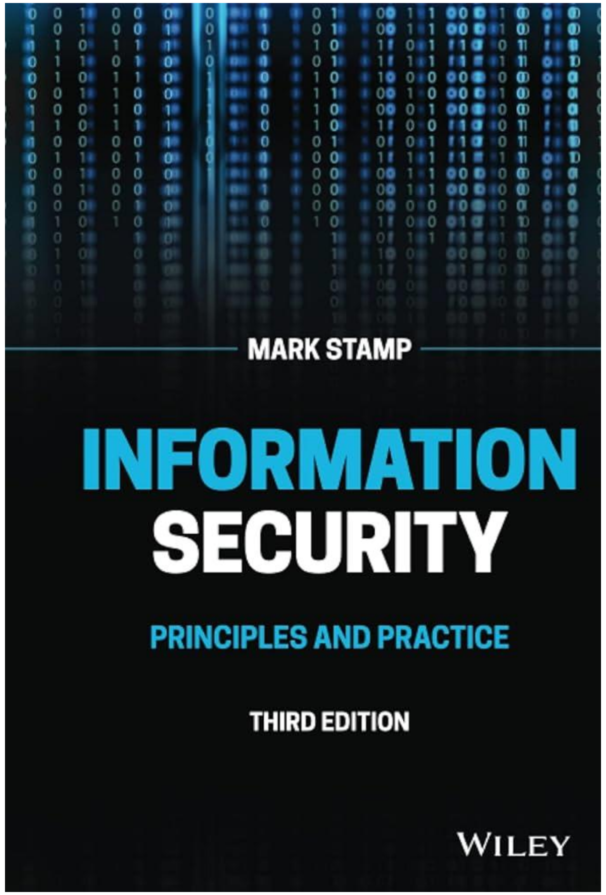
بسم الله الرحمن الرحیم

مبانی رایانش امن

جلسه ۱۶

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

- فصل ۴ استالینگ (شبکه)
- فصل ۱۰ استمپ



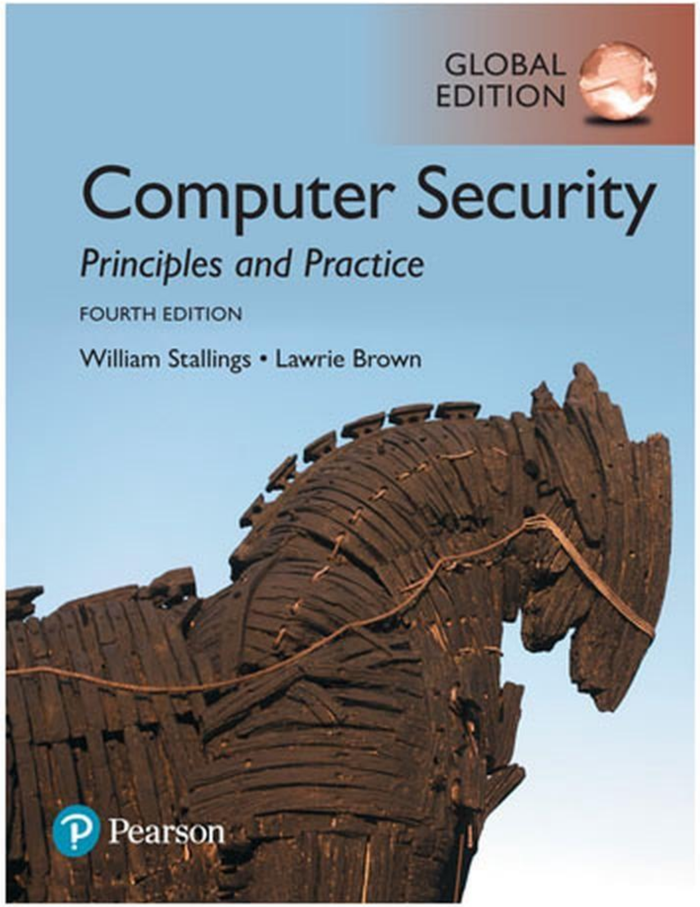
MARK STAMP

INFORMATION SECURITY

PRINCIPLES AND PRACTICE

THIRD EDITION

WILEY



