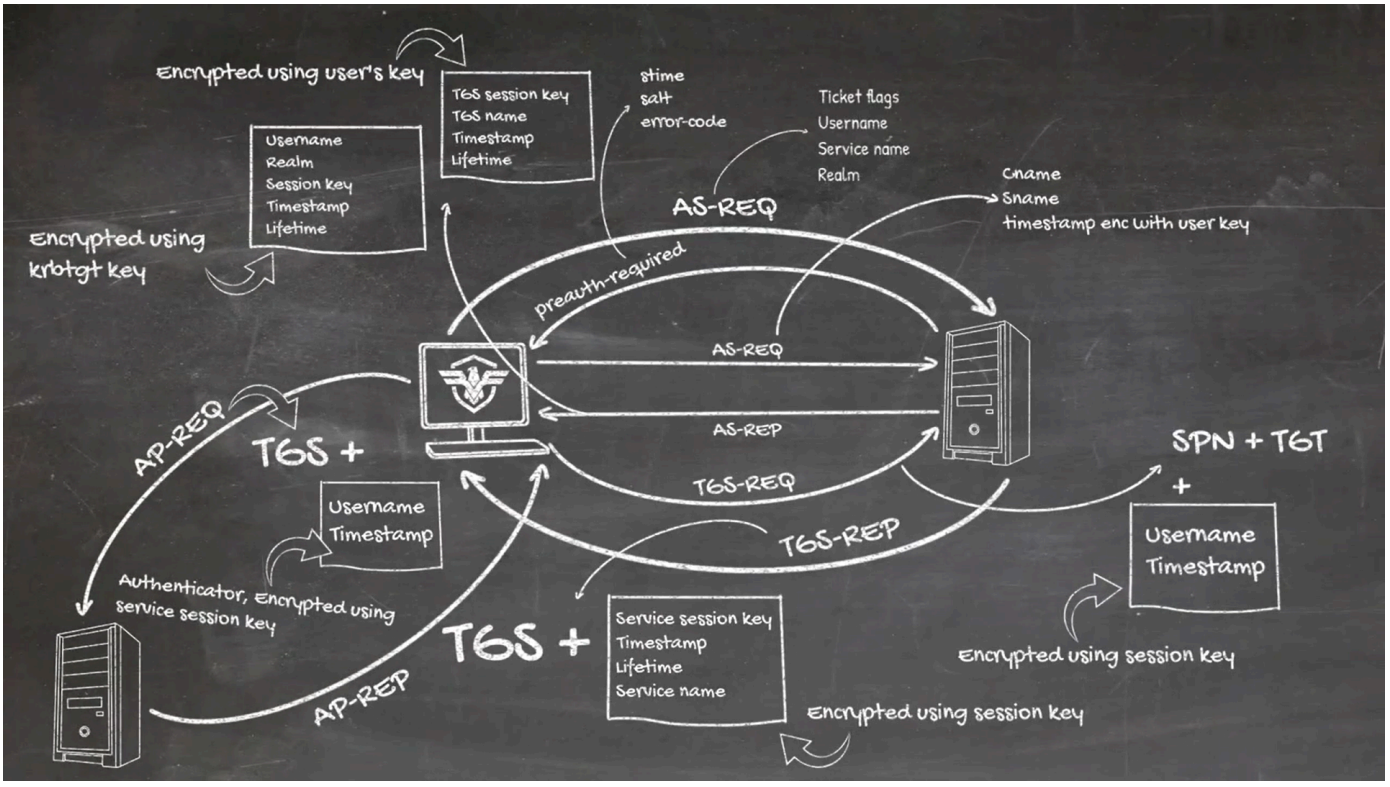
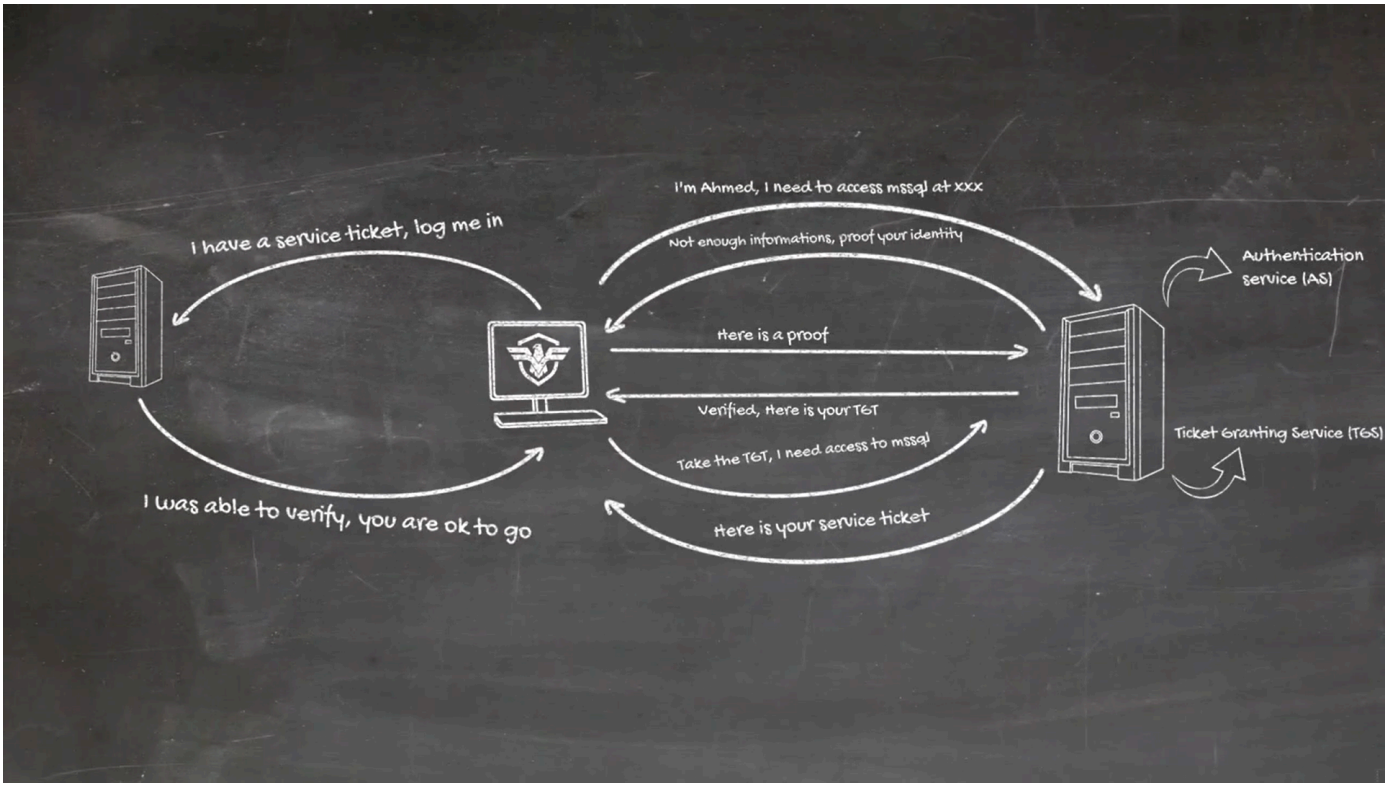
