Lecture 12
Byzantine Generals Problem

ECE 422: Reliable and Secure Systems Design

UNIVERSITY OF ALBERTA

Instructor: An Ran Chen
Term: 2024 Winter

Schedule for today

- Key concepts from last class
- Byzantine fault tolerance
- The Byzantine Generals Problem
  - Definition
  - Analogy to system design
  - The oral message solution
  - What happens when m = 2?

Failure classification

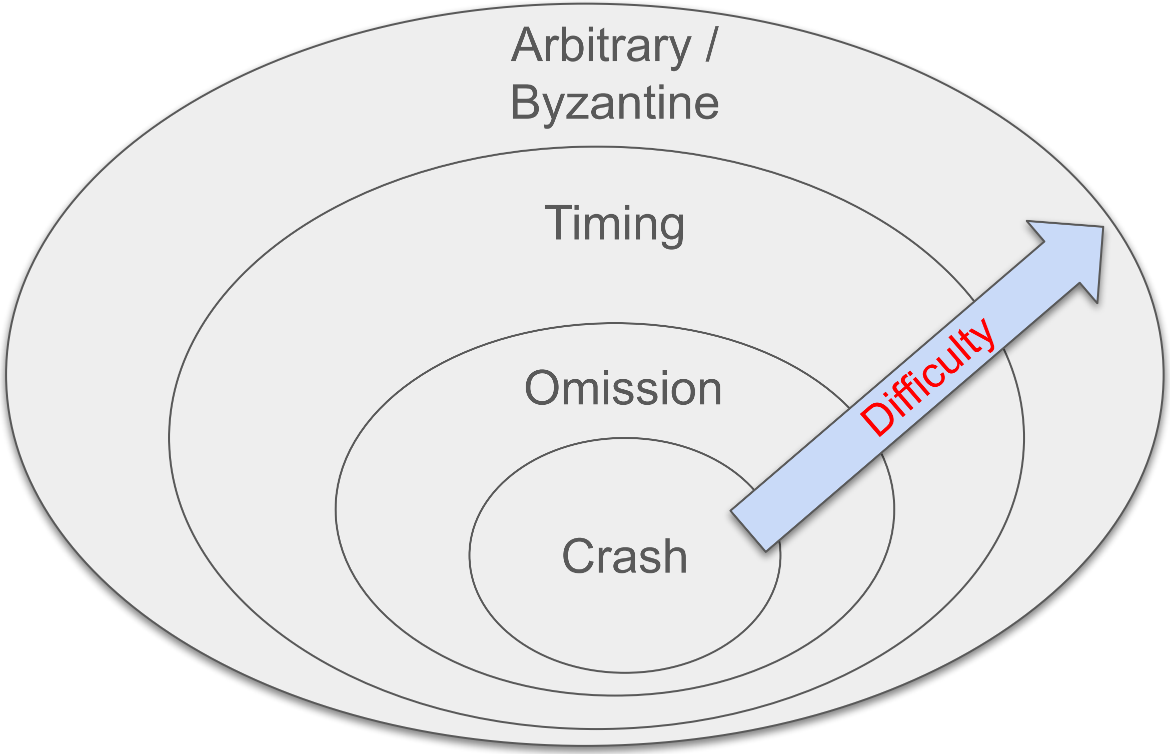
Failure classification describes the way how a system can fail that is perceived by the rest of the system.

- It aims to answer one question: “What kind of failures are we dealing with when taking the whole system into account.”
- It helps understand how to build and design for fault-tolerant systems.

There are four types of failure:

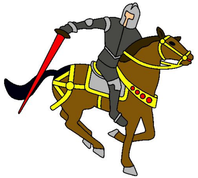
- Timing failure, also known as performance failure
- Omission failure
- Crash failure
- Arbitrary failure, also known as Byzantine failure

Difficulties of failures



Solution: TCP 3-way handshake

A1



A2

A1 wants to attack

If you respond, I'll attack!

A1 will attack;
A2 wants to attack

If you respond, I'll attack!

A1 will attack;
A2 will attack

We'll attack!

A1 will attack;
A2 will attack

There is no solution to the two general problem.

Byzantine failure

Byzantine failure: a node/component may fail arbitrary due to:

- Exhausted resources
- Conflicting information from different parts of the system

Why would nodes/components fail arbitrarily?

- Software bugs in the code
- Hardware failures
- Malicious attack on the system

There is an inconsistency in the functionality of the component. It presents different symptoms to different observers.

Byzantine fault tolerance

Byzantine fault tolerance: the ability of a system to continue functioning even if some of its nodes/components fail randomly or act maliciously.