GLOBAL EDITION

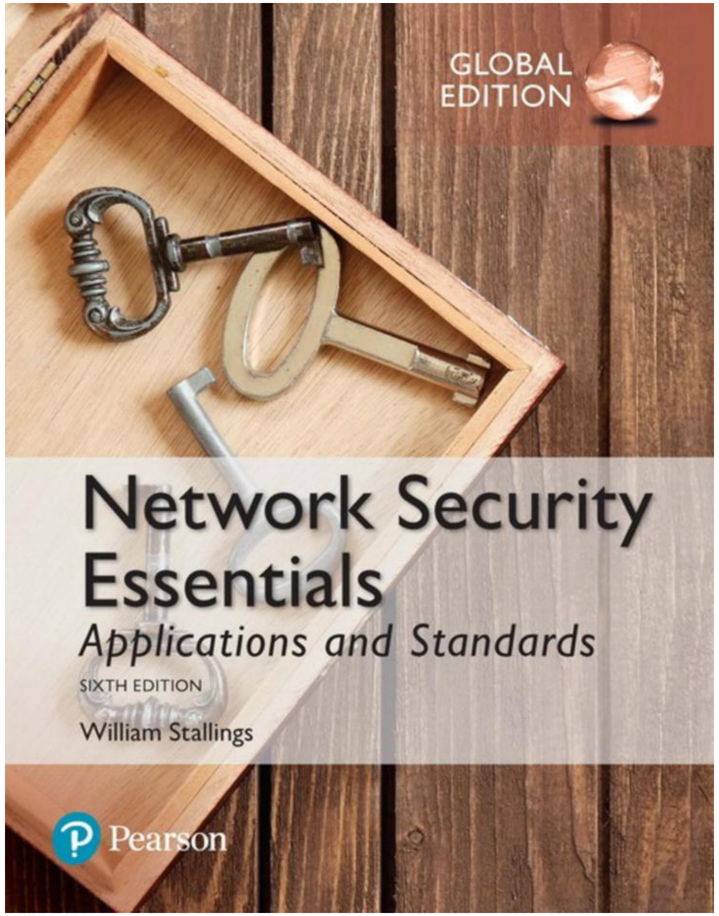
Computer Security

Principles and Practice

FOURTH EDITION

William Stallings • Lawrie Brown

Pearson



GLOBAL EDITION

Network Security Essentials

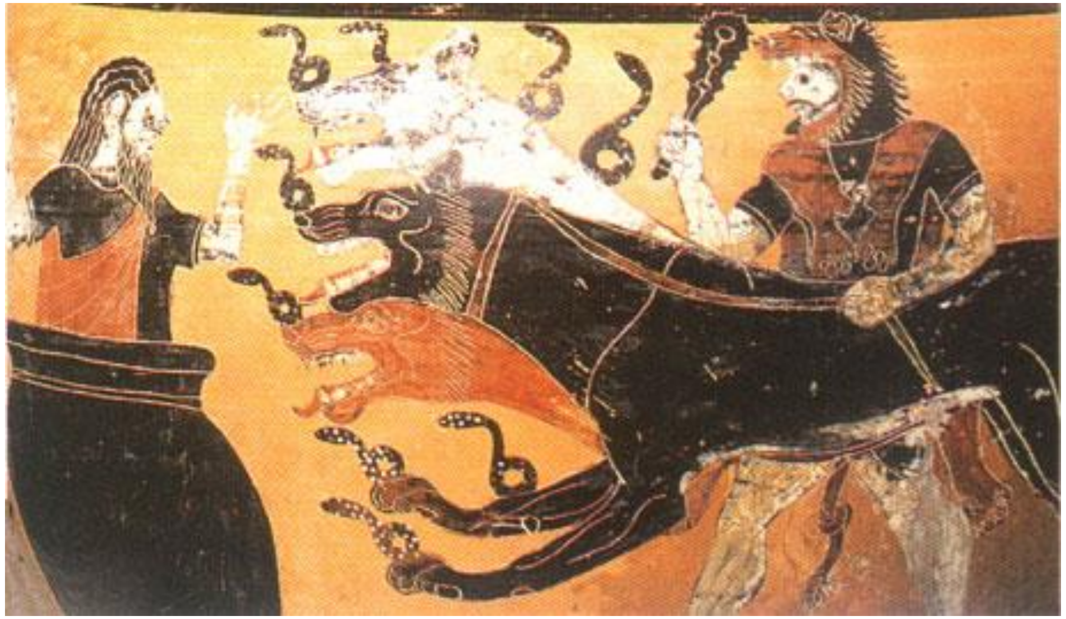
Applications and Standards

SIXTH EDITION

William Stallings

