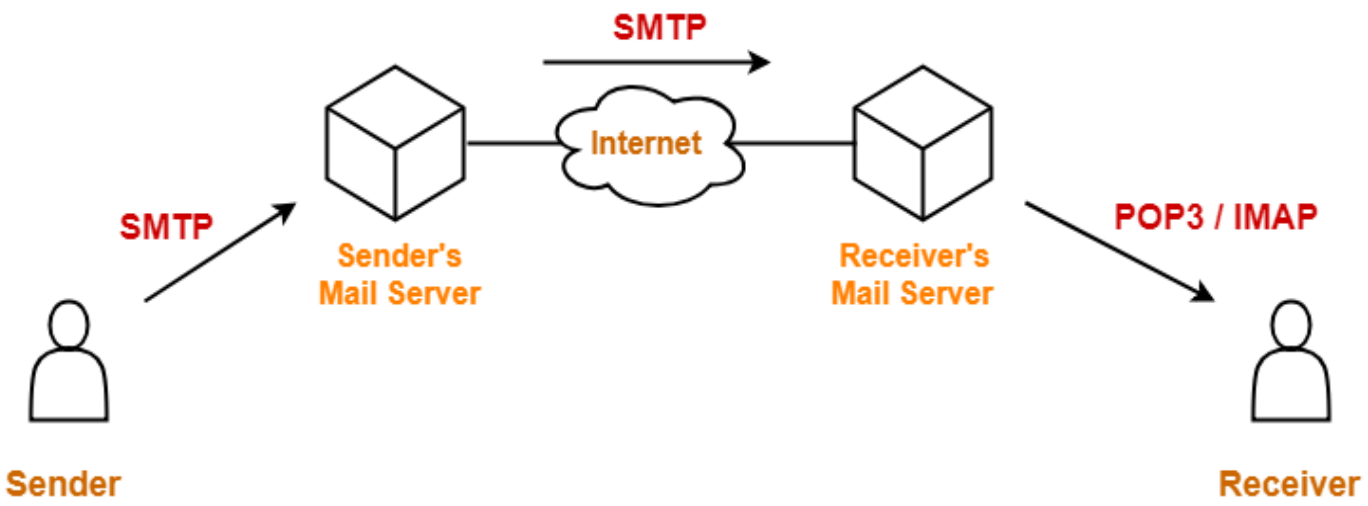
Tutorial 3

1. Email Protocols

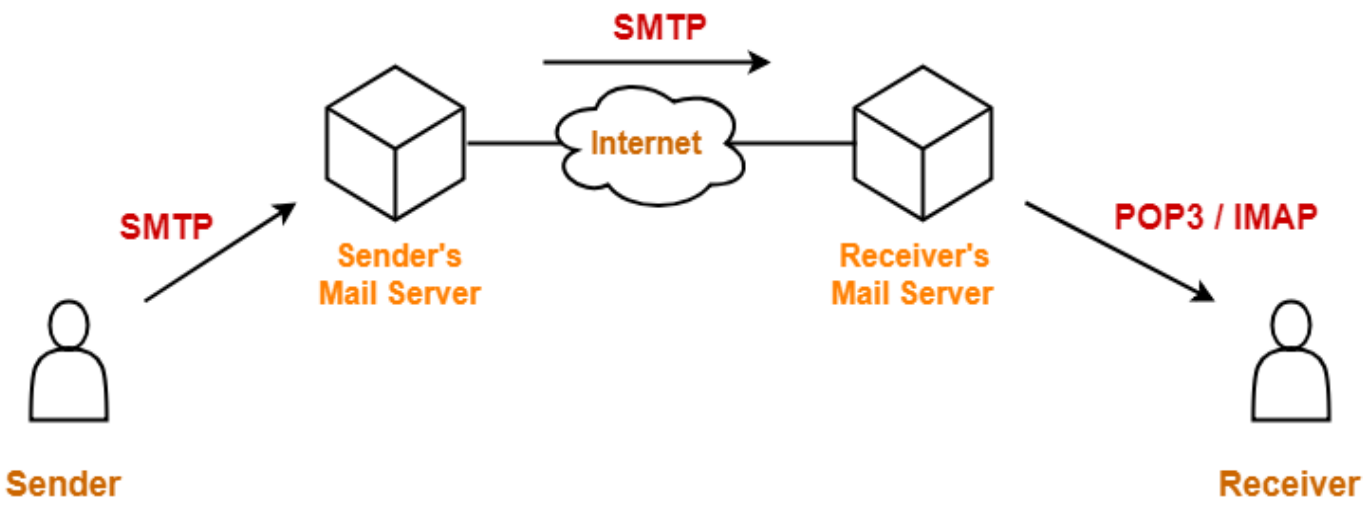
Email Protocols



1.1 How many email address(es) are involved in the SMTP protocol?

- Two email addresses. One is the sender and one is the receiver.