- The system can keep functioning even if certain components stop working
- For example, safety-critical systems need to be able to work when if some of its components fail
    - E.g., train, airplane, spacecraft
    - E.g., heart-lung machine, robotic surgery

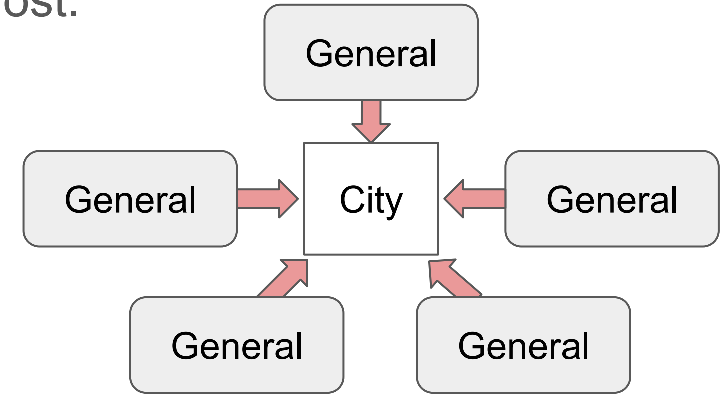
Schedule for today

- Key concepts from last class
- Byzantine fault tolerance
- The Byzantine Generals Problem
    - Definition
    - Analogy to system design
    - The oral message solution
    - What happens when m = 2?

The Byzantine Generals Problem

Byzantine Generals Problem: a group of Byzantine generals who must coordinate a attack by the means of messengers to reach consensus.

- The generals must decide as a group to attack or retreat.
- If they all make the same decision, they can eventually win the battle.
- But after the decision making phrase, if some of the generals attack while the other retreat, then the battle is lost.



The Byzantine Generals Problem

New assumptions:

- If a message is sent, it is always delivered correctly.
  - Sending lots of messengers increases the probability that one getting through
- One or more generals may be “traitors”
  - Analogous to a malicious node/component in the system
  - Traitors’ mission is to break the consensus among the “royal” generals (non-traitors) so that they lose the battle.

Consensus in the presence of a traitor

Suppose we have five generals, one of them is a traitor.

Each royal general shares their decision to others:

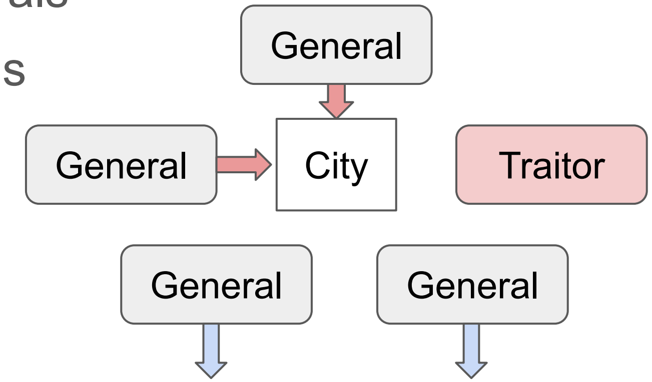
- Two generals send an “attack” message.
- Two other generals send a “retreat” message.

The traitor tries to break the consensus:

- Sends an “attack” message to the first two generals
- Send a “retreat” message to the last two generals

Next day: consensus is broken

- Two royal generals attack
- Two royal generals retreat



Analogy to system design

In a distributed system, some nodes can be:

- Flaky nodes
- Malicious nodes (e.g., Hackers)

The Byzantine problem attempts to propose a solution to these failures from two two main questions:

- How can we ensure the consistency?
  - All “royal” nodes produce the same results.
- How we can we ensure the validity?
  - All “royal” nodes produce a valid response.
- (Despite the presence of flaky or malicious nodes)

Byzantine Generals Problem

Lamport et al. proposed to solve the Byzantine Generals Problem with two solutions:

- Solution 1: Oral message
- Solution 2: Signed message

[PDF] **The Byzantine generals problem**
L Lamport, R Shostak, M Pease - Concurrency: **the** works of leslie …, 1982 - dl.acm.org
… three **generals** that coped with one traitor, then we could construct a three-**general** solution to **the Byzantine Generals Problem** that also worked in **the** presence of one traitor. Suppose …
Save Cite Cited by 9590 Related articles All 179 versions

The oral message solution simplifies the problem:

- A commanding general sends an order to his lieutenants.
  - A commanding general is the first person to send a decision
  - The rest become the lieutenants
  - Lieutenant can decide to execute or not execute the order from the command general
- Either of them can be traitors

Byzantine Generals Problem

When a commanding general sends an order to his lieutenants, there are two properties:

- Consistency: Loyal lieutenants must execute the same order.
- Validity: If the commanding general is loyal, then every loyal lieutenant must obeys his order.

Solution to Byzantine Generals Problem

Let m = # traitors (malicious nodes), n = total # generals (nodes)

Theorem: We need $n = (3m + 1)$ generals to tolerate (m) traitors.

- If $m < \frac{1}{3} n$, then it is solvable
- We can somehow achieve both consistency and validity

Example: m = 1, n = 3

Scenario 1: Why is it unsolvable?

