CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DIT357	Distributed Systems			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
DIT357-0016-zHR		2025-10-30	11	

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under eximinationen.

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	✓	8.25	8-1-1
2	✓	15	
3	✓	9	
4	✓	14	14-1-0
5	✓	15	
6	✓	10	
7	✓	15	
8	✓	10	
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen		96.25	

[redacted]

CHALMERS

Anonymous code

Anonym kod

DIT357 - 0016 - ZHR

Points for question (to be filled in by teacher)

Poäng på uppgiften (ifylles av lärare) 8.25

Consecutive page no. Löpande sid nr 1

Question no. Uppgift nr 1

Multiple choice Problem 1

1a) 2 1

1b) 3 1

1c) 3 1

1d) 1 1

1e) 4 1

1f) 1 1

1g) 4 1

1h) 5 0.25

1i) 1 —

1j) 4 1

5

a) Messages from mobile app to drone

Message	Purpuse
START	initiate video streaming
STOP	terminate video streaming

Messages from drone to app 2

Message	purpose
VIDEO <data packets>	send video back to the app

Actions done by app:

→ upon receiving the video, display the video

→ When user starts a stream, send START message to drone 2

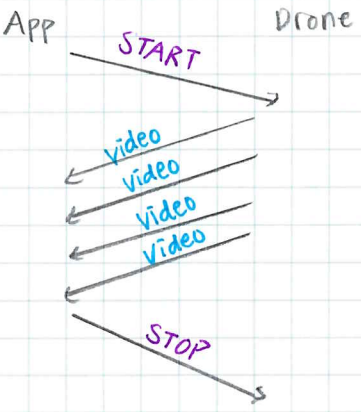
→ When user ends a stream, send STOP message to the drone

Actions done by drone:

→ Upon receiving START msg, start sending video data packets to the app.

→ Upon receiving STOP msg, terminate sending video data packets to app.

NORMAL operation



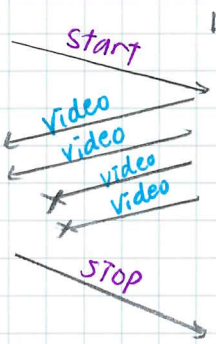
Packet loss situation

1.

→ lost, did not initiate streaming

App can react by sending the START msg again.

2.



2+2

} lost, parts of video did not reach the app

In this case, if the lost packets are less significant, the app does not need to do anything.

5

b) UDP could be used. In this case, we want SPEED! /or latency!

In a long video streaming, some packet losses can be acceptable, and do not negatively impact the flow of streaming. We want to see near real-time updates. So UDP is good, it offers fast delivery, some packet lost / wrong order delivery can be tolerated and create less overhead to achieve low latency!

2

c)

Checksum	1001	0110	0101	1011
+	0110	1110	0010	0001
	10000	0100	0111	1100
	0000	0100	0111	1100
+				1
	0000	0100	0111	1101
+	0000	1111	1100	0100
	0001	0100	0100	0001
1's complement:	1110	1011	1011	1110

wraps around

4

d) The receiver can compute the checksum from the msg to detect bit flips in the process of transmission

1

5

a) Seq = 10, 8 bytes of data (assume the retrasmission is still 8 bytes) 1

b) ACK = 40 (assume the retransmission is with 30 bytes of data) 1

c) Reply 1: ACK = 200
Reply 2: ACK = 300
Reply 3: ACK = 300 4p
Reply 4: ACK = 300

d) Reply 5: ACK = 400

e) Implicit NAKs: 3 same ACKs in a row creates an implicit negative acknowledgement. 1

f) This behaviour is called pipelining, to increse the utilization of the bandwidth so that sender does not need to stop and wait for ACKs for every single segment. 2

CHALMERS

Anonymous code
Anonym kod
DIT357 - 0016 - ZHR

Points for question (to be filled in by teacher)
Poäng på uppgiften (ifylles av lärare)
4

Consecutive page no. Löpande sid nr 5
Question no. Uppgift nr 4

Multiple choice Problem 4

4a) 3 —
4b) 2 |
4c) 1 |
4d) 4 |
4e) 4 |
4f) 4 |
4g) 4 |
4h) 2 |
4i) 1 |
4j) 2 |
4k) 2 |
4l) 2 |
4m) 4 |
4n) 3 |
4o) 1 |

5

a) $\lambda = 5$ req/sec $S = \mu^{-1} = 100$ millisec $= 0.1$ sec $\mu = 10$ req/sec

Processing capacity is $\mu = 10$ req/sec 1

b) $U = \frac{\lambda}{\mu} = \frac{5}{10} = \frac{1}{2} = 50\%$ utilization 2

c) $\bar{N} = \frac{U}{1-U} = \frac{0.5}{1-0.5} = 1$ average number of req. = 1 2

d) $R = \frac{\bar{N}}{X} = \frac{S}{1-U} = \frac{1}{5} = 0.2$ sec 2

x is λ (arrival rate)

e) unresponsive meaning Utilization is 1 or higher.

