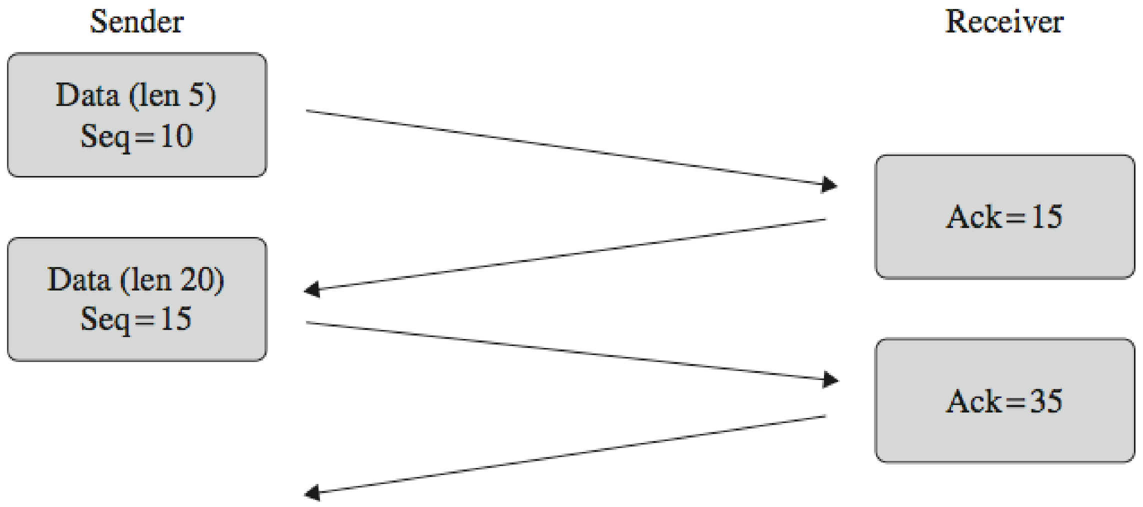
Session Hijacking

- After handshake, TCP uses Seq and Ack to ensure session correctness
  - Seq + Data_len of Sender == Ack of Receiver
  - If Seq and Ack don’t match, resynchronize or reestablish the connection