General A = commanding general, royal

General B = lieutenant, royal

General C = lieutenant, traitor

Royal

Commanding General A

Lieutenant B

Royal

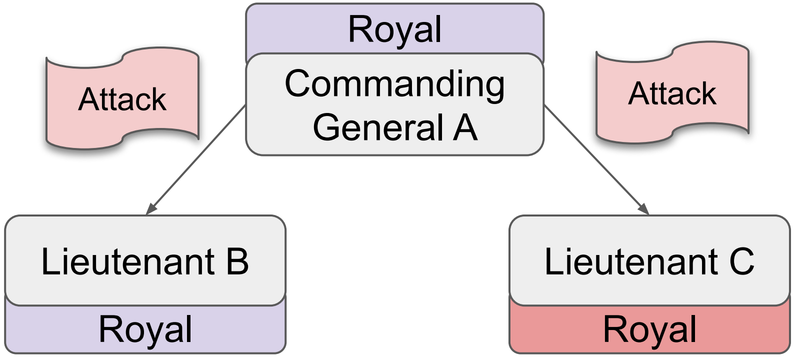
Lieutenant C

Royal

Example: m = 1, n = 3

Scenario 1: Why is it unsolvable?

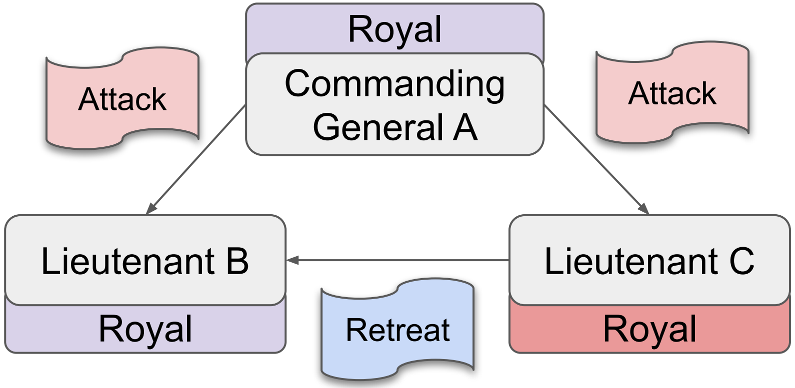
- A sends an “attack” message to the two others.
- B receives the message, and asks C to share the message that he/she received from A.



Example: m = 1, n = 3

Scenario 1: Why is it unsolvable?

- A sends an “attack” message to the two others.
- B receives the message, and asks C to share the message that he/she received from A.
- C is a traitor, thus changes the message to “retreat”.
- B cannot tell who is the traitor.
- Unsolvable



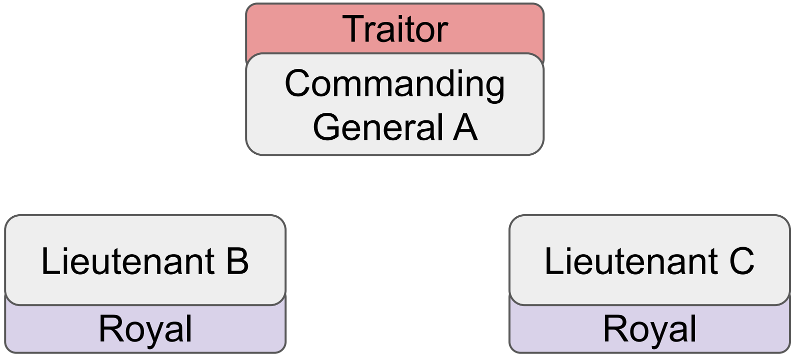
Example: $m = 1, n = 3$

Scenario 2: Why is it unsolvable?

General A = commanding general, traitor

General B = lieutenant, royal

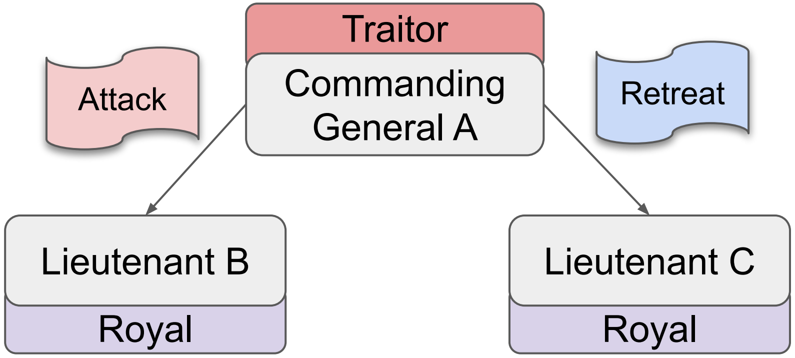
General C = lieutenant, royal



Example: m = 1, n = 3

Scenario 2: Why is it unsolvable?

- A sends different messages to others; “attack” to B, and “retreat” to C.
- B receives the message, and asks C to share the message that he/she received from A.

