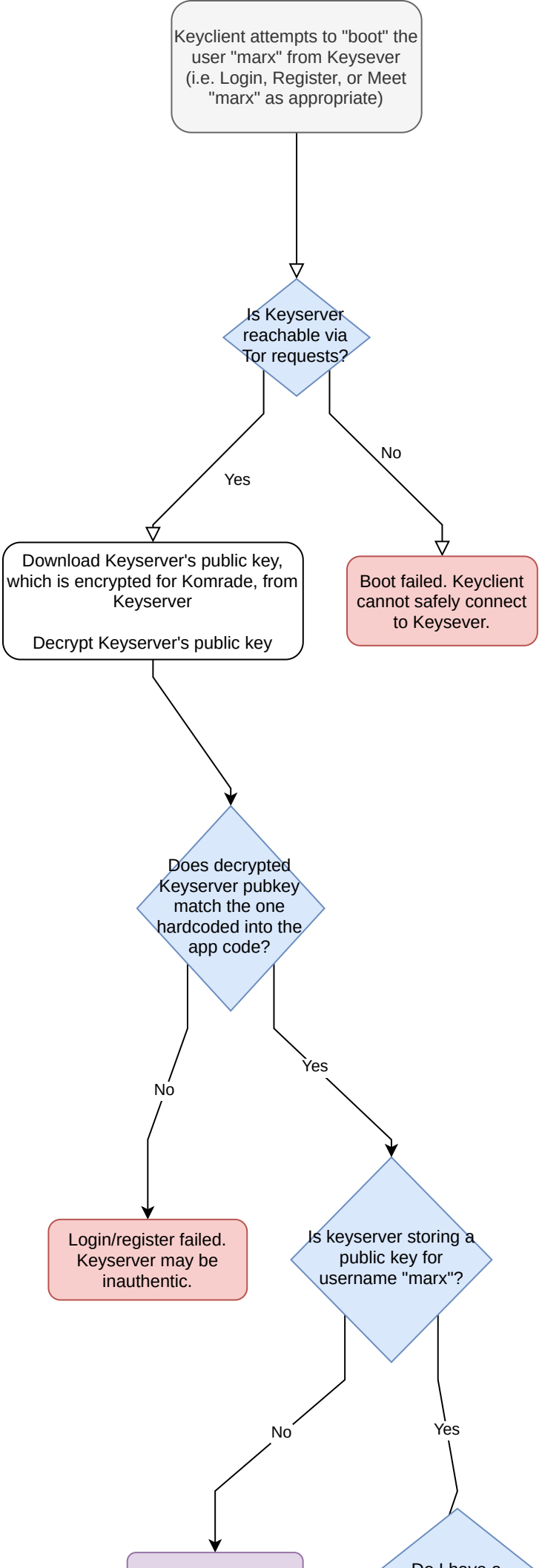
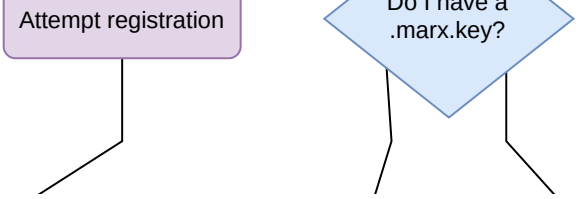
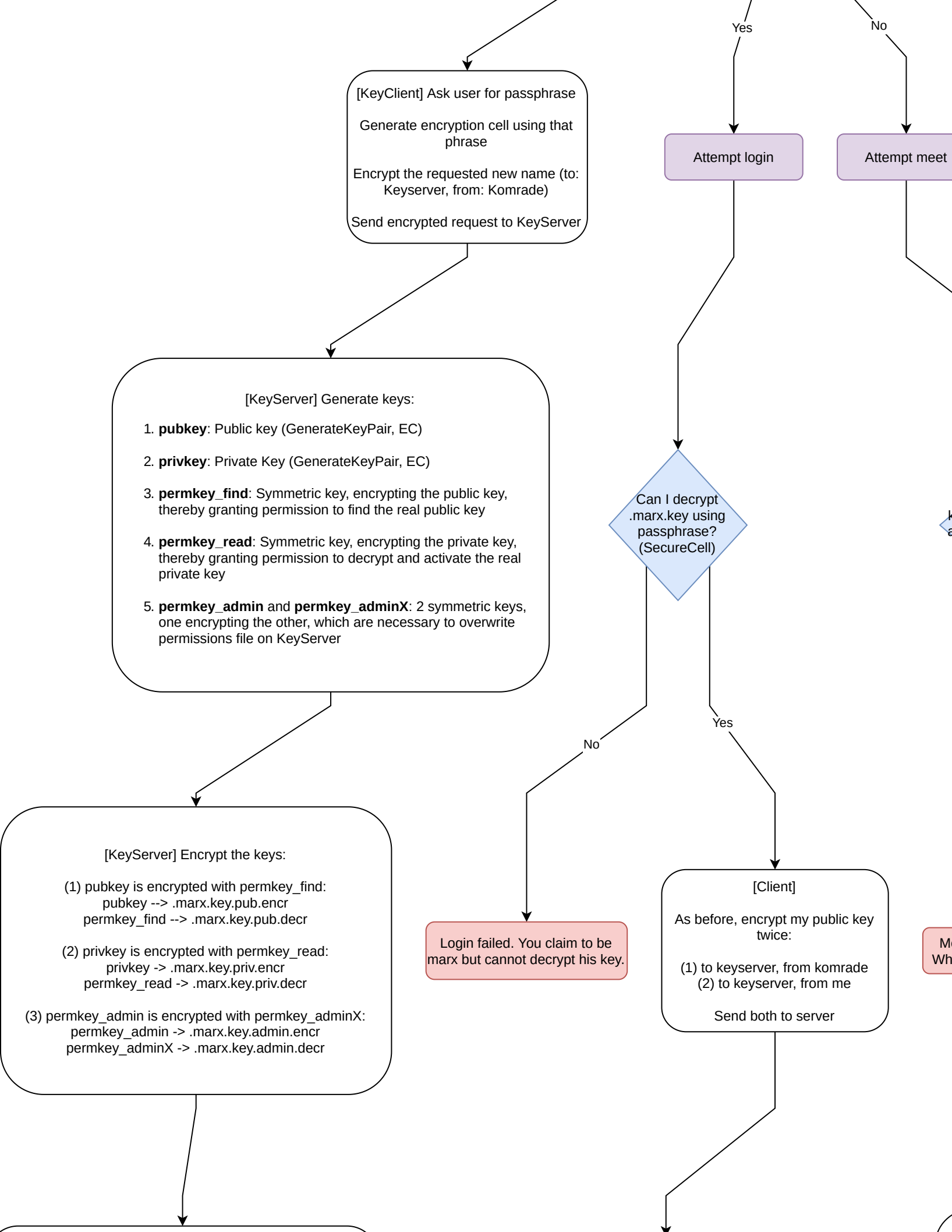