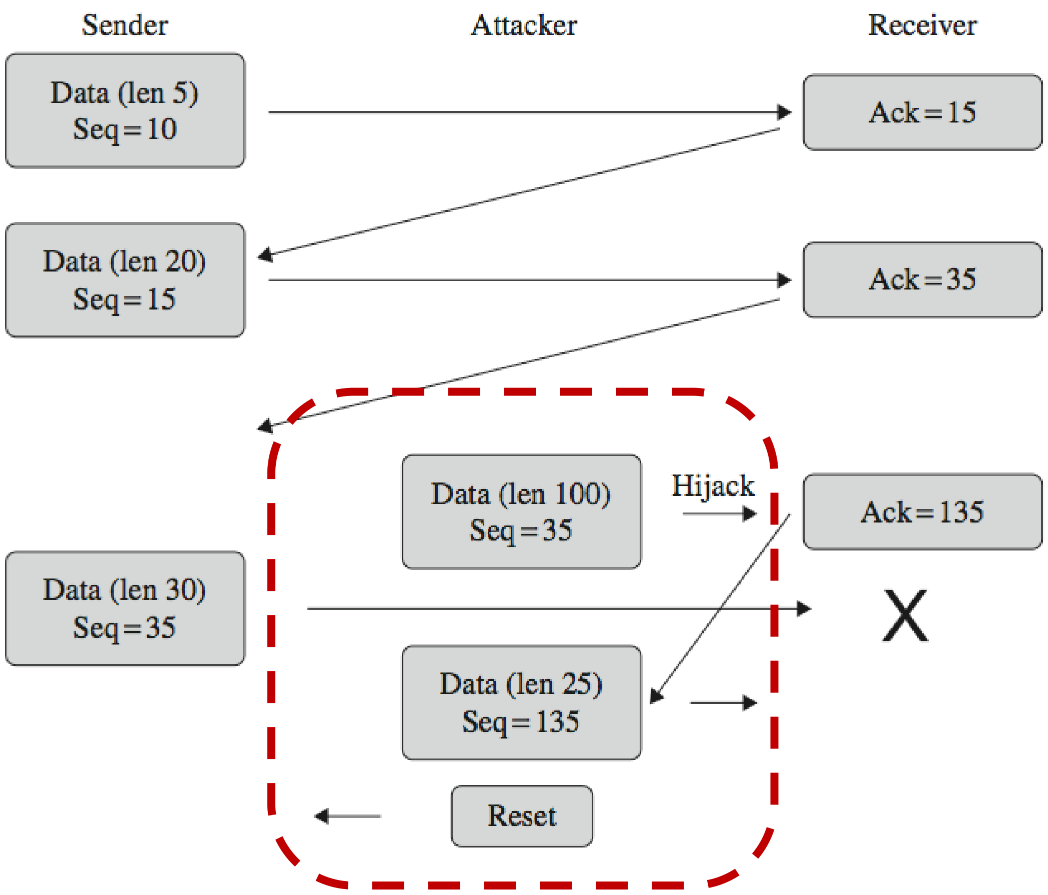


Normal TCP Exchange

Session Hijacking (Cond.)

- Attacker inserts a packet that maintains synchronization with receiver but destroys that with the real sender.
- Real sender's packet is rejected and receiver talks to attacker
- Attacker blocks sender through reset
- Attacker needs to guess the right Ack and computes Seq

