NR	Message Name		Ticketname/KeyName	verschlüsselt mit	Schlüssel ist bekannt				Bemerkungen
					Client	KDC		Server	
						AS	TGS		
1	**AS_REQ (1)** (Authentication Service Request) Client --> KDC		Username. Das Passwort wird in (2) benötigt	N/A					Der **AS_REQ** wird dem AS geschickt. Es enthält den Principal-Namen des Clients und den Principal-Namen des TGS (krbtgt/EXAMPLE.COM@EXAMPLE.COM)
2	**AS_REP (2)** (Authentication Service Reply)	**AS_REP** Client-Teil	TGS_$S_{A,KDC}$ (Session Key) + expiration Time + TGS Service Name	K_A = priv. Langzeitschlüssel Client	x	x			Mit dem TGS-Session-Key kann der Client seine Identität dem KDC beweisen, weil nur der Client und der KDC diesen Schlüssel kennen (**TGS_$S_{A,KDC}$**, expiration time, TGS Service Name, usw.). K_A Dieser Client-Teil des AS_REP kann der Client mit seinem Passwort entschlüsseln und kann somit den TGS_Session_Key aus der Meldung extrahieren.
		AS_REP TGT-Teil	TGT = TGS_$S_{A,KDC}$ + exp. Time + Principal-Name des Clients	K_{KDC} = Langzeitschlüssel des TGS (TGS_Key)		x	x		Der Client kann das TGT nicht entschlüsseln, weil er K_{KDC} nicht kennt. Somit kann der Client das TGT nicht manipulieren und vorallem nicht den TGS-Session-Key verändern.
		Client <-- KDC							Den TGS-Session-Key (**TGS_$S_{A,KDC}$**) und das TGT in seiner verschlüsselten Form speichert der Client in seinem Credential-Cache (**klist**) ab.
3	**TGS_REQ (3)** (Ticket Granting Server Request) Der REQ kommt zustande, wenn der Client auf einem kerberisierten Dienst zugreifen will. Dafür benötigt er ein Ticket vom TGS	Besteht aus 4 Elementen:	**Authenticator** - Client-Principal-Name - Timestamp - Checksumme	**TGS_$S_{A,KDC}$** (Session Key) Diesen Key kennt der Client aus Schritt 2 (**AS_REP**)	x		(x)		Da es sich beim TGS_REQ um einen kerberisierten Zugriff handelt, wird das TGT und der Authenticator gesendet. Der TGS prüft diese Angaben. Wenn ok, dann ist der Client authentifiziert und...
			Ticket Granting Ticket	K_{KDC} = Langzeitschlüssel des TGS (TGS_Key)			x		
		Client --> KDC	**Service Name (Dienst im Netz)**						
			Expiration Time des TGT						
4	**TGS_REP (4)** Ticket Granting Server Reply	Client-Teil	**Client-Ticket** -Principal-Name des Services -Service-Session-Key (Service_KAB) -Expiration Time	TGS_$S_{A,KDC}$ (=SessionKey)	x		x		... erstellt dann einen neuen Session Key (=Service-Session-Key) für lient und Service. Der TGS entnimmt der KDC-Datenbank den Langzeitschlüssel, dem TGS-Session-Key, welcher bei jeder Neuanmeldung immer wieder generiert wird. Der Client kann somit ohne weitere Interaktionen mit diesem Key, welcher in seinem Cache ist. Entschlüsselungen von weitern Service-Session-Keys vornehmen.
		Service-Teil Client <-- KDC	**Servie Ticket** -Service-Session-Key (Service_K_{AB}) -Client Name -Expiration Time	K_B			x	x	Er muss also (1) und (2) nicht mehr durchführen. Das in Schritt (2) eigegebene Passwort genügt. SSO wird mit diesem Trick mögclich.
5	**AP_REQ (5)** Application Server Request Client --> Server/Service		**Authenticator** -Timestamp -Checksumme	Service_$K_{A,B}$	x		x	(x)	Aus dem Service Ticket kann der Server den Service-Session-Key entnehmen. Mit diesem Key ist er in der Lage den Authenticator zu entschlüsseln, deshalb (X). Den Service-Session-Key kennen nur Dienst, Client und KDC. Dieser Key ist also ein gemeinsames Geheimnis zwischen Dienst und Client. Wenn der Service nun mit diesem Key den Authenticator entschlüsseln kann, het der Client seine Authentizität bewiesen. Kein anderer als der Client, hätte diesen Authenticator generieren können.
			Service Ticket -Service Session Key Service_$K_{A,B}$ -Client Name -Expiration Time	K_B			x	x	
6	**AP_REP (6)** Application Server Reply Optional Client <-- Server/Service		Timestamp	Service_$K_{A,B}$	x		x	x	Optional: Falls der Client sicher sein will, dass es der richtige Servr ist. Z.B: Telebanking-Server Wenn der Client den Zeitstempel mit Service_$K_{A,B}$ enstchlüsseln kann, weiss er, dass dieser vom "richtigen" Server gesendet wurde, weil nur dieser den Schlüsseö Service_$K_{A,B}$ mittels K_B kennt.

Legende

Schlüssel

K_A	private Key Client
K_{KDC}	private Key KDC = Langzeitschlüssel des TGS
TGS_$S_{A,KDC}$	TGS_Session Key (Authentisierung)
K_B	Langzeitschlüssel des Servers/Service
Service_$S_{A,B}$	Service-Session-Key