Kerberos V5

Technical Description

ASN.1 Data Representation Language

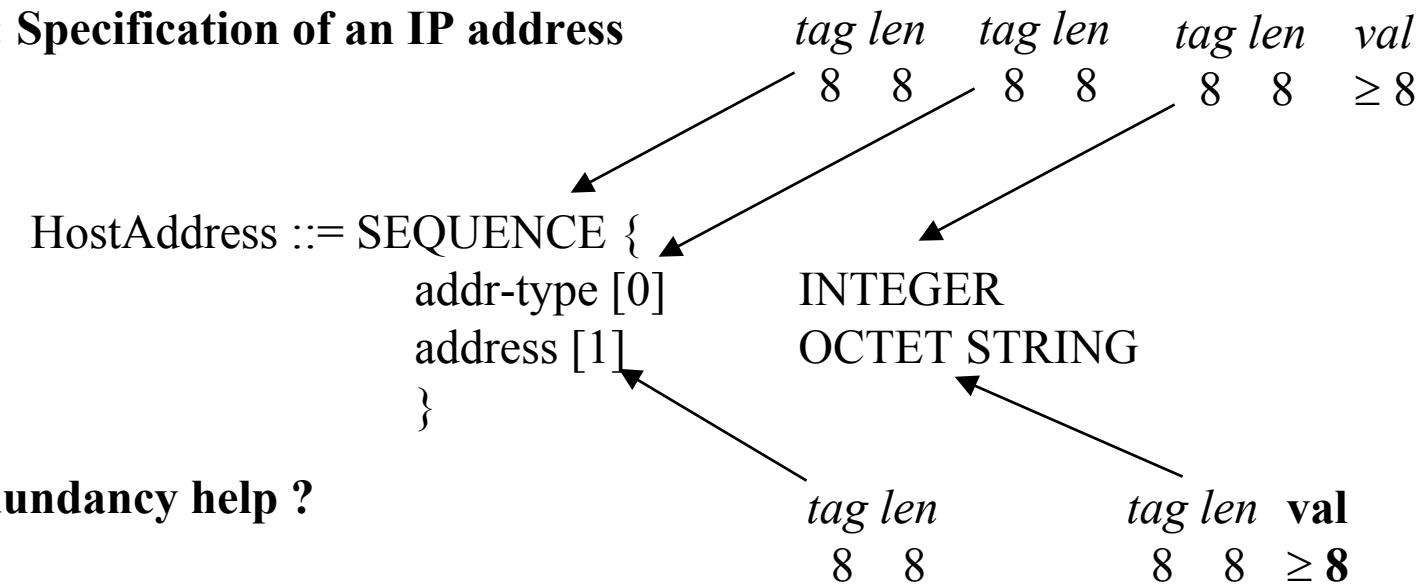
Basic Encoding Rules (BER) allow:

- o optional fields of data structures
- o variable-length data structures
- o typed data structures

Motivation:

- o independence of hardware data structure encodings
  - e.g., big- or little-endian byte ordering
- o standard definition

Example: Specification of an IP address



Does redundancy help ?

Delegation of Rights

- **Forwarding of TGTs**
  - forward*able* TGT => it can be exchanged for a TGT with one or more different network addresses (i.e., forward*ed* TGT)
  - limited time
  - option for transitive forwarding (adequate control ?)

- **Proxying of tickets**
  - proxy*able* TGT => it can be used to request tickets with one or more different network addresses (i.e., *proxy* tickets)
  - limited time
  - no transitive proxying

- **Forwarded and Proxy flags set in all derived TGT / tickets**
  - application awareness of delegation

Forwarding Protocol

- **AS_REQ includes:**
  - **"forwardable" option**
- **AS_REP includes:**
  - **"forwardable" TGT**
- **TGS_REQ includes:**
  - **"forwardable" TGT**
  - **"forwardable" option**
  - **"caddr list"**
- **TGS_REP includes:**
  - **"forwarded" TGT and "forwardable" flag (if requested)**
  - **set of "caddr"**
- **Forwarder constructs a KRB_CRED message to pass the forwarded ticket and ticket's session key to recipient**

Proxying Protocol

- **AS_REQ includes:**
  - **"proxyable" option**
- **AS_REP includes:**
  - **"proxyable" flag in TGT**
- **TGS_REQ includes:**
  - **"proxy" ticket for a *specific* application service**
  - **"caddr list"**
  - **additional access restrictions in "authorization_data"**
- **TGS_REP includes:**
  - **"proxy" ticket and set "authorization_data" (if requested)**
  - **set of "caddr"**
- **Proxy-er constructs a KRB_CRED message to pass the proxy ticket and ticket's session key to recipient**

Flag Checking in AS_REP and TGS_REP

- What if the "forwardable" requested option is not checked against the "forwardable" flag ?

  - o non-forwardable TGTs may become forwardable
  - o forwarded - only TGTs may become forwardable

- Note: the use of the "forwardable" feature may be dangerous as it may cause unrestricted propagation of a party's identity and permissions

- Other requested options vs. flag checks are necessary

  - o non-proxyable tickets may become proxyable
  - o non-renewable tickets may become renewable

Ticket Lifetimes

o starttime = time the ticket becomes valid
o endtime = time the ticket expires
o authtime = KDC (AS) time when TGT of AS_REP is created
starttime > authtime => postdated tickets
o postdated tickets are useful for batch / absentee computations
o long-lived tickets are necessary

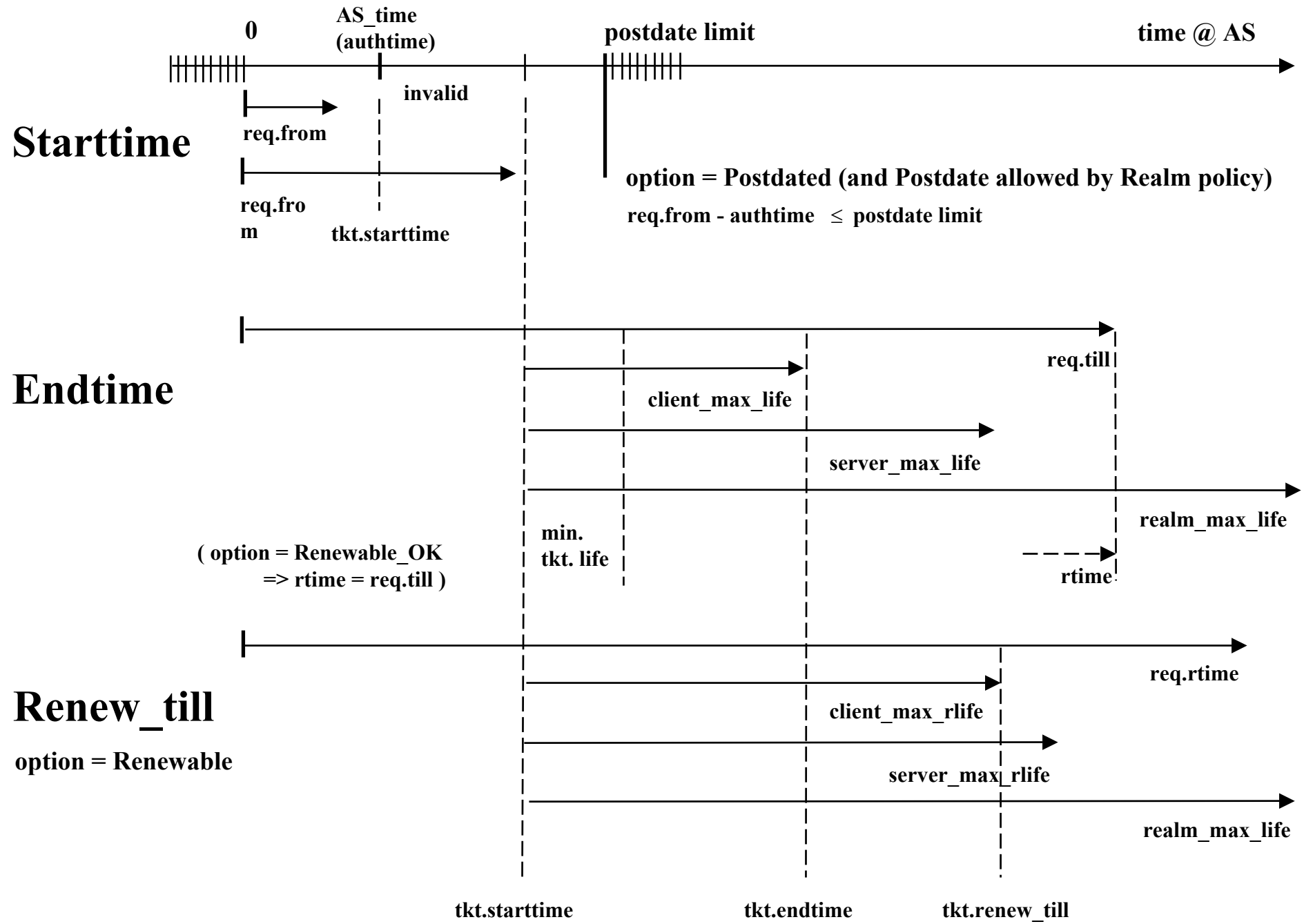
Problem

o long-lived tickets make revocation impossible
o postdated tickets must allow revocation before first use