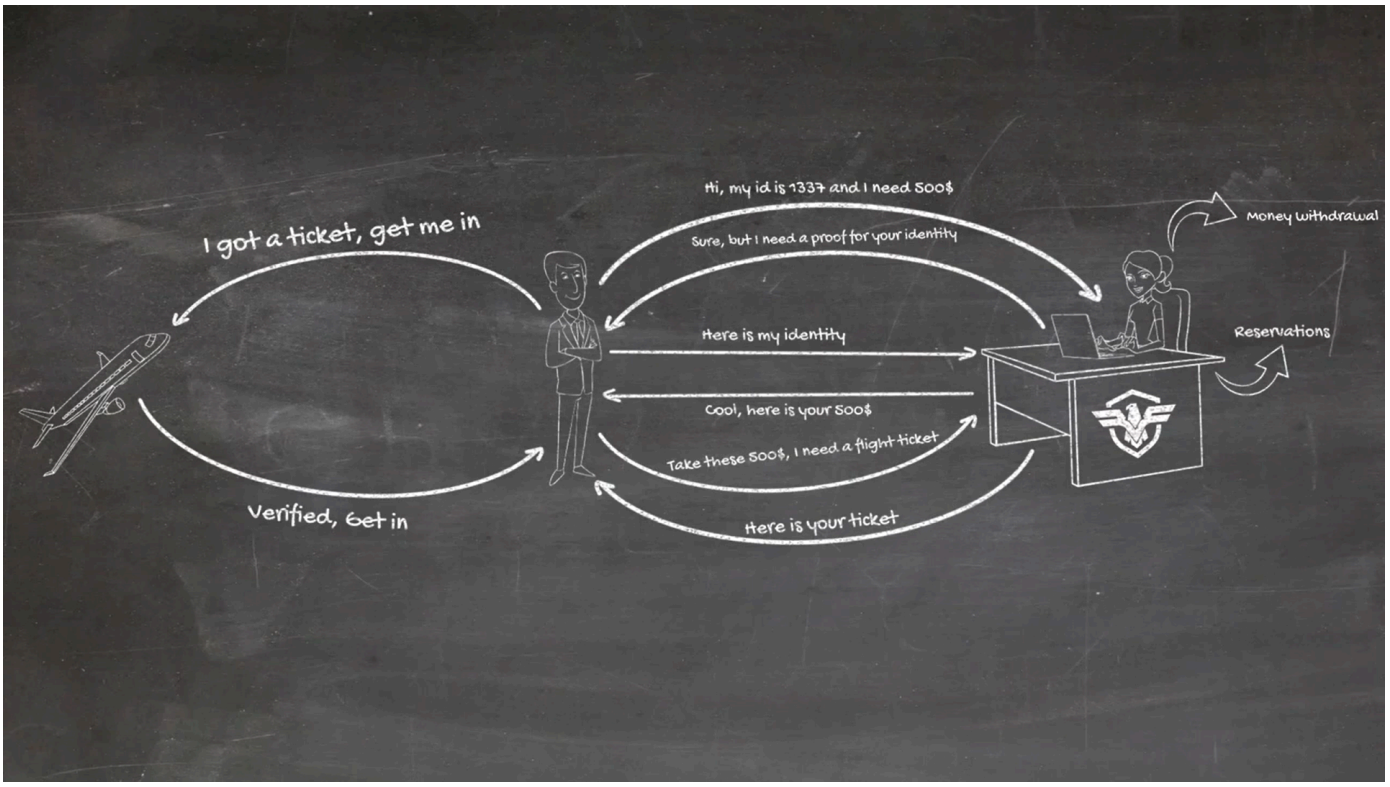
16-Kerber0s