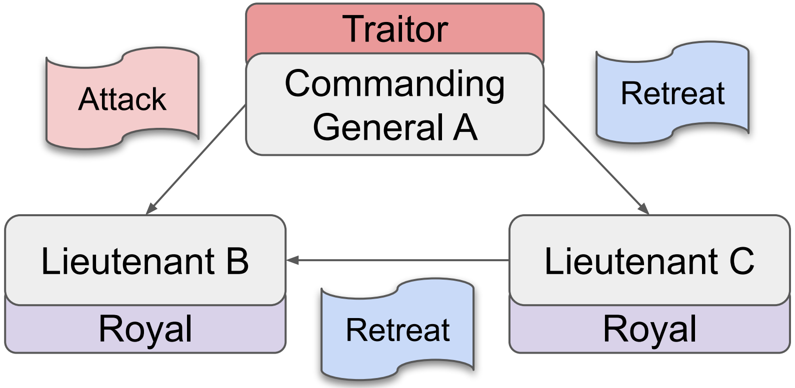


Example: $m = 1, n = 3$

Scenario 2: Why is it unsolvable?

- A sends different messages to others; “attack” to B, and “retreat” to C.
- B receives the message, and asks C to share the message that he/she received from A.
- C shares the message “retreat” with B.
- B still cannot tell who is the traitor.
- Unsolvable

If $m < \frac{1}{3} n$, then it is solvable, Otherwise it is unsolvable.



Example: m = 1, n = 4

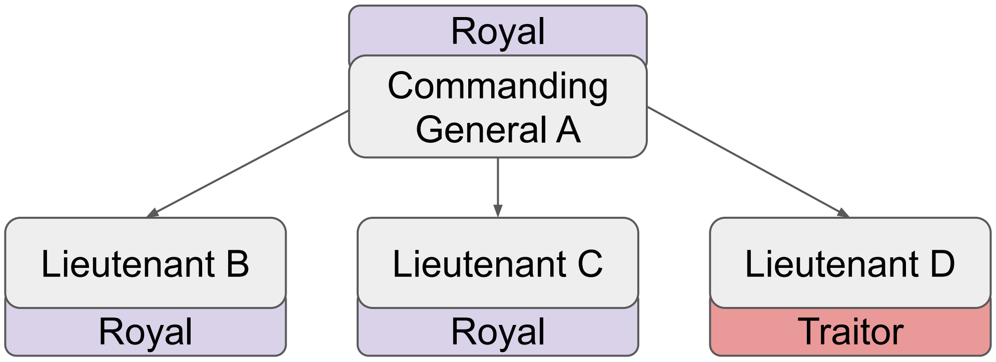
Scenario 1: Why is it solvable?

General A = commanding general, royal

General B = lieutenant, royal

General C = lieutenant, royal

General D = lieutenant, traitor



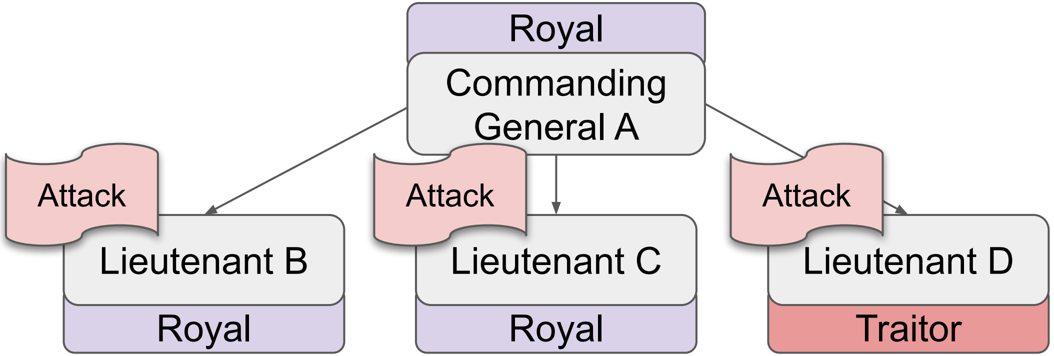
Example: m = 1, n = 4

Scenario 1: Why is it solvable?

- A sends an “attack” message to others.

- B decides to attack,
  V(B) = (A, A, R)

- B asks C and D for the message they received.
- B receives: “attack” from A, “attack” from C, and “retreat” from D
  - V(B) = (A, A, R)
- B decides to attack



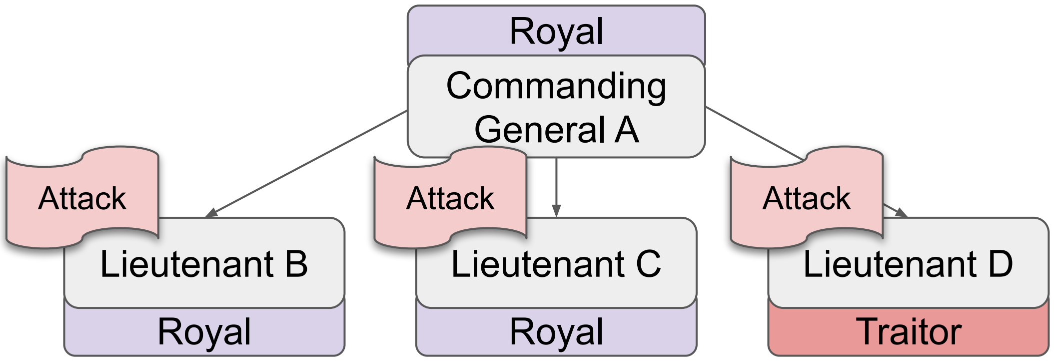
Example: m = 1, n = 4

Scenario 1: Why is it solvable?

