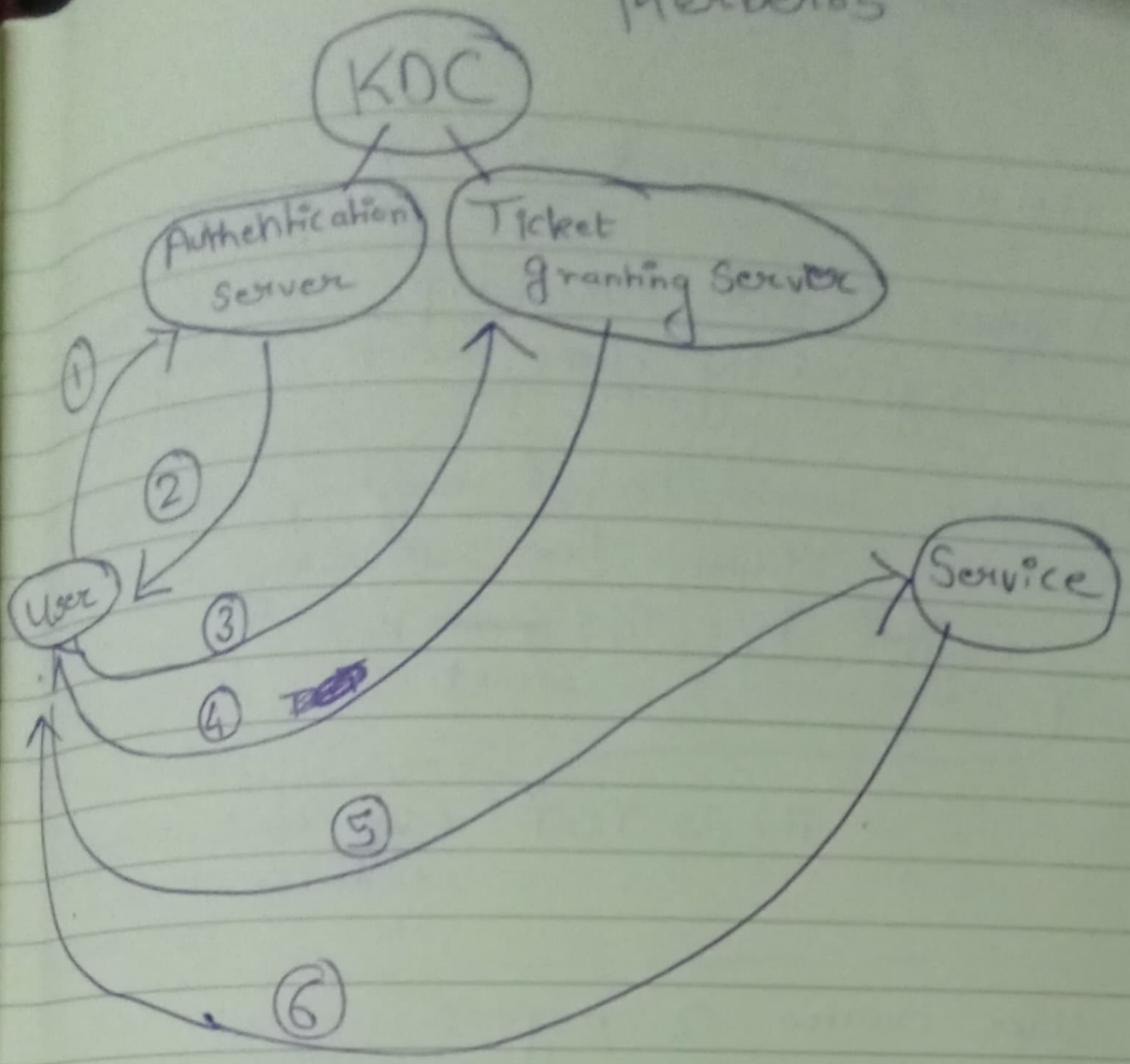
Kerberos



① User to Authn server

- contains attrbs: userid, service id, user-ip, TGT lifetime

- AuthN server checks userid against list/DB it has and grabs secret key from there (for users it modified password hash)

