Byzantine Generals Problem

Graham Swain

October 15, 2022
Math 479

- Byzantine armies are sieging a city.

- Byzantine armies are sieging a city.
- Generals can only communicate by sending messengers.

A. All loyal generals decide upon the same plan of action.

A. All loyal generals decide upon the same plan of action.

B. A small number of traitors cannot cause the loyal generals to adopt a bad plan.

- Condition A is met by having the generals use the same method of decision making.

- Condition A is met by having the generals use the same method of decision making.
- Condition B is met by having the generals use a robust decision making method.

To help ensure that Condition A and Condition B are met, we need to meet two other conditions:

To help ensure that Condition A and Condition B are met, we need to meet two other conditions:

1. Every loyal general must obtain the same information $v_1, ..., v_n$.

To help ensure that Condition A and Condition B are met, we need to meet two other conditions:

1. Every loyal general must obtain the same information $v_1, ..., v_n$.
2. If the i^{th} general is loyal, then the value that they send must be used by every loyal general as the value of v_i.

To help ensure that Condition A and Condition B are met, we need to meet two other conditions:

1. Every loyal general must obtain the same information $v_1, ..., v_n$.
2. If the i^{th} general is loyal, then the value that they send must be used by every loyal general as the value of v_i.

Condition 1 can be rewritten as:

1′. For every i, any two loyal generals use the same value of v_i.

- Loyal generals cannot take a value v_i at face value.

- Loyal generals cannot take a value v_i at face value.
- Condition $1'$ and Condition 2 are both contingent on a single v_i sent by the i^{th} general.

Byzantine Generals Problem

Byzantine Generals Problem

A commanding general must send an order to their $n-1$ lieutenants such that:

Byzantine Generals Problem

A commanding general must send an order to their $n-1$ lieutenants such that:

IC1. All loyal lieutenants obey the same order.

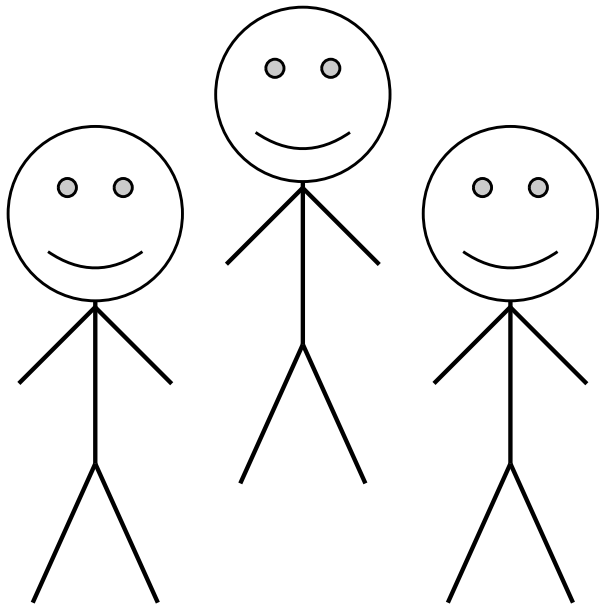
Byzantine Generals Problem

A commanding general must send an order to their $n-1$ lieutenants such that:

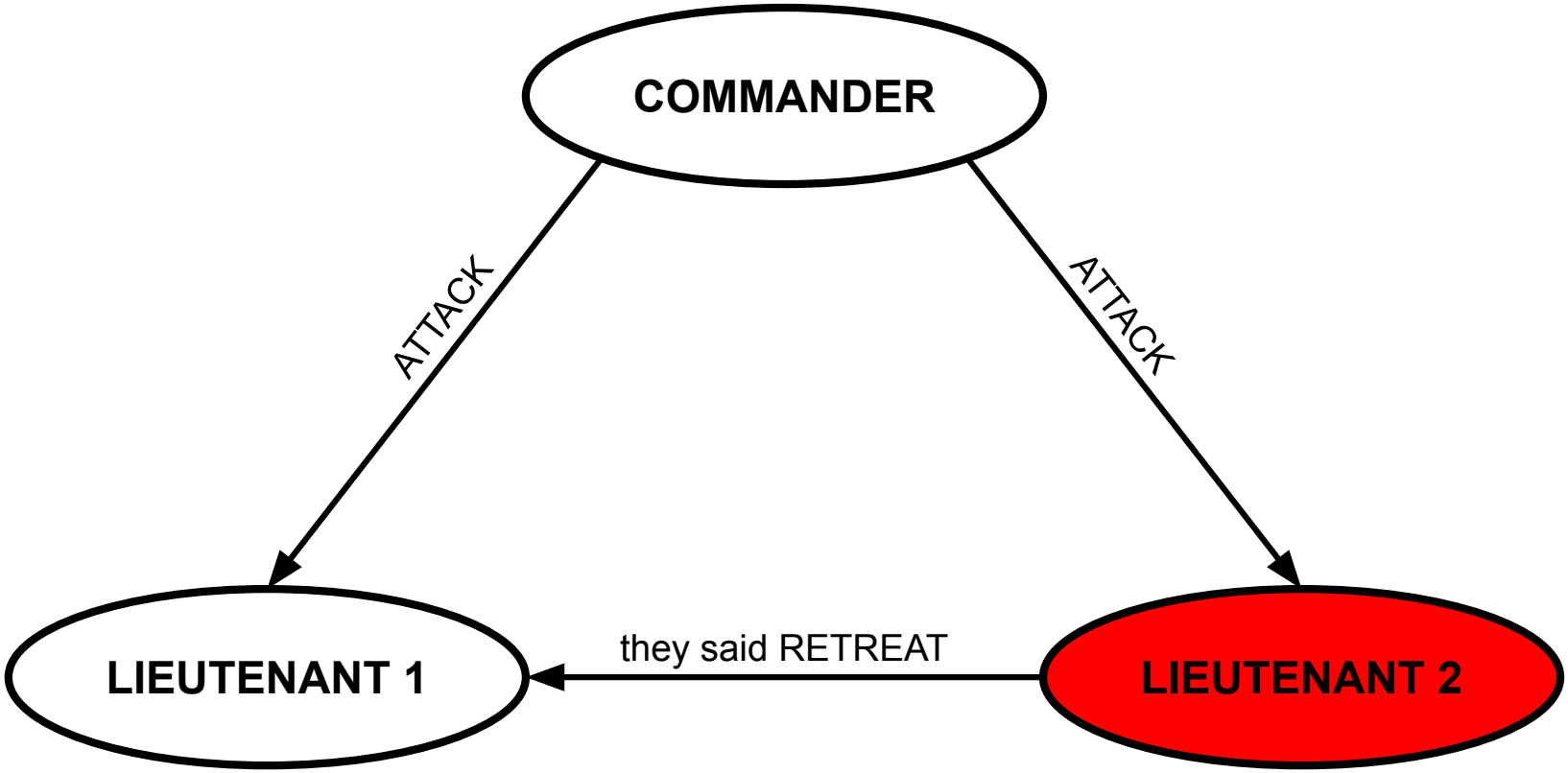
IC1. All loyal lieutenants obey the same order.

IC2. If the commander is loyal, then every loyal lieutenant obeys the order they send.