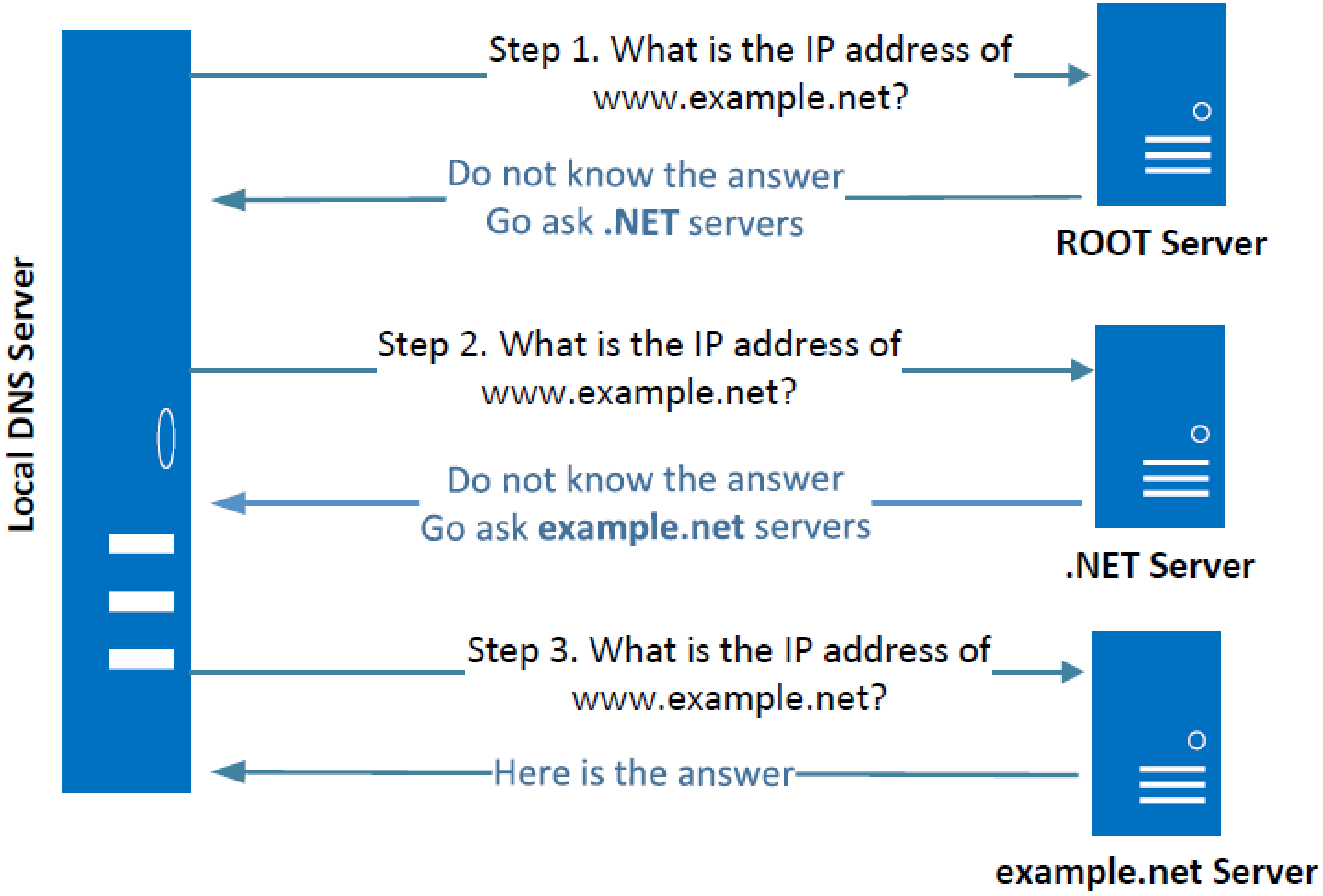
Question: *how to fix?*



TCP hijack

Attacks at Application layer

Background: DNS query

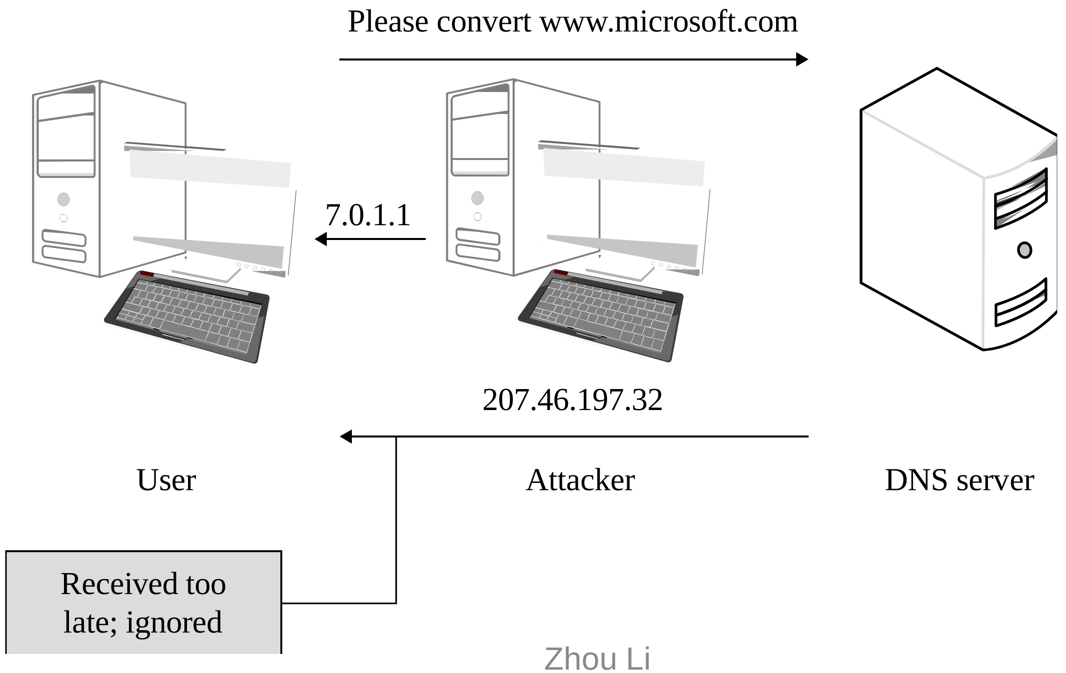


- The iterative process starts from the ROOT Server. If it doesn't know the IP address, it sends back the IP address of the nameservers of the next level server (.NET server) and then the last level server (example.net) which provides the answer.