+ salt + Kvno

↓ client has it

(2)

Auth N server send 2 msgs

i) Attrbs : TGSid, timestamp, lifetime of TGT, TGS session Key
encrypted by client secret

ii) Attrbs : Userid, TGSid, timestamp, Userip, lifetime, TGS session key
encrypted by TGS ~~session~~ key
secret

ii) is TGT (user can't decrypt as it doesn't have TGS secret)

---

(3) User creates 2 msgs :

i) Attrbs : service id, Requested lifetime

ii) The user authentication
attrbs : user-id, timestamp
encrypted with TGS session key
(received in (2) i))

Sends to TGS along with TGT
received as (2) ii)

④ TGS

first sees the service id in ③ i) and checks againts its table / DB, grabs the service secret.

Now TGS has TGS secret key

It decrypts TGT, grabs TGS session key

Uses TGS session key to decrypt user authenticator

Now, TGS will validate data

Makes sure

i) user id same in TGT and user authenticator

ii) compares timestamps (about 2min window)

iii) verify TGT's ip addr with ip addr, it received message from.

iv) Lifetime has not passed yet

If user authenticator not in cache, TGS will add it.

↓

provides replay protection

4 cont'd

TGS creates 2 msgs

i) User Authentication ~~msgs~~

~~ii) attribs : user id, Timestamp~~

- containing attribs service id, timestamp, lifetime, service session key
  - encrypted with TGS session key

ii) Service ticket

- attribs : userid, svcid, timestamp, user-ip, lifetime of service ticket, service session key
  - encrypted by service secret key

sent to user

Now, user has 2 msgs

User received TGS in (2), hence decrypts (4)(i), grabs service session key

(5) User cannot decrypt (4)(ii) as it encrypted by service secret key

So it encrypts user authenticator (containing session key, userid, timestamp) with service key

Send that along with service ticket (4)(i) to service.

Now service decrypts the service ticket with its secret key, grabs service session key and decrypts user authenticator.

Compares timestamp, ip, userid like in (4)

if everything ok → checks cache → if user authenticator NOT already in cache } replay protection