https://www.youtube.com/watch?v=xOpjjjuvqf4&ab_channel=IMinzarI







Active Directory Authentication(NET-NTLM (ip-based), Kerbros (name-based))

NTLM hash

md4(Unicode(pass))

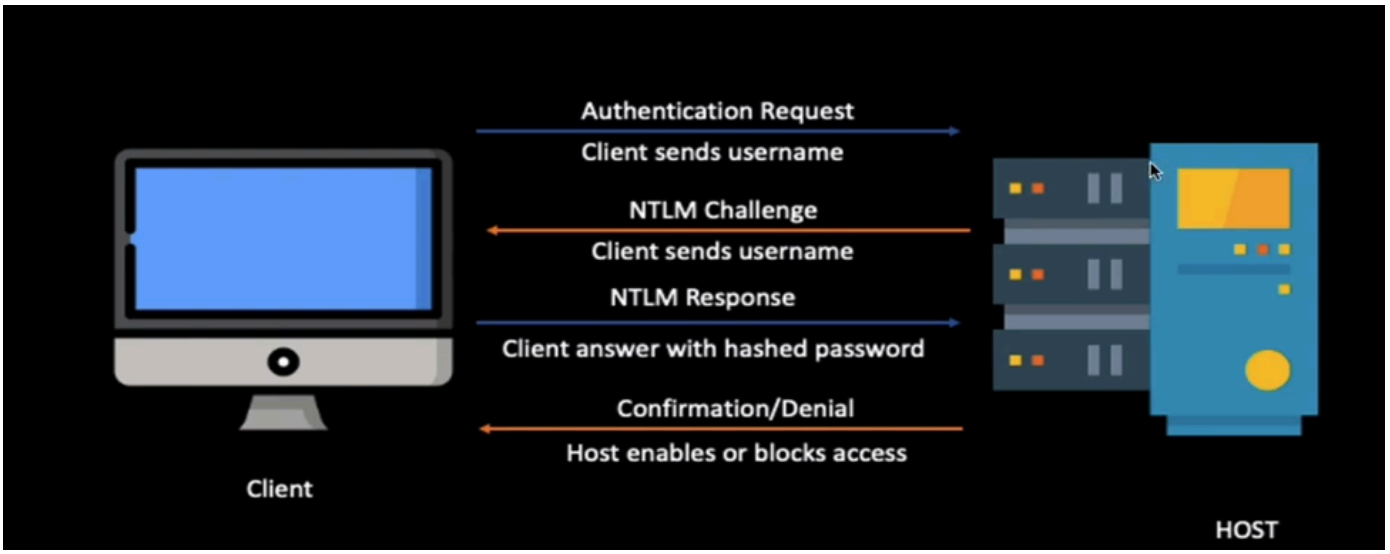
1-NET-NTLM Authentication

all password in windows stored in SAM file(Security Account Manager)

LSA(Local Security Authority): responsible for authentication process: compare between incoming data and the data stored in SAM file

before windows 7 LM hash was used.

the big problem in NTLM and LM that they does not using **salt** in **hash**



here i just use the hash not the password

so we can use **pass the hash attack**

we can dump hash from SAM using `mimikatz` and then pass the hash with it

---

2- NET-NTLMv2

as ver 1 but add the timestamp and domain name in the NTLM response to server ,to prevent replay attack

---

3-Kerbros (port: 88 TCP/UDP)--> Authentication not authorization

Kerberos

Kerberos authentication is **host-based**, **not IP based** like NTLM,

2-Key Distribution Center (KDC) which handle the Kerberos authentication requests, it's usually the domain controller or at least has access to the users and services secrets (Hashes) and consists of 2 services,
A – Authentication server (AS) which receives the client's authentication requests
B – Ticket Granting Service (TGS), which issue tickets to the client to access the services he needs.

Service Principal Names (SPNs)(ServiceClass/Host:Port for ex: Mysql/sql.iti.local:3306):

unique identifier for a service

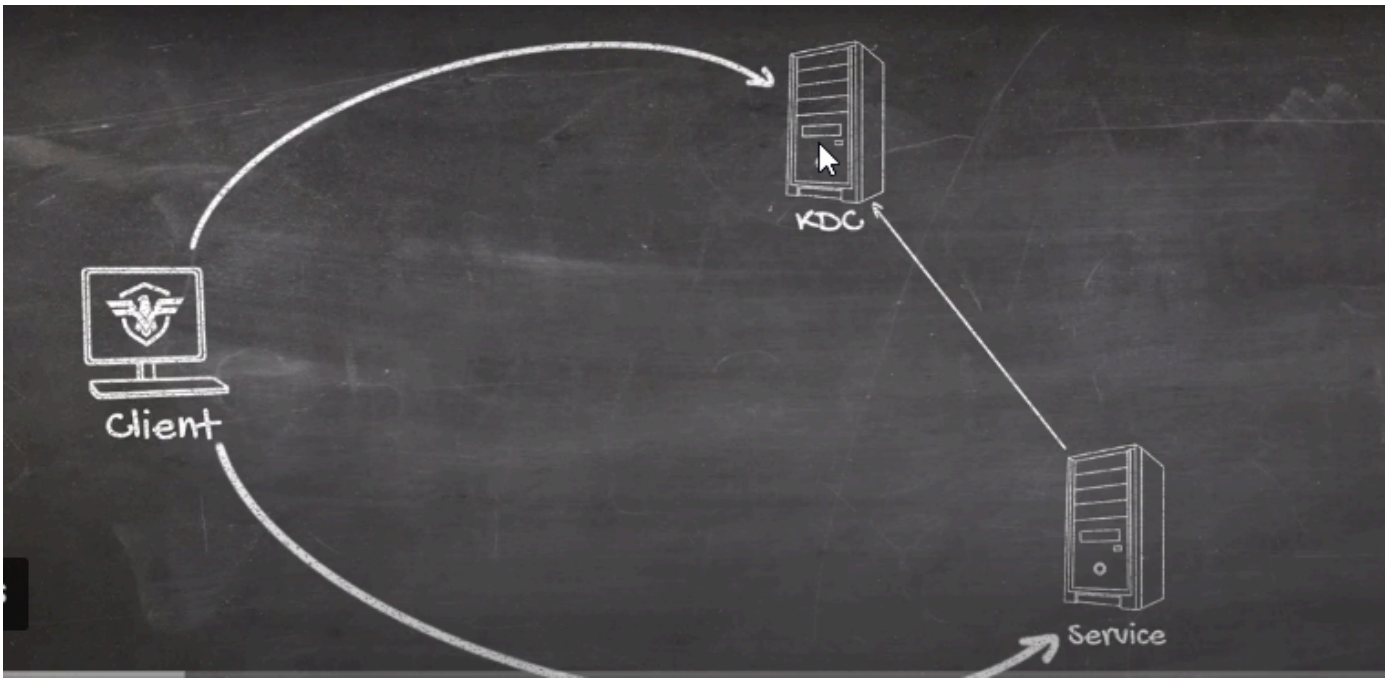
these principals should have a specific formatted name that complies with Kerberos requirements.