- A sends an “attack” message to others.
- C asks B and D for the message they received.
- C receives: “attack” from A, “attack” from B, and “retreat” from D
  - V(C) = (A A R)
- C decides to attack

- B decides to attack, V(B) = (A A R)
- C decides to attack

  V(C) = (A A R)



Example: m = 1, n = 4

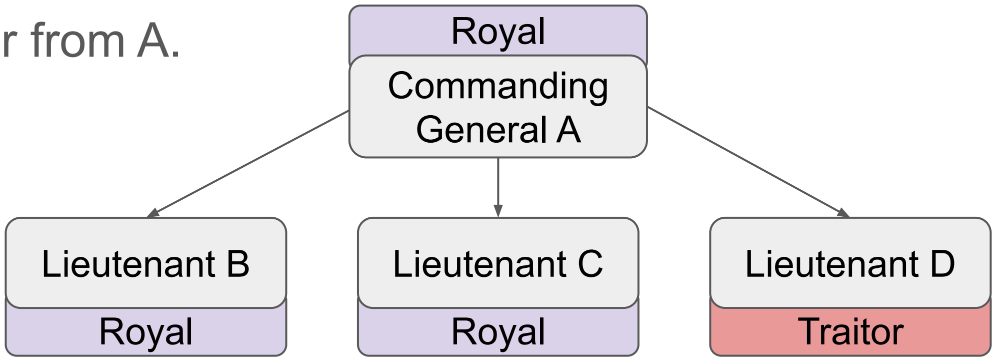
Scenario 1: Why is it solvable?

- A sends an “attack message” to other.

Problem solved!

- Consistency: B and C executed the same order.
- Validity: B and C obeyed the order from A.

- B decides to attack,
  V(B) = (A A R)
- C decides to attack
  V(C) = (A A R)



Example: m = 1, n = 4

Scenario 2: Is this solvable? Can we still achieve both properties?

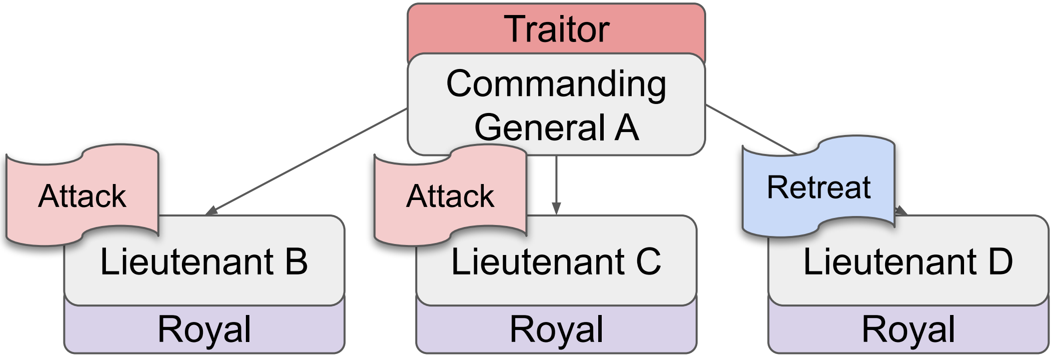
General A = commanding general, traitor

General B = lieutenant, royal

General C = lieutenant, royal

General D = lieutenant, royal

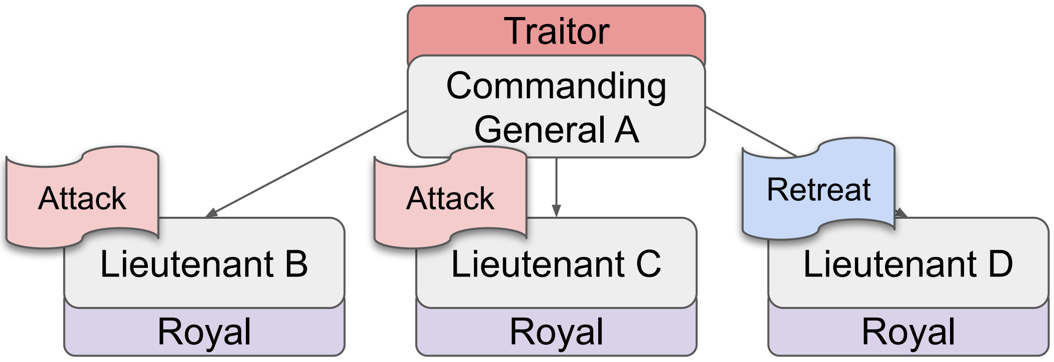
Validity: if and only if the commanding general is loyal, then every loyal lieutenant must obeys his order.



Example: m = 1, n = 4

Scenario 2: Why is it solvable?

