A review of the Byzantine problem

- There is an **enemy city** and a group of **General** i, each deciding to reach **an agreed upon plan** (which is the exact definition of **consensus**) to whether *Attack* or *Retreat*

A review of the Byzantine problem

- There is an **enemy city** and a group of **General** i, each deciding to reach **an agreed upon plan** (which is the exact definition of **consensus**) to whether *Attack* or *Retreat*
- and each general i is equipped with a messaging method for sending value $v(i)(\in \{\text{Attack, Retreat}\})$ for communicating with each other

A review of the Byzantine problem

- There is an **enemy city** and a group of **General** i, each deciding to reach **an agreed upon plan** (which is the exact definition of **consensus**) to whether *Attack* or *Retreat*
- and each general i is equipped with a messaging method for sending value $v(i)(\in \{\text{Attack, Retreat}\})$ for communicating with each other
- AND there are a bunch of **traitorous generals** sending **conflicting** messages, aim to prevent **loyal generals to reach a plan**

A review of the Byzantine problem

- There is an **enemy city** and a group of **General** i, each deciding to reach **an agreed upon plan** (which is the exact definition of **consensus**) to whether *Attack* or *Retreat*
- and each general i is equipped with a messaging method for sending value $v(i)(\in \{\text{Attack, Retreat}\})$ for communicating with each other
- AND there are a bunch of **traitorous generals** sending **conflicting** messages, aim to prevent **loyal generals to reach a plan**
- In fact, they send **arbitrary messages** to other generals.

A review of the Byzantine problem

- There is an **enemy city** and a group of **General** i, each deciding to reach **an agreed upon plan** (which is the exact definition of **consensus**) to whether *Attack* or *Retreat*
- and each general i is equipped with a messaging method for sending value $v(i)(\in \{\text{Attack, Retreat}\})$ for communicating with each other
- AND there are a bunch of **traitorous generals** sending **conflicting** messages, aim to prevent **loyal generals to reach a plan**
- In fact, they send **arbitrary messages** to other generals.
- In order for them to a reach consensus, two conditions must be satisfied:

A review of the Byzantine problem

- There is an **enemy city** and a group of **General** i, each deciding to reach **an agreed upon plan** (which is the exact definition of **consensus**) to whether *Attack* or *Retreat*
- and each general i is equipped with a messaging method for sending value $v(i)(\in \{\text{Attack, Retreat}\})$ for communicating with each other
- AND there are a bunch of **traitorous generals** sending **conflicting** messages, aim to prevent **loyal generals to reach a plan**
- In fact, they send **arbitrary messages** to other generals.
- In order for them to a reach consensus, two conditions must be satisfied:
  1. Every loyal general must obtain the same information $v(1), v(2) \ldots, v(n)$
  2. The value sent by a loyal general should be used by all loyal generals

A review of the Byzantine problem *cont.*

How should the generals send their messages?

A review of the Byzantine problem *cont.*

How should the generals send their messages?
Let's examine how **a single general i should send the message $v(i)$** that is formally defined by:

A review of the Byzantine problem *cont.*

How should the generals send their messages?
Let's examine how **a single general i should send the message $v(i)$** that is formally defined by: *why??*

A review of the Byzantine problem *cont.*

How should the generals send their messages?
Let's examine how **a single general i should send the message $v(i)$** that is formally defined by: (which is made by grouping generals into two groups, namely **commander and lieutenant generals**)

Definition (Byzantine Generals Problem)

A **commanding general** must send an order to his $n-1$ **lieutenant generals** s.t.:
IC1. All loyal lieutenants obey the same order.
IC2. If the commanding general is loyal, then every loyal lieutenant obeys the order he sends.

A review of the Byzantine problem *cont.*

How should the generals send their messages?
Let's examine how **a single general i should send the message $v(i)$** that is formally defined by: (which is made by grouping generals into two groups, namely **commander and lieutenant generals**)

Definition (Byzantine Generals Problem)

A **commanding general** must send an order to his $n-1$ **lieutenant generals** s.t.:
IC1. All loyal lieutenants obey the same order.
IC2. If the commanding general is loyal, then every loyal lieutenant obeys the order he sends.

Note that IC2 $\implies$ IC1

A review of the Byzantine problem *cont.*

How should the generals send their messages?
Let's examine how **a single general i should send the message $v(i)$** that is formally defined by: (which is made by grouping generals into two groups, namely **commander and lieutenant generals**)

Definition (Byzantine Generals Problem)

A **commanding general** must send an order to his $n-1$ **lieutenant generals** s.t.:
IC1. All loyal lieutenants obey the same order.
IC2. If the commanding general is loyal, then every loyal lieutenant obeys the order he sends.

Note that IC2 $\implies$ IC1
* Also note that,E.g. when General n sends $v(n)$ lieutenant $n-1$ retrieves a message from $n-2$ generals and then apply a function $Majority(v(1), v(2), \ldots, v(n-2))$ and adds $v(n)$ to the list V_1

A review of the Byzantine problem *cont.*

Lamport gave a recursive algorithm based on **majority function** for the mentioned problem in case of having **oral messages** whose content is solely managed by the sender.

1. **It is expensive** $\rightarrow O(n!)$
2. It only works only, in case of having m traitors, for $n \geq 3m + 1$ generals

A review of the Byzantine problem *cont.*

Lamport gave a recursive algorithm based on **majority function** for the mentioned problem in case of having **oral messages** whose content is solely managed by the sender.

Unfortunately, the algorithm there are two problems concerning this algorithm:

1. **It is expensive** $\rightarrow O(n!)$
2. It only works only, in case of having m traitors, for $n \geq 3m + 1$ generals

A review of the Byzantine problem *cont.*

Lamport gave a recursive algorithm based on **majority function** for the mentioned problem in case of having **oral messages** whose content is solely managed by the sender.

Unfortunately, the algorithm there are two problems concerning this algorithm:

1. **It is expensive** $\rightarrow O(n!)$
2. It only works only, in case of having m traitors, for $n \geq 3m + 1$ generals

In order to deal with problem 2, he proposed an algorithm based on **unforgeable messages.**