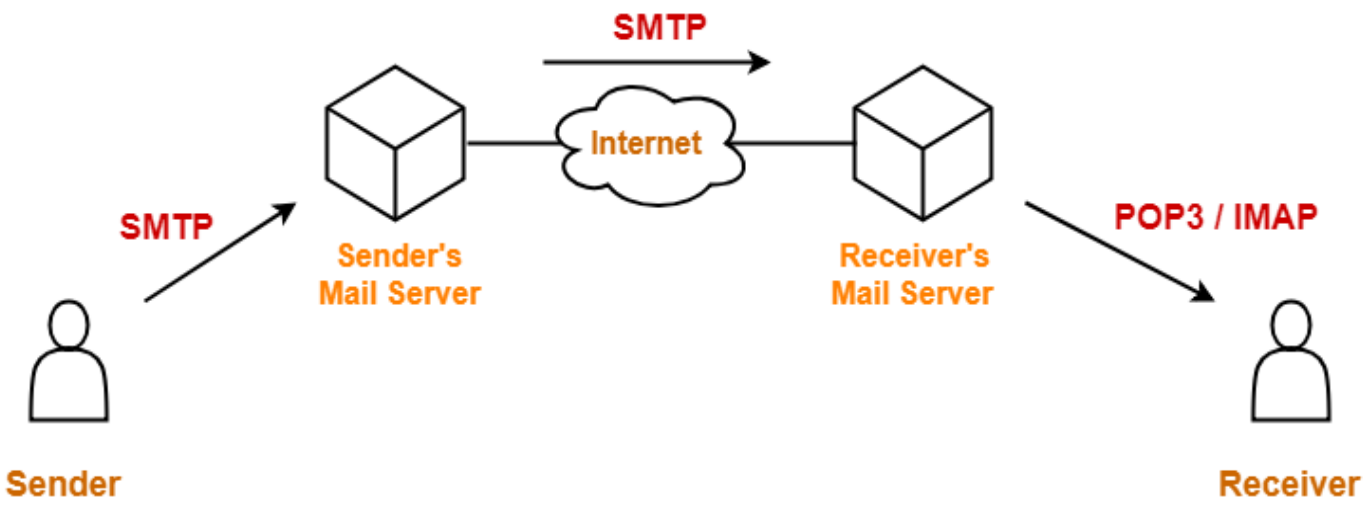
Email Protocols



1.2 Does SMTP require any password?

- No. Sending emails to a receiver doesn't need to know the receiver's password. No authentication on sender either.

