The Implemented Byzantine Algorithm

Preconditions: There are  generals whose public-keys are known to everyone and private keys are known only to themselves (e.g. consider RSA-512). The first general in the list is the commander who broadcasts a command (e.g. "attack") to the rest of generals. Any of the  generals (commander or others) can either by a byzantine general or an honest general.

Postconditions: All honest generals are in consensus at the end (either they all output "attack", or they all output "retreat", or they all output "None")

1. The commander (first general) signs the "attack" message and broadcasts it to other generals along with the signature and the commander's public key. For the sake of simplicity, the simulation assumes that the commander always sends the "attack" message. Obviously, you can change the commander's command to "retreat" if you wish. The commander won't be active after this step (no message send/receive).
2. Each general () that receives a message first verifies the correctness of the message and its signature(s). If the message and its signature(s) are verified successfully, the general signs the message to create a new signature (if the verification fails, the message should be discarded). Then,  appends the new signature and 's public key to the message's list of signatures. Finally, the general relays the message and its list of signatures by sending them to everyone else in the network. To verify the list of signatures received along with a message, the general needs to do the following checks:
   * The accompanying public key of every signature must verify that the signature belongs to the message.
   * No general should sign a message more than once, i.e. public keys of signatures must be unique
   * The signature of the commander should be in the list of all signatures.
   * The signature of  should not be in the list of all signatures (otherwise, a party may sign a message more than once)
   * The accompanying public key of every signature should be one of the  public keys assigned to the generals (no unknown public key should be found).
3. After multiple message-passing operations, no message will be exchanged between generals anymore. In this state, which is the final state of the protocol, the honest/non-byzantine generals output their message in the following way:
   * If a general has received and verified conflicting messages (i.e., "attack" and "retreat"), it outputs "None".
   * If a general has not received any verified message, it outputs "None".
   * If a general has received and verified a single message (either "attack" or "retreat") once or multiple times, then it outputs the message.

Detailed of the Given Code

In the given Python code, the *generals*list contains  generals where  can be modified by changing the value of variable *no\_of\_generals*. The commander is stored at the beginning of the list (*generals [0]*). Each general, including the commander, can be byzantine with a probability of 80%.

To represent generals, a class called *General* is implemented that stores the general's index *(index:* int), private key *(d:* *int*), and public key *(n, e: int,* *int*). It also stores a bool variable called *byzantine*which is *True* if the general is byzantine.

Also, the messages exchanged by the mentioned algorithm are instantiated from the given class *Message*. The class stores the source (*src: int*), destination (*trg: int*), message text (*text: str*), and its list of order pairs of signatures and public keys (*sig\_set: list*). It also stores the bool variable *relay* which is *True*if the message is not a broadcast from the commander.

Each general is a thread running concurrently with other generals. All generals have the same main function called *general\_run*. The messages are exchanged using a global variable called *shared\_inbox* whose access is synchronized by a Lock object (If you are not familiar with the concept of multithreading and synchronization, you don't need to be worried about it as the code is already implemented).

In the given code, the byzantine commander tries to sabotage the consensus of honest generals by sending the first half of generals the "attack" message and sending the second half of generals the "retreat" message. Also, the byzantine generals (other than the commander) try to sabotage the consensus of honest generals by relaying the opposite message whenever they receive a message (relay the "attack" message when the received message is "retreat" and relay the "retreat" message otherwise).

Assignments Testing Scenarios

Please do the following:

1. (30 points) Run the given code 3 times for each of the following cases. Record and report the generals' output messages produced by the program and specify whether the commander was byzantine or not in each run:

Scenario 1: no\_of\_generals = 3, byzantine\_prob = 0.5

Run 1

Honest general 1's output: None

Honest general 2's output: None

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 2

Run2

Honest general 1's output: None

Byzantine general 2's output doesn't matter!

Commander is byzantine!

Are all honest generals in-consensus?

Number of exchanged messages: 2

Run2

Honest general 1's output: None

Byzantine general 2's output doesn't matter!

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 2

Scenario 2: no\_of\_generals = 6, byzantine\_prob = 0.3

Run 1

Honest general 1's output: None

Honest general 3's output: None

Honest general 2's output: None

Byzantine general 4's output doesn't matter!

Honest general 5's output: None

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 5

Run 2

Honest general 1's output: None

Honest general 4's output: None

Honest general 3's output: None

Byzantine general 2's output doesn't matter!

