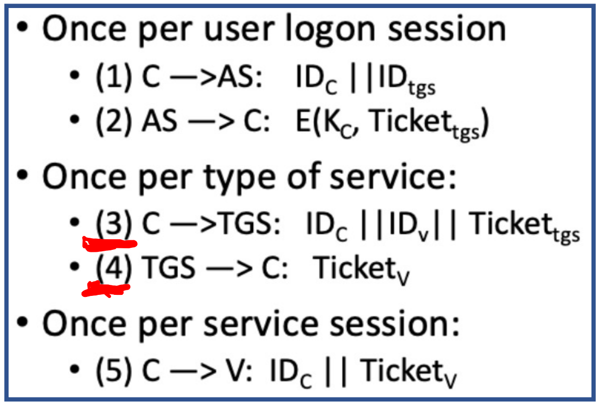
Secure?

no user authentication

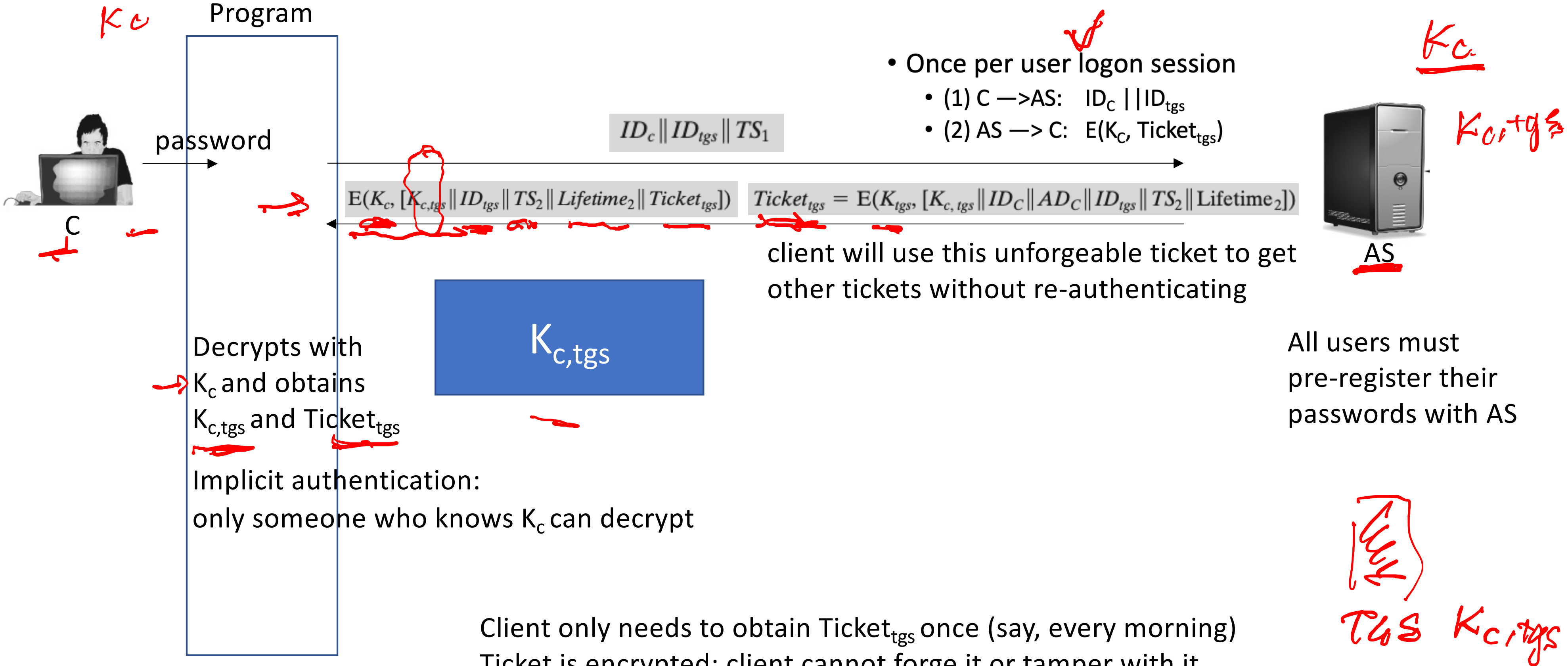
- Ticket hijacking
  - Malicious user may steal the service ticket of another user on the same workstation and try to use it
    - Network address verification does not help
  - Servers must verify that the user who is presenting the ticket is the same user to whom the ticket was issued
- No server authentication
  - Attacker may misconfigure the network so that he receives messages addressed to a legitimate user – man in the middle attack
    - Cause a denial of service
  - Servers must prove their identity to users
- **Solution: session key**