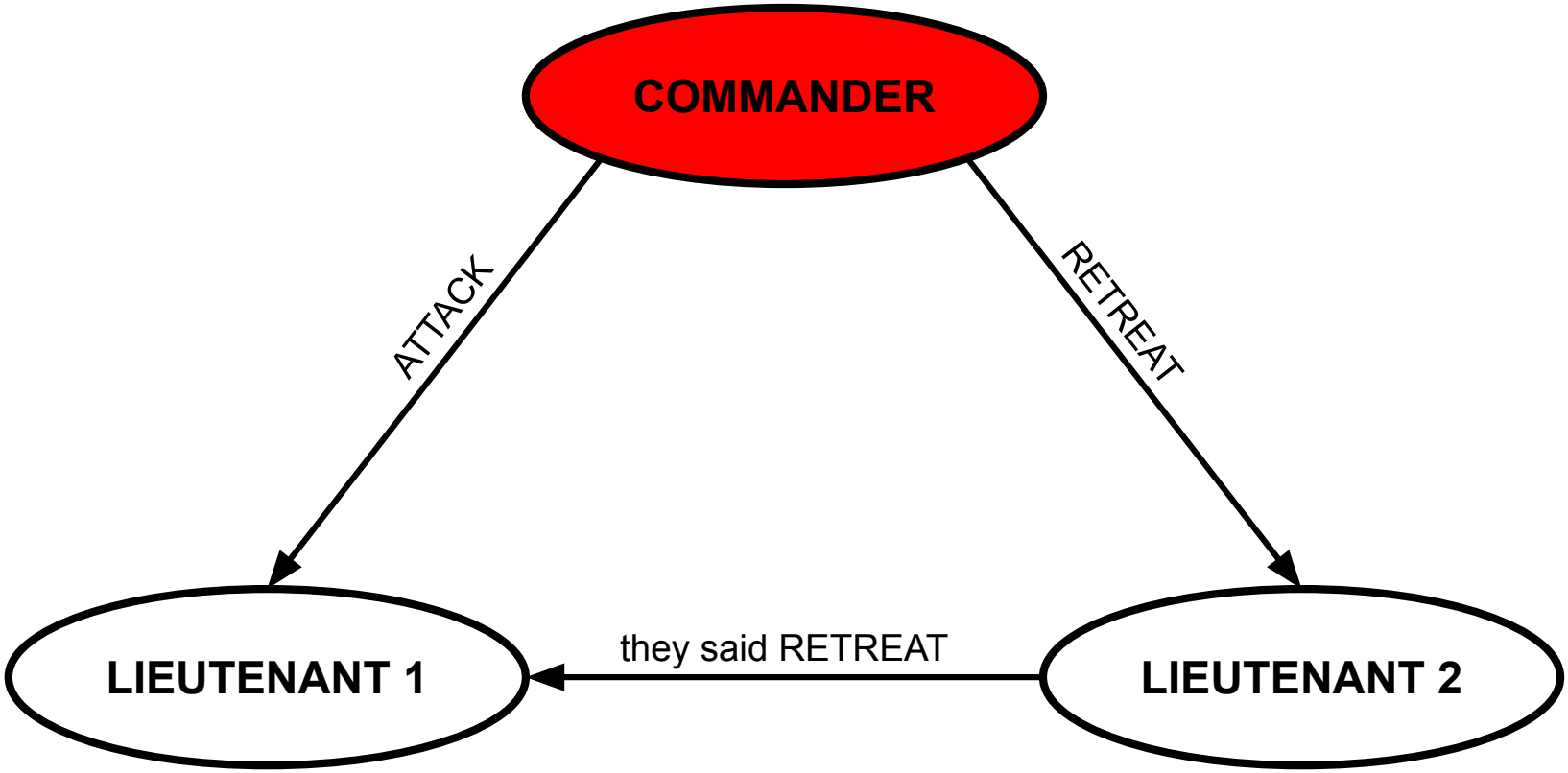
Three Generals Problem

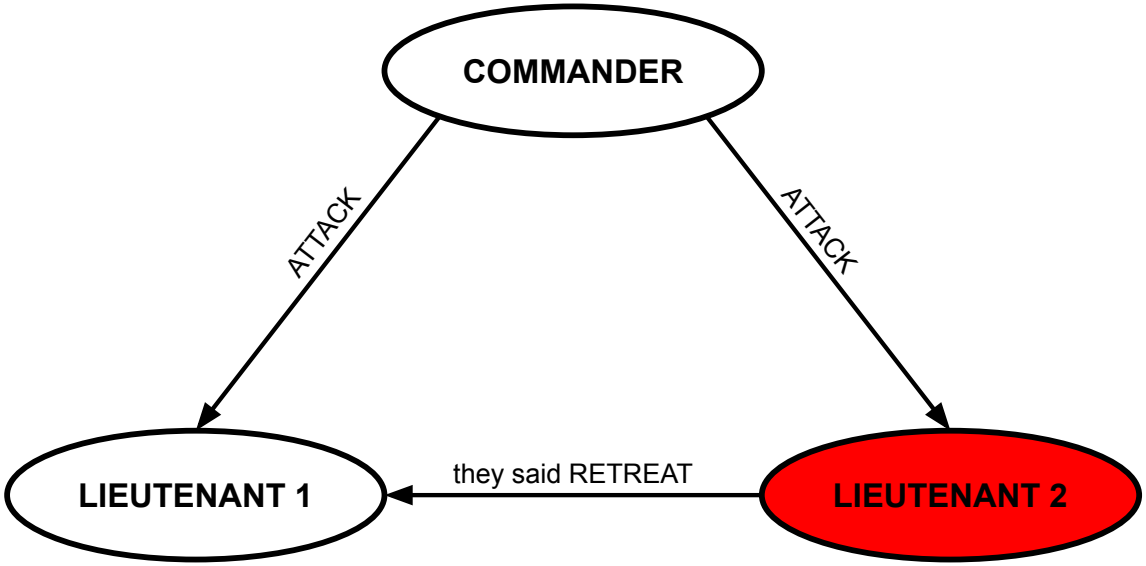


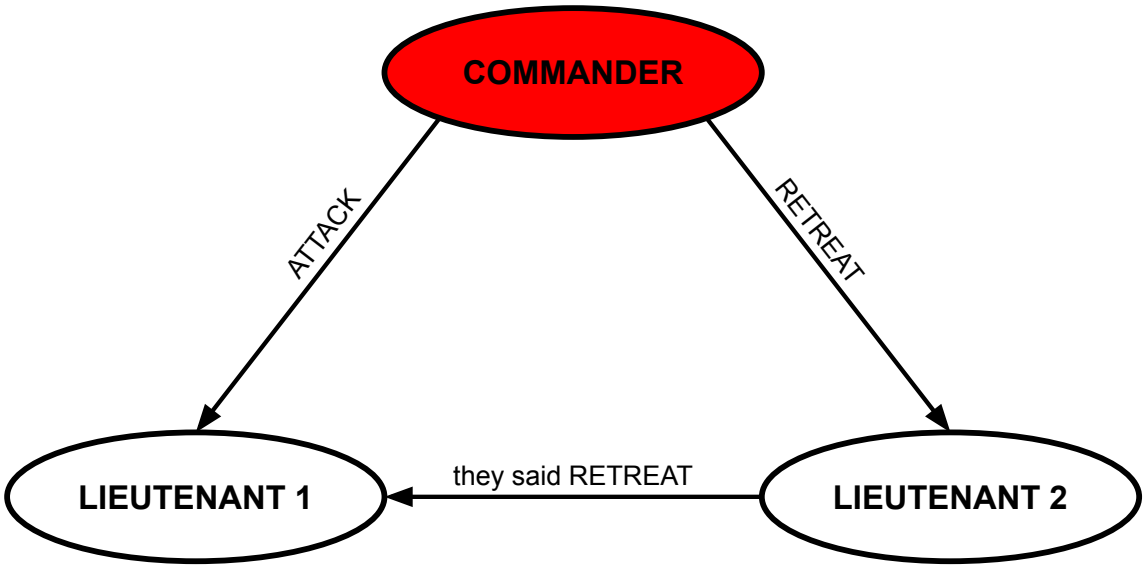
Situation 1: Commander is Loyal



Situation 1: Commander is a Traitor




COMMANDER
ATTACK
ATTACK
LIEUTENANT 1
they said RETREAT
LIEUTENANT 2


COMMANDER
ATTACK
RETREAT
LIEUTENANT 1
they said RETREAT
LIEUTENANT 2

Theorem (Three Generals)

No solution exists for $n < 3m + 1$ generals with m traitors and $n > 3$.

Proof.

Assume that a solution exists for $3m$ or less Albanian generals.
We will show a solution exists for three Byzantine generals with a single traitor.

Proof.

Each Byzantine general represents at most m Albanian generals.

Proof.

Each Byzantine general represents at most m Albanian generals.

Since the Albanian commander needs to be represented as well, the Byzantine commander represents the Albanian commander as well as at most $m - 1$ Albanian lieutenants.

Proof.

Each Byzantine general represents at most m Albanian generals.

Since the Albanian commander needs to be represented as well, the Byzantine commander represents the Albanian commander as well as at most $m - 1$ Albanian lieutenants.

