**STEP 1**

**SENDS USER 🡪 AUTHENTICATION SERVER**

[ ENCRYPTED VIA SSL ] **MESSAGE 1**

* User name / ID
* Service name / service ID
* User IP address
* Requested lifetime for TGT

**STEP 2**

**RECEIVES AUTHENTICATION SERVER 🡨 USER**

Here AS checks for the USER ID in its database and SERVICE ID in the database. If both exists then AS moves forward in creating the message for the user.

**STEP 3**

**SENDS AUTHETICATION SERVER 🡪 USER**

AS generates random TGS session key to share with the client.

[ENCRYPTED VIA CLIENT SECRET KEY ] **MESSAGE 2**

* TGS name/ID
* Timestamp
* Lifetime of ticket
* TGS session key

[ ENCRYPTED VIA TGS SECRET KEY ] **MESSAGE 3**

* User name / ID
* TGS name ID
* Time Stamp
* User IP address
* Lifetime for TGT
* TGS session Key

**STEP 4**

**RECEIVES USER 🡨 AUTHENTICATION SERVER**

Now for opening the first msg of AS the client must create a client secret key. Secret key is a combination of user password + salt + KVNO.

So when the user puts up the password, the AS adds salt to it and then a hashing function is run on it , to create a secret key. Now user opens the msg using this secret if the password was right then the message is opened else the message does not open and wrong password pop up come on the screen.

After the secret key is made and password is correct then the user opens the message and it contains TGS session key.

**STEP 5**

**SENDS USER 🡪 TICKET GRANTING SERVER**

[ NOT ENCRYPTED ] **MESSAGE 4**

* SERVICE NAME / ID
* REQUESTED LIFE TIME FOR THE TICKET

[ ENCRYPTED VIA TGS SESSION KEY ] **MESSAGE 5**

* User name / ID
* Time stamp

[ ENCRYPTED VIA TGS SECRET KEY ] **MESSAGE 6**

* User name / ID
* TGS name ID
* Time Stamp
* User IP address
* Lifetime for TGT
* TGS session Key

**STEP 6**

**RECEIVES TICKET GRANTING SERVER 🡨 USER**

Now the TGS look for the service ID in its database and fetch secret key for that service ID.

If the service ID matches from the database, then it goes ahead with further authentication process.

TGS opens MESSAGE 6 using the TGS secret key from the database. And then opens MESSAGE 5 using TGS session key present in MESSAGE 6.

If the user authentication details for the service is not in the TGS cache server then it will the details in its cache server.

And if the authentication details already their in the cache server and are not expired then the TGS will allow the user to move forward.

**STEP 7**

TGS generates random service session key and share with the client in the message.

**SENDS TICKET GRANTING SERVER 🡪 USER**

[ ENCRYPTED VIA TGS SESSION KEY ]  **MESSAGE 7**

* Service id
* Time stamp
* Lifetime of ticket
* Service session key

[ENCRYPTED VIA SERVICE SECRET KEY ] **MESSAGE 8**

* Username / ID
* Service name
* Time stamp
* User IP address
* Lifetime for service ticket
* Service session key

**STEP 8**

**RECEIVES USER 🡨 TICKET GRANTING SERVER**

Now since the user already have the TGS session key from earlier messages. It will open up

MESSAGE 7. On opening MESSSAGE 7 the user will get service session key. But the user till now do not have the service secret key. So the user will just open up MESSAGE 7.

Also the user will get service session key from MESSAGE 7.

**STEP 9**

**SENDS USER 🡪 SERVICE**

[ENCRYPTED VIA SERVICE SECRET KEY ] **MESSAGE 9**

* Username / ID
* Service name
* Time stamp
* User IP address
* Lifetime for service ticket
* Service session key

[ ENCRYPTED VIA SERVICE SESSION KEY ] **MESSAGE 10**

* Username/ ID
* Time stamp

**STEP 10**

**RECEIVES SERVICE 🡨 USER**

Service already have the service secret key. So it will open up MESSAGE 9. From there service will get the service session key with which it will open MESSAGE 10. Now the service starts to authenticate the username /ID , service name, time stamp [max 2 min] , User IP address etc.

Service also maintains cache server of its own where it marks entries for the already visited user. If the user is already in visited then it is allowed to access the service and authenticated else the process of authentication continues.

**STEP 11**

**SENDS SERVICE 🡪 USER**

[ ENCRYPTED VIA SERVICE SESSION KEY ] **MESSAGE 11**

* Service name / ID
* Time stamp

MESSAGE 11 was the final message from the service. And if all the authentication goes well the service give confirmation to the user to access the service and provides the feature of SSO till the time the data is expired in the cache server.

**AUTHETICATION DONE HERE.**