Yusuf – 201703786
Felix Rundstrøm – 202208224
Thomas Andersen - 202207574

Handin 2

DISSY IT 5

Exercise 4.6

Git: https://gitlab.au.dk/dissy_it_5/hand_in_2

```
type Peer struct {
	Port        int
	ledger      *account.Ledger
	open        bool
	name        string
	ip          string
	connections map[int]net.Conn
	mu          sync.Mutex
}
```

Each peer contains its port number, an account ledger, a Boolean value open (true = port is open for communication), its name, its ip and a map of connections.

NewPeer creates a new instance of peer

StartNewNetwork creates a new network with the peer itself as the only member

Connect connects peers to another peer
If the peer is already connected to the peer,
it returns the connection If the peer is not connected to the peer,
it connects to the peer
If the peer is not connected to any peers, it starts a new network

The peers communicate over Handleconnection and handleMessage
When a new peer is initiated it will create its own network and wait for messages
When another peer join the network it'll ask for the firs peers messages using the AskForPeers function. It will then flood a join message to the list of peers

Messages can have 5 different actions.
Join action will make the receiving peer connect to new peer. It is the first message a new peer will send.
askForPeers action sends a list of connected peers
peers action will flood join message containing "join" and a list of connect peers
Transaction action will update the ledger about a transaction that has happen.

Yusuf – 201703786
Felix Rundstrøm – 202208224
Thomas Andersen - 202207574

Floodmessage and floodtransaction both flood a message or a transaction to all connected peers

TO RUN CODE: go run ./ main.go
TO TEST: go test -v

10. For the report: If you did the simple flooding solution, give a scenario of use which leads to eventual consistency and give a scenario of use which does not lead to eventual consistency. For instance, what happens if all peers join the network one by one before any messages are sent?

When each peer join one by one they will receive the set of peers to connect to when they send out a join message. They will thereby have consistency because all the connections are up to date and any messages in the network will reach the peer eventually.

The peers do not send acknowledgements of successful received messages, this can break consistency if a new peer losses connection, messages are dropped or messages are blocked.

11. For the report: Assume we made the following change to the system: When a transaction arrives, it is rejected if executing the transaction would make the `From` account go below 0. Does your system still have eventual consistency? Why or why not?

The same rule would apply to all peers and therefore the transaction request would be rejected by all peers. If for instance a peer has not updated the from account to the same amount as the other peers, and it allows this transaction, the flooding of this mistake does not have any checks to ensure that this is not put in the ledgers of the other peers. A validation process in the ledger updates from other peers or check ins with other peers if this transaction is allowed would mitigate this.

Exercise 5.1

1)

if the original message from the sender was:

100100111101000100000

Yusuf – 201703786
Felix Rundstrøm – 202208224
Thomas Andersen - 202207574

and one time pad was:

0101010111001010101**0**

this means that the encrypted message would be:

1100011000011011101**0**

then the hacker could change the encrypted message to:

1100011000011011101**1**

and the decrypted message(salary would be)

1001001111010001000**1** = 1.084.361

2)

it is an authenticity problem.

The hacker still has no idea what the message contains.

The receiver would know not to use the message if he couldn't authenticate the sender.

3)

because in the book they define it to be broken if and only if the receiver can decrypt the message and read it. but here he is simply changing one bit because he knows how the message is structured.

4)

Yes he could do the attack with stream ciphers, it is very similar to one-time pad.

The attack would is possible.

The next 128 block of bits after the salary, would not be able to be decrypted properly by the receiver.

AES in counter mode is also possible, since we xor the key with the message in a similar way to otp. and the same key is used to decrypt.