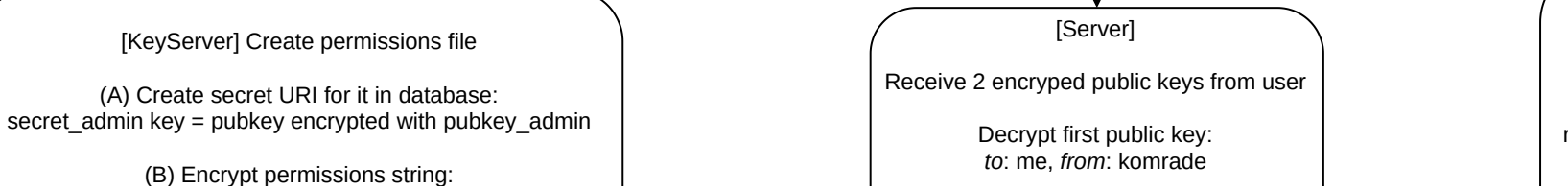
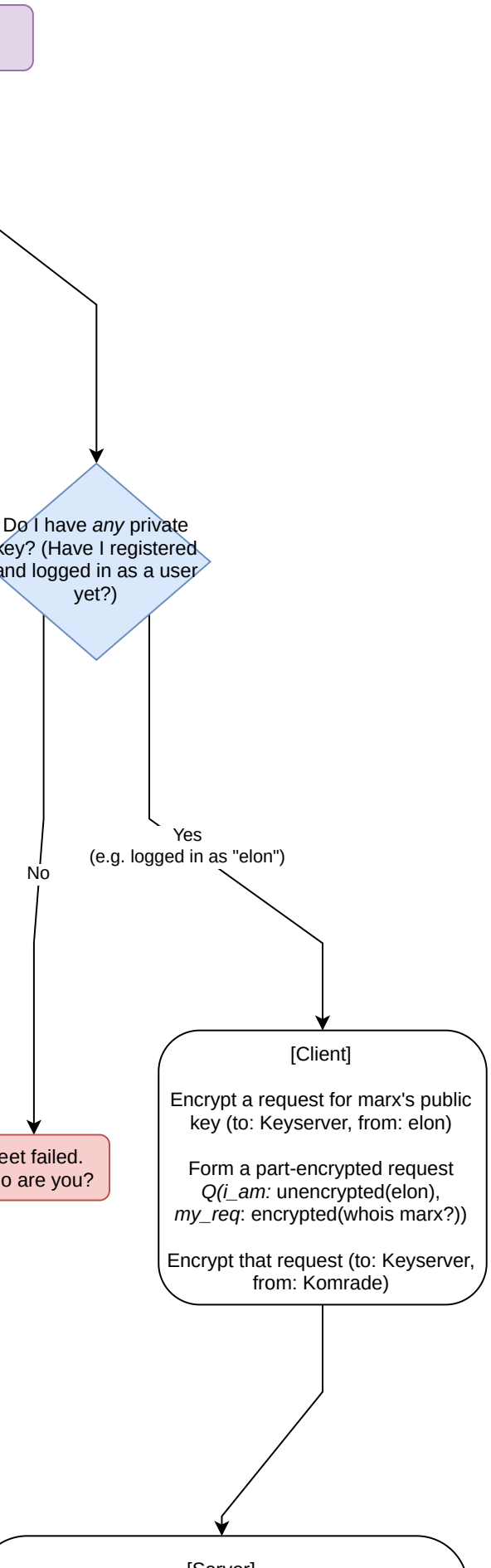