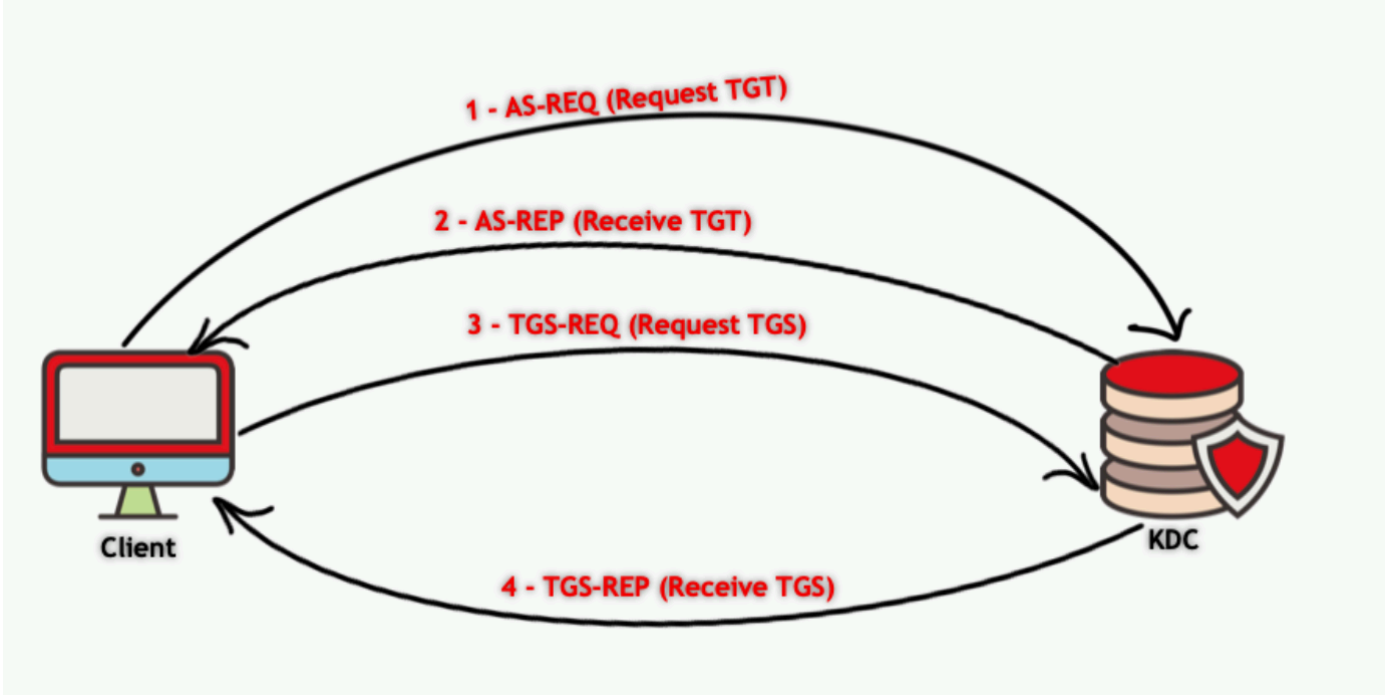
SPNs are used when authenticating to any service using Kerberos, the service must have a registered SPN in order for Kerberos to be used for authentication.