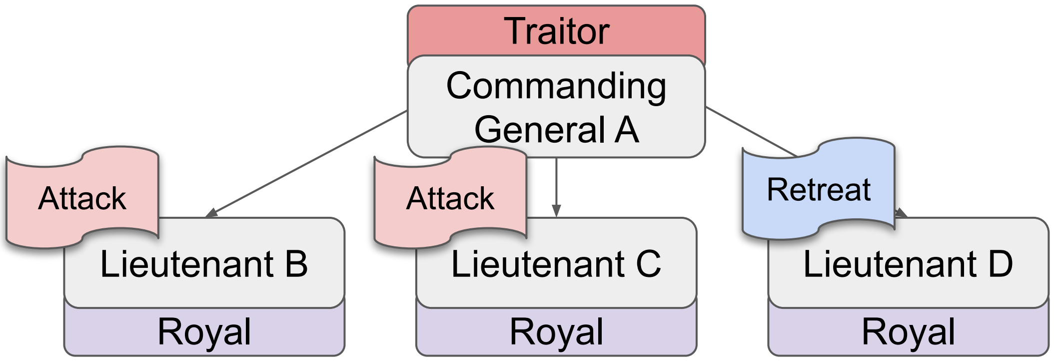
- A sends “attack” to the first two lieutenants, and “retreat” to the last.
- B receives: “attack” from A, “attack” from C, and “retreat” from D
    - V(B) = (A A R)
- B decides to attack



Example: m = 1, n = 4

Scenario 2: Why is it solvable?

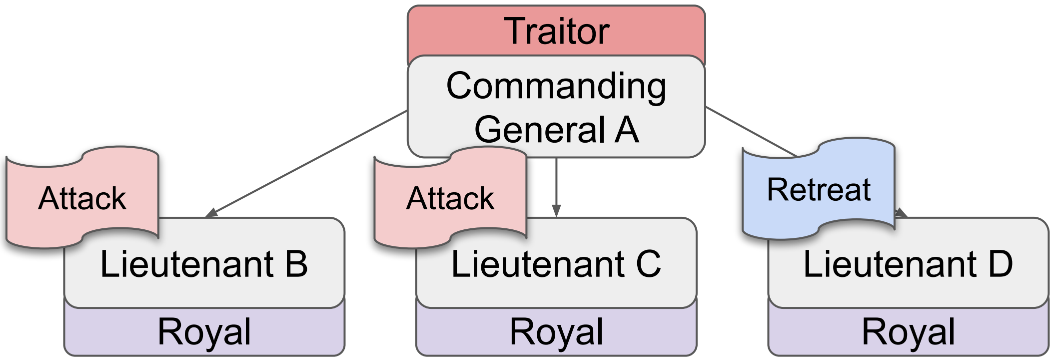
- A sends “attack” to the first two lieutenants, and “retreat” to the last.
- C receives: “attack” from A, “attack” from B, and “retreat” from D
  - V(C) = (A A R)
- C decides to attack



Example: m = 1, n = 4

Scenario 2: Why is it solvable?

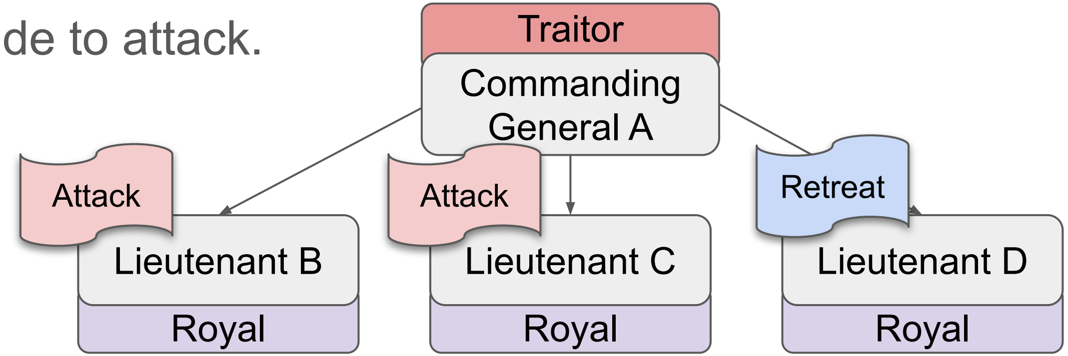
- A sends “attack” to the first two lieutenants, and “retreat” to the last.
- D receives: “retreat” from A, “attack” from B, and “attack” from C
  - V(D) = (R A A)
- D decides to attack



Example: m = 1, n = 4

Scenario 2: Why is it solvable?

- A sends “attack” to the first two lieutenants, and “retreat” to the last.
- V(B) = (A A R)
- V(C) = (R A A)
- V(D) = (R A A)
- All three royal lieutenants decide to attack.



Problem solved!

- Consistency fulfilled.

Schedule for today

- Key concepts from last class
- Byzantine fault tolerance
- The Byzantine Generals Problem
  - Definition
  - Analogy to system design
  - The oral message solution
  - What happens when $m = 2$?