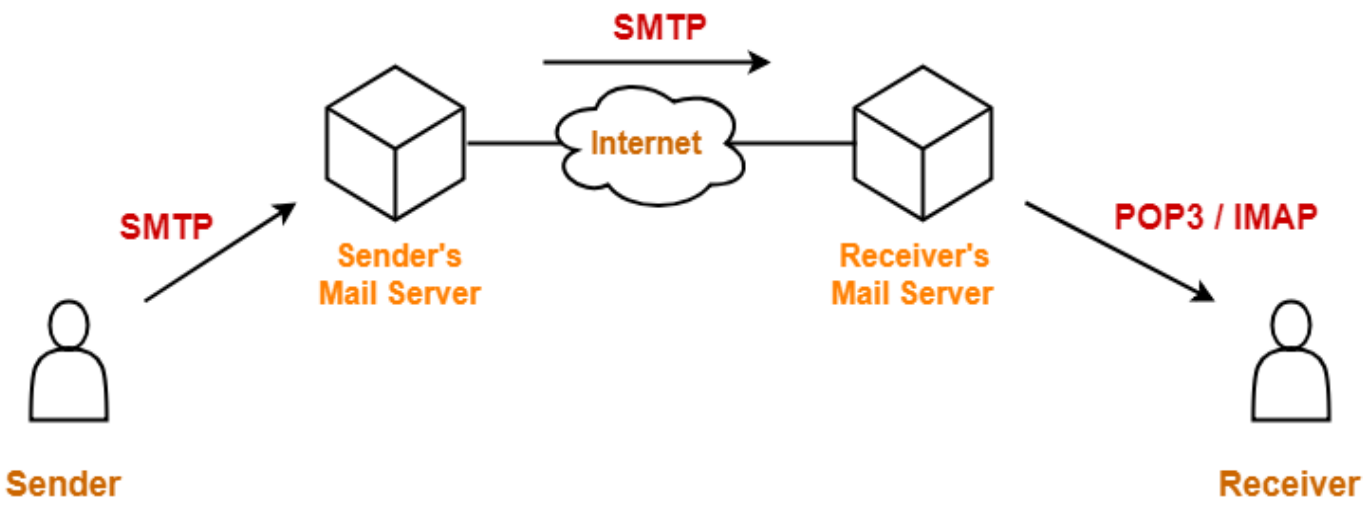
Email Protocols



1.3 Does POP3/IMAP require any password?

- YES. The receiver must show the password of the corresponding email account (receiver's email address)

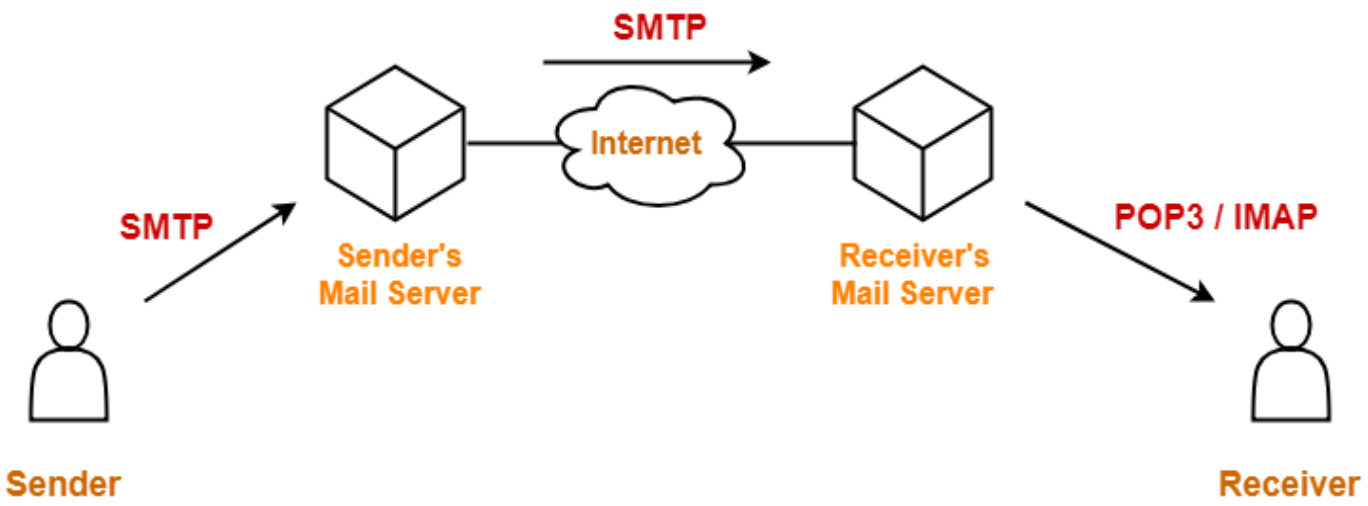
Email Protocols



1.4 Who can read the email sent from the sender to the receiver?

❖ The sender's mail server and also the receiver's mail server.