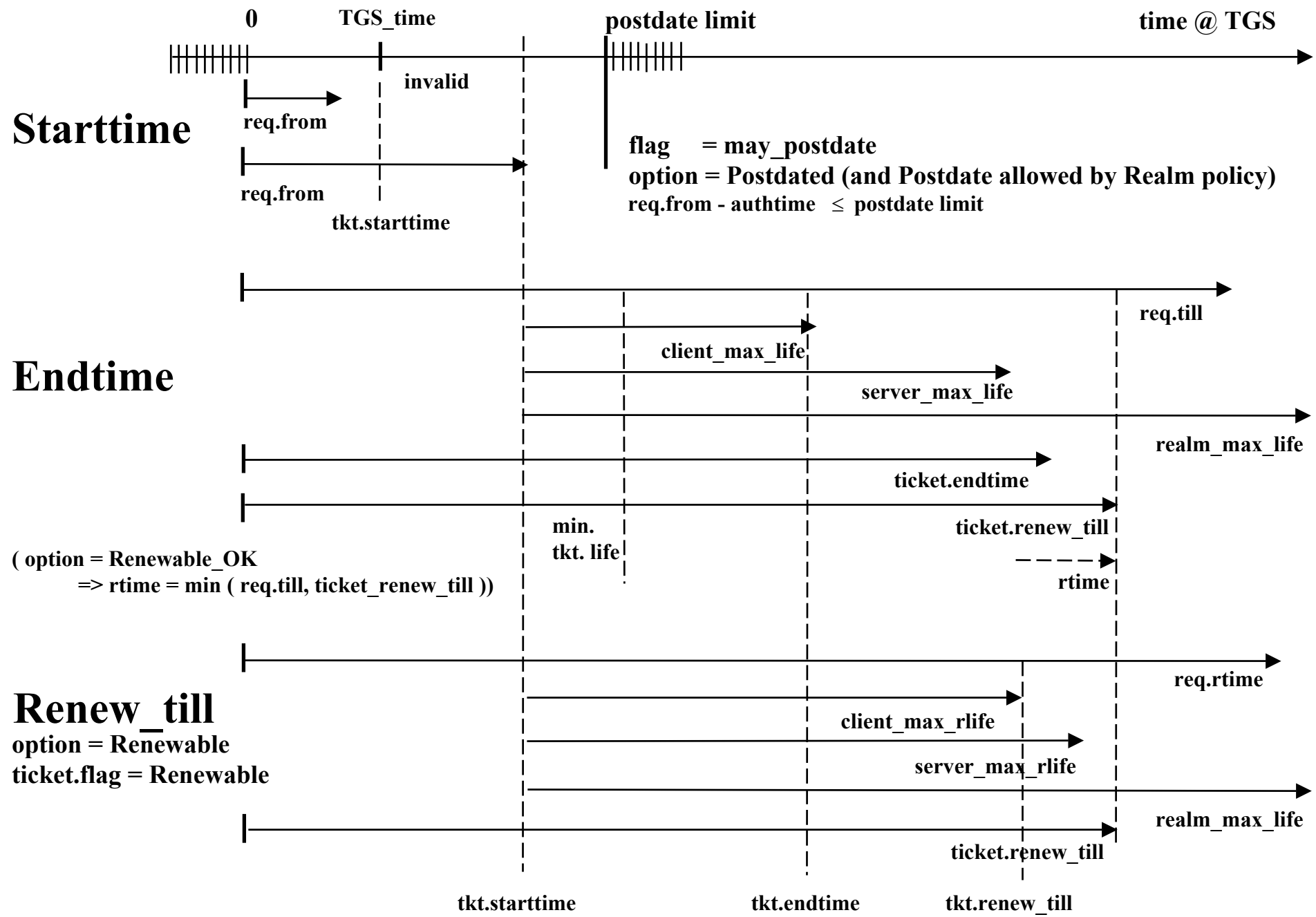
Solution

o introduce renewable tickets and “renew_til” limit
- renewal => *endtime = min { max_renewable_life, renew_until }*
o introduce “invalid” ticket status, postdate limit, and ticket validation
o tickets are renewed and/or validated unless
they are placed on ticket revocation list already
o finite ticket holding time on revocation list

TGT Lifetime Determination at AS



Ticket Lifetime Determination at TGS



Message Options and Ticket Flags

ap_options	kdc_options	who interprets option	ticket flags	who checks ticket flags
USE_SESSION_KEY	FORWARDABLE	AS, TGS	FORWARDABLE	TGS
MUTUAL_REQUIRED	FORWARDED	TGS	FORWARDED	TGS, SERVICE
	PROXIABLE	AS, TGS	PROXIABLE	TGS
	PROXY	TGS	PROXY	TGS, SERVICE
	ALLOW-POSTDATE	AS, TGS	MAY-POTDATE	TGS
	POSTDATED	AS, TGS	POSTDATED	TGS
	RENEWABLE	AS, TGS	RENEWABLE	TGS
	RENEWABLE-OK	AS, TGS		
	ENC-TKT-IN-SKEY	TGS		
	RENEW	TGS		
	VALIDATE	TGS	INVALID	TGS, SERVICE
			INITIAL (tkt was issued byAS_REQ)	SERVICE (E.G., PASSWD)
			PRE-AUTHENT	TGS, SERVICE
			HW-AUTHENT	TGS, SERVICE

Key Versions in Kerberos V5

Per Principal (p) set of triples **< { p_key }K_{KDC}, p_kvno, k_kvno >**
TGS_REQ returns tickets encrypted in key with *highest* p_kvno
Encoding in the KDC database: separate principal entry per key

Motivation for KDC support of multiple p_kvno

Ticket renewal by KDC

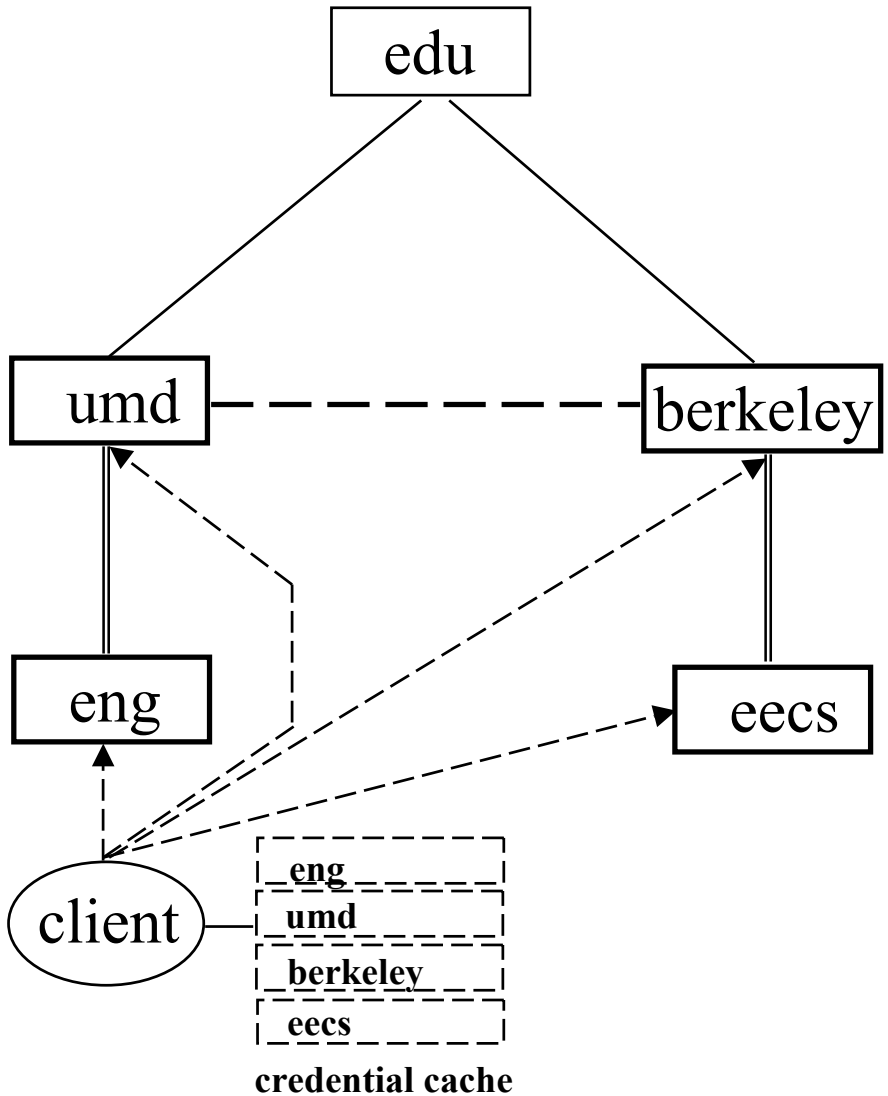
- Scenario:
  1. Server changes its key
  2. Client, which has a renewable ticket encrypted in old server key, requests renewal
  3. KDC needs to remember old server key to decrypt ticket and renew it.
  4. KDC verifies ticket renewability, renews ticket, and re-encrypts it in key with highest p_kvno

Ticket postdating (similar scenario)