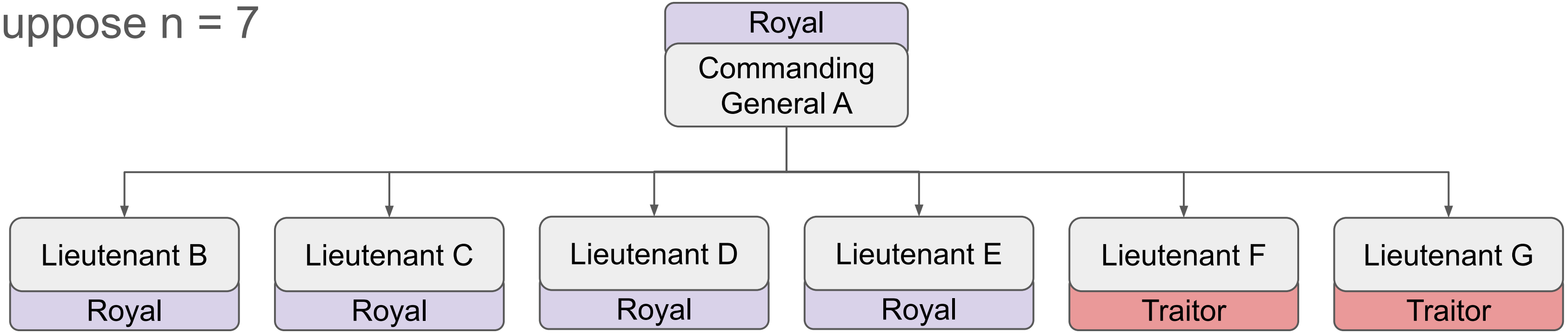
What happens when m = 2?

A recursive solution must be used to solve m > 1:

- The commanding general sends the message to the other lieutenants.
- Each lieutenant becomes a commanding general and sends the message they received from the original commanding general to the other lieutenants.
- …

What happens when m = 2?

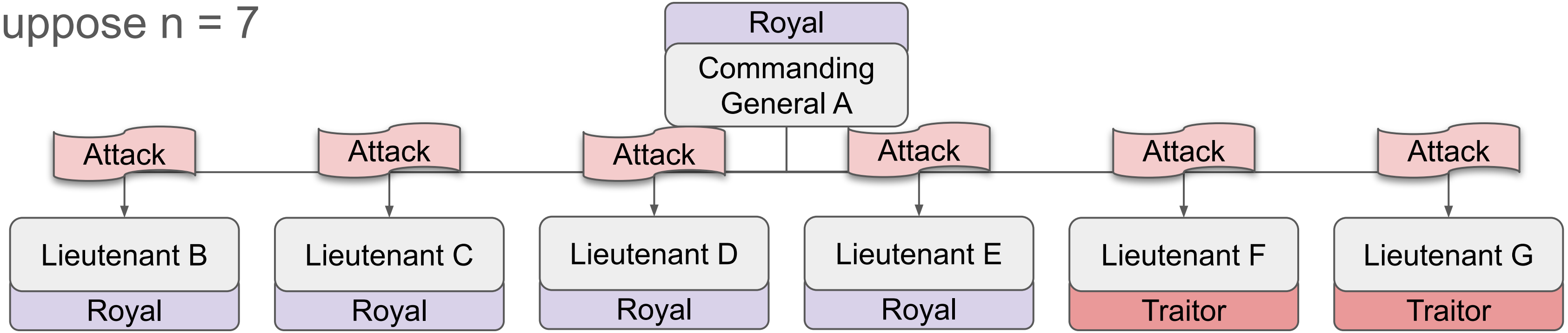
Suppose n = 7



- General A = commanding general, royal
- General B = lieutenant, royal
- General C = lieutenant, royal
- General D = lieutenant, royal
- General E = lieutenant, royal
- General F = lieutenant, traitor
- General G = lieutenant, traitor

What happens when m = 2?

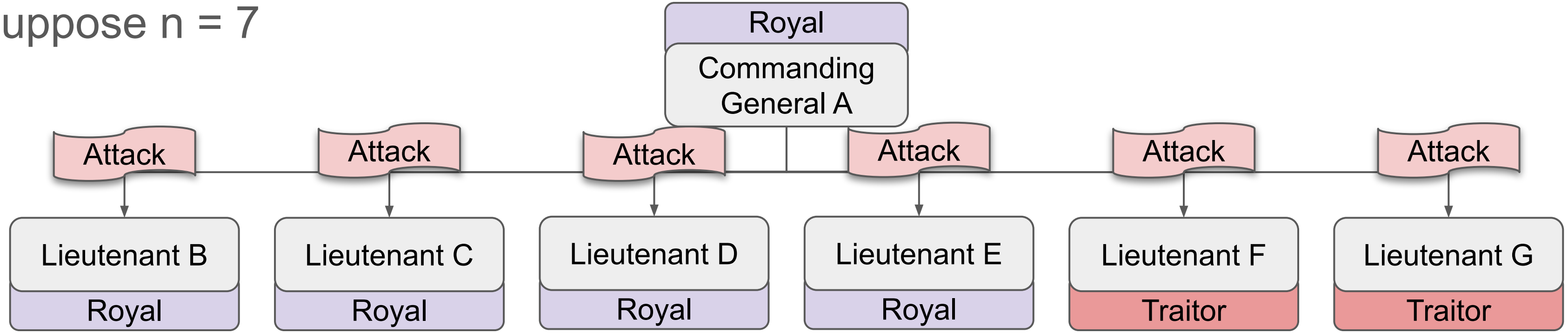
Suppose n = 7



- A sends “attack”
- B asks C the message he/she received
  - C replies “attack”
- B asks D, E, F, G: “what C received from A?”
  - D replies “attack”, E replies “attack”, F replies “retreat”, G replies “retreat”
  - V_C = (A, A, A, R, R), B confirms that C received “attack”