We know that there is a single Byzantine traitor.
Since each Byzantine general represents at most m Albanian generals, we know there is at most m Albanian traitors.

Proof.

The assumed solution means that IC1 and IC2 is true for the Albanian generals.

Since up to m Albanian generals are represented by a Byzantine general, then IC1 and IC2 must also be true for the Byzantine generals, which we know is impossible, forming a contradiction. □

We know we need $n \geq 3m + 1$ generals if we have m traitors.

Oral Solution

We make some assumptions about the messages:

We make some assumptions about the messages:

A1. Every message that is sent is delivered correctly.

We make some assumptions about the messages:

A1. Every message that is sent is delivered correctly.

A2. The receiver of a message knows who sent it.

We make some assumptions about the messages:

A1. Every message that is sent is delivered correctly.

A2. The receiver of a message knows who sent it.

A3. The absence of a message can be detected.

Algorithm OM(0)

Algorithm OM(0)

1. The commander sends their value to every lieutenant.

Algorithm OM(0)

1. The commander sends their value to every lieutenant.
2. Each lieutenant uses the value they received from the commander. If they received no value, default to RETREAT.

Algorithm OM(m), $m > 0$

1. The commander sends their value to every lieutenant.

Algorithm OM(m), $m > 0$

1. The commander sends their value to every lieutenant.
2. For each i,

Algorithm OM(m), $m > 0$

1. The commander sends their value to every lieutenant.
2. For each i,
    a. Lieutenant i receives a value v_i from the commander. Default to RETREAT if they receive no value.

Algorithm OM(m), $m > 0$

1. The commander sends their value to every lieutenant.
2. For each i,
   a. Lieutenant i receives a value v_i from the commander. Default to RETREAT if they receive no value.
   b. Lieutenant i acts as the commander in OM($m-1$) to send the message to each of the remaining $n-2$ lieutenants.

Algorithm OM(m), $m > 0$

1. The commander sends their value to every lieutenant.
2. For each i,
   a. Lieutenant i receives a value v_i from the commander. Default to RETREAT if they receive no value.
   b. Lieutenant i acts as the commander in OM($m-1$) to send the message to each of the remaining $n-2$ lieutenants.
3. For each i, and each j not equal to i,