Keyclient attempts to "boot" the user "marx" from Keysever (i.e. Login, Register, or Meet "marx" as appropriate)
Is Keyserver reachable via Tor requests?
Yes
No
Download Keyserver's public key, which is encrypted for Komrade, from Keyserver
Decrypt Keyserver's public key
Boot failed. Keyclient cannot safely connect to Keysever.
Does decrypted Keyserver pubkey match the one hardcoded into the app code?
No
Yes
Login/register failed. Keyserver may be inauthentic.
Is keyserver storing a public key for username "marx"?
No
Yes


Attempt registration
Do I have a .marx.key?


Yes
No
[KeyClient] Ask user for passphrase
Generate encryption cell using that phrase
Encrypt the requested new name (to: Keyserver, from: Komrade)
Send encrypted request to KeyServer
Attempt login
Attempt meet
[KeyServer] Generate keys:
1. pubkey: Public key (GenerateKeyPair, EC)
2. privkey: Private Key (GenerateKeyPair, EC)
3. permkey_find: Symmetric key, encrypting the public key, thereby granting permission to find the real public key
4. permkey_read: Symmetric key, encrypting the private key, thereby granting permission to decrypt and activate the real private key
5. permkey_admin and permkey_adminX: 2 symmetric keys, one encrypting the other, which are necessary to overwrite permissions file on KeyServer
Can I decrypt .marx.key using passphrase? (SecureCell)
No
Yes
[KeyServer] Encrypt the keys:
(1) pubkey is encrypted with permkey_find:
pubkey --> .marx.key.pub.encr
permkey_find --> .marx.key.pub.decr
(2) privkey is encrypted with permkey_read:
privkey -> .marx.key.priv.encr
permkey_read -> .marx.key.priv.decr
(3) permkey_admin is encrypted with permkey_adminX:
permkey_admin -> .marx.key.admin.encr
permkey_adminX -> .marx.key.admin.decr
Login failed. You claim to be marx but cannot decrypt his key.
[Client]
As before, encrypt my public key twice:
(1) to keyserver, from komrade
(2) to keyserver, from me
Send both to server