Pearson

Kerberos



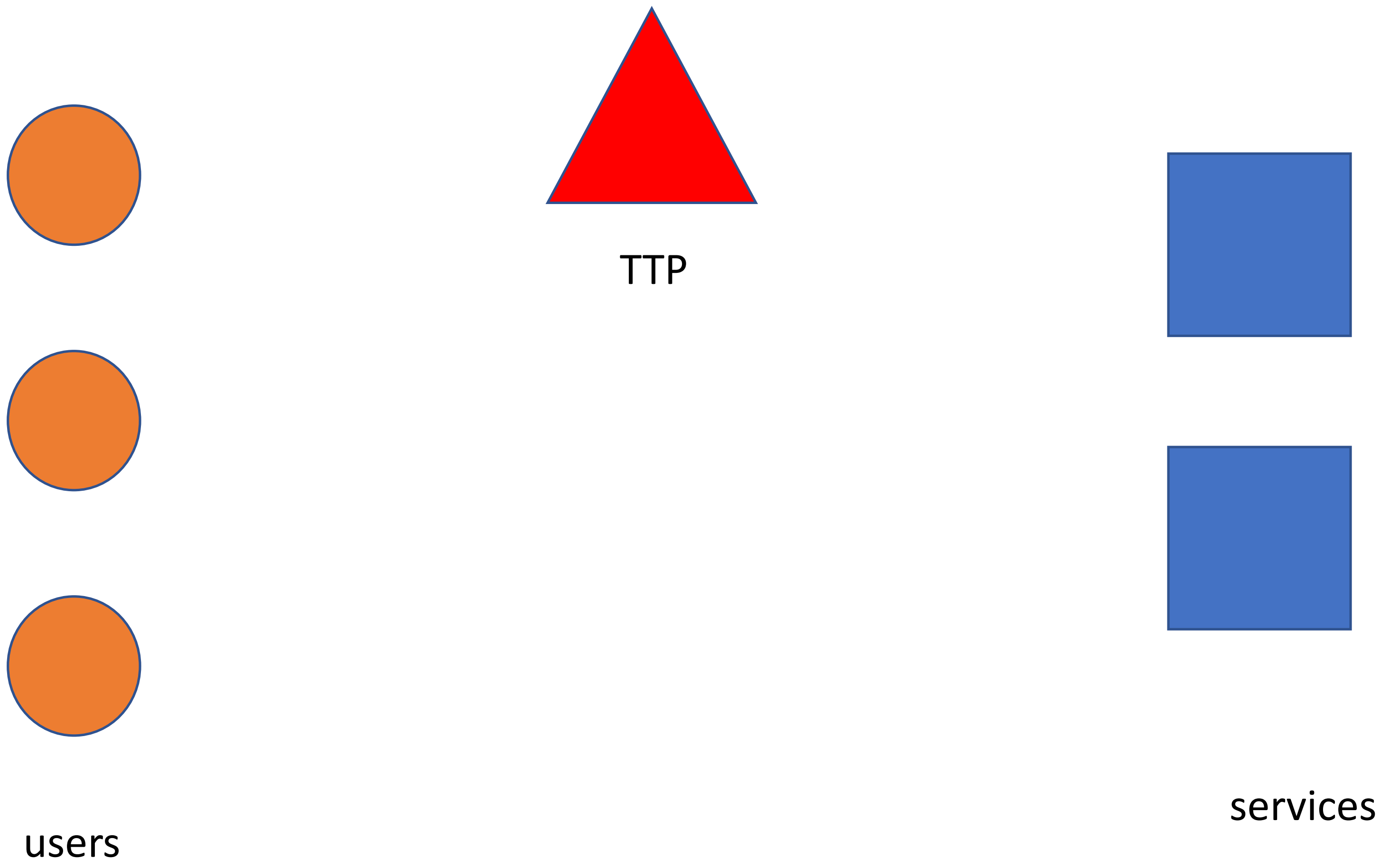
Many to many authentication

users
services

Kerberos

- نیازمندی های ما در این سیستم:
  - ❑ کسی با شنود یا فعالانه نتواند جعل هویت کند.
  - ❑ از دید کاربران، کل سیستم شبیه یک سیستم مبتنی بر پسورد ساده باشد.
  - ❑ تعداد زیادی کاربر را پشتیبانی کند.