If $U = 1$, μ remains unchanged, find λ

$1 = \frac{\lambda}{\mu} = \frac{\lambda}{10} \Rightarrow \lambda = 10$ req/sec 2

max 10 req/sec before the system becomes unresponsive.

f) new $S = 200$ millisec $= 0.2$ sec $\lambda = 5$ req/sec

new $\mu = 5$ req/sec

$U = \frac{\lambda}{\mu} = \frac{5}{5} = 1$ $R = \frac{S}{1-U} = \frac{0.2}{1-1}$ division by zero error

The utilization is 1, the system is overloaded. 2

g) Amdahl's law $\text{speedup} \leq \frac{1}{S + \frac{1-S}{N}}$

S = serial portion = 0.4

speedup = 2x faster

from 0.2 sec → 0.1 sec

$2 \leq \frac{1}{0.4 + \frac{0.6}{N}}$

$0.8 + \frac{1.2}{N} \leq 1$

$0.8N + 1.2 \leq N$

$1.2 \leq 0.2N$

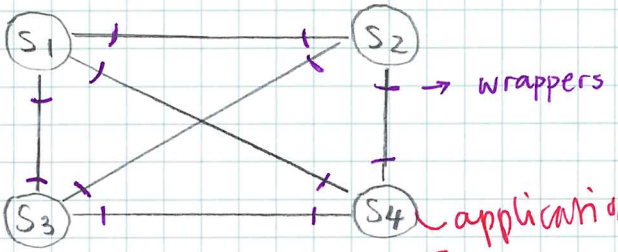
$6 \leq N$

$N \geq 6$ cores needed

3+1p

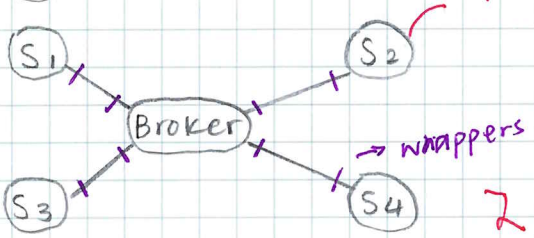
5

a) Service 1, 2, 3, 4 represents Auth, Payment, Catalog and Shipping respectively



2

b)



2

It reduces referential coupling because the services do not need to know each other's address / location / ID. They all talk to the broker, which then transfer the msg onwards to those who are intrested example: pub-sub

c) Without broker: $N \cdot (N-1)$ wrappers $= 4 \cdot 3 = 12$ wrappers $O(N^2)$

With broker: $2N$ wrappers $= 2 \cdot 4 = 8$ wrappers $O(N)$

Broker's disadvantage: Single point of failure! If the broker is down, then all communication will stop.

2

d) Max capacity of broker = 2000 msg / sec

Arrival rate to broker = 5 msg / sec per service

Every time a msg comes in, the broker needs to output the msg to $N-1$ services (all services except the sender)

So: $5 \cdot N \cdot (N-1) = 2000$

$N^2 - N = 400$

$N^2 - N - 400 = 0$

Taking the root of N

N should be roughly around 20.