Honest general 5's output: None

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 5

Run 3

Honest general 2's output: None

Honest general 1's output: None

Byzantine general 5's output doesn't matter!

Honest general 4's output: None

Byzantine general 3's output doesn't matter!

Commander is byzantine!

Are all honest generals in-consensus?

Number of exchanged messages: 5

Scenario 3: no\_of\_generals = 10, byzantine\_prob = 0.7

Run 1

Honest general 1's output: None

Honest general 2's output: None

Byzantine general 3's output doesn't matter!

Byzantine general 4's output doesn't matter!

Honest general 5's output: None

Honest general 8's output: None

Byzantine general 6's output doesn't matter!

Honest general 7's output: None

Byzantine general 9's output doesn't matter!

Commander is byzantine!

Are all honest generals in-consensus?

Number of exchanged messages: 9

Run 2

Byzantine general 1's output doesn't matter!

Byzantine general 2's output doesn't matter!

Byzantine general 3's output doesn't matter!

Byzantine general 7's output doesn't matter!

Byzantine general 5's output doesn't matter!

Honest general 8's output: None

Honest general 9's output: None

Byzantine general 4's output doesn't matter!

Honest general 6's output: None

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 9

Run 3

Byzantine general 1's output doesn't matter!

Honest general 2's output: None

Honest general 3's output: None

Honest general 4's output: None

Byzantine general 5's output doesn't matter!

Byzantine general 8's output doesn't matter!

Byzantine general 7's output doesn't matter!

Byzantine general 6's output doesn't matter!

Honest general 9's output: None

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 9

2. (20 points) Find out the average # of exchanged messages under each of the three previous scenarios.

**Average of Exchanged messages: 2**

**Average of Exchanged messages: 5**

**Average of Exchanged messages: 9**

3. (25 points) Modify the code so that each byzantine general (except the commander) sabotages the consensus of honest generals by relaying two conflicting messages whenever it receives a verified message. Then, test your code by running it three times for no\_of\_generals = 7, and byzantine\_prob = .5. Record and report the result. Does the given algorithm fail to provide consensus?

Run1

Honest general 1's output: None

Honest general 4's output: None

Honest general 3's output: None

Byzantine general 5's output doesn't matter!

Byzantine general 6's output doesn't matter!

Byzantine general 2's output doesn't matter!

Commander is byzantine!

Are all honest generals in-consensus?

Number of exchanged messages: 6

Run2

Honest general 1's output: None

Byzantine general 5's output doesn't matter!

Honest general 2's output: None

Honest general 6's output: None

Byzantine general 3's output doesn't matter!

Byzantine general 4's output doesn't matter!

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 6

Run3

Byzantine general 3's output doesn't matter!

Honest general 2's output: None

Honest general 4's output: None

Byzantine general 1's output doesn't matter!

Byzantine general 5's output doesn't matter!

Honest general 6's output: None

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 6

4. (25 points) Modify the code so that each byzantine general (except the commander) sabotages the consensus of honest generals by not relaying any message whenever it receives a verified message. Then, test your code by running it three times for no\_of\_generals = 7, and byzantine\_prob = .3. Record and report the result. Does the given algorithm fail to provide consensus?

Run1

General # 3 received None

General # 4 received None

General # 5 received None

General # 6 received None

Honest general 1's output: None

Byzantine general 4's output doesn't matter!

Honest general 6's output: None

Honest general 3's output: None

Honest general 5's output: None

Honest general 2's output: None

Commander is byzantine!

Are all honest generals in-consensus?

Number of exchanged messages: 6

Run2

General # 4 received None

General # 5 received None

General # 6 received None

Honest general 3's output: None

Honest general 1's output: None

Byzantine general 2's output doesn't matter!

Byzantine general 4's output doesn't matter!

Honest general 5's output: None

Honest general 6's output: None

Commander is byzantine!

Are all honest generals in-consensus?

Number of exchanged messages: 6

Run3

Honest general 1's output: None

Byzantine general 3's output doesn't matter!

Honest general 4's output: None

Honest general 5's output: None

Byzantine general 2's output doesn't matter!

Byzantine general 6's output doesn't matter!

Commander is honest and wants to attack!

Are all honest generals in-consensus?

Number of exchanged messages: 6