TTP saves password

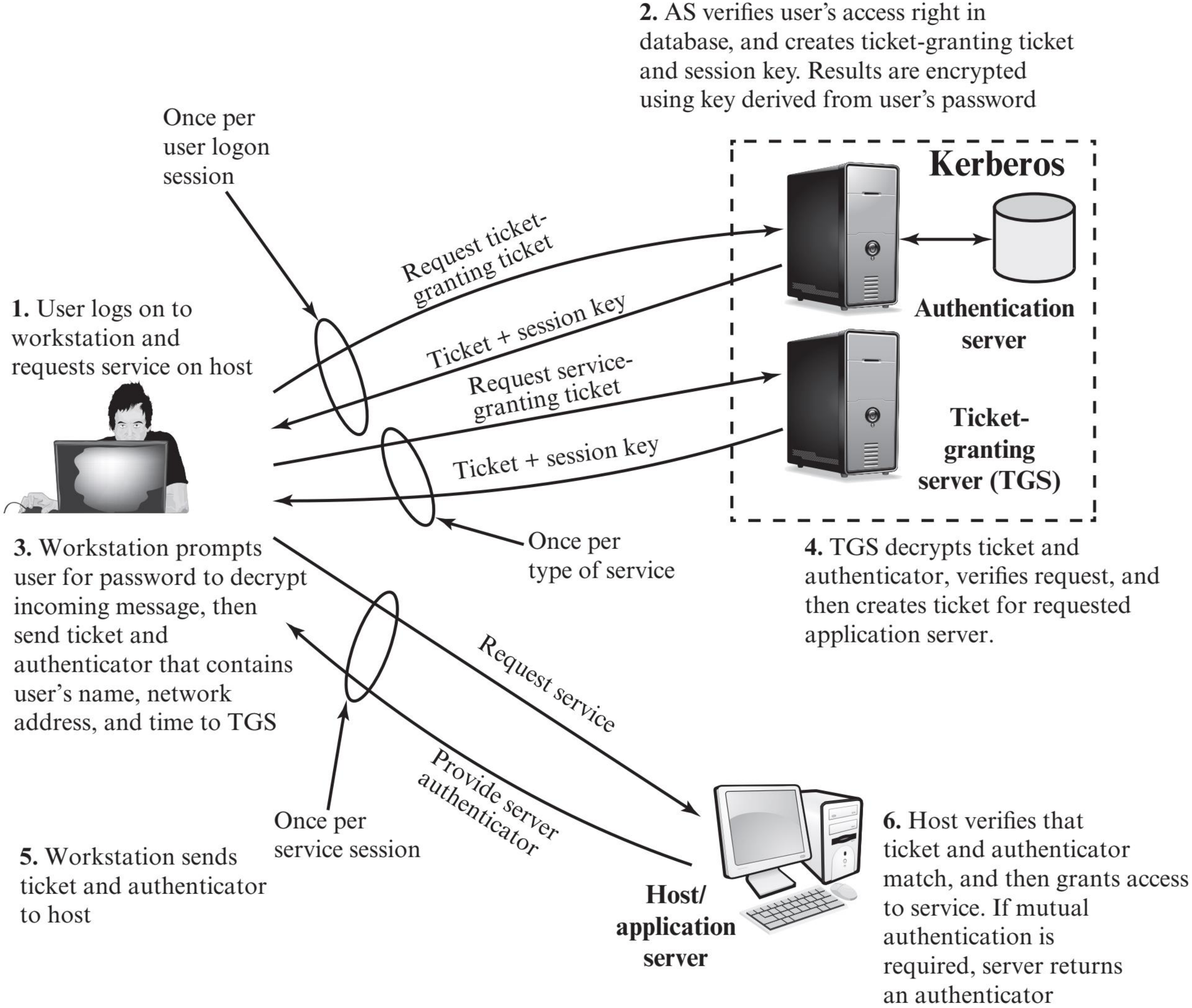


Kerberos

- In Greek mythology, Kerberos is 3-headed dog that guards entrance to Hades
  - "Wouldn't it make more sense to guard the exit?"
- In security, Kerberos is an authentication protocol based on symmetric key crypto
  - Originated at MIT
  - Based on Needham-Schroeder protocol
  - Relies on a **Trusted Third Party (TTP)**

Motivation for Kerberos

- Authentication using public keys
  - N users ⇒ N key pairs
- Authentication using symmetric keys
  - N users requires (on the order of) N^2 keys
- Symmetric key case **does not scale**
- Kerberos based on symmetric keys but only requires N keys for N users
  - - Security depends on TTP
  - + No PKI is needed