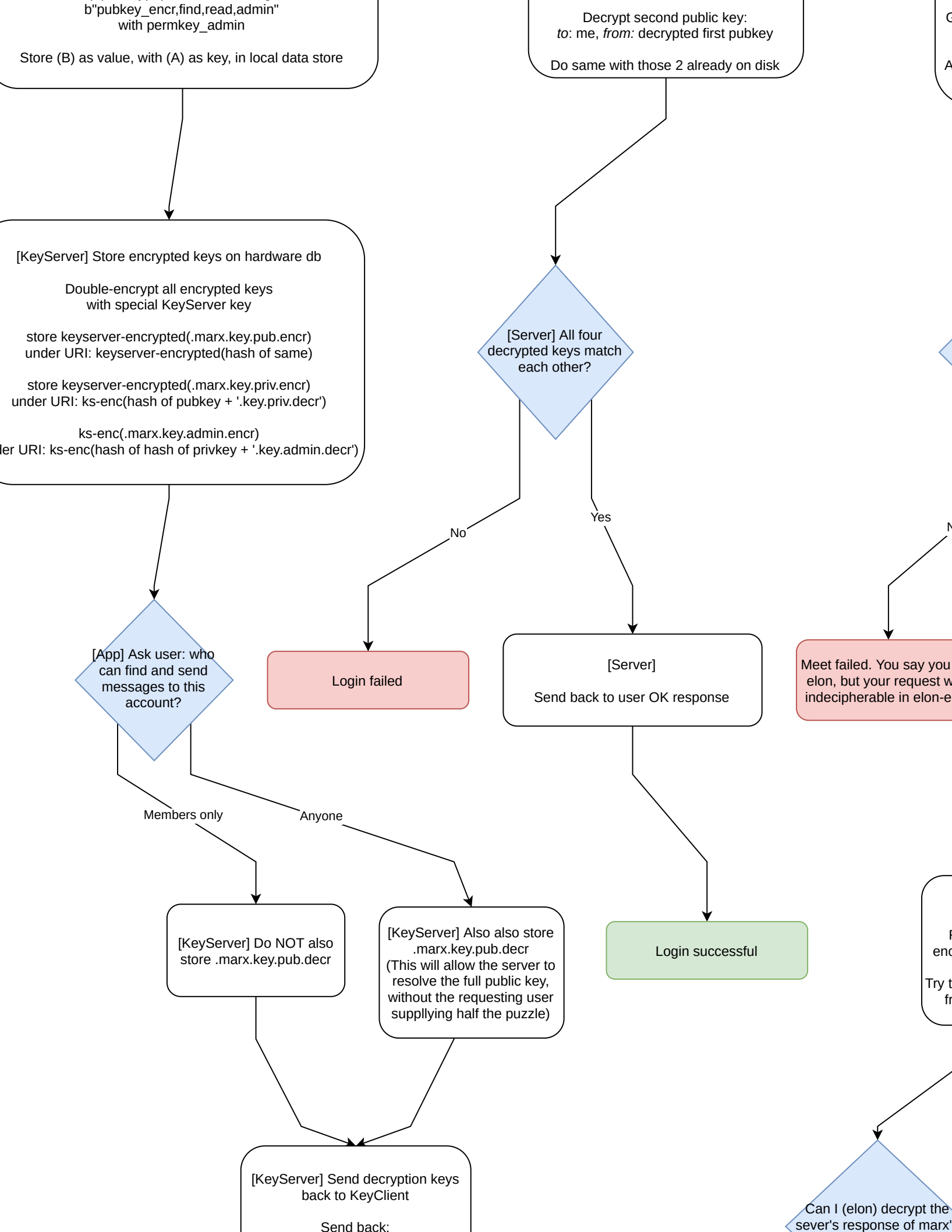

[KeyServer] Create permissions file
(A) Create secret URI for it in database:
secret_admin key = pubkey encrypted with pubkey_admin
(B) Encrypt permissions string:
[Server]
Receive 2 encryped public keys from user
Decrypt first public key:
to: me, *from*: komrade


Do I have *any* private key? (Have I registered and logged in as a user yet?)
No
Yes
(e.g. logged in as "elon")
eet failed.
o are you?
[Client]
Encrypt a request for marx's public key (to: Keyserver, from: elon)
Form a part-encrypted request *Q(i_am:* unencrypted(elon), *my_req*: encrypted(whois marx?))
Encrypt that request (to: Keyserver, from: Komrade)

