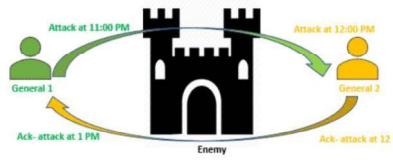
Two General Problem

- This problem seems to be very simple, but this was unsolved as of today. So, let's understand this in detail.
- This problem states a scenario where two generals are attacking a common enemy, both the generals has its own army and they will be able to defeat the enemy only if they both attack at same time, if any one of them does not attack then they will not be able to win this battle.
- Now the problem here is the communication between two generals, for them to communicate they need to exchange the messages.

Two General Problem

- First general sends a messenger across the enemy camp that need to share the time of the attack to second general, now there may be chance that messenger is captured by enemy army and they distort the message and the correct timing details is not passed to second general as shown in above example.
- Once the information is received by second general then acknowledgement of that need to be send to first general and again that messenger can be captured by army and messenger share some other timing of the attack and this acknowledgement cycle will keep on going. So, this problem seems to be unsolved.

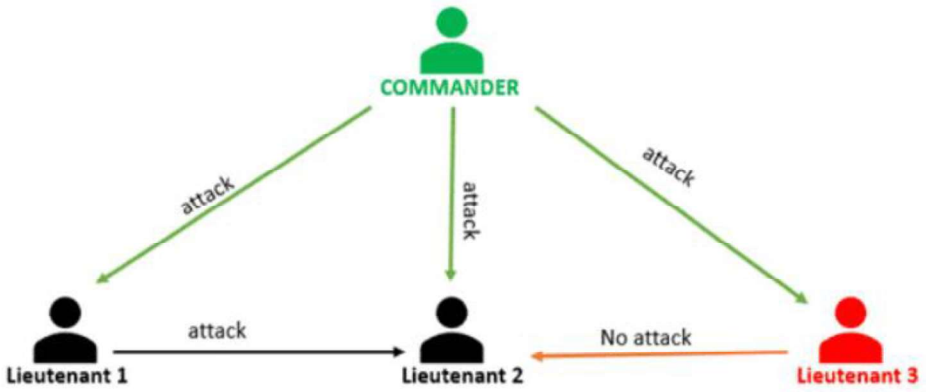


Byzantine Generals PRoblem

- Byzantine Generals Problem is advance version of “Two general problem” where there can be many generals and they need not to agree only on time of attack but here one or more than one general can be traitor.
- So the question comes how consensus can be reached, answer to that is, consensus is reached when 2/3 of the actors are honest. If the traitors are more than 1/3, consensus is not reached, the armies do not coordinate their attack and the enemy wins.

