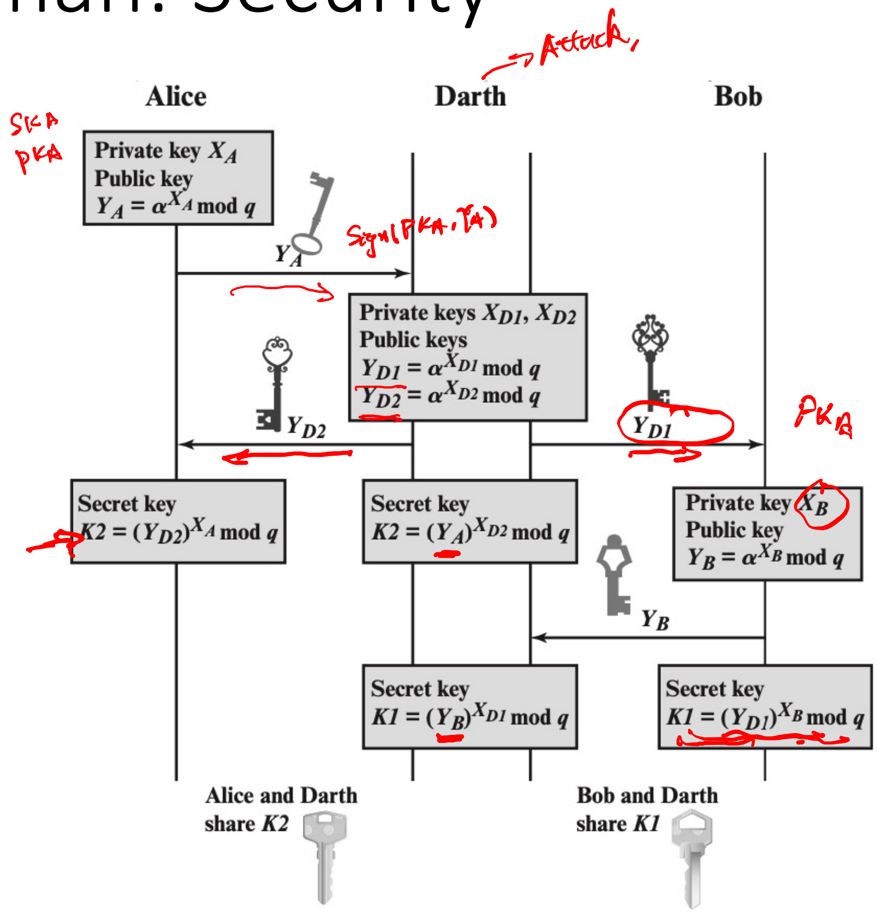
Diffie-Hellman is susceptible to man-in-the-middle attacks

- David can alter messages, block messages, and send her own messages
- **DH is not** secure against a MITM attacker: David can just do a DH with both sides!

Diffie-Hellman: Security

TLS.



Reason:
Lack of authentication

Defense:
① Symmetric Encryption
② MAC code.
√③ Digital Signature

Diffie-Hellman: issues

- Diffie-Hellman is not secure against a MITM adversary
- DHE is an *active protocol*: Alice and Bob need to be online at the same time to exchange keys
    - What if Bob wants to encrypt something and send it to Alice for her to read later?
- Diffie-Hellman does not provide *authentication*
    - You exchanged keys with someone, but Diffie-Hellman makes no guarantees about who you exchanged keys with; it could be David!

$P(K_1 \mid K_3) = 0$

Ephemerality of Diffie-Hellman

PK | | PK

DHKE — K_1 ✓

DHKE — K_2 ✓

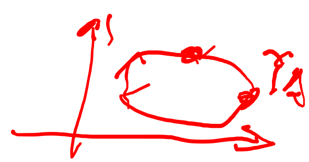
DHKE — K_3

forward secrecy

- Diffie-Hellman can be used ephemerally (called Diffie-Hellman ephemeral, or DHE)
  - **Ephemeral**: Short-term and temporary, not permanent
  - Alice and Bob discard X_A, X_B and $K = \alpha^{X_A X_B} \ mod\ q$ when they're done
  - Because you need X_A and X_B to derive K, you can never derive K again!
  - Sometimes K is called a **session key**, because it's only used for an ephemeral session
- Eve can't decrypt any messages she recorded: Nobody saved X_A, X_B or K, and her recording only has $\alpha^{X_A} \ mod\ q$ and $\alpha^{X_B} \ mod\ q$!