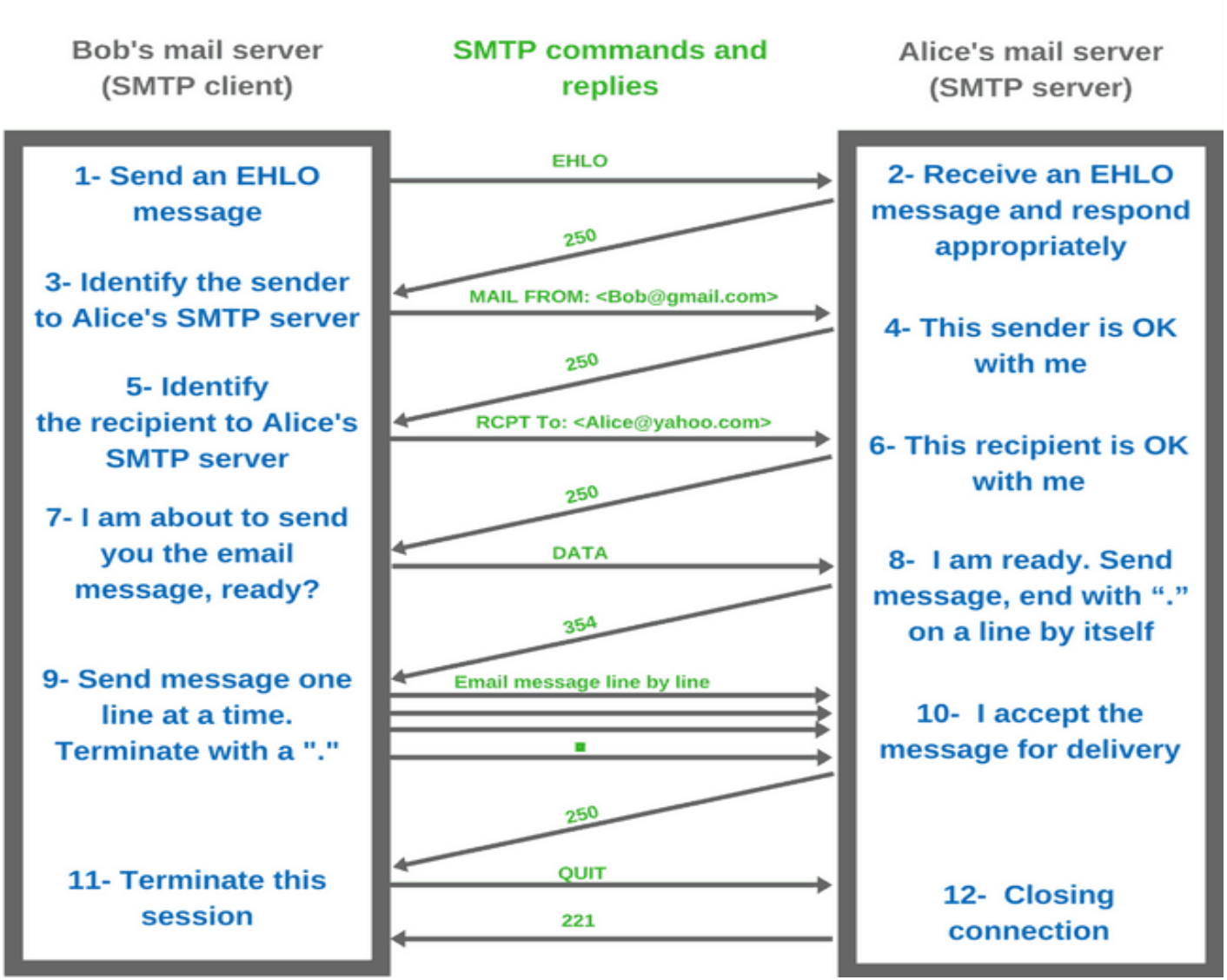
Email Protocols



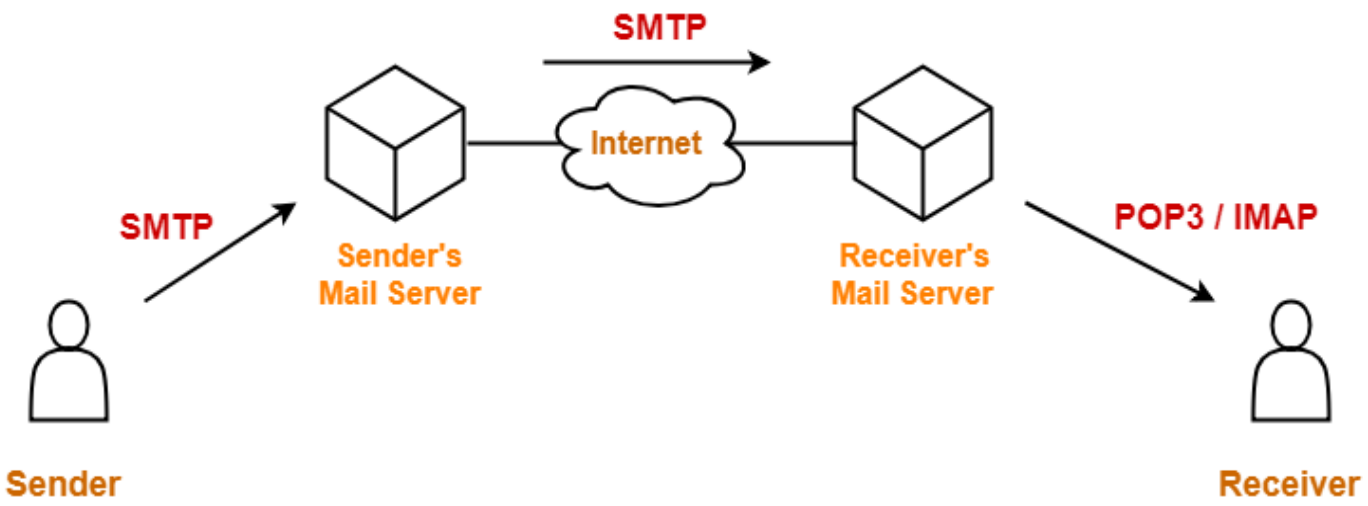
1.5 How to send an email ANONYMOUSLY to a receiver?

- When running the SMTP protocol between Sender and Receiver-Server, put the sender email address blank.

Email Protocols (SMTP)



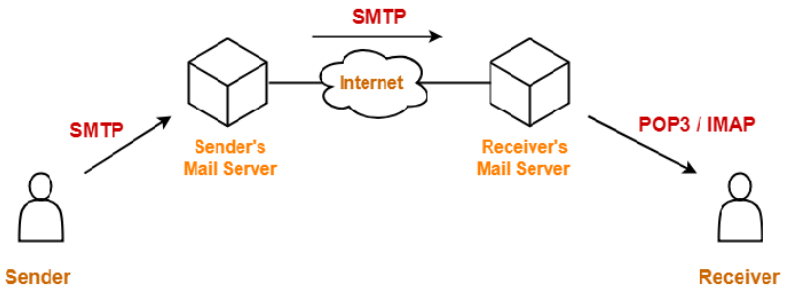
Email Protocols



1.6 Sender (Alice) and Receiver (Bob) share a secret key K. How to send an email to Bob such that

- Bob knows that the email is from Alice
- Bob's email server doesn't know who sent that email.