- Once per user logon session
  - (1) C —>AS: $ID_C || ID_{tgs}$
  - (2) AS —> C: $E(K_C, Ticket_{tgs})$
- Once per type of service:
  - (3) C —>TGS: $ID_C || ID_v || Ticket_{tgs}$
  - (4) TGS —> C: $Ticket_V$
- Once per service session:
  - (5) C —> V: $ID_C || Ticket_V$

Kerberos v4. - once per user logon session

- Once per user logon session
  - (1) C —>AS: ID_C | | ID_{tgs}
  - (2) AS —> C: $E(K_C, Ticket_{tgs})$



$ID_c \| ID_{tgs} \| TS_1$

$E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$ $Ticket_{tgs} = E(K_{tgs}, [K_{c,\,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| \mathrm{Lifetime}_2])$

client will use this unforgeable ticket to get other tickets without re-authenticating

Decrypts with K_c and obtains $K_{c,tgs}$ and $Ticket_{tgs}$

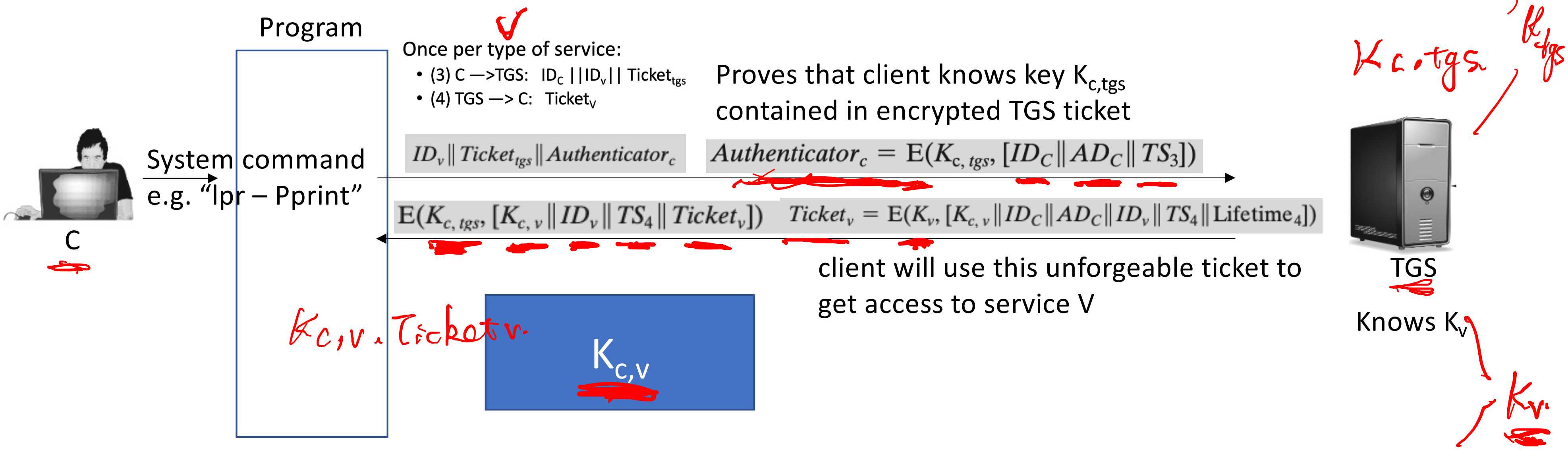
Implicit authentication:
only someone who knows K_c can decrypt