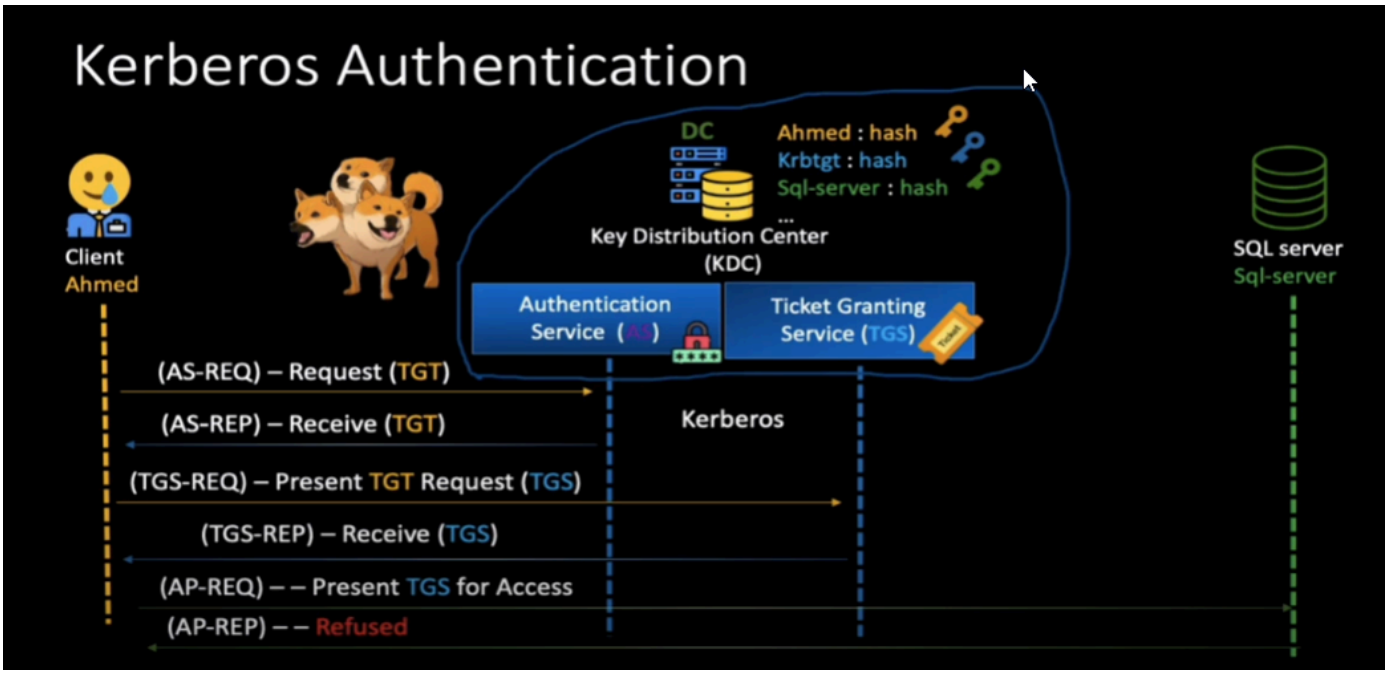
passwords is encrypted using NTLM hash

Let’s split the connection into 2 parts,
1 – Client <—-> KDC(key distribution center)(بيضمن هوية المستخدم) **service y3tbr**
2 – Client <—-> Service

The 1st part (Client <—-> KDC) involves the following







AS(authentication service): bt2kd your identity on network

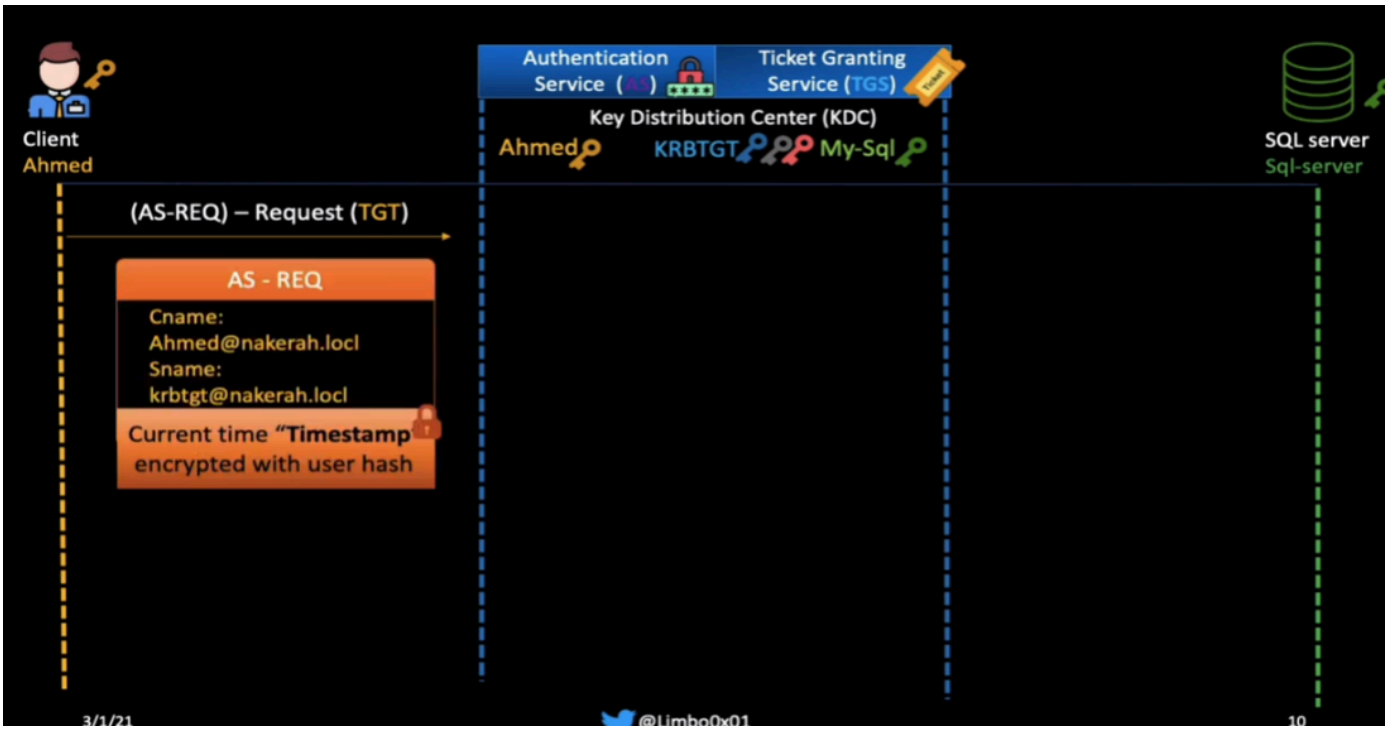
L service hya bt2oly ana lya permission a3mlha access wla la 7ata lw m3aya TGS

---

1 – AS-REQ (Authentication request):