DNS Spoofing

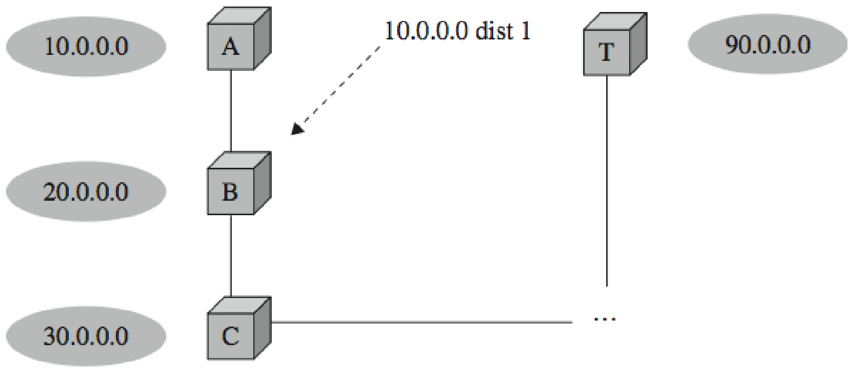
- A MitM attacker intercepts and replies to a query before the real DNS server can respond
- Response from authentic DNS server ignored
- Attack can be persistent because of DNS cache

Question: *how to fix?*

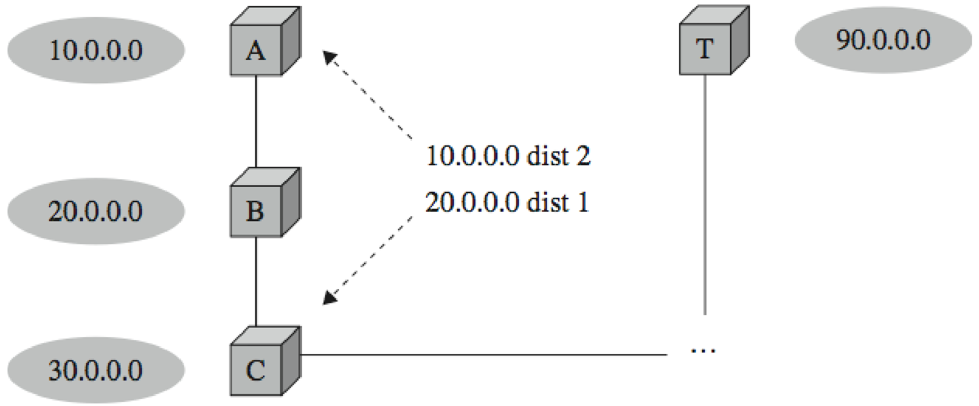


Background: Routing

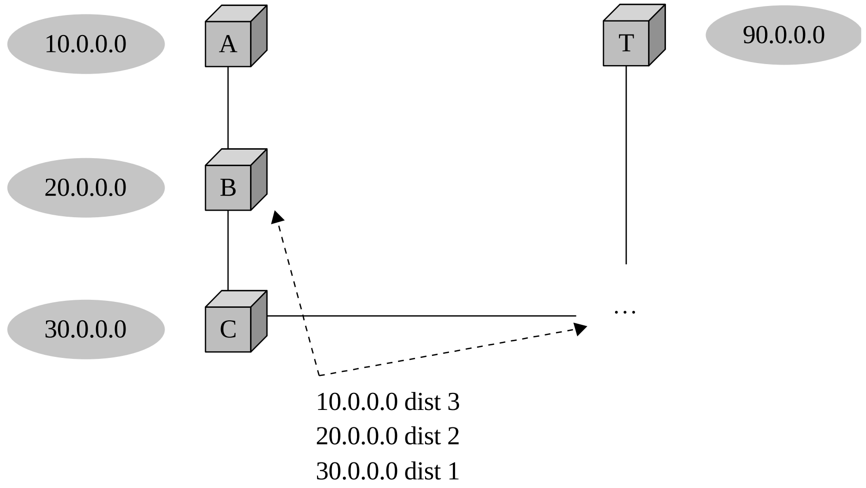
- Network routing updates
  - Border Gateway Protocol (BGP)
  - Router sends a msg to other routers, listing addresses it has a path
  - Other routers add paths and propagate the info iteratively



Step 1: Router A advertise it's distance 1 to any machine from 10.0.0.0 subnet



Step 2: Router B advertises to A&C that it's distance 2 to 10.0.0.0 and 1 to its own subnet