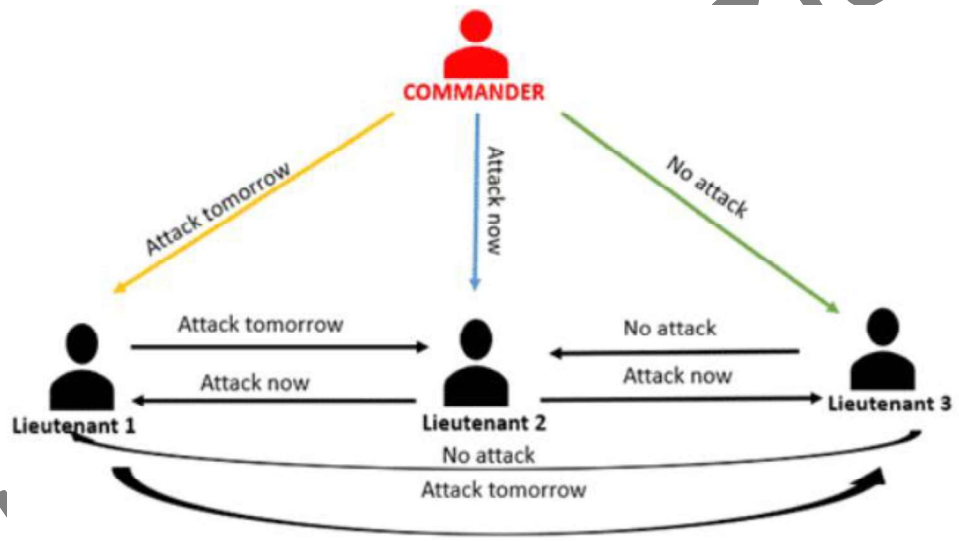
Byzantine Generals PRoblem

- This should be clear from below diagram where we are seeing Lieutenant 2's point of view.
- In this scenario, Lieutenant 3 is traitor. As shown in above diagram,
  - Commander sends "attack" command to all Lieutenants
  - Lieutenant-1 sends "attack" command to Lieutenant-2
  - Lieutenant-3 sends "no attack" command to Lieutenant-2
- Now Lieutenant-2 has 2 "attack" command and 1 "no attack" command so majority is "attack" so he will go with "attack" command.



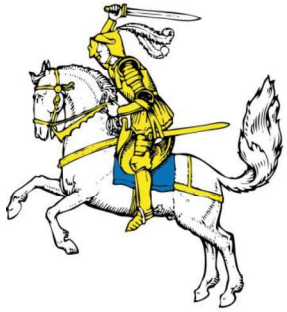
Byzantine Generals Problem

- Let's consider another scenario where commander is traitor.
- In this scenario, commander is traitor. As shown in diagram,
  - Commander sends "attack tomorrow" command to Lieutenant-1, "attack now" command to Lieutenant-2 and "no attack" command to Lieutenant-3
  - Lieutenant-1 sends "attack tomorrow" command to Lieutenant-2 and Lieutenant-3.
  - Lieutenant-2 sends "attack now" command to Lieutenant-1 and Lieutenant-3.
  - Lieutenant-3 sends "no attack" command to Lieutenant-1 and Lieutenant-2.
- Now Lieutenant-1 will have "attack tomorrow", "attack now" and "no attack" command. since there is no majority so he will retreat.
- same apply to other Lieutenants as well.