Diffie-Hellman Key Exchange: Summary

- Algorithm:
  - Alice chooses X_A and sends $\alpha^{X_A}\ mod\ q$ to Bob
  - Bob chooses X_B and sends $\alpha^{X_B}\ mod\ q$ to Alice
  - Their shared secret is $(\alpha^{X_A})^{X_B} = (\alpha^{X_B})^{X_A} = \alpha^{X_A X_B}\ mod\ q$
- Diffie-Hellman provides forwards secrecy: Nothing is saved or can be recorded that can ever recover the key
- Diffie-Hellman can be performed over other mathematical groups, such as elliptic-curve Diffie-Hellman (ECDH)
- Issues
  - **Not** secure against MITM
  - Both parties must be online
  - Does not provide authenticity



A

x_A

$Y_A = x_A \cdot G$

$K = x_A \cdot Y_B$

B

x_B

$Y_B = x_B \cdot G$

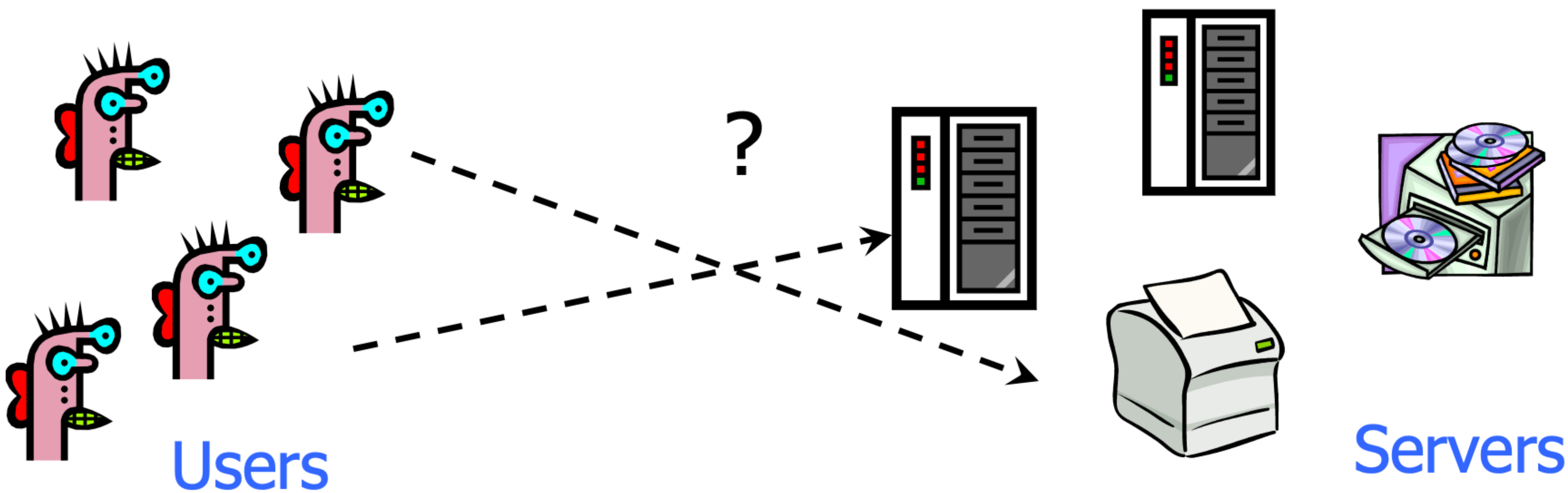
$K = x_B \cdot Y_A$

$= x_A \cdot x_B \cdot G$

Kerberos

4.3

Many-to Many Authentication



How do users prove their identities when requesting services from machines on the network?

Threats

- User impersonation
  - Malicious user with access to a workstation pretends to be another user from the same workstation
- Network address impersonation
  - Malicious user changes network address of his workstation to impersonate another workstation
- Eavesdropping, tampering, replay
  - Malicious user eavesdrops, tampers, or replays other users’ conversations to gain unauthorized access

Requirements

- Security
  - against attacks by eavesdroppers and malicious users
- Transparency
  - users shouldn’t notice authentication taking place
  - entering password is ok, if done rarely
- Scalability
  - Large number of users and servers

Kerberos

- scenario: users at workstations wish to access services on servers distributed throughout the network – many to many authentication

Kerberos

- a centralized authentication server provides mutual authentication between users and servers
  - a key distribution and user authentication service developed at MIT
  - works in an open distributed environment
- client-service model
- Kerberos protocol messages are protected against eavesdropping and replay attacks
- Kerberos v4 and v5 [RFC 4120]