What happens when m = 2?

Suppose n = 7



- A sends “attack”
- B asks C, D, E, F, G: “what D received from A?”
    - C replies “attack”, D replies “attack”, E replies “attack”, F replies “retreat”, G replies “retreat”
    - V_D = (A, A, A, R, R), B confirms that D received “attack”

What happens when m = 2?

Suppose n = 7

For Lieutenant B

- Lieutenant B pick the most popular action from the other generals:
    - V_B = A, B receives “attack” from A
    - V_C = (A, A, A, R, R), B confirms that C received “attack”
    - V_D = (A, A, A, R, R), B confirms that D received “attack”
    - V_E = (A, A, A, R, R), B confirms that E received “attack”
    - V_F = (R, R, R, R, R), B confirms that F received “retreat”
    - V_G = (R, R, R, R, R), B confirms that G received “retreat”
- Lieutenant B decides to attack, because there are 4 attacks and 2 retreats

For Lieutenant C …

What happens when m = 2?

Suppose n = 7

- Commanding general A will attack.
- Lieutenant B decides to attack, because there are 4 attacks and 2 retreats
- Lieutenant C decides to attack, because there are 4 attacks and 2 retreats
- Lieutenant D decides to attack, because there are 4 attacks and 2 retreats
- Lieutenant E decides to attack, because there are 4 attacks and 2 retreats

All four royal lieutenant decide to attack, obeying A.

Problem solved!

- Consistency fulfilled
- Validity fulfilled

Cost of solving Byzantine Generals Problem

Solving the Byzantine Generals Problem is expensive

- m = 0, sending n messages to every lieutenant
- m = 1, sending n^2 messages to every lieutenant
- m = 2, sending n^3 messages to every lieutenant
- …

m	Messages Sent
0	$O(n)$
1	$O(n^2)$
2	$O(n^3)$
3	$O(n^4)$

Timeline

- In 1978, Robert Shostak worked on a NASA-sponsored project about fault-tolerant for aircraft control.

SIFT: Design and analysis of a fault-tolerant computer for aircraft control
JH Wensley, L **Lamport**, J Goldberg… - Proceedings of the …, 1978 - ieeexplore.ieee.org
SIFT (Software Implemented Fault Tolerance) is an ultrareliable computer for critical aircraft control applications that achieves fault tolerance by the replication of tasks among
☆ Save 99 Cite Cited by 857 Related articles All 17 versions

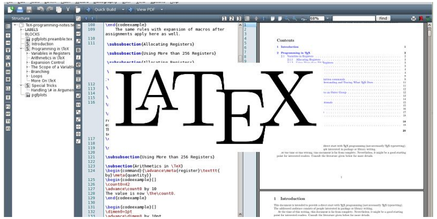
- In 1980, Leslie Lamport, Robert Shostak, and Marshall Pease together worked on an algorithm for processors to reach consensus in the presence of faults.

Reaching agreement in the **presence** of **faults**
M Pease, R Shostak, L Lamport - Journal of the ACM (JACM), 1980 - dl.acm.org
… In the absence of **faults reaching** a satisfactory mutual **agreement** is usually an easy matter. In … include those of **reaching** approximate **agreement** and **reaching agreement** under various …
☆ Save 99 Cite Cited by 3167 Related articles All 5 versions

- In 1982, they later come up with the analogy of the Byzantine generals problem.

[PDF] **The Byzantine generals problem**
L Lamport, R Shostak, M Pease - Concurrency: **the** works of leslie …, 1982 - dl.acm.org
… three **generals** that coped with one traitor, then we could construct a three-**general** solution to **the Byzantine Generals Problem** that also worked in **the** presence of one traitor. Suppose …
☆ Save 99 Cite Cited by 9590 Related articles All 179 versions

Leslie Lamport is also the initial developer of LaTeX.



Timeline

- In 1998, “Practical Byzantine Fault Tolerance” (PBFT) was published.
    - PBFT was proposed as a practical solution for Byzantine fault tolerance that takes into consideration there can be delays in the network.

- In 2008, “Bitcoin: A peer-to-peer electronic cash system” was published.
    - Key idea: A ledger via some complex math records the transactions.
    - Proof of work (PoW) to validate transactions on a blockchain.
    - Individuals around the world can come to a consensus without relying on any third-party as an intermediate for trust.