Realm Hierarchy

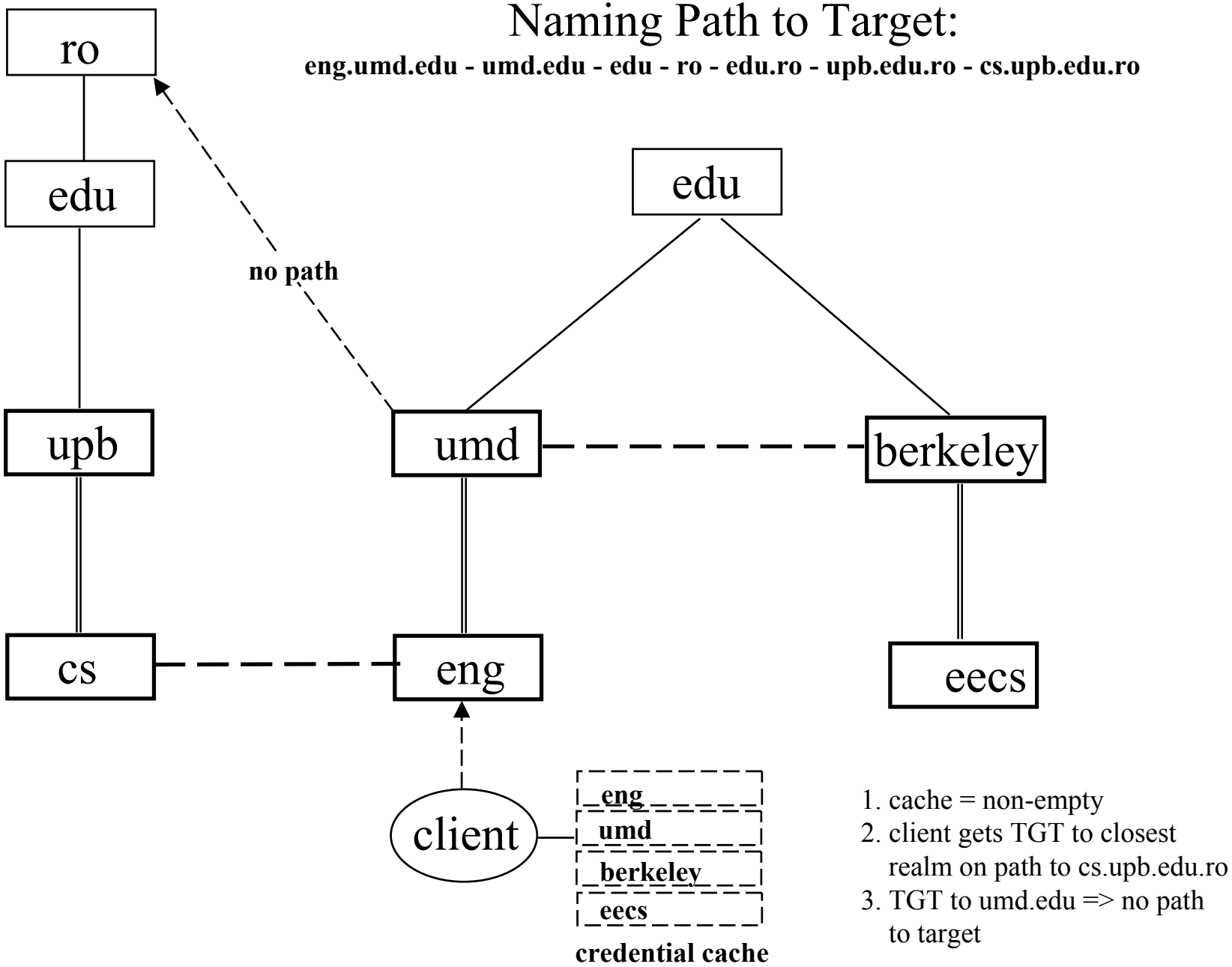
Naming Path to Target Realm:

eng.umd.edu - umd.edu -edu - berkeley.edu - eecs.berkeley.edu



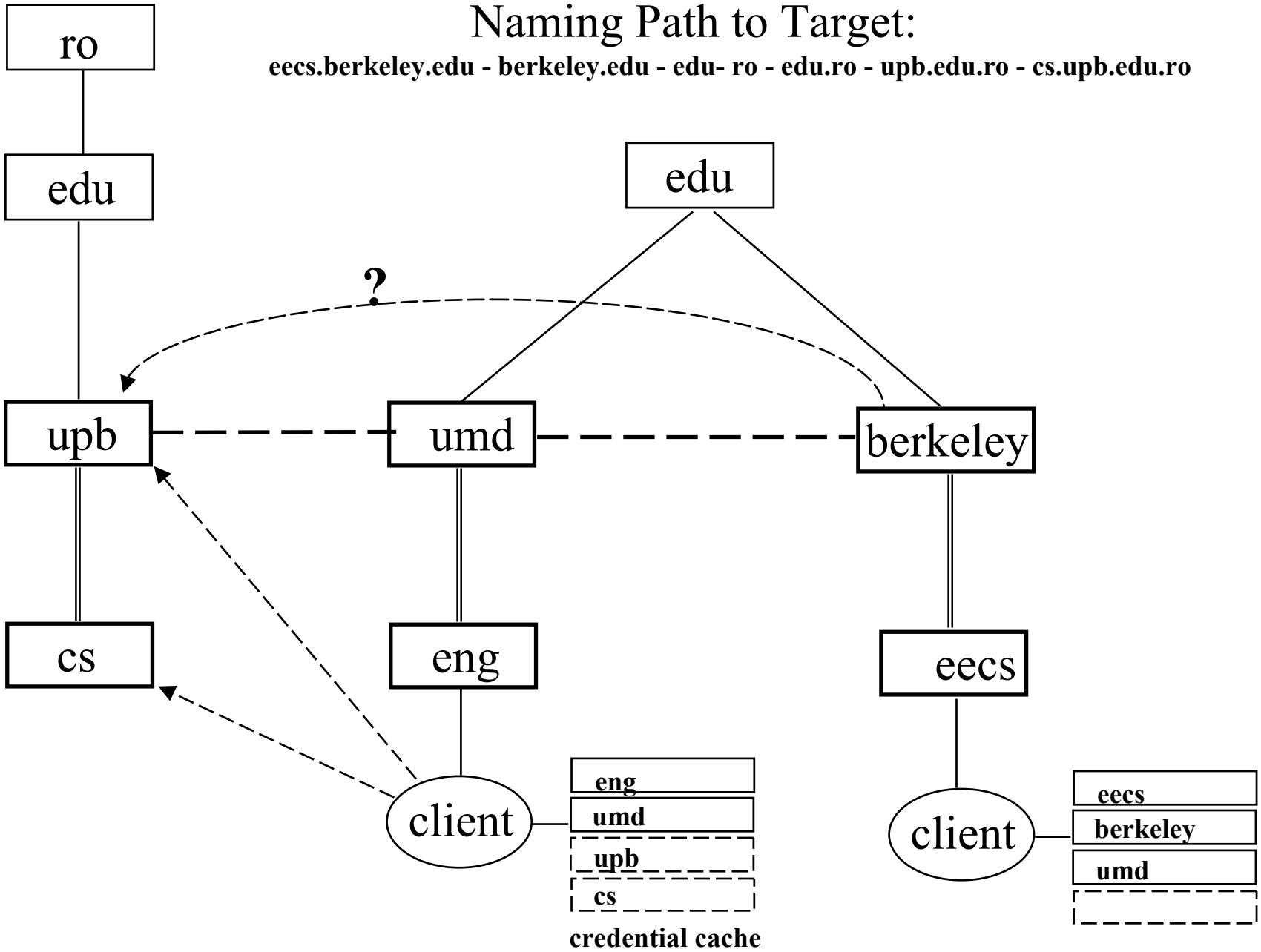
1. cache = empty
2. client traverses trust path and obtains TGTs (e.g., TGT to target realms eecs.berkeley.edu)

Inter-realm Authentication Algorithm - An Example



Naming Path to Target:

eecs.berkeley.edu - berkeley.edu - edu- ro - edu.ro - upb.edu.ro - cs.upb.edu.ro



Pre-authentication
(and other password-discovery countermeasures)

Motivation

- o AS_REQ/AS_REP generate any number of known plaintext - chiphertext pairs
- o Off-line password guessing attacks

Solution

- o PADATA = $\{ \text{ctime} \}^{K_{client}}$ required in AS_REQ
- o AS_REP sent only if plaintext ctime of AS_REQ = decrypted PADATA

Separation of Human vs. Server Principals

Motivation

- o TGS_REQ specifies a human principal instead of a server principal
- o Effect of pre-authentication is circumvented

Solution

- o no_ap_tkt flag set for human principals

Pre-authentication etc. (continued)

Separation of principal keys per realm

Motivation

- o Principals registered in multiple realms may use the same key
- o Theft of key in one realm => compromised keys in all realms

Solution

- o key = { OWF (p_name, p_ realm, passwd) }$^{K_{KDC}}$ is stored in KDC database

- o default "salt": < p_name, p_realm> ; new realm name => obsolete "salt"
  => wrong "pdata" in AS_REQ => user cannot login
- o obsolete "salt" => second chance login => KDC includes new "salt" in error message
- o "pdata" of AS_REP contains new "salt" if any

Double TGT Authentication - Motivation

Kerberos V4 : User-to-Host Authentication

- User inputs decryption key (i.e., password) ; Server gets its key from *srvtab*

PROBLEM:

€ User-to-User Authentication

- Workstations cannot offer authenticated services; *srvtab* cannot be protected
- Idle public workstations cannot be authenticated