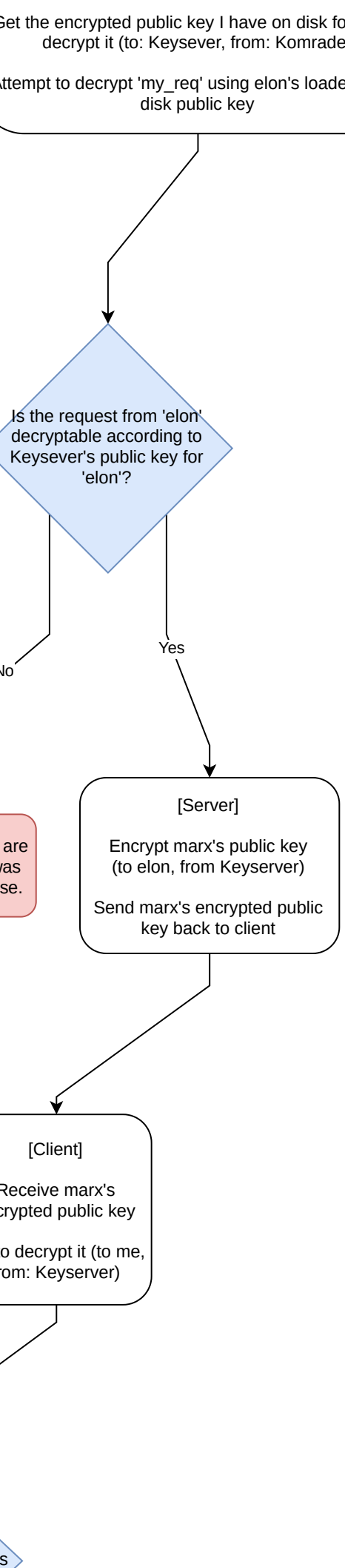
[Server]

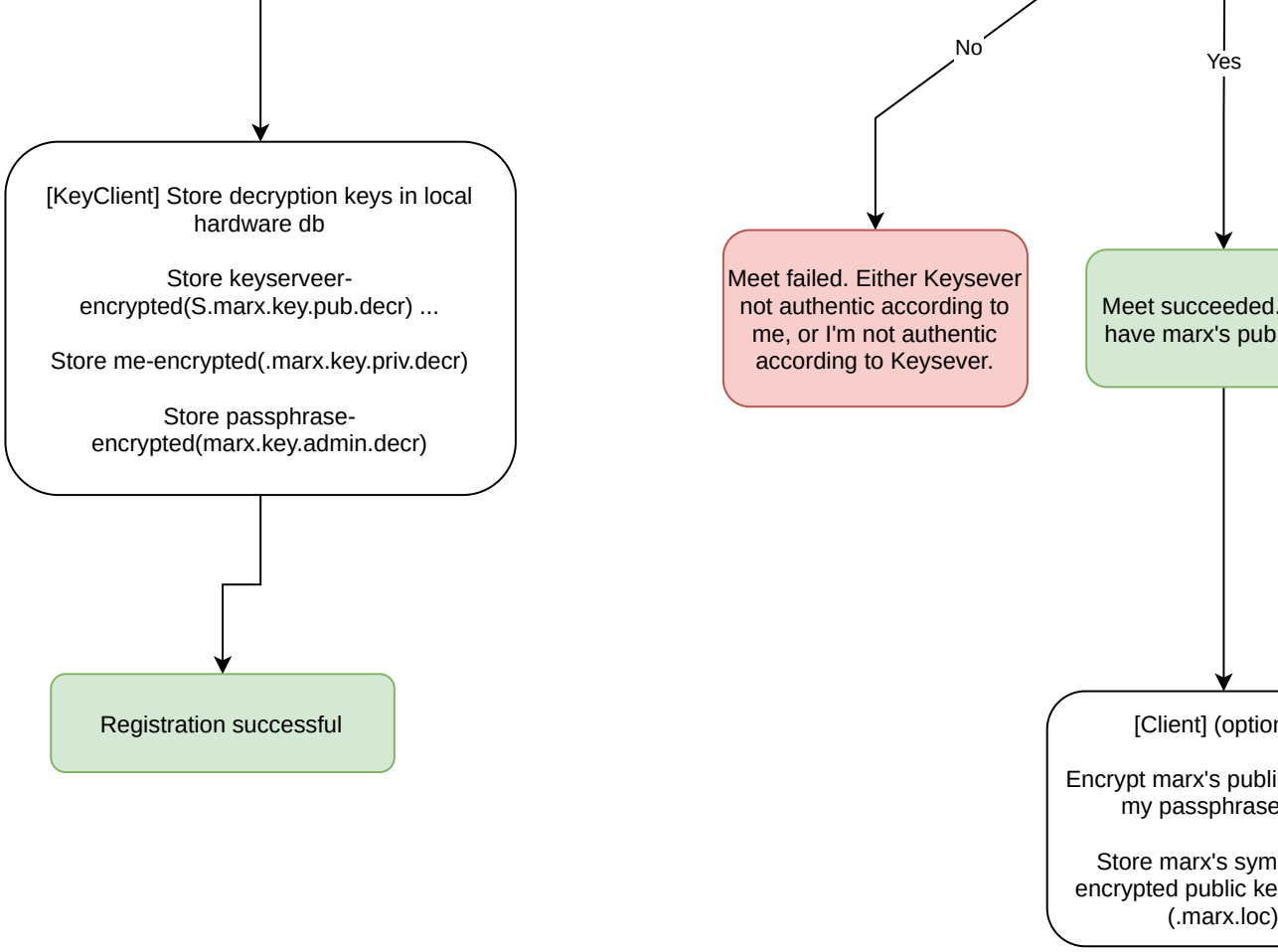
Receive double-encrypted request from client