1. User logs on to workstation and requests service on host

2. AS verifies user's access right in database, and creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password

3. Workstation prompts user for password to decrypt incoming message, then send ticket and authenticator that contains user's name, network address, and time to TGS

4. TGS decrypts ticket and authenticator, verifies request, and then creates ticket for requested application server.

5. Workstation sends ticket and authenticator to host

6. Host verifies that ticket and authenticator match, and then grants access to service. If mutual authentication is required, server returns an authenticator

Kerberos KDC

- Kerberos **Key Distribution Center** or **KDC**
  - KDC acts as the TTP
  - TTP is trusted, so it must not be compromised
- KDC shares symmetric key K_A with Alice, key K_B with Bob, key K_C with Carol, etc.
- And a master key K_{KDC} known *only* to KDC
- KDC enables authentication, session keys
  - Session key for confidentiality and integrity
- In practice, crypto algorithm is DES

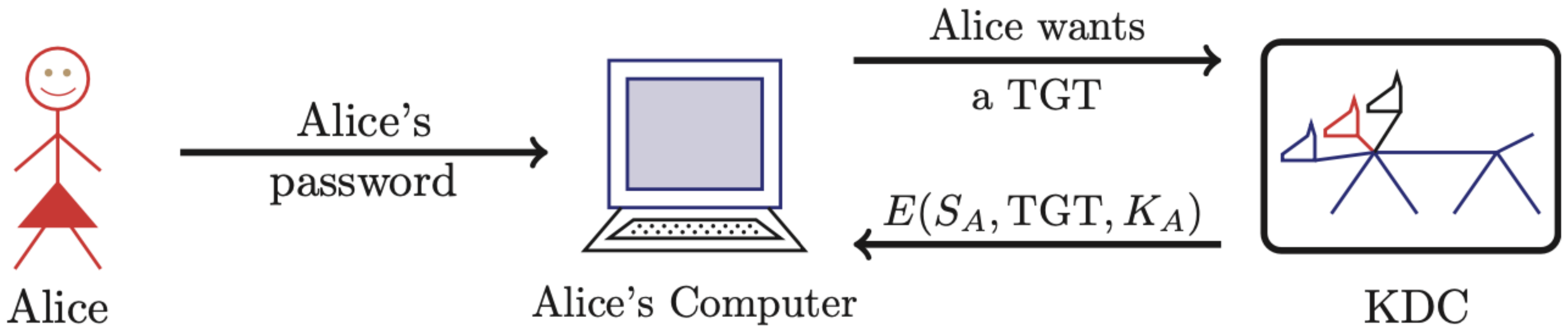
Kerberos Tickets

- KDC issues **tickets** containing info needed to access network resources
- KDC also issues **Ticket-Granting Tickets** or **TGTs** that are used to obtain tickets
- Each TGT contains
  - Session key
  - User's ID
  - Expiration time
- Every TGT is encrypted with K_{KDC}
  - So, TGT can only be read by the KDC

Kerberized Login

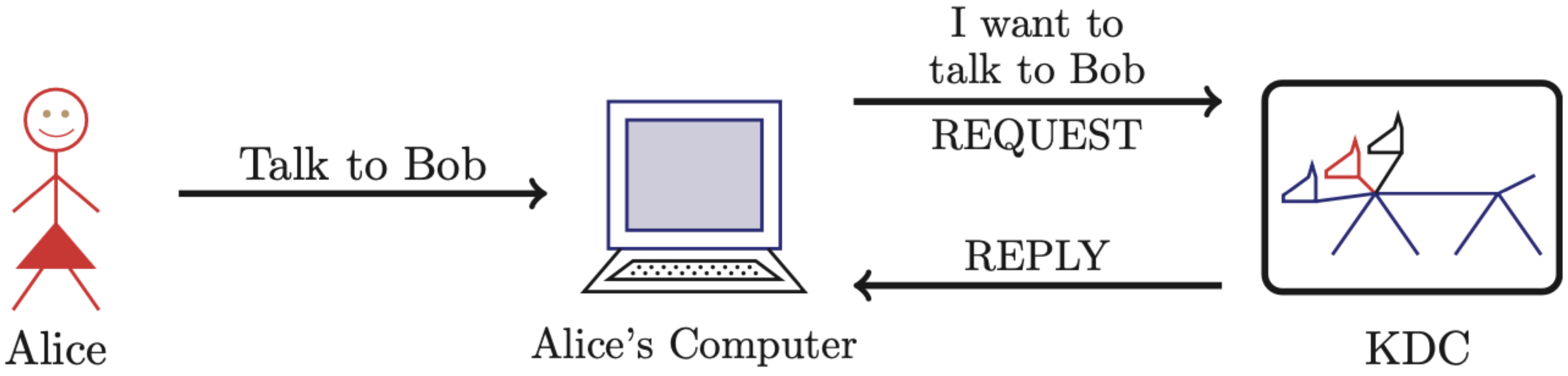
- Alice enters her password
- Then Alice's computer does following:
  - Derives K_A from Alice's password
  - Uses K_A to get TGT for Alice from KDC
- Alice then uses her TGT (credentials) to securely access network resources
- **Plus:** Security is transparent to Alice
- **Minus:** KDC ***must*** be secure — it's trusted!

