Messages échangés par le protocole KERBEROS

(a) Authentication Service Exchange: to obtain ticket-granting ticket
(1) C → AS: $ID_c \parallel ID_{tgs} \parallel TS_1$
(2) AS → C: $E_{K_c}\left[K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}\right]$
$Ticket_{tgs} = E_{K_{tgs}}\left[K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2\right]$
(b) Ticket-Granting Service Exchange: to obtain service-granting ticket
(3) C → TGS: $ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$
(4) TGS → C: $E_{K_{c,tgs}}\left[K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v\right]$
$Ticket_{tgs} = E_{K_{tgs}}\left[K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2\right]$
$Ticket_v = E_{K_v}\left[K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4\right]$
$Authenticator_c = E_{K_{tgs}}\left[ID_C \parallel AD_C \parallel TS_3\right]$
(c) Client/Server Authentication Exchange: to obtain service
(5) C → V: $Ticket_v \parallel Authenticator_c$
(6) V → C: $E_{K_{c,v}}\left[TS_5 + 1\right]$ (for mutual authentication)
$Ticket_v = E_{K_v}\left[K_{c,v} \parallel ID_C \parallel AD_C \parallel ID_v \parallel TS_4 \parallel Lifetime_4\right]$
$Authenticator_c = E_{K_{c,v}}\left[ID_C \parallel AD_C \parallel TS_5\right]$

Détails et justification des éléments employés dans les messages du protocole KERBEROS

(a) Authentication Service Exchange	
Message (1)	Client requests ticket-granting ticket
ID_C:	Tells AS identity of user from this client
IDtgs:	Tells AS that user requests access to TGS
TS_1:	Allows AS to verify that client's clock is synchronized with that of AS
Message (2)	AS returns ticket-granting ticket
E_{K_c}:	Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2)
$K_{c,tgs}$:	Copy of session key accessible to client; created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key
ID_{tgs}:	Confirms that this ticket is for the TGS
TS_2:	Informs client of time this ticket was issued
$Lifetime_2$:	Informs client of the lifetime of this ticket
$Ticket_{tgs}$:	Ticket to be used by client to access TGS
(b) Ticket-Granting Service Exchange	
Message (3)	Client requests service-granting ticket
ID_V:	Tells TGS that user requests access to server V
$Ticket_{tgs}$:	Assures TGS that this user has been authenticated by AS
$Authenticator_c$:	Generated by client to validate ticket
Message (4)	TGS returns service-granting ticket
$K_{c,tgs}$:	Key shared only by C and TGS; protects contents of message (4)
$K_{c,v}$:	Copy of session key accessible to client; created by TGS to permit secure exchange between client and server without requiring them to share a permanent key
ID_V:	Confirms that this ticket is for server V
TS_4:	Informs client of time this ticket was issued
$Ticket_V$:	Ticket to be used by client to access server V
$Ticket_{tgs}$:	Reusable so that user does not have to reenter password
$E_{K_{tgs}}$	Ticket is encrypted with key known only to AS and TGS, to prevent tampering
$K_{c,tgs}$:	Copy of session key accessible to TGS; used to decrypt authenticator, thereby authenticating ticket
ID_C:	Indicates the rightful owner of this ticket
AD_C:	Prevents use of ticket from workstation other than one that initially requested the ticket
ID_{tgs}:	Assures server that it has decrypted ticket properly
TS_2:	Informs TGS of time this ticket was issued
$Lifetime_2$:	Prevents replay after ticket has expired