Decrypt first time (to: Keysever, from: Komrade) to reveal a second encryted request [i.e. *Q(i_am:* elon, *my_req*: elon-encrypted 'whois marx']


b"pubkey_encr,find,read,admin"
with permkey_admin
Store (B) as value, with (A) as key, in local data store
Decrypt second public key:
to: me, from: decrypted first pubkey
Do same with those 2 already on disk
[KeyServer] Store encrypted keys on hardware db
Double-encrypt all encrypted keys
with special KeyServer key
store keyserver-encrypted(.marx.key.pub.encr)
under URI: keyserver-encrypted(hash of same)
store keyserver-encrypted(.marx.key.priv.encr)
under URI: ks-enc(hash of pubkey + '.key.priv.decr')
ks-enc(.marx.key.admin.encr)
er URI: ks-enc(hash of hash of privkey + '.key.admin.decr')
[Server] All four decrypted keys match each other?
No
Yes
[App] Ask user: who can find and send messages to this account?
Login failed
[Server]
Send back to user OK response
Members only
Anyone
[KeyServer] Do NOT also store .marx.key.pub.decr
[KeyServer] Also also store .marx.key.pub.decr
(This will allow the server to resolve the full public key, without the requesting user suppllying half the puzzle)
Login successful
[KeyServer] Send decryption keys back to KeyClient
Send back:


.marx.key.pub.decr
.marx.key.priv.decr
.marx.key.admin.decr
encrypted pubkey?


Get the encrypted public key I have on disk for 'elon', decrypt it (to: Keysever, from: Komrade)
Attempt to decrypt 'my_req' using elon's loaded-from-disk public key
Is the request from 'elon' decryptable according to Keysever's public key for 'elon'?
No
Yes
are
was
se.
[Server]
Encrypt marx's public key (to elon, from Keyserver)
Send marx's encrypted public key back to client
[Client]
Receive marx's crypted public key
o decrypt it (to me, rom: Keyserver)
s