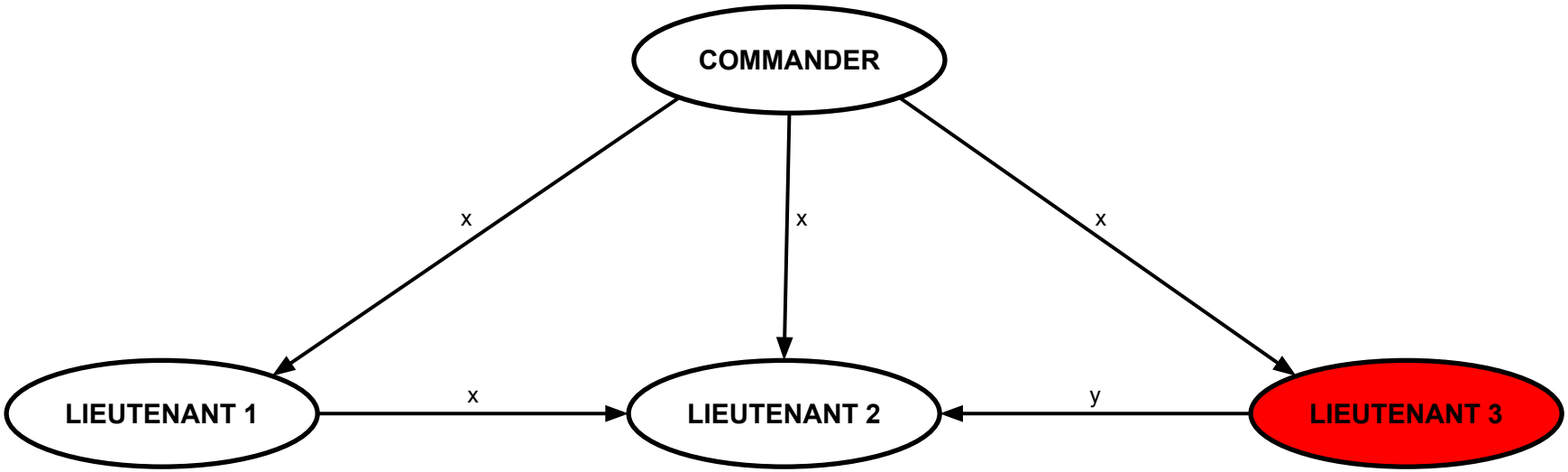
Algorithm OM(m), $m > 0$

1. The commander sends their value to every lieutenant.
2. For each i,
   a. Lieutenant i receives a value v_i from the commander. Default to RETREAT if they receive no value.
   b. Lieutenant i acts as the commander in OM($m-1$) to send the message to each of the remaining $n-2$ lieutenants.
3. For each i, and each j not equal to i,
   a. let v_j be the value Lieutenant i received from Lieutenant j in step (2b). Default to RETREAT if Lieutenant i received no value from Lieutenant j.

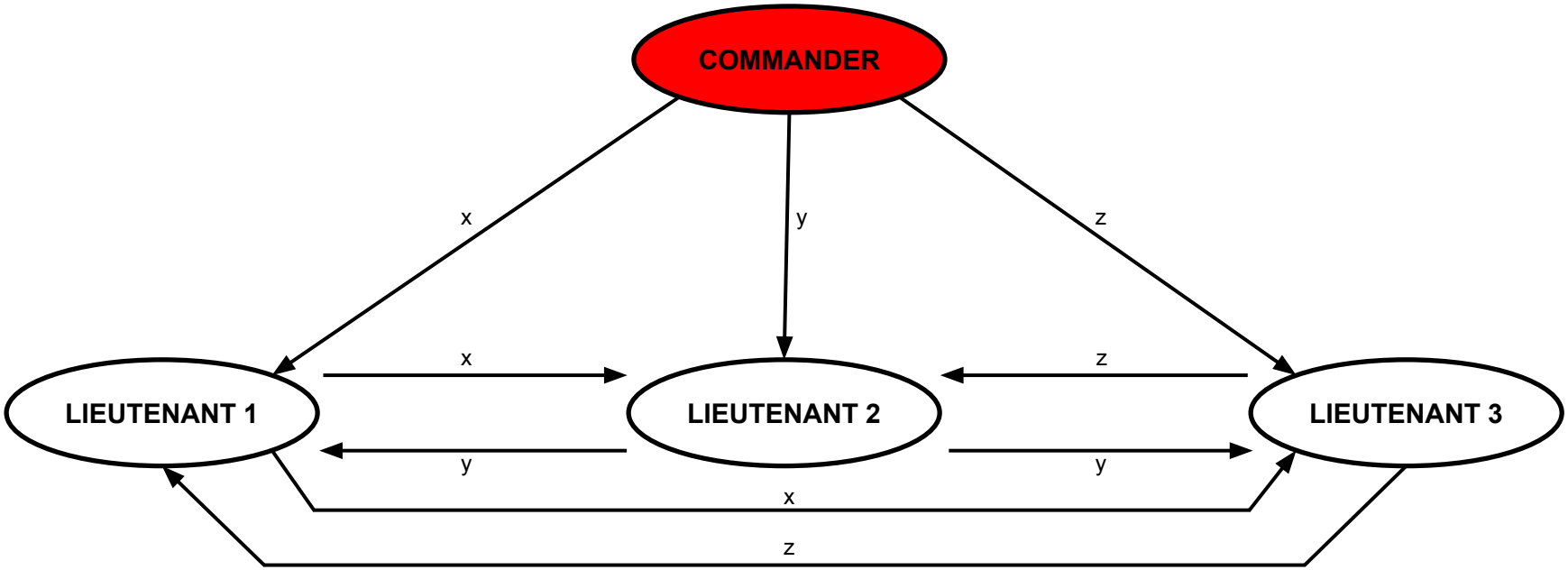
Algorithm OM(m), $m > 0$

1. The commander sends their value to every lieutenant.
2. For each i,
   a. Lieutenant i receives a value v_i from the commander. Default to RETREAT if they receive no value.
   b. Lieutenant i acts as the commander in OM($m-1$) to send the message to each of the remaining $n-2$ lieutenants.
3. For each i, and each j not equal to i,
   a. let v_j be the value Lieutenant i received from Lieutenant j in step (2b). Default to RETREAT if Lieutenant i received no value from Lieutenant j.
   b. Lieutenant i uses the value $majority(v_1, ..., v_{n-1})$.