$Authenticator_c$:	Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay
$E_{K_{c,tgs}}$	Authenticator is encrypted with key known only to client and TGS, to prevent tampering
ID_C:	Must match ID in ticket to authenticate ticket
AD_C:	Must match address in ticket to authenticate ticket
TS_2:	Informs TGS of time this authenticator was generated
(c) Client/Server Authentication Exchange	
Message (5)	Client requests service
$Ticket_V$:	Assures server that this user has been authenticated by AS
$Authenticator_c$:	Generated by client to validate ticket
Message (6)	Optional authentication of server to client
$E_{K_{c,v}}$:	Assures C that this message is from V
$TS_5 + 1$:	Assures C that this is not a replay of an old reply
$Ticket_v$:	Reusable so that client does not need to request a new ticket from TGS for each access to the same server
E_{K_v}:	Ticket is encrypted with key known only to TGS and server, to prevent tampering
$K_{c,v}$:	Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket
ID_C:	Indicates the rightful owner of this ticket
AD_C:	Prevents use of ticket from workstation other than one that initially requested the ticket
ID_V:	Assures server that it has decrypted ticket properly
TS_4:	Informs server of time this ticket was issued
$Lifetime_4$:	Prevents replay after ticket has expired
$Authenticator_c$:	Assures server that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay
$E_{K_{c,v}}$:	Authenticator is encrypted with key known only to client and server, to prevent tampering
ID_C:	Must match ID in ticket to authenticate ticket
AD_c:	Must match address in ticket to authenticate ticket
TS_5:	Informs server of time this authenticator was generated

Messages échangés dans le protocole KERBEROS v5

(a) Authentication Service Exchange: to obtain ticket-granting ticket
(1) C → AS: Options ‖ ID_c ‖ $Realm_c$ ‖ ID_{tgs} ‖ $Times$ ‖ $Nonce_1$
(2) AS → C: $Realm_c \| ID_C \| Ticket_{tgs} \| E_{K_c}\left[K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}\right]$
$Ticket_{tgs} = E_{K_{tgs}}\left[Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times\right]$
(b) Ticket-Granting Service Exchange: to obtain service-granting ticket
(3) C → TGS: Options ‖ ID_v ‖ $Times$ ‖ ‖ $Nonce_2$ ‖ $Ticket_{tgs}$ ‖ $Authenticator_c$
(4) TGS → C: $Realm_c \| ID_C \| Ticket_v \| E_{K_{c,tgs}}\left[K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_V\right]$
$Ticket_{tgs} = E_{K_{tgs}}\left[Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times\right]$
$Ticket_v = E_{K_v}\left[Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times\right]$
$Authenticator_c = E_{K_{c,tgs}}\left[ID_C \| Realm_c \| TS_1\right]$
(c) Client/Server Authentication Exchange: to obtain service
(5) C → V: Options ‖ $Ticket_v$ ‖ $Authenticator_c$
(6) V → C: $E_{K_{C,V}}$ [TS_2 ‖ Subkey ‖ Seq#]
$Ticket_v = E_{K_v}\left[Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times\right]$
$Authenticator_c = E_{K_{c,V}}\left[ID_C \| Realm_c \| TS_2 \| Subkey \| Seq\#\right]$

Flags utilisés dans le KERBEROS v5

INITIAL	This ticket was issued using the AS protocol and not issued based on a ticket-granting ticket.
PRE-AUTHENT	During initial authentication, the client was authenticated by the KDC before a ticket was issued.
HW-AUTHENT	The protocol employed for initial authentication required the use of hardware expected to be possessed solely by the named client.
RENEWABLE	Tells TGS that this ticket can be used to obtain a replacement ticket that expires at a later date.
MAY-POSTDATE	Tells TGS that a postdated ticket may be issued based on this ticket-granting ticket.
POSTDATED	Indicates that this ticket has been postdated; the end server can check the authtime field to see when the original authentication occurred.
INVALID	This ticket is invalid and must be validated by the KDC before use.
PROXIABLE	Tells TGS that a new service-granting ticket with a different network address may be issued based on the presented ticket.
PROXY	Indicates that this ticket is a proxy.
FORWARDABLE	Tells TGS that a new ticket-granting ticket with a different network address may be issued based on this ticket-granting ticket.
FORWARDED	Indicates that this ticket has either been forwarded or was issued based on authentication involving a forwarded ticket-granting ticket.