€ Scaling Constraints

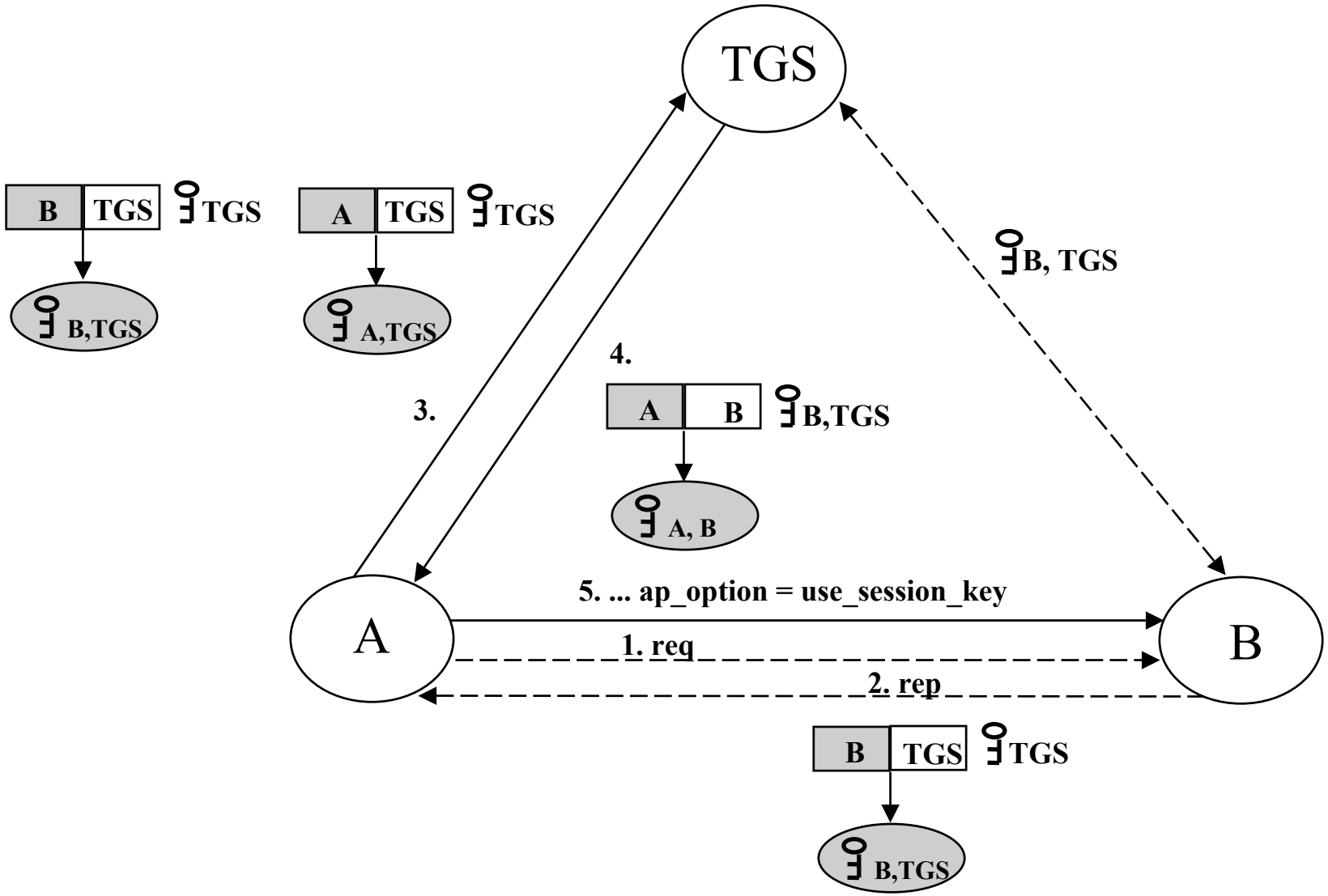
- Neither added state nor added load to Kerberos
- No added frequently changing fields to KDC database
- One transaction per connection

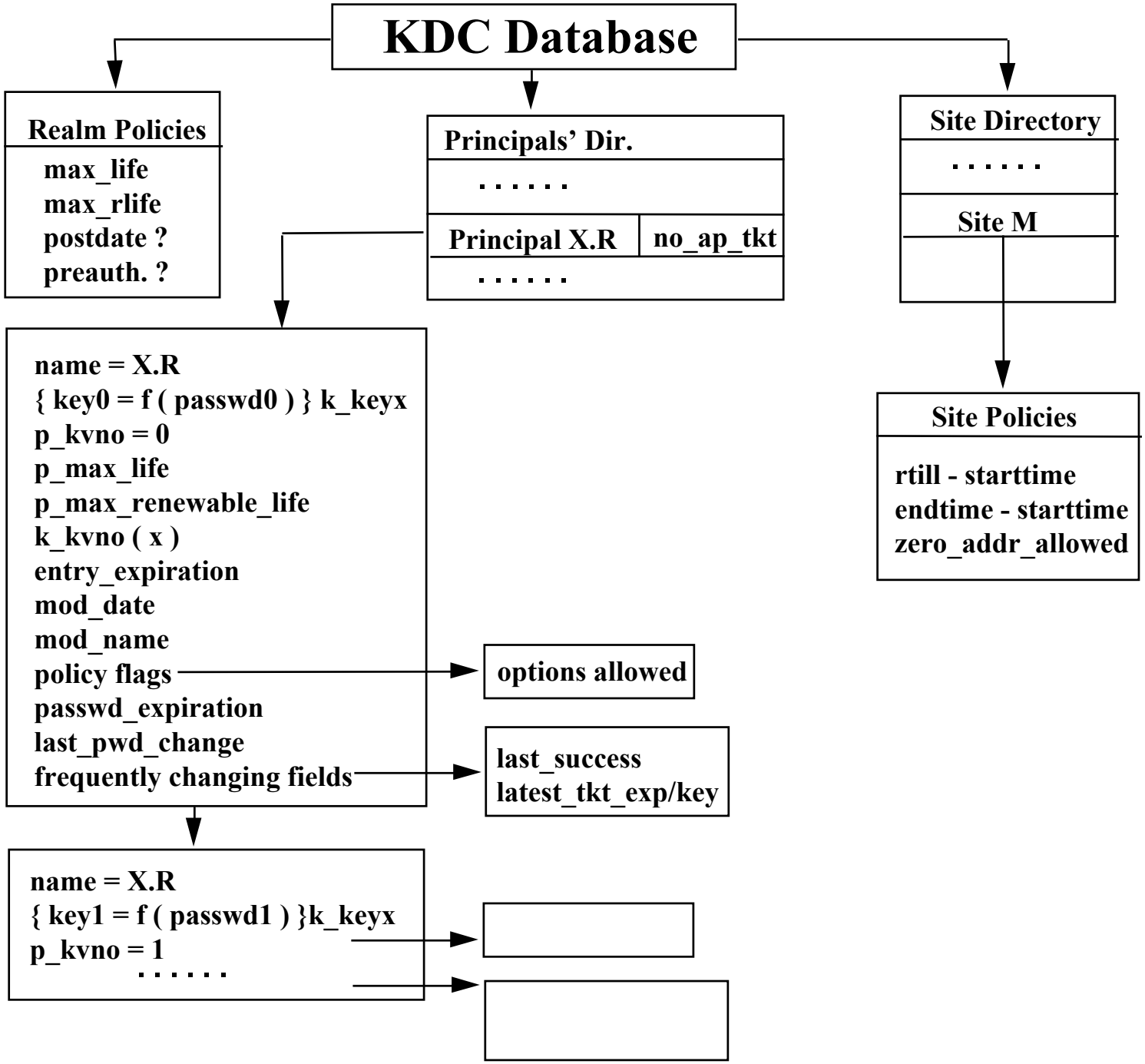
SOLUTION:

€ Client initiates protocol with Kerberos

€ Use Double TGT Authentication (aka. ENC-TKT-IN-SKEY)