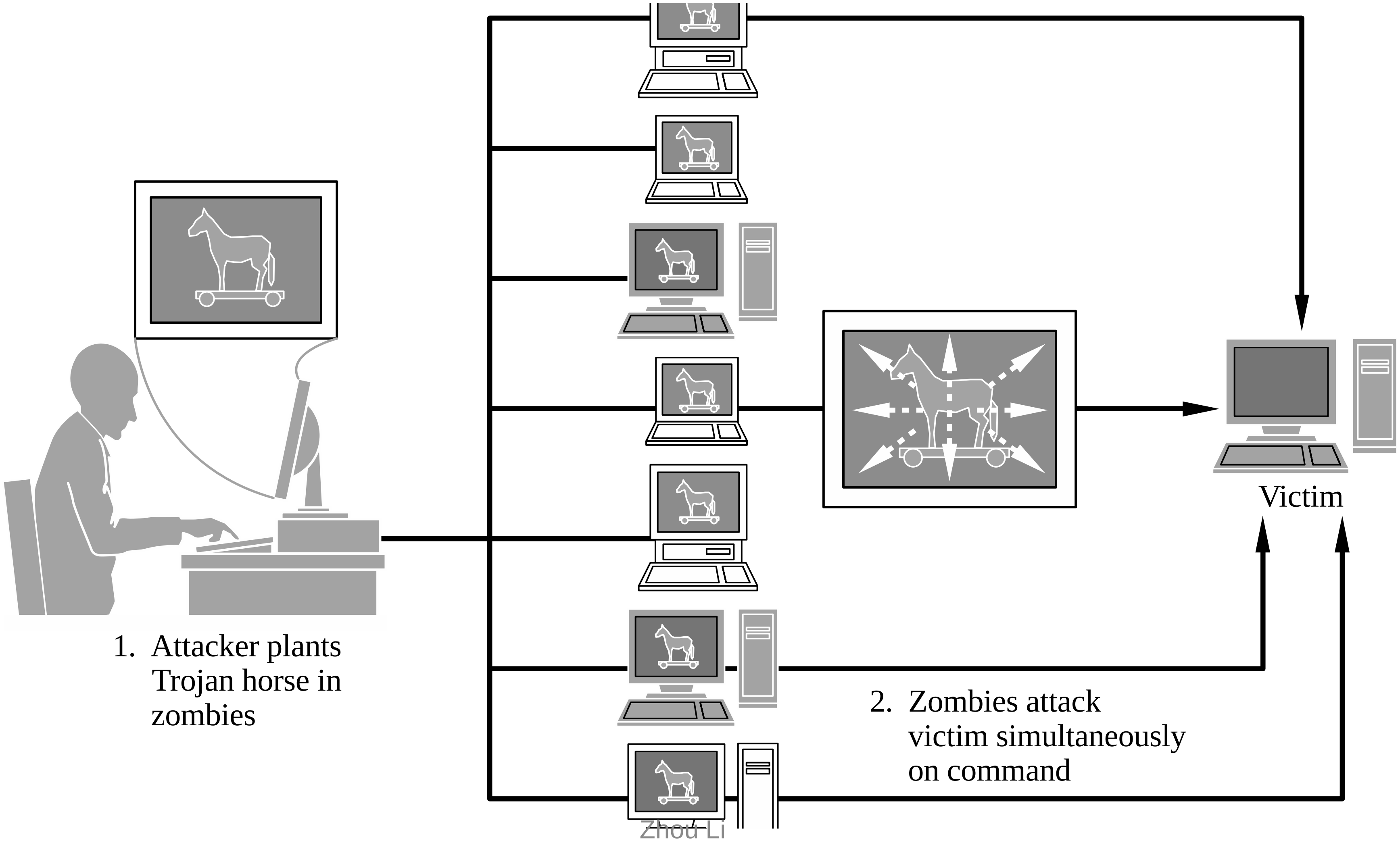
Step 3: Router C does similar things

Attack routing protocol

- Routers operate on implicit trust
- Routing table can be manipulated but no way to authenticate the changes under BGP
- Attack can succeed when knowing the right timing and sequence numbers
- Attack needs to be on edge of a large subnet
  - E.g., impersonating ISP

Question: *how to fix?*

Distributed Denial of Service (DDoS)