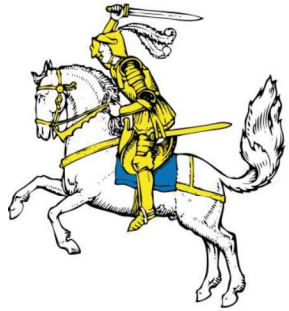
Byzantine Generals Problem

Byzantine Generals Problem

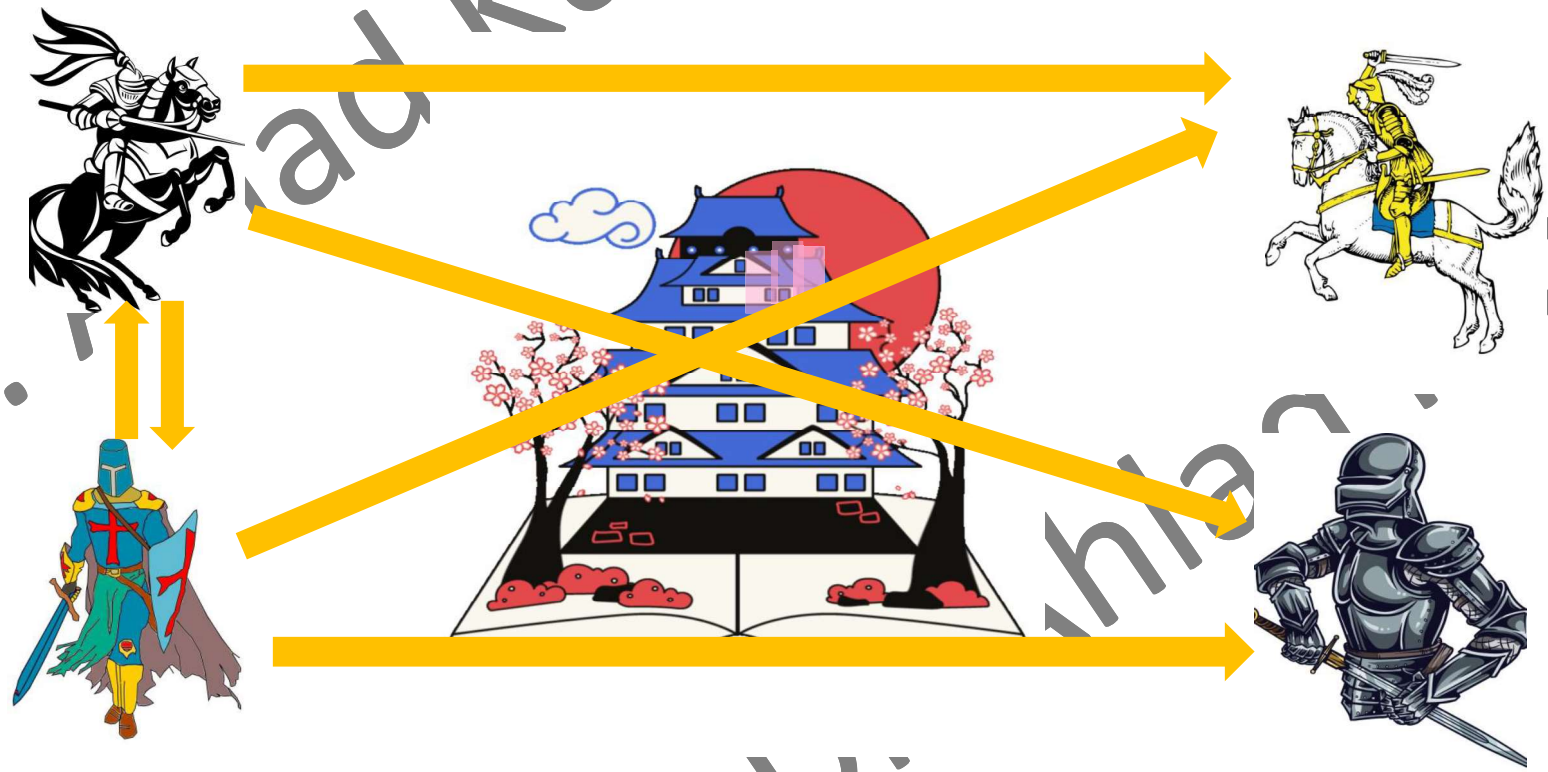


Byzantine Generals Problem

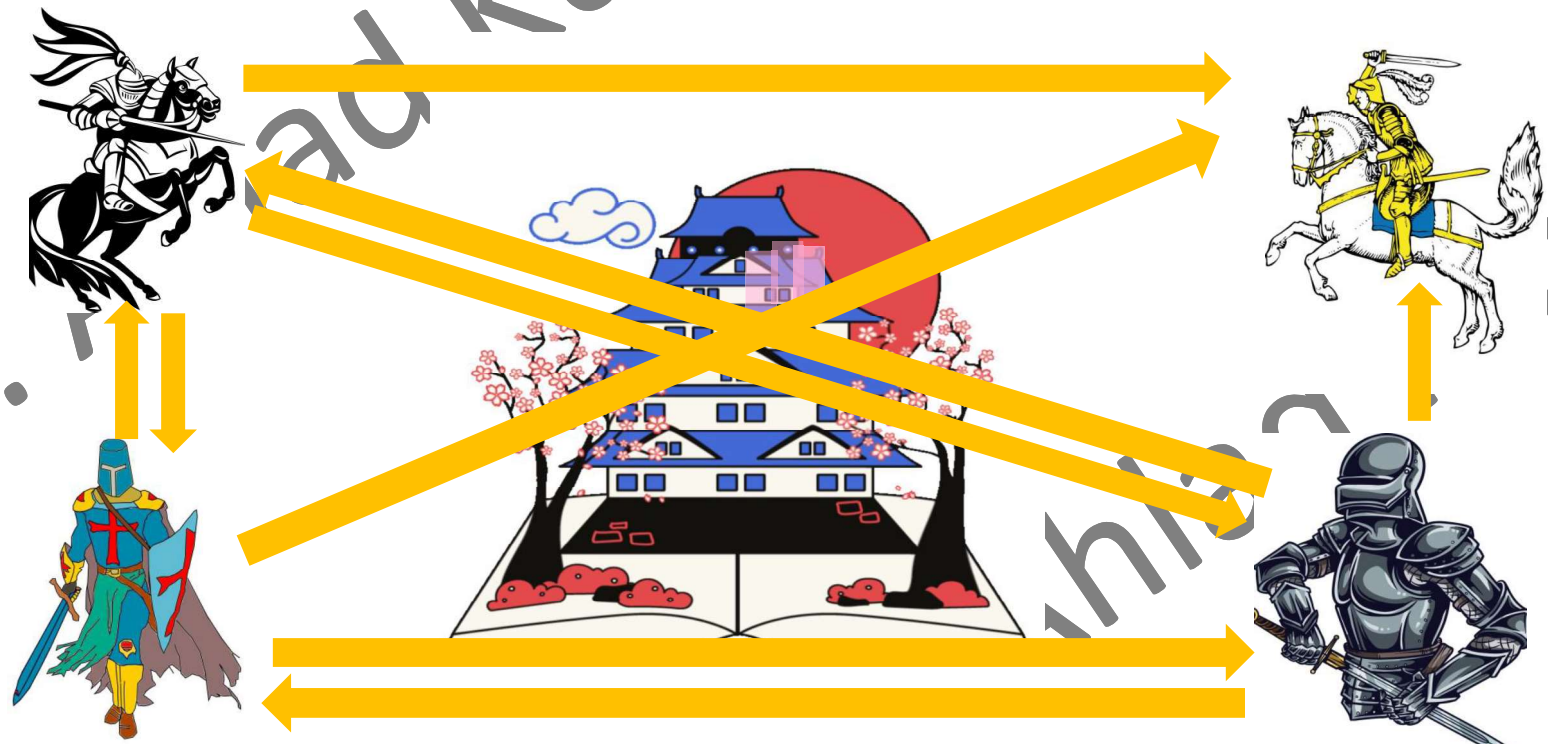
Yellow: Fight



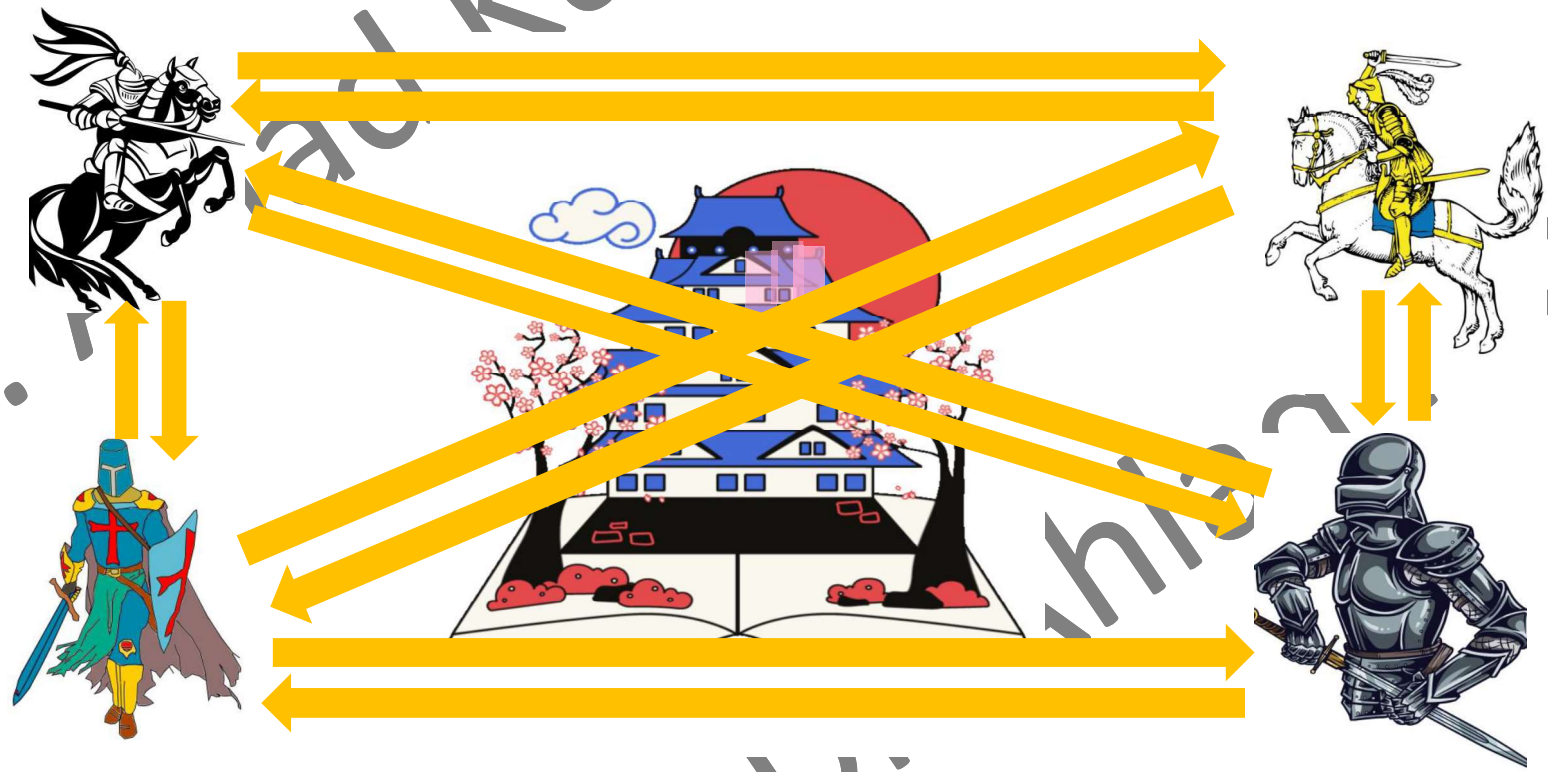
Byzantine Generals Problem



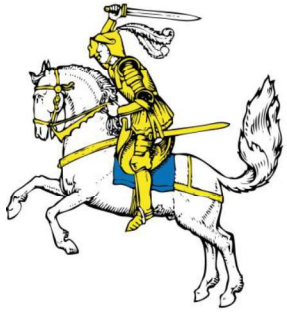
Byzantine Generals Problem

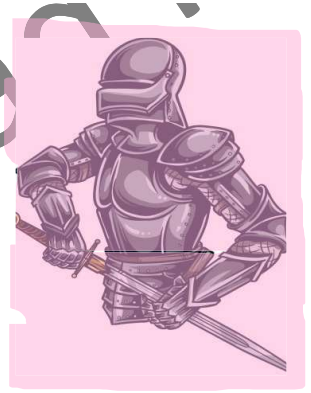


Byzantine Generals Problem



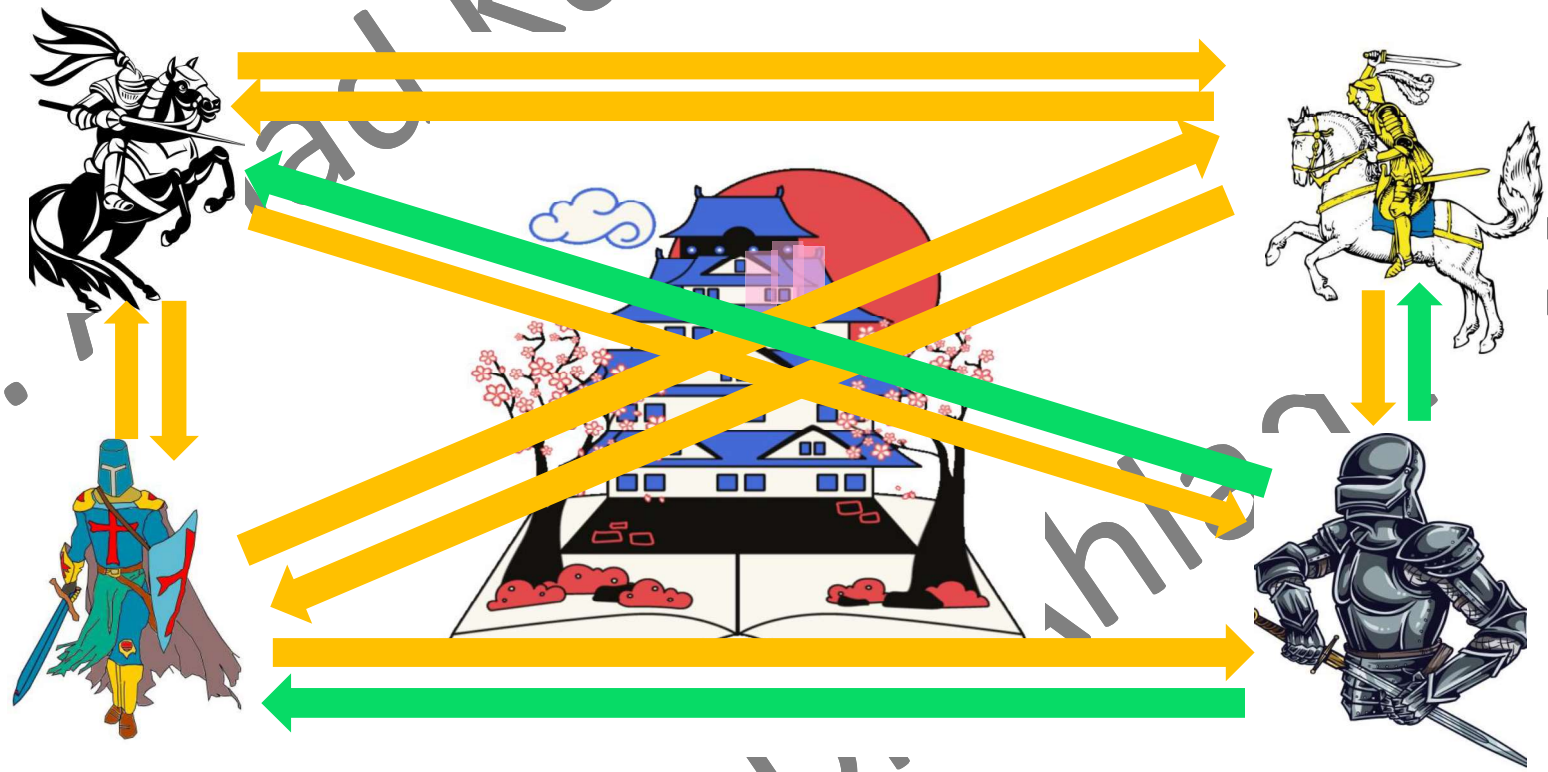
Byzantine Generals Problem



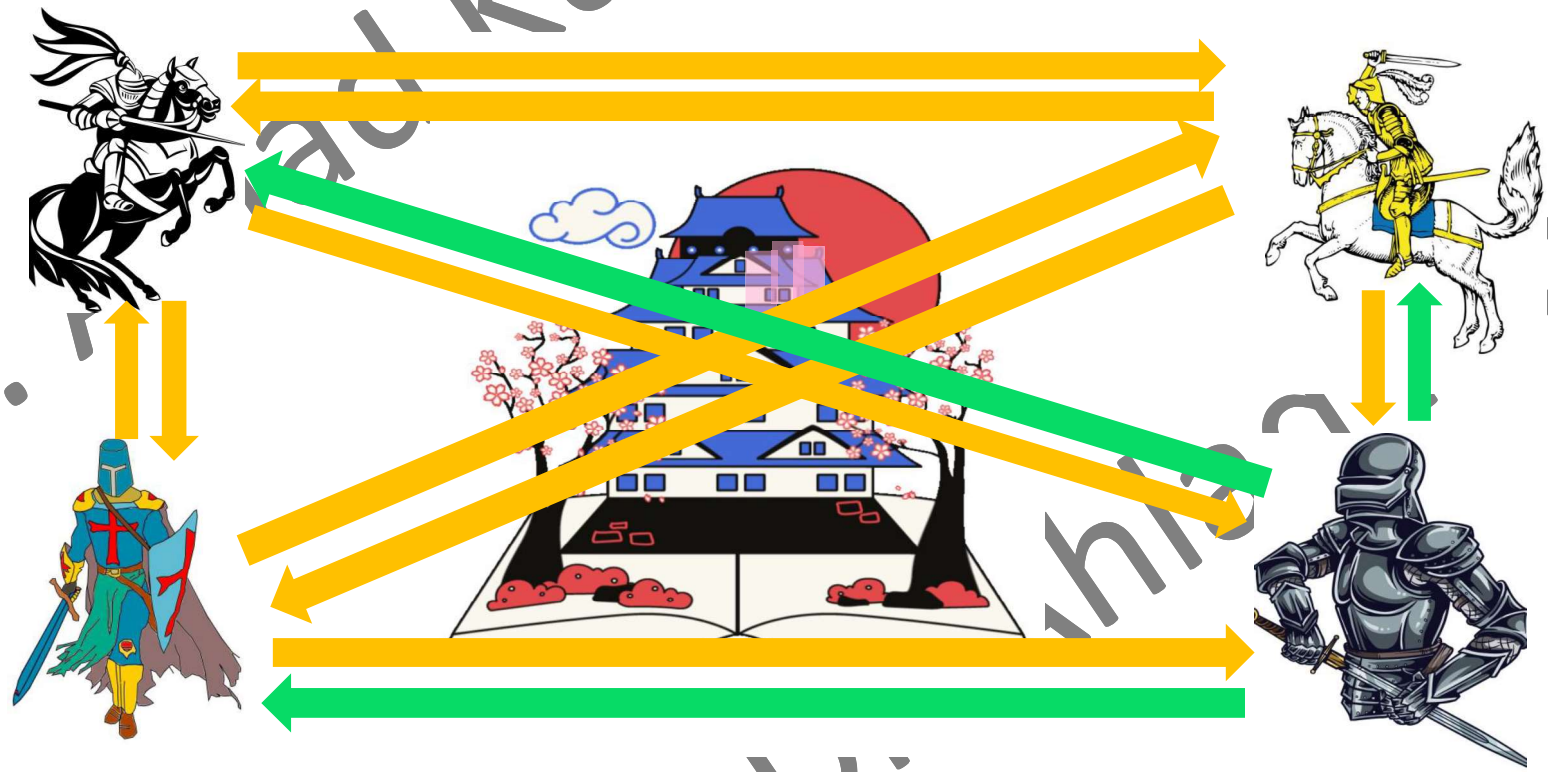


Traitor

Byzantine Generals Problem



Byzantine Generals Problem



Practical Byzantine Fault Tolerance

by

Miguel Castro

Abstract

Our growing reliance on online services accessible on the Internet demands highly-available systems that provide correct service without interruptions. Byzantine faults such as software bugs, operator mistakes, and malicious attacks are the major cause