Email Protocols



1.6 Sender (Alice) and Receiver (Bob) share a secret key K. How to send an email to Bob such that

- Bob knows that the email is from Alice
- The Bob's email server doesn't know who sent that email.

- Suppose that M is the email content to be sent.
- Alice computes CT=E_K(H(M), Alice's email address) and attaches it after M.
- Send both M and CT to Bob and leave the sender's email address blank.
- Bob can decrypt and verify that M was indeed sent from Alice.

Note: Bob need to try all his keys if he shared keys with many others.

2. Email Security

Sign Then Encrypt

- A has a pair of keys (d, e), where d is private and e is public
- B has a pair of keys (d', e'), where d' is private and e' is public

  $A \rightarrow B: E_{e'}(A, M, Sign_d(M))$

- B believes that A sent the message, if the message and signature can be verified with e.
- A believes that only B can receive the signed M
- It provides authentication, non-repudiation, confidentiality and sender anonymity

Encrypt-then-Sign

- A has a pair of keys (d, e), where d is private and e is public
- B has a pair of keys (d', e'), where d' is private and e' is public

  $A \rightarrow B: E_{e'}(A, M), Sign_{d}(E_{e'}(A, M))$

- Is there any problem with this approach?

Encrypt-then-Sign

- A has a pair of keys (d, e), where d is private and e is public
- B has a pair of keys (d', e'), where d' is private and e' is public

  $A \rightarrow B: E_{e'}(A, M), Sign_d(E_{e'}(A, M))$

- Is there any problem with this approach?
  - B cannot directly prove to a third party A has signed M if $E_{e'}()$ is a randomized PKE such as ElGamal encryption, unless B reveals his private key d'
  - There is no anonymity for A: the signature is not encrypted and can be verified against each possible public key. Sender's identity is revealed if the signature is valid w.r.t. a particular public key.

PGP Public Key Management

PGP uses the web of trust to manage public keys.