All users must pre-register their passwords with AS

TGS Kc,tgs

Client only needs to obtain $Ticket_{tgs}$ once (say, every morning)
Ticket is encrypted; client cannot forge it or tamper with it

Kerberos v4. - once per type of service



Once per type of service:

- (3) C —>TGS: ID_C | | ID_v | | $Ticket_{tgs}$
- (4) TGS —> C: $Ticket_V$

Proves that client knows key $K_{c,tgs}$ contained in encrypted TGS ticket

$ID_v \| Ticket_{tgs} \| Authenticator_c$

$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$

$E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| \text{Lifetime}_4])$

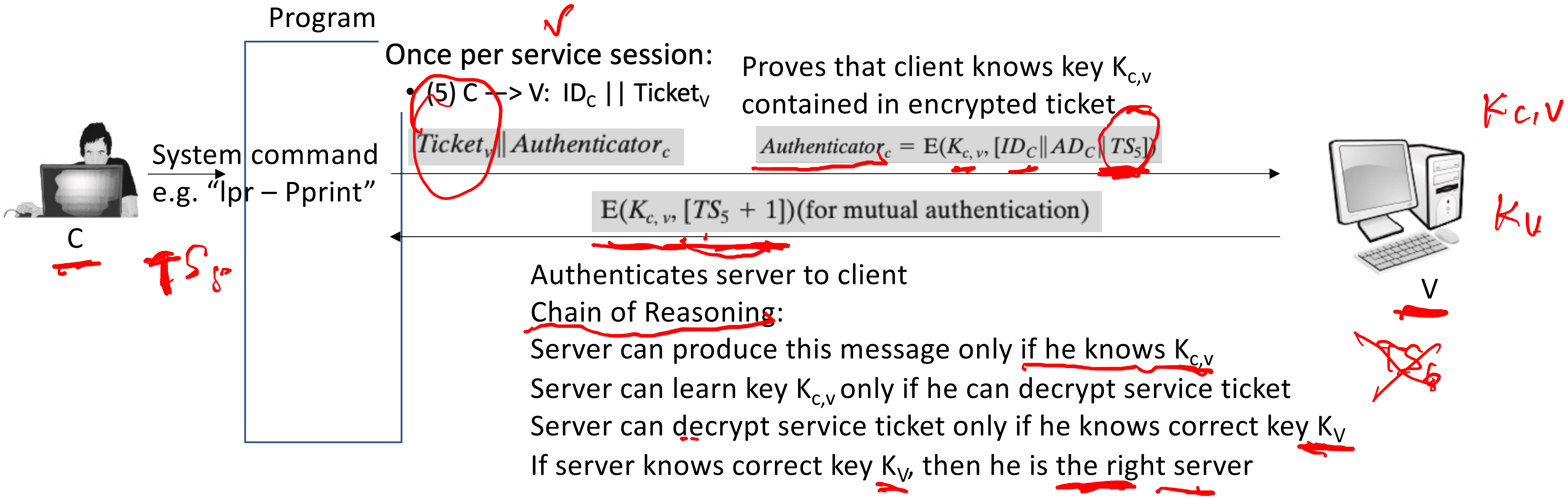
client will use this unforgeable ticket to get access to service V

Client uses $Ticket_{tgs}$ to obtain a service ticket, $Ticket_v$ and a short-term session key for each network service (printer, email, etc.)