Double TGT Authentication - Message Flows

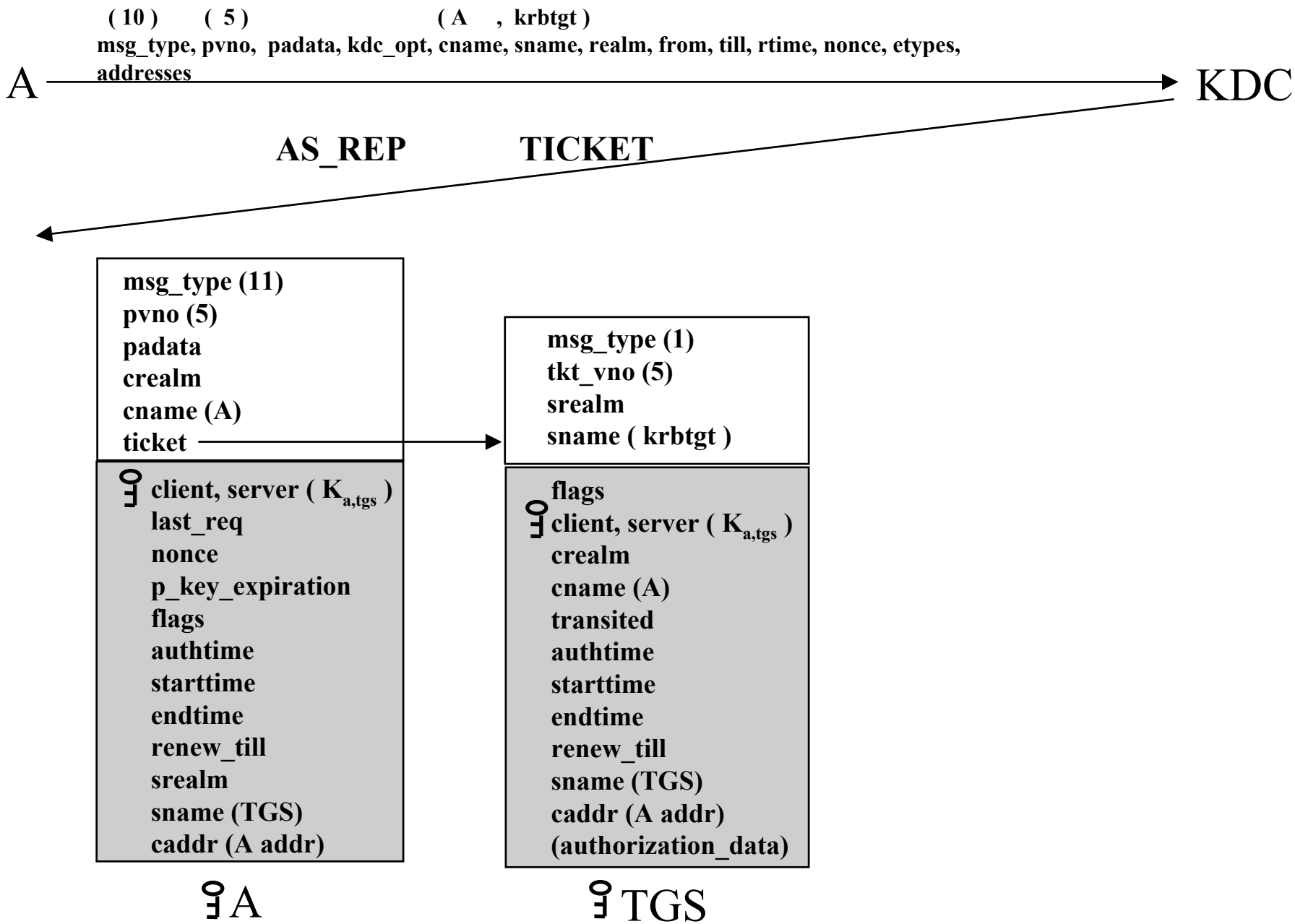



KDC Database
Realm Policies
max_life
max_rlife
postdate ?
preauth. ?
Principals' Dir.
. . . . . .
Principal X.R
no_ap_tkt
. . . . . .
Site Directory
. . . . . .
Site M
Site Policies
rtill - starttime
endtime - starttime
zero_addr_allowed
name = X.R
{ key0 = f (passwd0) } k_keyx
p_kvno = 0
p_max_life
p_max_renewable_life
k_kvno (x)
entry_expiration
mod_date
mod_name
policy flags
passwd_expiration
last_pwd_change
frequently changing fields
options allowed
last_success
latest_tkt_exp/key
name = X.R
{ key1 = f (passwd1) }k_keyx
p_kvno = 1
. . . . . .

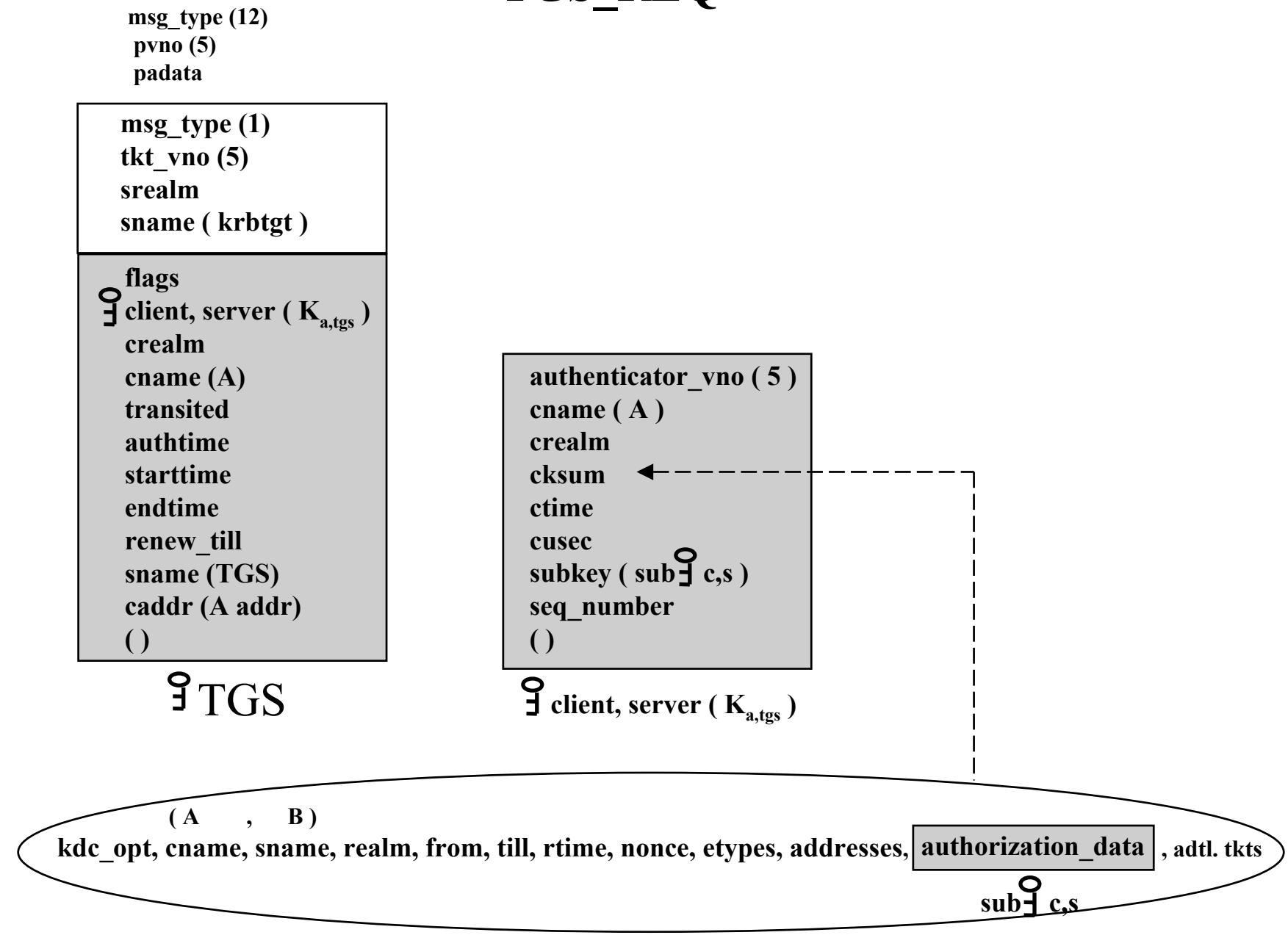
Kerberos V 5

Message Formats and Protocol Flows

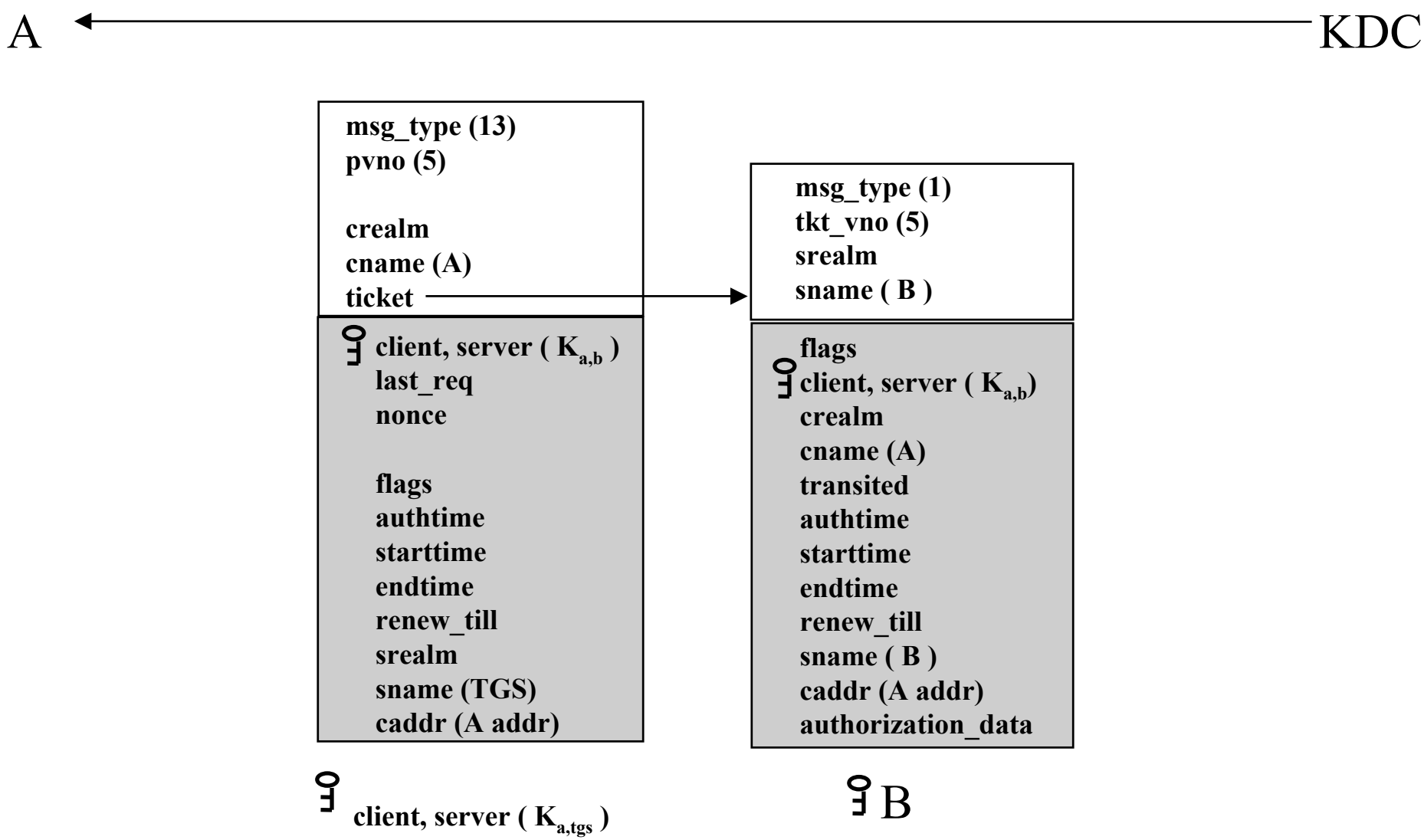
AS_REQ / AS_REP


(10) (5) (A , krbtgt)
msg_type, pvno, padata, kdc_opt, cname, sname, realm, from, till, rtime, nonce, etypes, addresses
A
KDC
AS_REP
TICKET
msg_type (11)
pvno (5)
padata
crealm
cname (A)
ticket
client, server (K a,tgs)
last_req
nonce
p_key_expiration
flags
authtime
starttime
endtime
renew_till
srealm
sname (TGS)
caddr (A addr)
msg_type (1)
tkt_vno (5)
srealm
sname (krbtgt)
flags
client, server (K a,tgs)
crealm
cname (A)
transited
authtime
starttime
endtime
renew_till
sname (TGS)
caddr (A addr)
(authorization_data)
A
TGS