of service interruptions. This thesis describes a new replication algorithm, BFT, that can be used to build highly-available systems that tolerate Byzantine faults. It shows, for the first time, how to build Byzantine-fault-tolerant systems that can be used in practice to implement real services because they do not rely on unrealistic assumptions and they perform well. BFT works in asynchronous environments like the Internet, it incorporates mechanisms to defend against Byzantine-faulty clients, and it recovers replicas proactively. The recovery mechanism allows the algorithm to tolerate any number of faults over the lifetime of the system provided fewer than 1/3 of the replicas become faulty within a small window of vulnerability.

Quantum Computing in Blockchain

- Quantum computing is the use of quantum phenomena such as superposition and entanglement to perform computation. Computers that perform quantum computations are known as quantum computers.
- Quantum superposition is a fundamental principle of quantum mechanics. It states that, much like waves in classical physics, any two (or more) quantum states can be added together ("superposed") and the result will be another valid quantum state; and conversely, that every quantum state can be represented as a sum of two or more other distinct states.
- Quantum entanglement is a physical phenomenon that occurs when a pair or group of particles is generated, interact, or share spatial proximity in a way such that the quantum state of each particle of the pair or group cannot be described independently of the state of the others, including when the particles are separated by a large distance.
- In quantum computing, a qubit or quantum bit (sometimes qbit) is the basic unit of quantum information—the quantum version of the classical binary bit physically realized with a two-state device. A qubit