4

CHALMERS | Anonymous code: DIT357 - 0016 - ZHR | Points for question (to be filled in by teacher) | Poäng på uppgiften (ifylles av lärare): 15 | Consecutive page no. Löpande sid nr: 8 | Question no. Uppgift nr: 7

a) $3! = 3 \times 2 \times 1 = 6$ combinations 2

b)

P1: x ← 1 P2: y ← 1 P3: print(x,y)	P1: x ← 1 P3: print(x,y) P2: y ← 1	P2: y ← 1 P1: x ← 1 P3: print(x,y)	P2: y ← 1 P3: print(x,y) P1: x ← 1	P3: print(x,y) P1: x ← 1 P2: y ← 1	P3: print(x,y) P2: y ← 1 P1: x ← 1
Prints: 11	10	11	01	00	00
e) 2 valid	x	valid	x	x	x

6

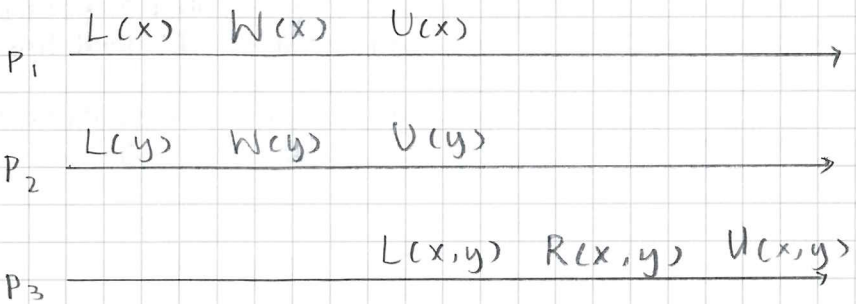
c) 4 different print outputs : 00 , 11 , 10 , 01 2

d) All of them! there is 1 execution per process, no internal execution order for each process. So P_1, P_2, P_3 can run in any order. 1

e) see blue pen above for the valid/invalid marking.

only 2 valid combinations left, both with P3 executing last.

f) We can use grouping operations. Lock the variables until the write is completely. 2



5

a) If Alice and Bob do not have a shared symmetric key already ($K_{A,B}$), then the answer is no. Because Alice's public key K_A^+ is known for everyone, anyone can get K_A^+ from CA, and use it to encrypt data to send to Alice. Alice would not be able to tell who is the sender. But if they have a symmetric key shared, then they can use nonce R_A / R_B to make sure they are talking to each other by using shared key to encrypt / decrypt the nonces. HOWEVER! there is a risk in this where Mallory tries to intercept by pretending to be one of them using multiple sessions to get the nonces (encrypted & decrypted)

2

b) Yes, because of the CA.

3

Alice —(in-person ID verification)→ CA —(K_{CA}^- private key (encrypt); CA encrypt K_a^+ using its own private key)→ certificate for Alice's public key

Bob —(get certificate)→ Certificate —(Use CA's public key to decrypt; K_{CA}^+ to decrypt and get Alice's public key)→ K_a^+

So Bob knows for sure that is Alice's key!

Assume CA is trustworthy and K_{CA}^- is NOT compromised!

Plus: If Bob's use K_A^+ to encrypt data, ONLY Alice - the owner of K_A^-, can decrypt the data!

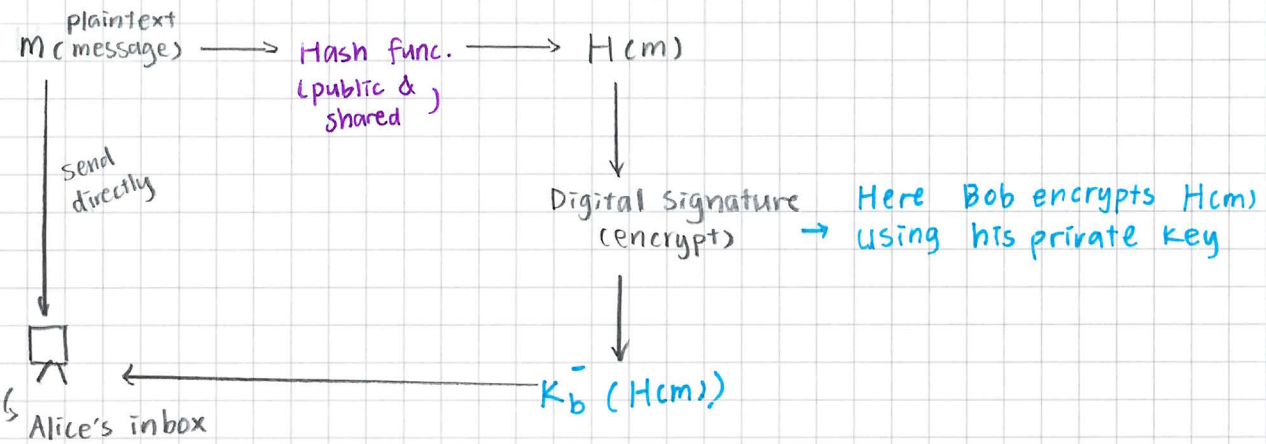
ASSUME! ALICE'S PRIVATE KEY! IS! NOT! COMPROMISED!