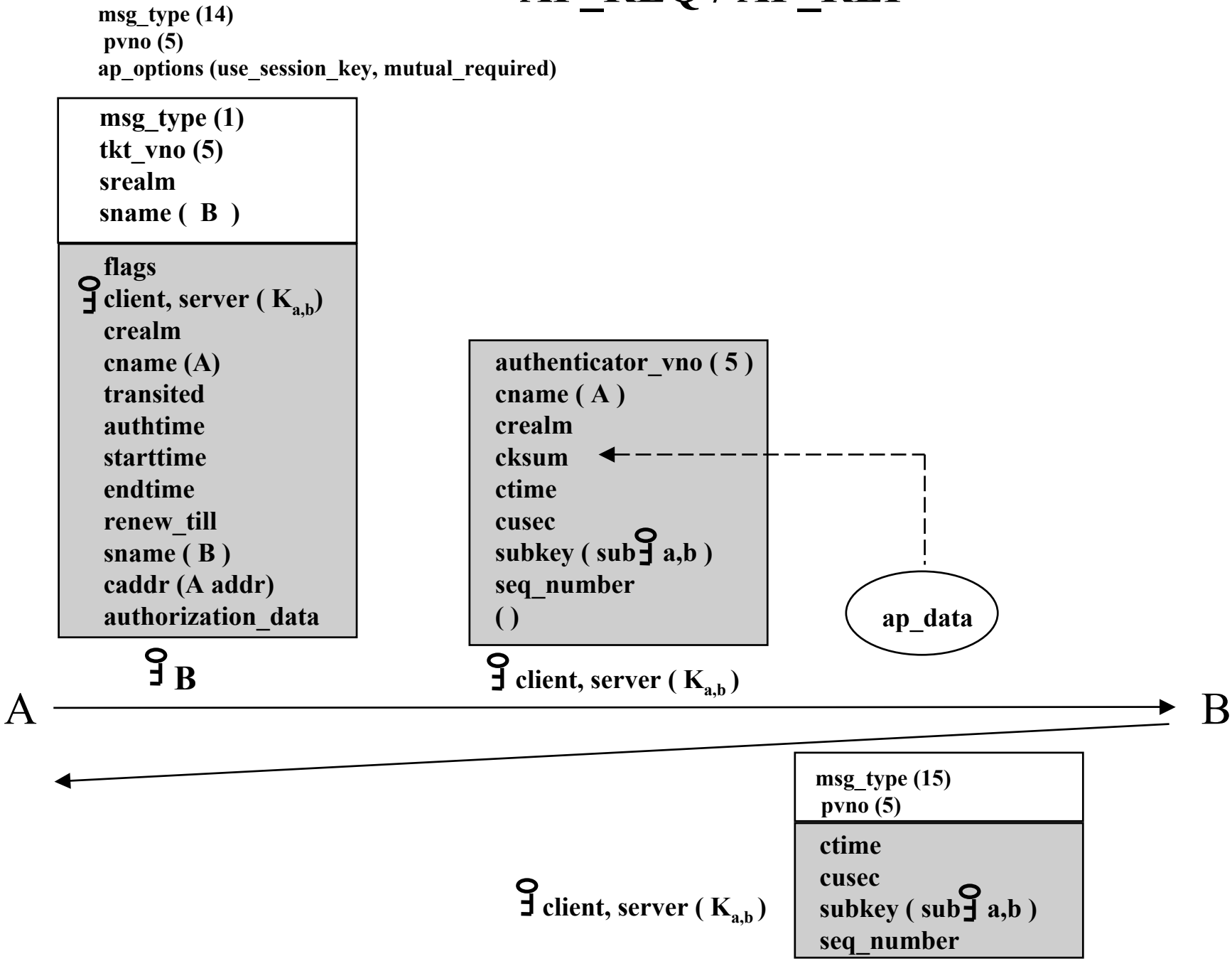
TGS_REQ


msg_type (12)
pvno (5)
padata
msg_type (1)
tkt_vno (5)
srealm
sname (krbtgt)
flags
client, server ($K_{a,tgs}$)
crealm
cname (A)
transited
authtime
starttime
endtime
renew_till
sname (TGS)
caddr (A addr)
( )
TGS
authenticator_vno (5)
cname (A)
crealm
cksum
ctime
cusec
subkey (sub c,s)
seq_number
( )
client, server ($K_{a,tgs}$)
(A , B)
kdc_opt, cname, sname, realm, from, till, rtime, nonce, etypes, addresses, authorization_data , adtl. tkts
sub c,s
A
KDC

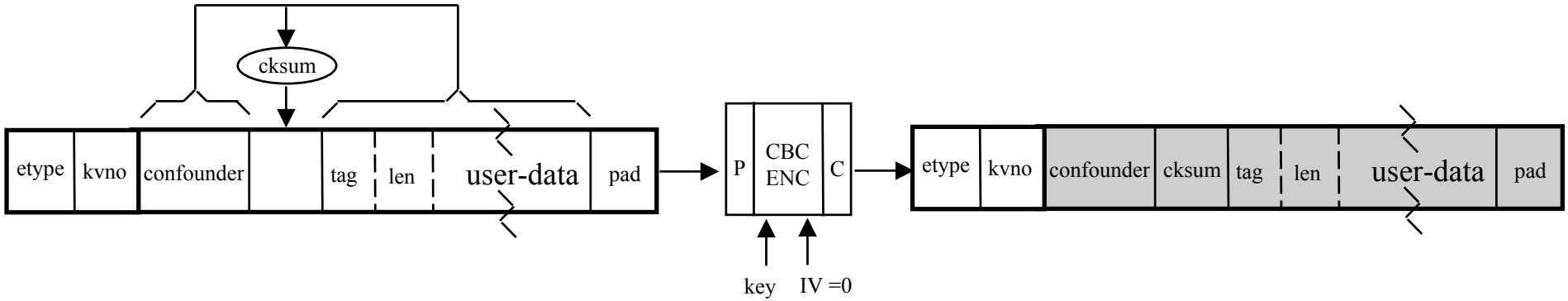
TGS_REP


A
KDC
msg_type (13)
pvno (5)
crealm
cname (A)
ticket
client, server ($K_{a,b}$)
last_req
nonce
flags
authtime
starttime
endtime
renew_till
srealm
sname (TGS)
caddr (A addr)
client, server ($K_{a,tgs}$)
msg_type (1)
tkt_vno (5)
srealm
sname (B)
flags
client, server ($K_{a,b}$)
crealm
cname (A)
transited
authtime
starttime
endtime
renew_till
sname (B)
caddr (A addr)
authorization_data
B