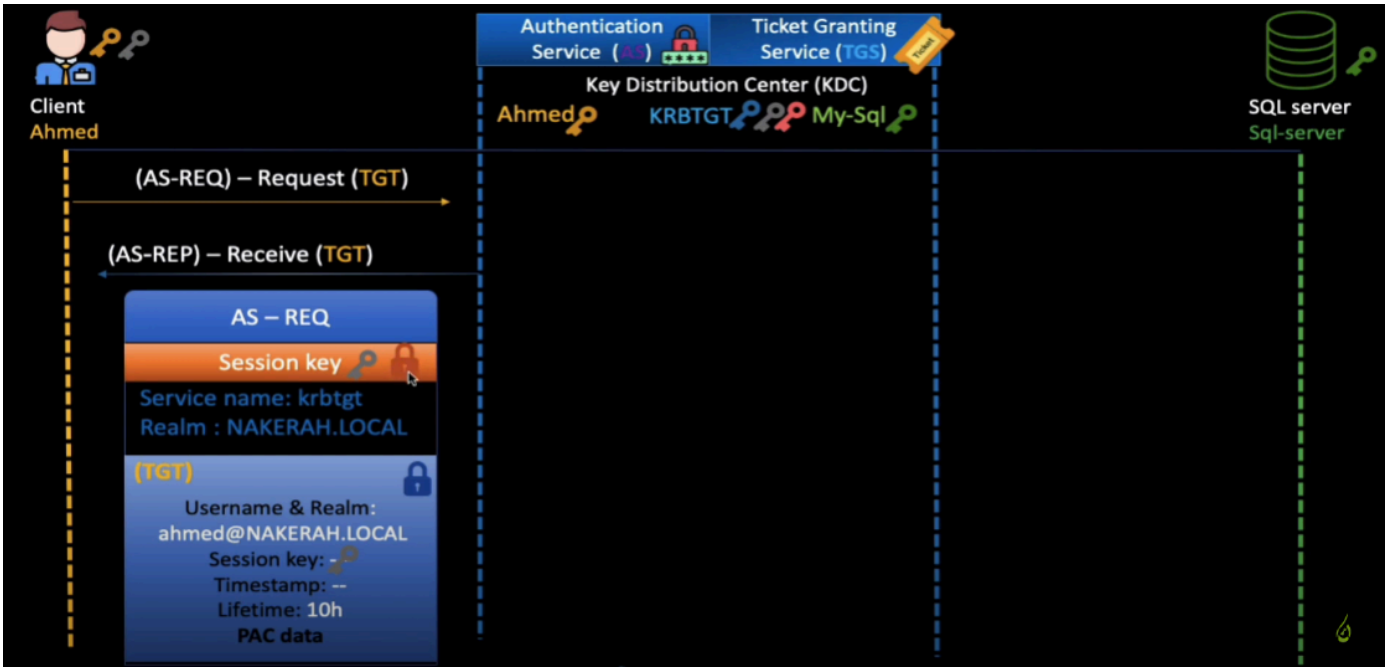
The client hashes the user's password, uses that hash to encrypt the current timestamp, and sends the encrypted timestamp to the KDC.
The KDC already has a copy of the user's hash so it uses the hash and tries to decrypt that message to retrieve the timestamp.
If the decryption is successful, then the KDC knows that the client used the correct hash and hence proved his identity to that KDC **(key distribution center)** on domain controller.

encrypt timestamp to prevent replay attack



2 – AS-REP (Authentication reply):the Authentication service (AS) replies with two messages:

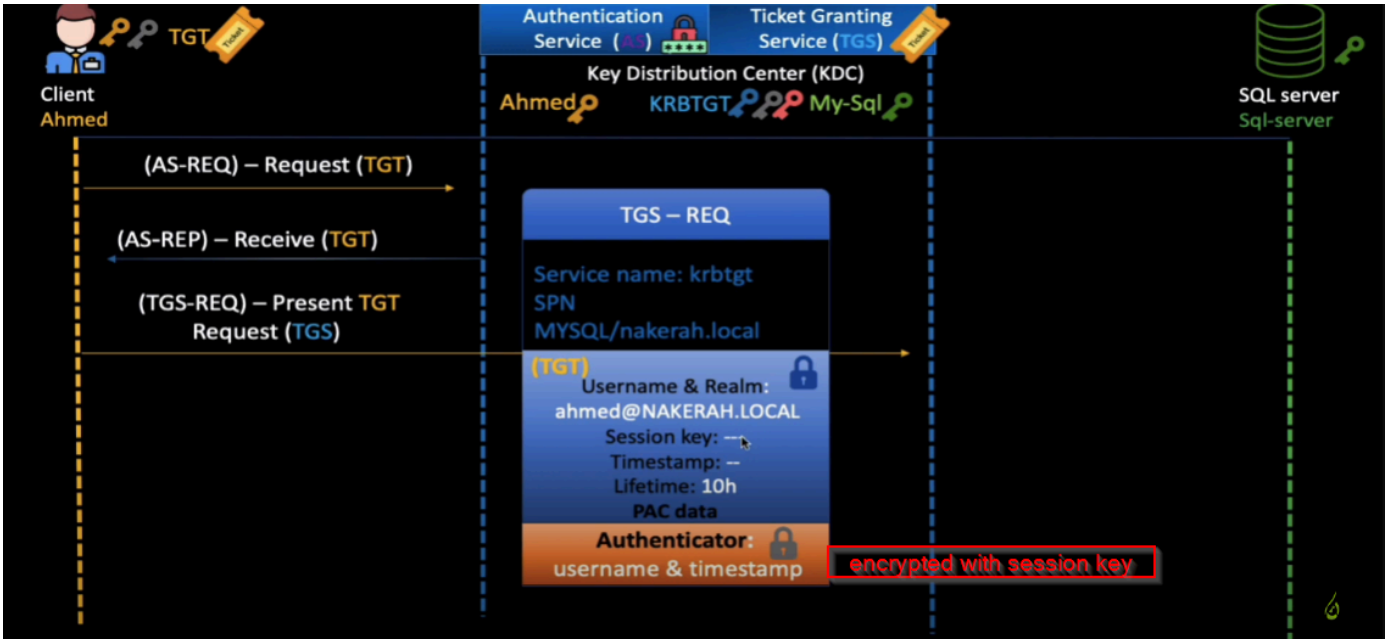
A – A session key encrypted using the user's hash, that key will be used for future messages.
B-TGT (التيكت ال هتسحملي اخد بعد كدة ساب تيكتس زي تيكت بتاعت دخول دريم بارك وبعد كدة في شوية سيرفسس كل واحدة ليها تيكت تانية بس مش هقدر اخدها لو معيش التيكت الاسا) (ticket-granting ticket), That TGT contains information about the user and his privileges on the domain, This message is encrypted using the hash of the KRBTGT account's password. That hash is known only to the KDC, so only the KDC can decrypt the TGT.