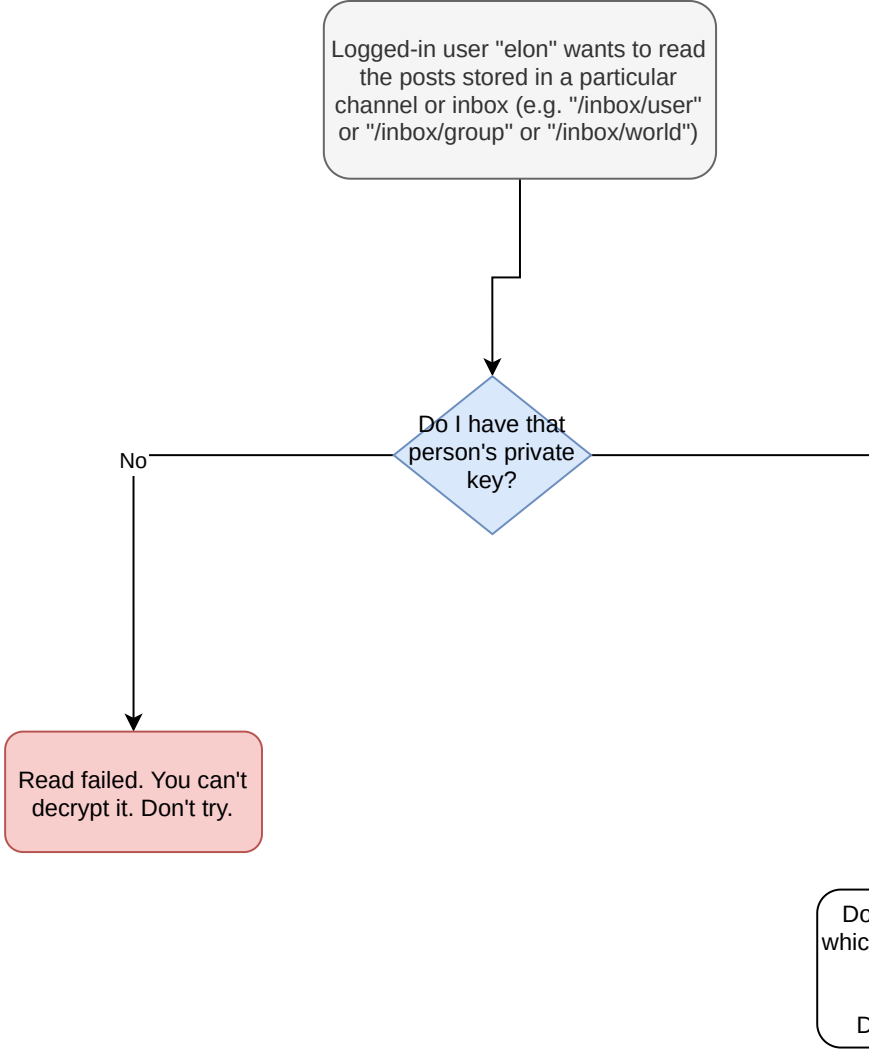

No
Yes
[KeyClient] Store decryption keys in local hardware db
Store keyserveer-encrypted(S.marx.key.pub.decr) ...
Store me-encrypted(.marx.key.priv.decr)
Store passphrase-encrypted(marx.key.admin.decr)
Meet failed. Either Keysever not authentic according to me, or I'm not authentic according to Keysever.
Meet succeeded.
have marx's pub
Registration successful
[Client] (optior
Encrypt marx's publi
my passphrase
Store marx's sym
encrypted public ke
(.marx.loc)

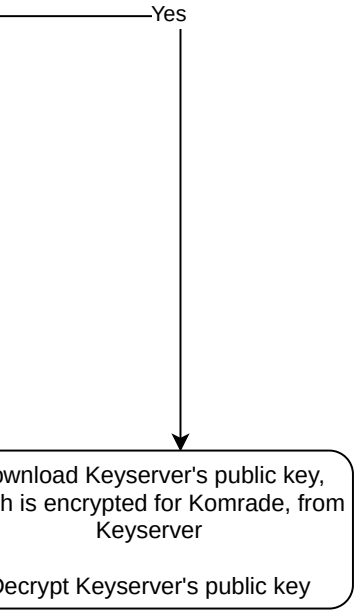
. I now
ic key.

nal)

c key with
e-key

metric-
y on disk


Logged-in user "elon" wants to read the posts stored in a particular channel or inbox (e.g. "/inbox/user" or "/inbox/group" or "/inbox/world")
Do I have that person's private key?
No
Read failed. You can't decrypt it. Don't try.
Do
whic
D


Yes
wnload Keyserver's public key,
h is encrypted for Komrade, from
Keyserver
ecrypt Keyserver's public key