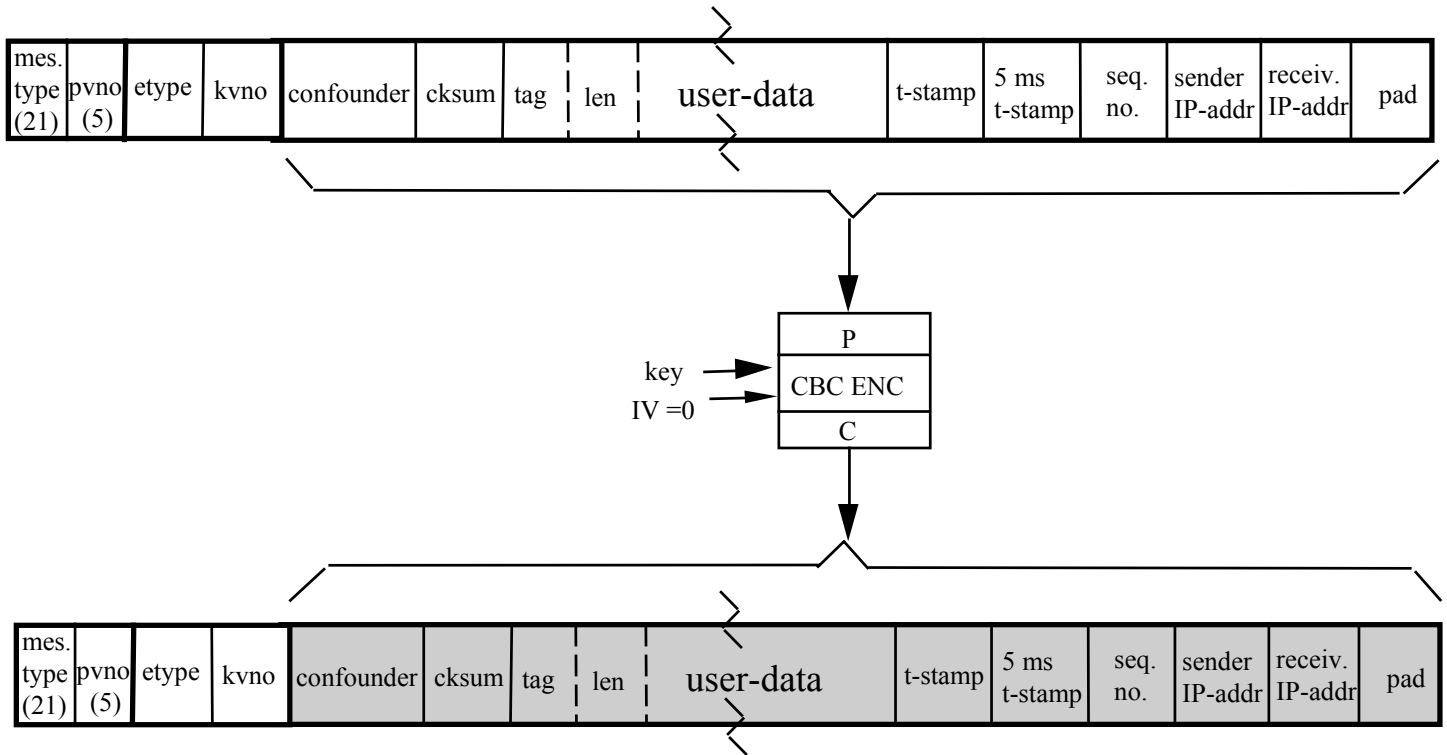
AP_REQ / AP_REP


msg_type (14)
pvno (5)
ap_options (use_session_key, mutual_required)
msg_type (1)
tkt_vno (5)
srealm
sname (B)
flags
client, server ($K_{a,b}$)
crealm
cname (A)
transited
authtime
starttime
endtime
renew_till
sname (B)
caddr (A addr)
authorization_data
B
authenticator_vno (5)
cname (A)
crealm
cksum
ctime
cusec
subkey (sub a,b)
seq_number
( )
ap_data
client, server ($K_{a,b}$)
A
B
msg_type (15)
pvno (5)
client, server ($K_{a,b}$)
ctime
cusec
subkey (sub a,b)
seq_number

Data Encryption (for Confidentiality)