PAC data : mktob feha enta meen w a l permissions bt3tk

3-TGS-REQ (Ticket Granting Service request):(تيكت لسيرفس معينه بوريله التيكت الرئيسية الاول وبعدين يديني التيكت التانية دي)

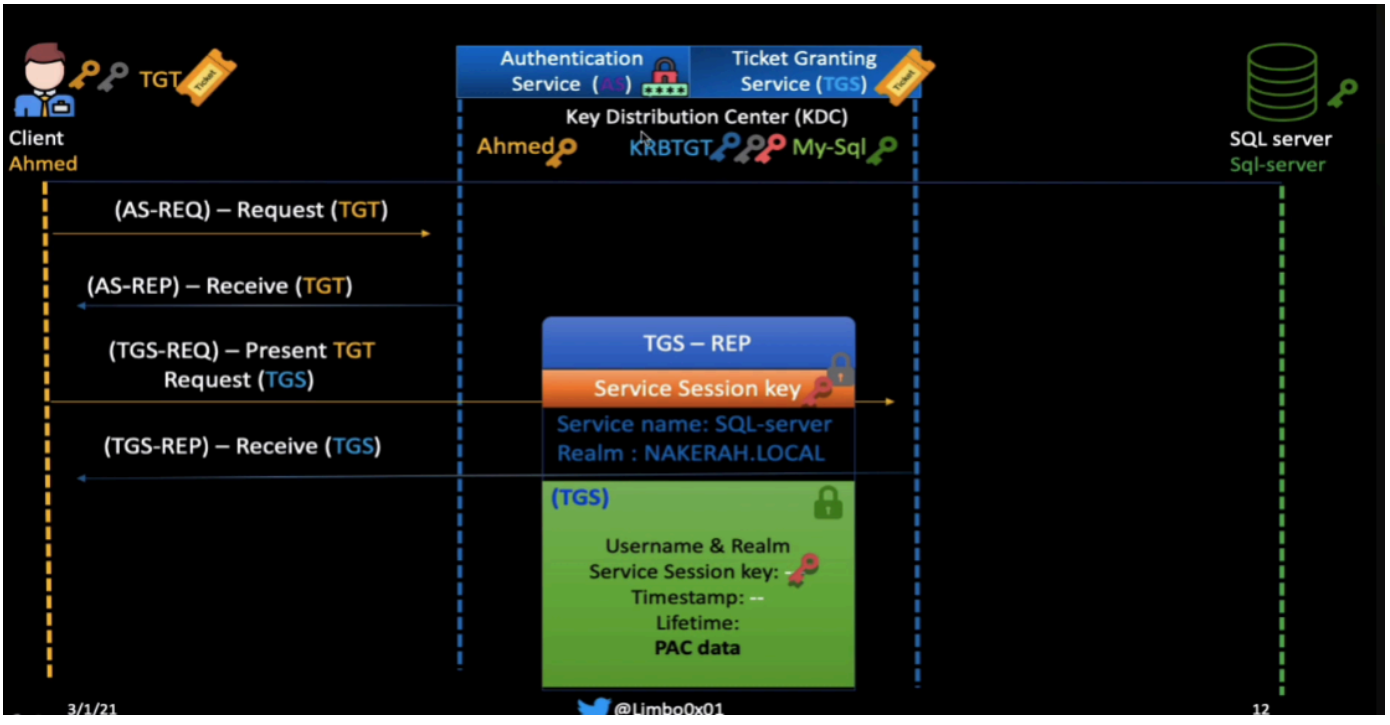
the client now has the TGT, he then requests a ticket to access the service he wants, so the client **encrypts that request** using the session key and sends it to the KDC which will decrypt and validate it. The TGT is also sent in that request.