Kerberized Login



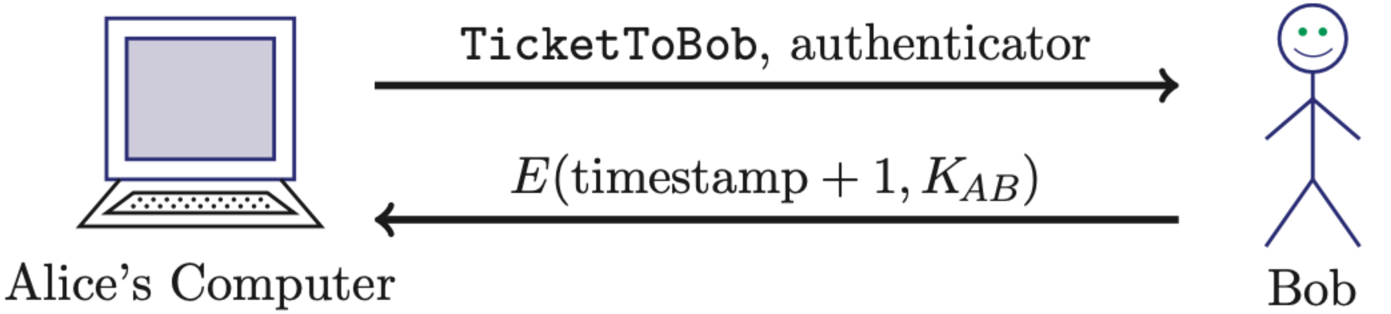
- Key K_A = h(Alice's password)
- KDC creates session key S_A
- Alice's computer decrypts S_A and TGT
  - Then it forgets K_A
- TGT = E("Alice", S_A, K_{KDC})

Alice Requests "Ticket to Bob"



- REQUEST = (TGT, authenticator)
  - authenticator = E(timestamp, S_A)
- REPLY = E("Bob", K_{AB}, ticket to Bob, S_A)
  - ticket to Bob = E("Alice", K_{AB}, K_B)
- KDC gets S_A from TGT to verify timestamp

Alice Uses Ticket to Bob



- ❑ ticket to Bob = E("Alice", K_{AB}, K_B)
- ❑ authenticator = E(timestamp, K_{AB})
- ❑ Bob decrypts "ticket to Bob" to get K_{AB} which he then uses to verify timestamp

Kerberos

- Key S_A used in authentication
  - For confidentiality/integrity
- Timestamps for authentication and replay protection
- Recall, that with timestamps…
  - Reduce the number of messages — like a nonce that is known in advance
  - But, “time” is a security-critical parameter

Questions about Kerberos

- When Alice logs in, KDC sends $E(S_A, TGT, K_A)$ where $TGT = E(\text{"Alice"}, S_A, K_{KDC})$

  Q: Why is TGT encrypted with K_A?

  A: Enables Alice to remain anonymous when she (later) uses her TGT to request a ticket

- In Alice's "Kerberized" login to Bob, why can Alice remain anonymous?
- Why is "ticket to Bob" sent to Alice?
  - Why doesn't KDC send it directly to Bob?

Kerberos Alternatives

- Could have Alice's computer remember password and use that for authentication
  - Then no KDC required
  - But hard to protect passwords
  - Also, does not scale
- Could have KDC remember session key instead of putting it in a TGT
  - Then no need for TGT
  - But **stateless** KDC is major feature of Kerberos