OM(m) Commander is Loyal



OM(m) Commander is a Traitor



Signed Messages

A4. (a) A loyal general's signature cannot be forged, and any alterations of the contents of their signed messages can be detected.

A4. (a) A loyal general's signature cannot be forged, and any alterations of the contents of their signed messages can be detected.
(b) Anyone can verify the authenticity of a general's signature.

We need some requirements for how the generals decide which order to follow:

We need some requirements for how the generals decide which order to follow:

1. If the set V consists of the single element v, then $\mathit{choice}(V) = v$.

We need some requirements for how the generals decide which order to follow:

1. If the set V consists of the single element v, then $\mathit{choice}(V) = v$.
2. $\mathit{choice}(\emptyset) =$ RETREAT, where $\emptyset$ is the empty set.

- The value x signed by General i is denoted as $x : i$.

- The value x signed by General i is denoted as $x : i$.
- That means $x : i : j$ is the value x signed by General i and then General j.

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.
2. For each i,

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.
2. For each i,
    a. If Lieutenant i receives a message from the commander of form $v : 0$ and they have not received any other order, then:

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.
2. For each i,
    a. If Lieutenant i receives a message from the commander of form $v : 0$ and they have not received any other order, then:
        i. they set $V_i = \{v\}$.

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.
2. For each i,
    a. If Lieutenant i receives a message from the commander of form $v : 0$ and they have not received any other order, then:
        i. they set $V_i = \{v\}$.
        ii. they send the message $v : 0 : i$ to every other lieutenant.

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.
2. For each i,
   a. If Lieutenant i receives a message from the commander of form $v : 0$ and they have not received any other order, then:
      i. they set $V_i = \{v\}$.
      ii. they send the message $v : 0 : i$ to every other lieutenant.
   b. If Lieutenant i receives a message of the form $v : 0 : j_1 : ... : j_k$ and v is not in V_i, then:

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.
2. For each i,
   a. If Lieutenant i receives a message from the commander of form $v : 0$ and they have not received any other order, then:
      i. they set $V_i = \{v\}$.
      ii. they send the message $v : 0 : i$ to every other lieutenant.
   b. If Lieutenant i receives a message of the form $v : 0 : j_1 : \ldots : j_k$ and v is not in V_i, then:
      i. they add v to V_i.