5

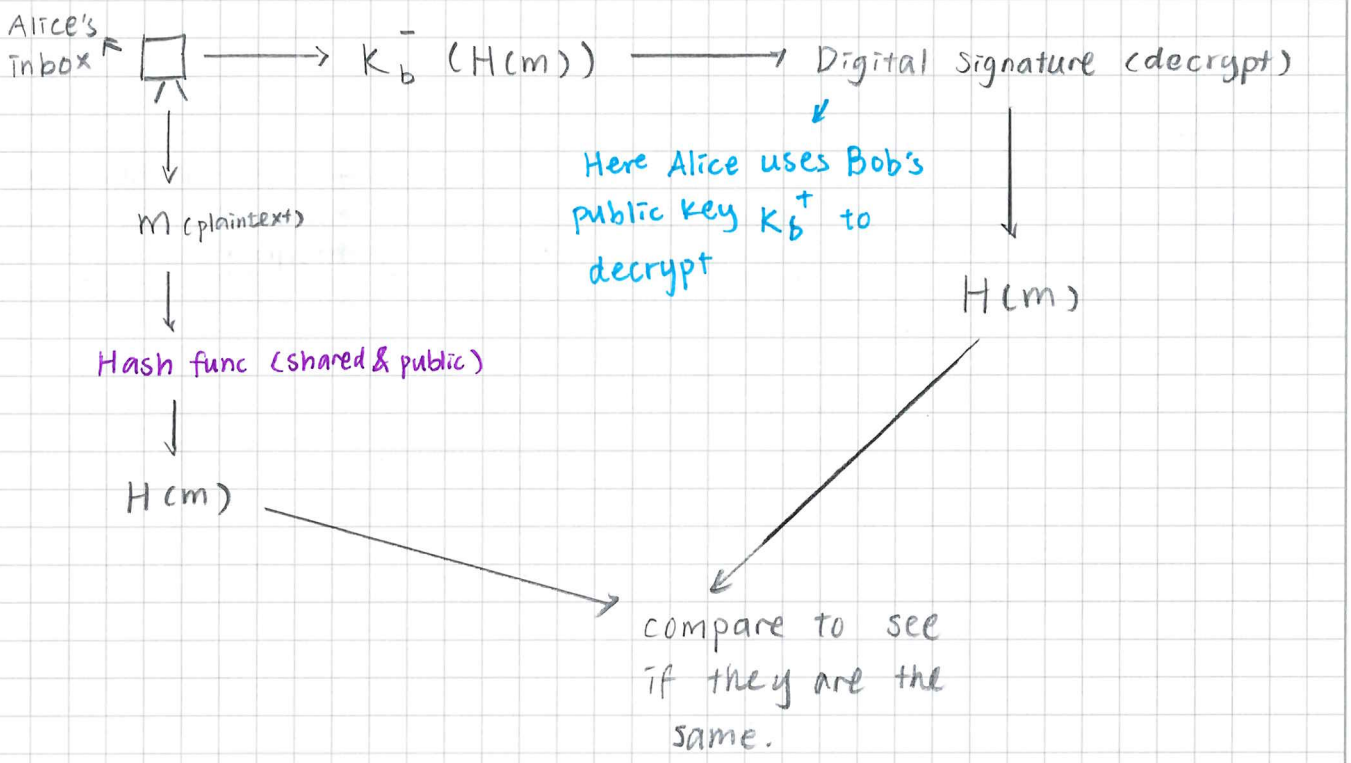
c) I will use diagram to explain:

3

First, Bob needs to send a digitally signed message.



Then, Alice got the msgs and verify the integrity:



Nice clear figure!!
☺

5

d) Authenticity 2

Yes, for authenticity, we need to know the identities and make sure the data is not corrupted.

By using Public - Private key pairs, CA certificates for public keys and the digital signature, we can make sure:

1) Bob and Alice can verify each other's identity through CA certificates (trustworthy) to make sure this public key actually belongs to this person.
2) After obtaining the public key, we make sure only the owner of the private key in this pair can encrypt / decrypt the data, making sure we are actually talking to this person
3) The digital signature + Hash function provides verification of the correctness of the data.