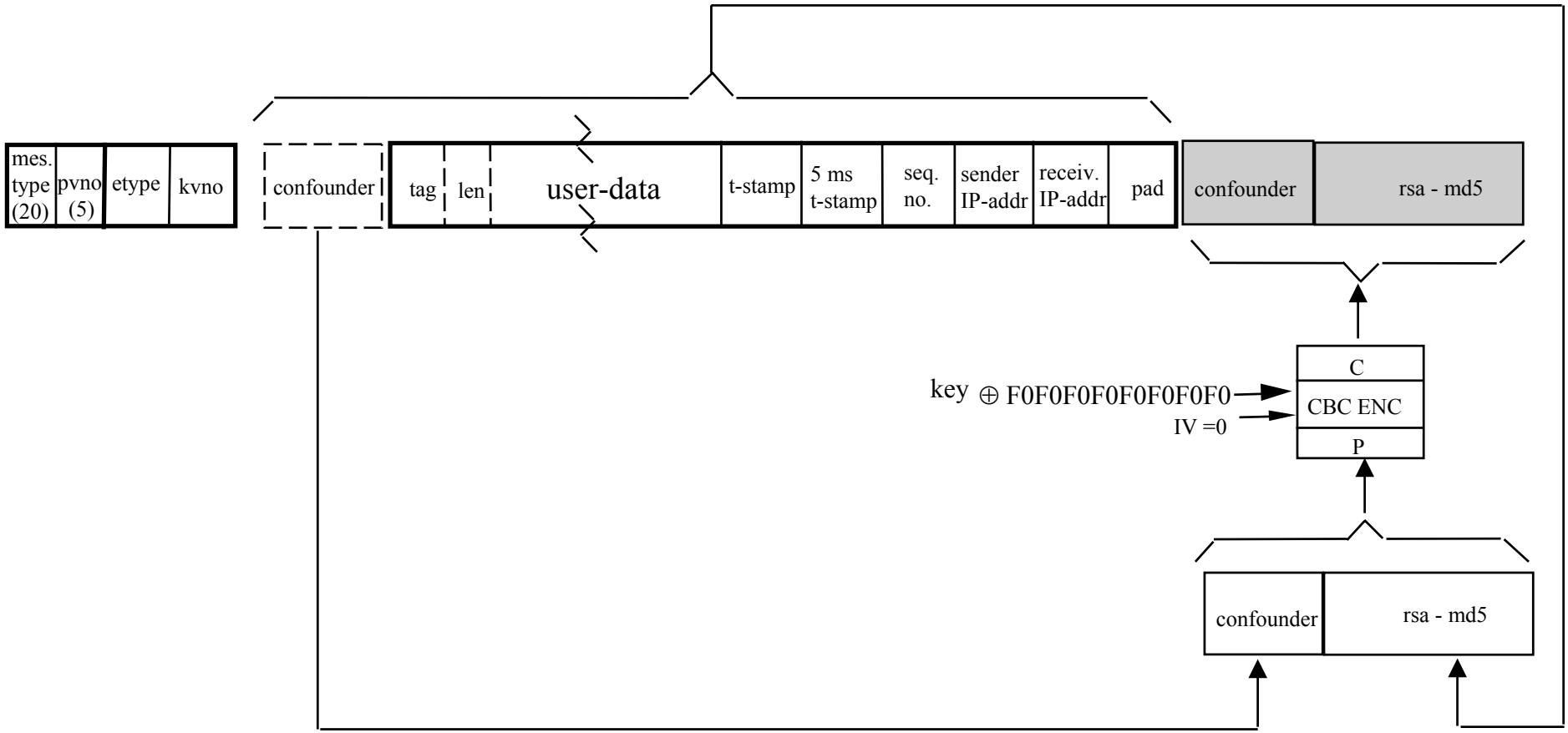


krb_priv Messages

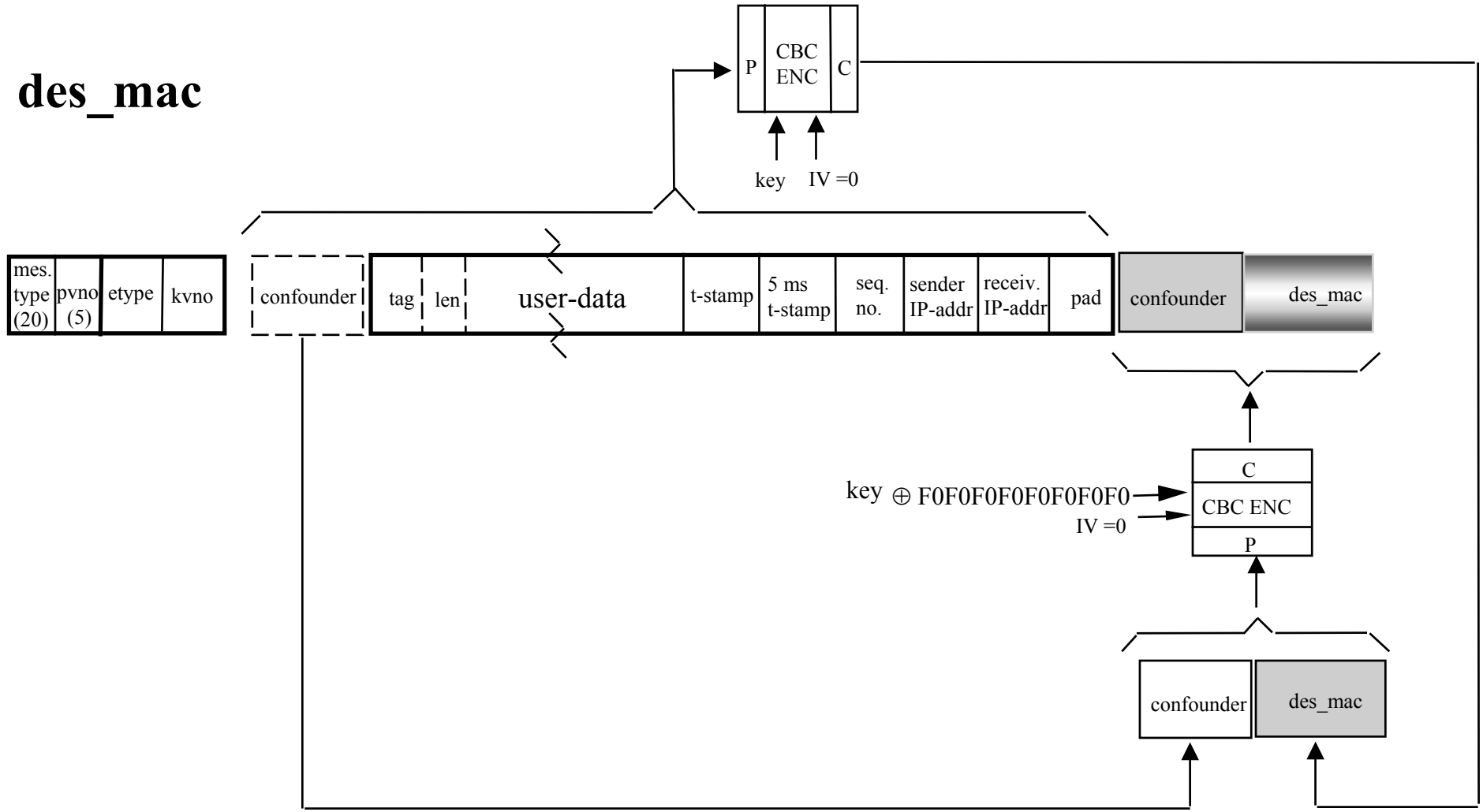


Data Integrity - kerb_safe Messages

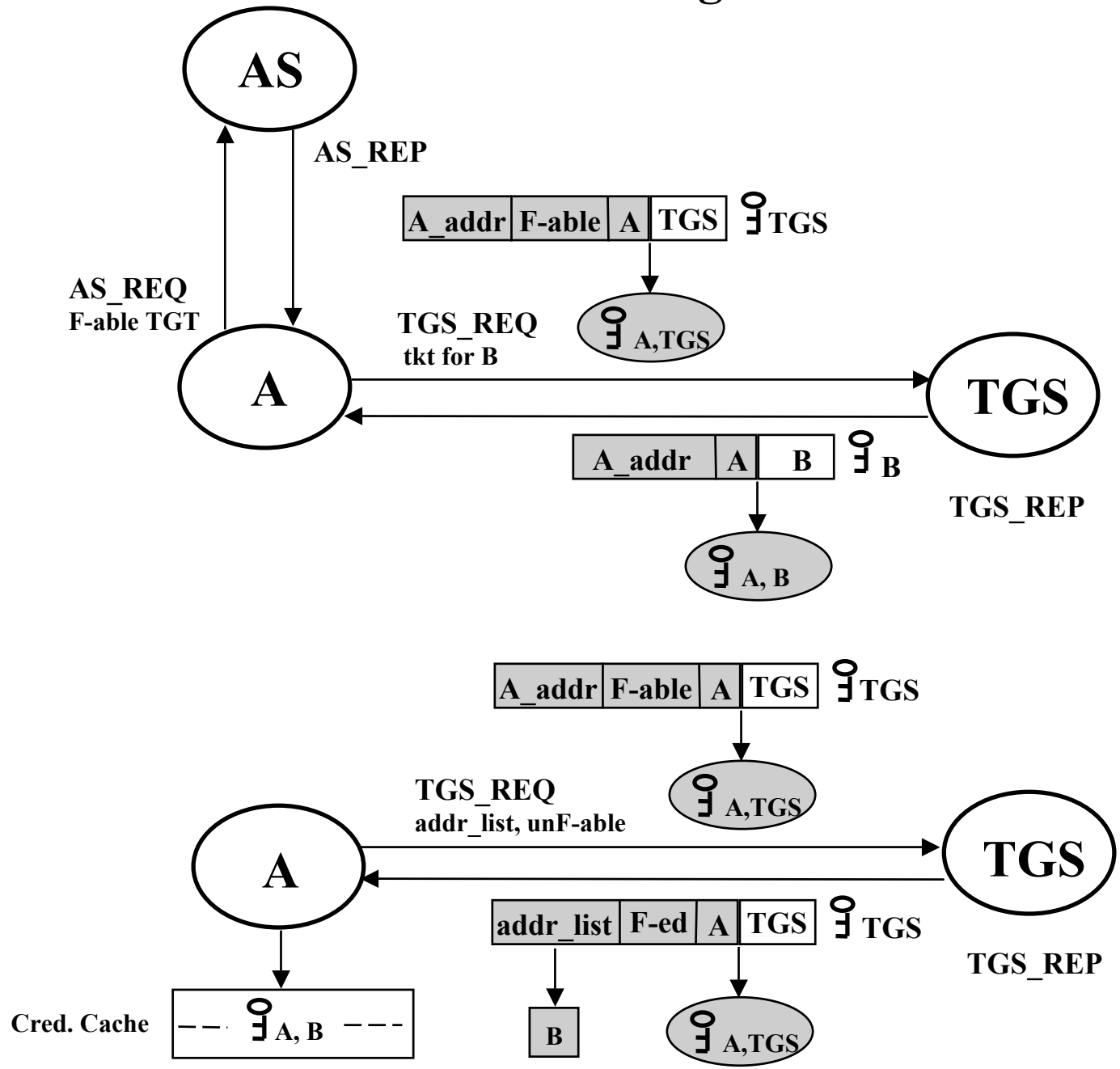
rsa_md5_des

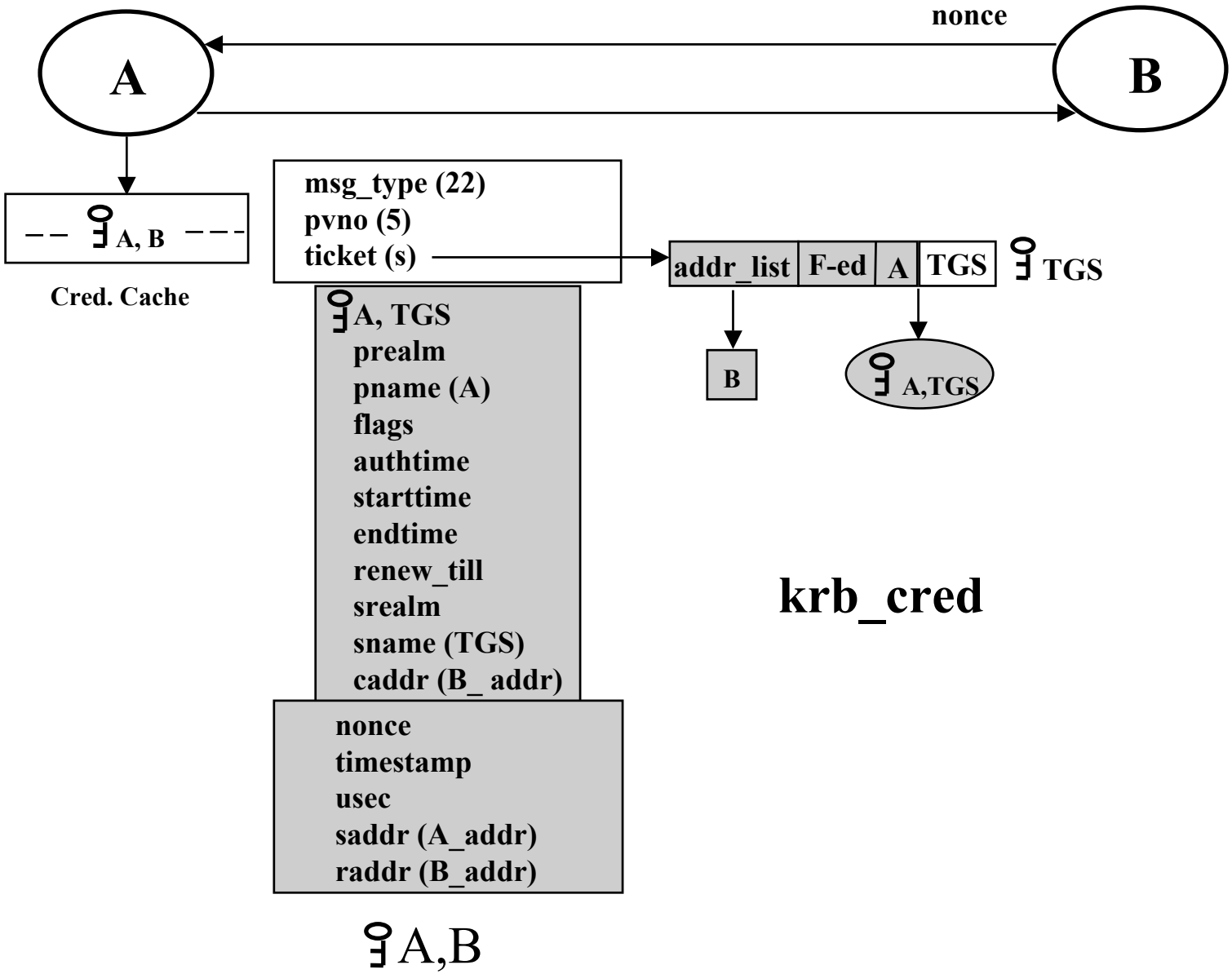


Data Integrity - kerb_safe Messages (ctnd.)

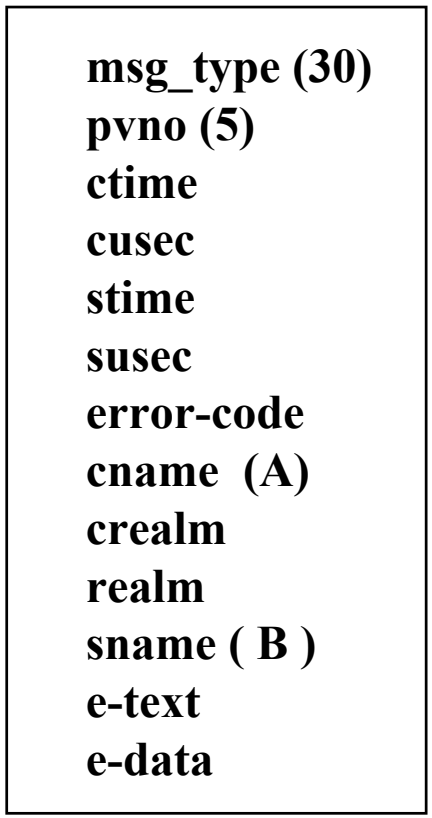


Forwarding


AS
AS_REP
A_addr F-able A TGS
TGS
AS_REQ
F-able TGT
TGS_REQ
tkt for B
A,TGS
A
TGS
A_addr A B
B
TGS_REP
A, B
A_addr F-able A TGS
TGS
TGS_REQ
addr_list, unF-able
A,TGS
A
TGS
addr_list F-ed A TGS
TGS
TGS_REP
Cred. Cache
A, B
B
A,TGS


nonce
A
B
A, B
Cred. Cache
msg_type (22)
pvno (5)
ticket (s)
addr_list
F-ed
A
TGS
TGS
B
A,TGS
A, TGS
prealm
pname (A)
flags
authtime
starttime
endtime
renew_till
srealm
sname (TGS)
caddr (B_ addr)
nonce
timestamp
usec
saddr (A_addr)
raddr (B_addr)
A,B
krb_cred

krb_error


msg_type (30)
pvno (5)
ctime
cusec
stime
susec
error-code
cname (A)
crealm
realm
sname (B)
e-text
e-data