6.4 Question 4

6.4.A In a Kerberos infrastructure, what is the difference between an Authentication Server (AS) and a Ticket-granting Server (TGS)?

The Authentication Server uses a centralized database in order to store all its known passwords. The AS is also sharing unique secret keys with every server connected to it. The first thing in the authentication process is that a client ($\mathcal{A}$) is authenticating itself to the AS. The AS sends a encrypted message to the client containing the shared secret key between client and TGS ($K_{\mathcal{A},TGS}$) and a ticket for the TGS. Only if the client know its secret key it gets access to the information in the received message.
The client then requests the TGS with the use of the received shared key and ticket a shared key (and ticket) to a service provider so the client can use it. Only if the client is authenticated by the AS the TGS will provide such tickets. If this authentication check is successful the TGS will send a message containing a shared key between the client and the service provider and an eventual ticket to this service encrypted with the shared key $K_{\mathcal{A},TGS}$, so only if the client is authenticated and knows its own secret key it can encrypt this message in order to access the services that are provided.

6.4.B What is a Kerberos realm? What are the requirements for supporting authentication even across different realms?

A Kerberos realm is a set of managed nodes that share the same Kerberos database (Lecture, slide 43).
Requirements:

- User ID and hashed password stored in ccentralized database
- Each server shares secret key with one another
- Interoperating realms share secret keys with the Kerberos server in other realms
- Mutual registration between each TGS across each realm

6.4.C What is wrong with my claim above? Why?

I will analyse the statement sentence by sentence:

- *In a decentralized authentication service such as Kerberos V4, the Authentication server encrypts the Ticket that the clients require for accessing a Ticket-granting server (TGS), using AES and the Public Key of each TGS.*
  Kerberos uses a **centralized** AS. Furthermore, the AS encrypts the Ticket and the shared-key between client and the TGS using the **public key of the client**. Furthermore, **DES** is used for encryption instead of AES.
- *Its biggest strength is that it provides authentication which is time independent.*
  All requests and responses contain the timestamps, even those from the TGS and the service provider, hence the authentication request is **time-dependent**
- *On the other side, one of its main technical deficiencies is that it is prone to master database changes that are originated outside the master computer system (e.g. from another Realm). This is what Kerberos 5 attempts to address.*
  The technical deficiencies are unneeded double encryption, usage of vulnerable PCBC, session keys which could be used for replay attacks, and password attacks. These deficiencies are tried to be overcome in Kerberos 5.