFT: Redundant Byzantine Fault Tolerance
- 2013: Veronese et al. – Efficient Byzantine Fault-Tolerance
- 2015: Bahsoun et al. – Making BFT Protocols Really Adaptive

Review

- m = number of faults tolerated
- n = number of replicas needed

Today

Thursday

		Synchronous	Semi-Synchronous	Asynchronous
Oral Messages	Sufficient	$n \geq 3m + 1$ [LSP80]		
	Impossible	$n \leq 3m$ [LSP80]		$m \geq 1$ [FLP82]
Authenticated Messages	Sufficient	$1 \leq n$ [LSP80]	$n \geq 3m + 1$ [CL99]	
	Impossible			$m \geq 1$ [FLP82]

Credit: Eleanor Birrell, CS 6410, Fall 2010