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

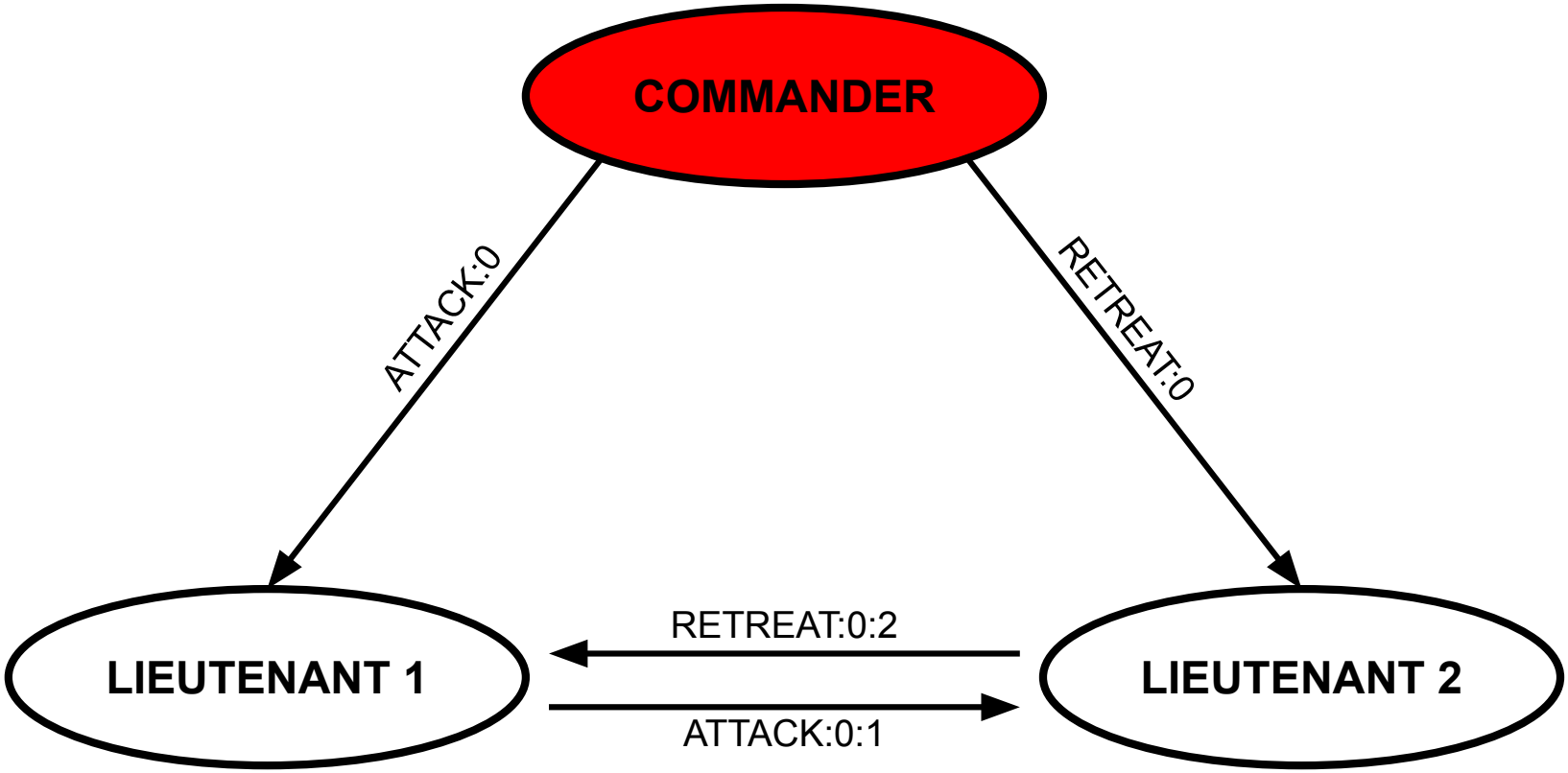
1. The commander signs and sends their value to every lieutenant.
2. For each i,
   a. If Lieutenant i receives a message from the commander of form $v : 0$ and they have not received any other order, then:
      i. they set $V_i = \{v\}$.
      ii. they send the message $v : 0 : i$ to every other lieutenant.
   b. If Lieutenant i receives a message of the form $v : 0 : j_1 : ... : j_k$ and v is not in V_i, then:
      i. they add v to V_i.
      ii. if $k < m$, then they send the message $v : 0 : j_1 : ... : j_k : i$ to every other lieutenant, except for $j_1, ..., j_k$.

Algorithm SM(m), $m > 0$

Initially $V_i = \{\}$

1. The commander signs and sends their value to every lieutenant.
2. For each i,
   a. If Lieutenant i receives a message from the commander of form $v : 0$ and they have not received any other order, then:
      i. they set $V_i = \{v\}$.
      ii. they send the message $v : 0 : i$ to every other lieutenant.
   b. If Lieutenant i receives a message of the form $v : 0 : j_1 : ... : j_k$ and v is not in V_i, then:
      i. they add v to V_i.
      ii. if $k < m$, then they send the message $v : 0 : j_1 : ... : j_k : i$ to every other lieutenant, except for $j_1, ..., j_k$.
   c. For each i, when Lieutenant i receives no more messages, they follow the result from $choice(V_i)$.

OM(m) Commander is a Traitor



Applications

- Computer components.

Applications

- Computer components.
- Nodes on a network.

Applications

- Computer components.
- Nodes on a network.
- Blockchain

References

[1] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem", ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, 1982, 382-401.

[2] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults.", J. ACM Transactions 27, 2, 1980, 228-234