if not → adds it

(6) Service creates

i) service authenticator

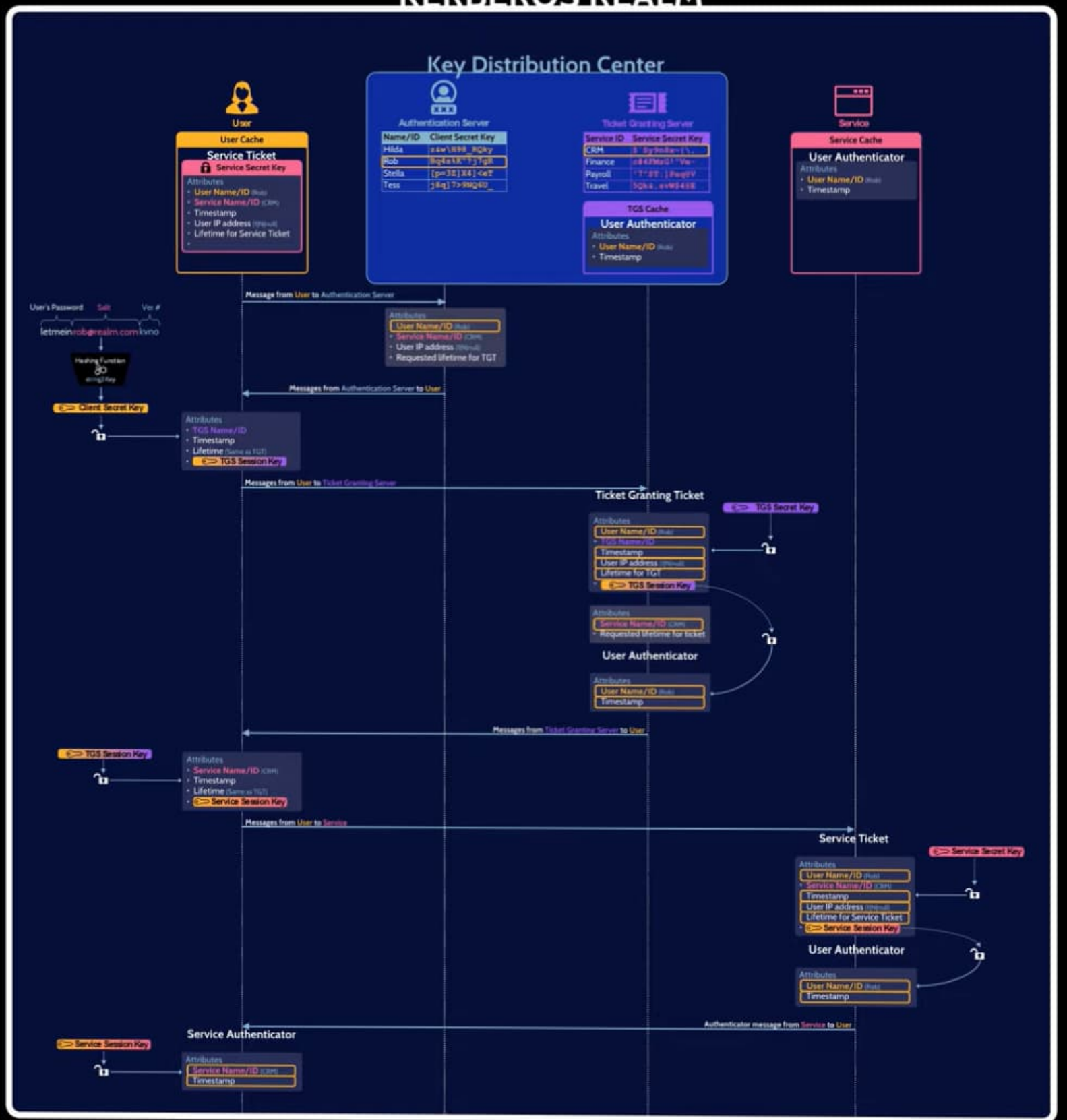
service id, timestamp

encrypted by service session key

sent to user.

User verifies service id, timestamp ~~thus forth~~ and caches an encrypted copy of service ticket for future use.

Thus completing mutual authentication


KERBEROS REALM
Key Distribution Center
Authentication Server
Name/ID
Client Secret Key
Hilda
Rob
Stella
Tess
Ticket Granting Server
Service ID
Service Secret Key
CRM
Finance
Payroll
Travel
TGS Cache
User Authenticator
Attributes
User Name/ID
Timestamp
User
User Cache
Service Ticket
Service Secret Key
Attributes
User Name/ID
Service Name/ID
Timestamp
User IP address
Lifetime for Service Ticket
Service
Service Cache
User Authenticator
Attributes
User Name/ID
Timestamp
Message from User to Authentication Server
User's Password
Salt
Ver #
letmein
rob@realm.com
kvno
Hashing Function
string2key
Client Secret Key
Attributes
User Name/ID
Service Name/ID
User IP address
Requested lifetime for TGT
Messages from Authentication Server to User
Attributes
TGS Name/ID
Timestamp
Lifetime
TGS Session Key
Messages from User to Ticket Granting Server
Ticket Granting Ticket
TGS Secret Key
Attributes
User Name/ID
Timestamp
User IP address
Lifetime for TGT
TGS Session Key
Attributes
Service Name/ID
Requested lifetime for ticket
User Authenticator
Attributes
User Name/ID
Timestamp
Messages from Ticket Granting Server to User
TGS Session Key
Attributes
Service Name/ID
Timestamp
Lifetime
Service Session Key
Messages from User to Service
Service Ticket
Service Secret Key
Attributes
User Name/ID
Service Name/ID
Timestamp
User IP address
Lifetime for Service Ticket
Service Session Key
User Authenticator
Attributes
User Name/ID
Timestamp
Authenticator message from Service to User
Service Authenticator
Service Session Key
Attributes
Service Name/ID
Timestamp