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| \text{Lifetime}_2])$

K_{tgs}

Kerberos v4. - once per service session



Once per service session: ✓

- (5) C --> V: ID_C || $Ticket_V$

$Ticket_v \| Authenticator_c$

Proves that client knows key $K_{c,v}$ contained in encrypted ticket

$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$

System command e.g. “lpr – Pprint”

$E(K_{c,v}, [TS_5 + 1])$(for mutual authentication)

Authenticates server to client

Chain of Reasoning:

Server can produce this message only if he knows $K_{c,v}$

Server can learn key $K_{c,v}$ only if he can decrypt service ticket

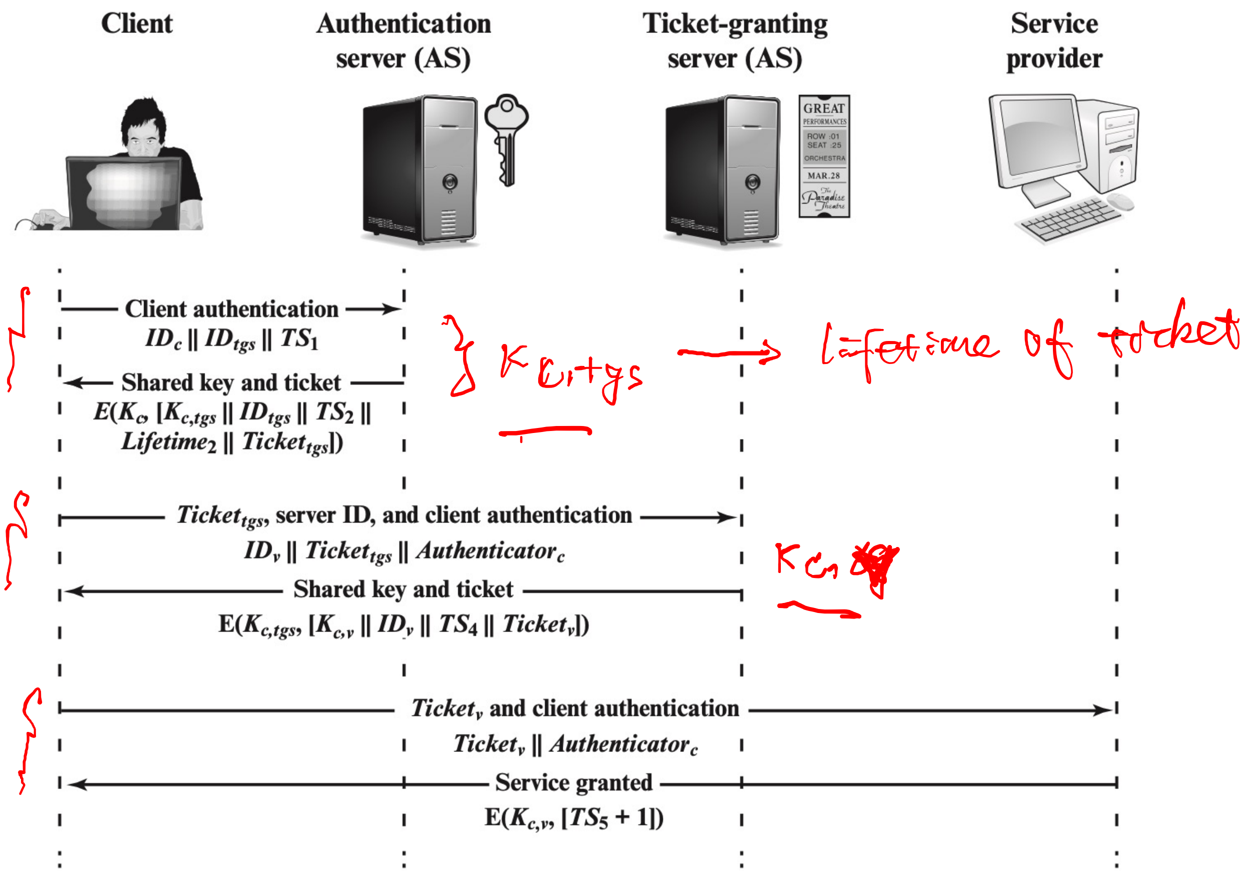
Server can decrypt service ticket only if he knows correct key K_V

If server knows correct key K_V, then he is the right server

For each service request, client uses the short-term key, $K_{c,v}$, for that service and the ticket he received from TGS

$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$

Overview of Kerberos



Client

Authentication server (AS)

Ticket-granting server (AS)