is a two-state (or two-level) quantum-mechanical system, one of the simplest quantum systems displaying the peculiarity of quantum mechanics.

Quantum Blockchain

- Quantum blockchain can be understood as a decentralized, encrypted and distributed database based on quantum computation and quantum information theory.
- Once the data is recorded in the quantum blockchain, it will not be maliciously tampered with.
- In recent years, the development of quantum computation and quantum information theory makes more and more researchers focus on the research of quantum blockchain.

Effects of Quantum Computing on Blockchain

- In the context of quantum computing, we are confronted with two aspects of invalidating the promises of blockchain. First, the inversion of hashes is assumed to be computationally difficult.
- If this can be dramatically simplified by a quantum computer, the authenticity of the upstream blockchain can no longer be guaranteed and the authenticity of entries in the blockchain is compromised.
- Blockchain relies on the computation of hashes to provide security against modification of the past blocks.

Effects of Quantum Computing on Blockchain

- Grover's algorithm is specifically a solution to the problem of finding a pre-image of a value of a function that is difficult to invert.
- If we are given a signature that is the hash value of some data $s = H(d)$, and the function $H(d)$ can be implemented on a quantum computer, then Grover's algorithm allows us to find d for a given s in time of order $O(\sqrt{n})$ where n is the size of the space of valid hashes.
- In other words, it allows us to generate hash collisions more efficiently than brute force search, which would be $O(n)$. For a hash of length k bits this means that we have a significant speedup by a factor of $2^{k/2}$.
- This can be very large even for small values of k.

Effects of Quantum Computing on Blockchain

- Grover's algorithm can be used in two ways to attack the blockchain.
- The first, and most obvious, is that it can be used to search for hash collisions which can be used to replace blocks without disturbing the integrity of the blockchain.
- The second is that it can speed up the generation of hashes, potentially to the point that entire chains of records can be recreated with consistent modified hashes sufficiently quickly to undermine the integrity of the chain.
- In both cases the algorithm is used to find the pre-image of a given value under a difficult to invert function.

Effects of Quantum Computing on Blockchain

- As a secondary threat, in any aspect of a blockchain implementation that uses public/private key cryptography, whether it be in information exchange between parties or in digital signatures, a quantum computer may be able to break the security of the encryption.