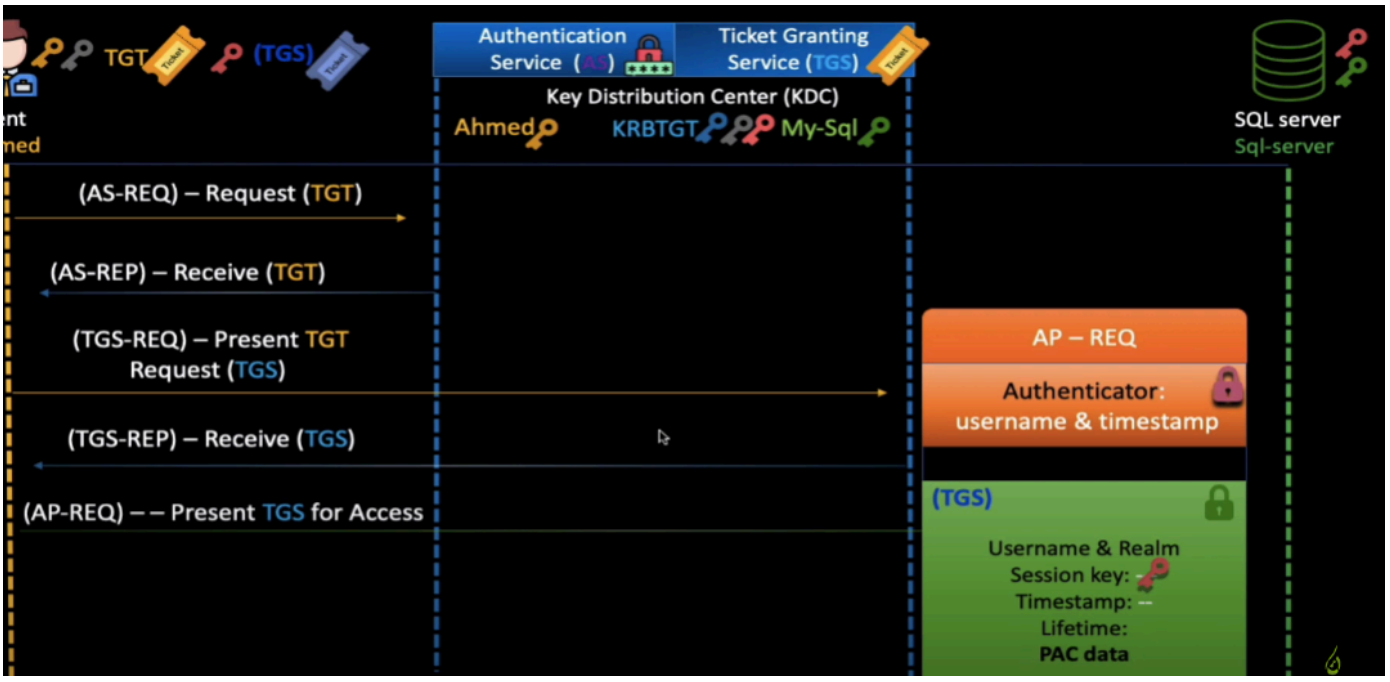


4 – TGS-REP: After validating the TGT the KDC responds with two messages

A – message specialized for the targeted service, encrypted with the service's hash which is stored at the KDC, this includes the information in the TGT as well as a session key

B – message for the client containing a service session key for further requests between the client and the service he asked to access, which is encrypted using the key retrieved from the AS-REP step.





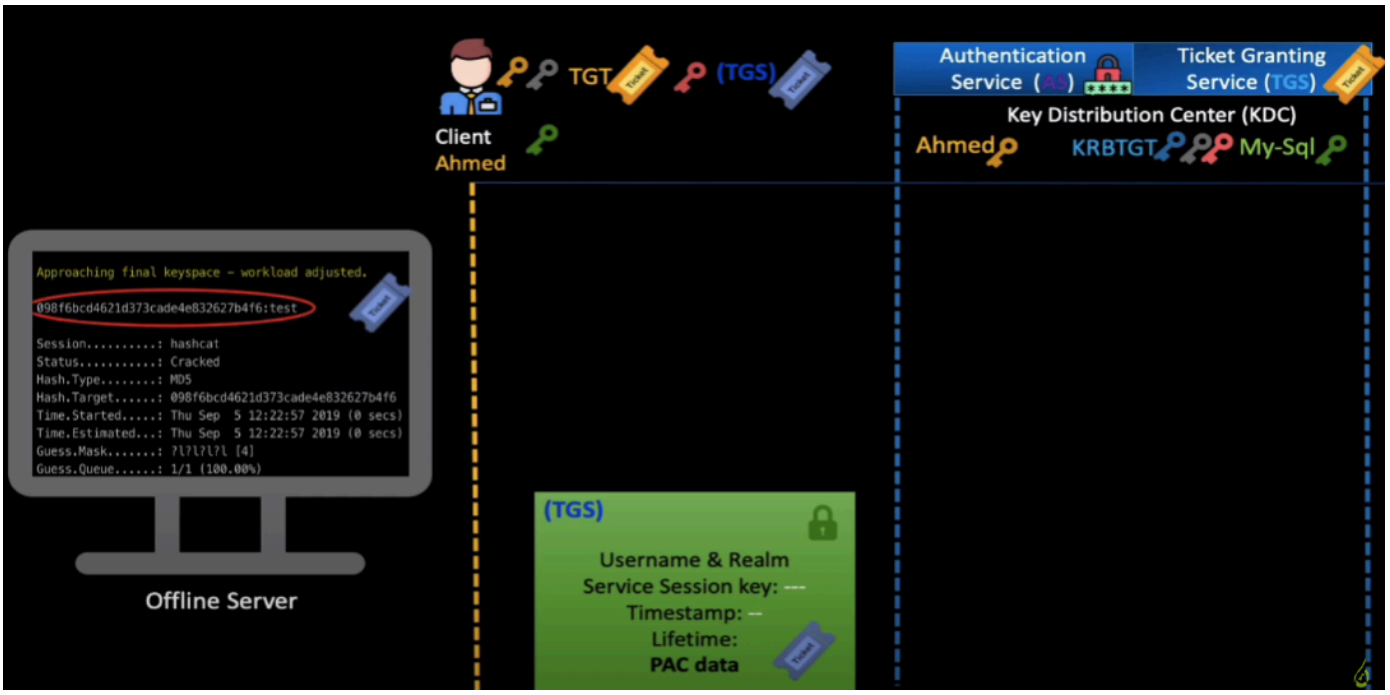
encrypt authenticator with service session key that is sent before and then compare the values in AP-REQ and TGS (timestamp and username)

The 2nd part Client <—-> Service
The client presents the message (TGS) from the TGS-REP step while connecting to the service along with an encrypted part, called authenticator message, this part includes the user's name and timestamp which was encrypted and will be decrypted using the service session key.
Then compare the username and timestamp from the TGS with the username and timestamp from the authenticator message.

AS req for (authentication service request):contains username,SPN(),timestamp

---

Kerbroasting



steal the TGS and try to crack the hashed password to log in as a service user **or create a silver ticket with high privilege**

We can issue a TGS ticket on our own machine, dump the ticket and start an offline bruteforce attack against it to retrieve the plaintext password for that user (service account)!

we got the SPN hash through power view or kali (impacket tool)