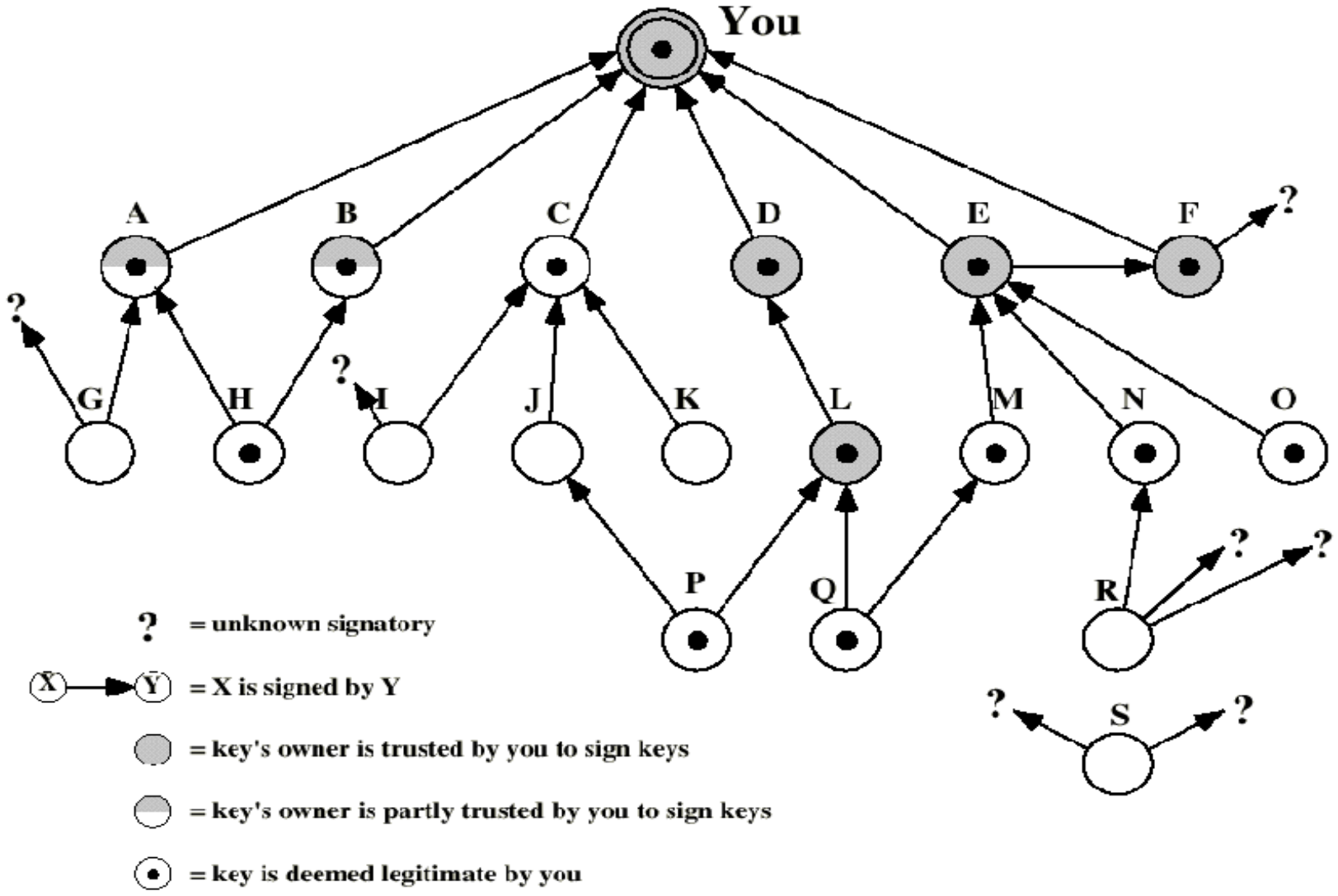
- **Owner Trust**: Do YOU trust all the public keys certified by this user?
- **Key Legitimacy**: Do YOU believe that this is the public key of this user?
- **Signatures**: All "certificates" for this public key issued by PGP users, collected by YOU.
- **Signature Trust(s)**: Do YOU trust all these "certificates"?

PGP Public Key Management

According to the info given by the figure below, complete the blank cells in the form. Here we assume that a public key is also trusted if it has been certified by at least two partially trusted users.

- **U**: Untrusted or Undefined
- **P**: Partially trusted
- **T**: Trusted
- **Sign(PRa, PUb||IDb)**: User A signs (or certifies) B's public key.

PGP Public Key Management



PGP Public Key Management

Public Key Ring of YOU

User ID	Public Key	Owner Trust	Key Legitimacy	Signatures	Signature Trusts	…
A	PKa					…
C	PKc					…
D	PKd					…
E	PKe					…
J	PKj					…
L	PKl					…
N	PKn					…
P	PKp					…
…	…	…	…	…	…	…

Public Key Ring of YOU

User ID	Public Key	Owner Trust	Key Legitimacy	Signatures	Signature Trusts	…
A	PKa	P	T	Sign(PRyou, PUa||IDa)	T	…
C	PKc	U	T	Sign(PRyou, PUc||IDc)	T	…
D	PKd	T	T	Sign(PRyou, PUd||IDd)	T	…
E	PKe	T	T	Sign(PRyou, PUe||IDe)	T	…
J	PKj	U	U	Sign(PRc, PUj||IDj)	U	…
L	PKl	T	T	Sign(PRd, PUl||IDl)	T	…
N	PKn	U	T	Sign(PRe, PUn||IDn)	T	…
P	PKp	U	T	Sign(PRj, PUp||IDp) Sign(PRl, PUp||IDp)	U T	…
…	…	…	…	…	…	…