Service provider

Client authentication

$ID_c \| ID_{tgs} \| TS_1$

Shared key and ticket

$E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$K_{c,tgs}$ → lifetime of ticket

$Ticket_{tgs}$, server ID, and client authentication

$ID_v \| Ticket_{tgs} \| Authenticator_c$

Shared key and ticket

$E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$K_{c,v}$

$Ticket_v$ and client authentication

$Ticket_v \| Authenticator_c$

Service granted

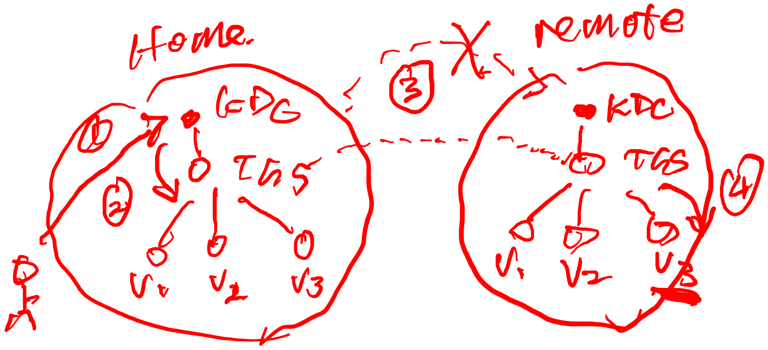
$E(K_{c,v}, [TS_5 + 1])$

Important Ideas in Kerberos

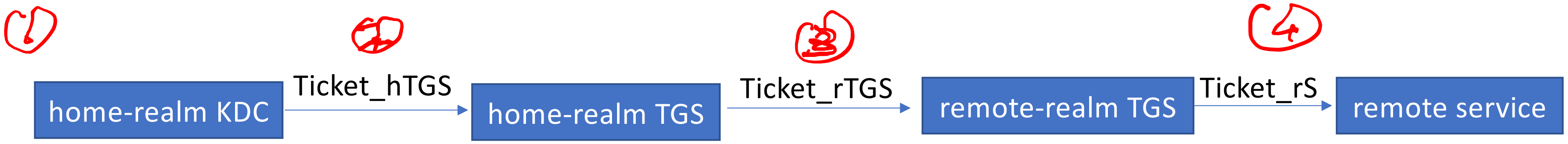
- Short-term session keys
  - Long-term secrets used only to derive short-term keys
  - Separate session key for each user-server pair
    - Re-used by multiple sessions between same user and server
- Proofs of identity based on authenticators
  - Client encrypts his identity, addr, time with session key; knowledge of key proves client has authenticated to KDC/AS
    - Also prevents replays (if clocks are globally synchronized)
  - Server learns this key separately (via encrypted ticket that client can't decrypt), then verifies client's authenticator
- Symmetric cryptography only

Kerberos in Large Networks

→ scalability



- One KDC isn’t enough for large networks
- Network is divided into realms
  - KDCs in different realms have different key databases
- To access a service in another realm, users must...
  - Get ticket for home-realm TGS from home-realm KDC
  - Get ticket for remote-realm TGS from home-realm TGS
    - As if remote-realm TGS were just another network service
  - Get ticket for remote service from that realm’s TGS
  - Use remote-realm ticket to access service



home-realm KDC → Ticket_hTGS → home-realm TGS → Ticket_rTGS → remote-realm TGS → Ticket_rS → remote service

Practical Uses of Kerberos

- Microsoft Windows – Active Directory
- Email, FTP, network file systems, many other applications have been kerberized
  - Use of Kerberos is transparent for the end user
  - Transparency is important for usability!
- Authentication for network protocols
  - rsh
- Local authentication
  - login and su in OpenBSD
- Secure windowing systems

Readings

- Kerberos: The Network Authentication Protocol
  https://web.mit.edu/kerberos/

Practice – no submission

- William Stallings, “Network Security Essentials”, 6 Edition, 2017
  - Chapter 4’s problems: 4.8, 4.9, 4.10