Botnet

- Botnet: logical collection of internet-connected devices such as computers, smartphones or IoT devices whose security has been breached and control ceded to a third party.
- Mirai botnet
  - Composed of compromised IoT devices
  - Large DDoS attack in 2016

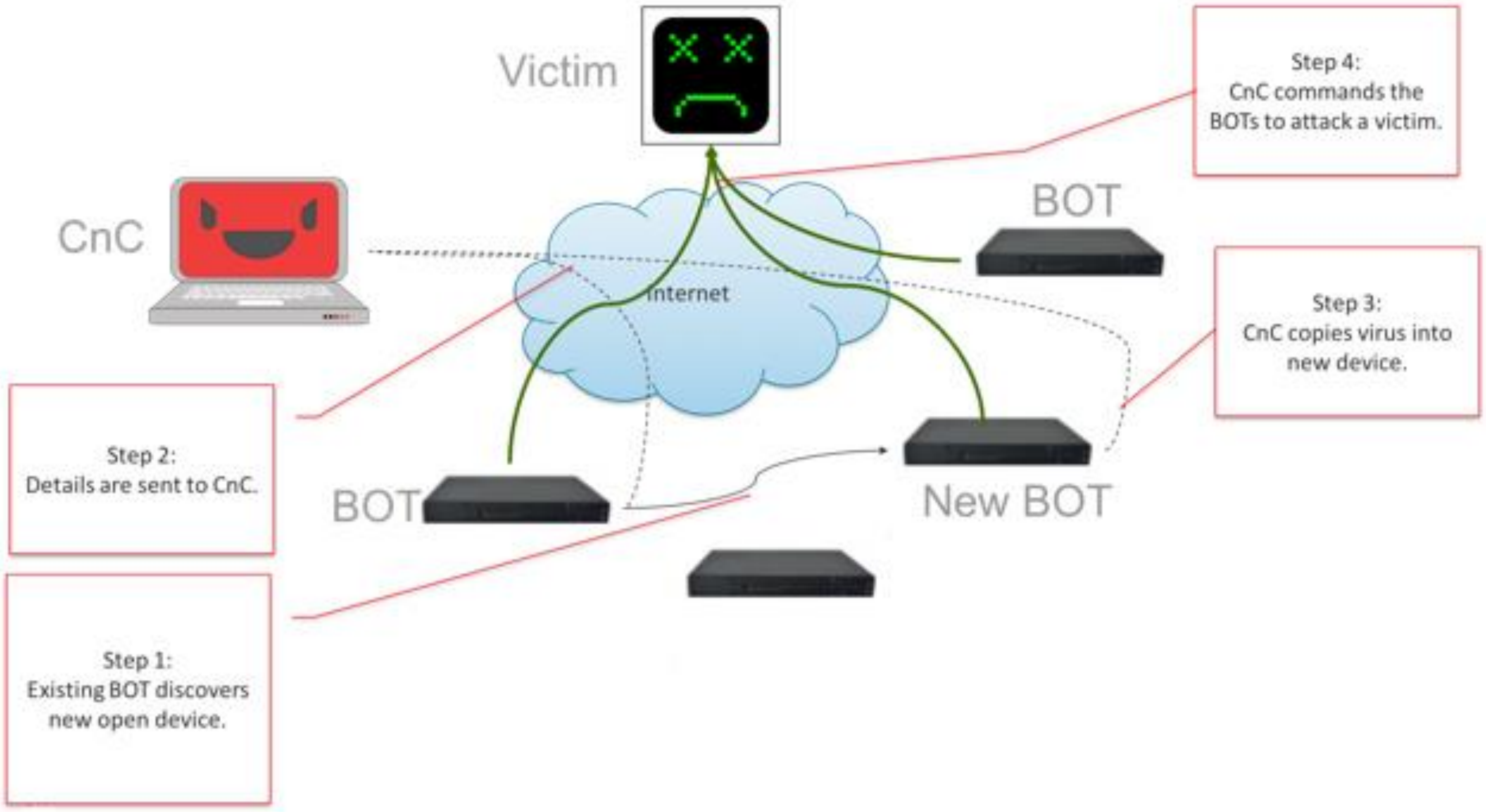


Figure 1 Mirai System

https://www.corero.com/resources/ddos-attack-types/mirai-botnet-ddos-attack.html

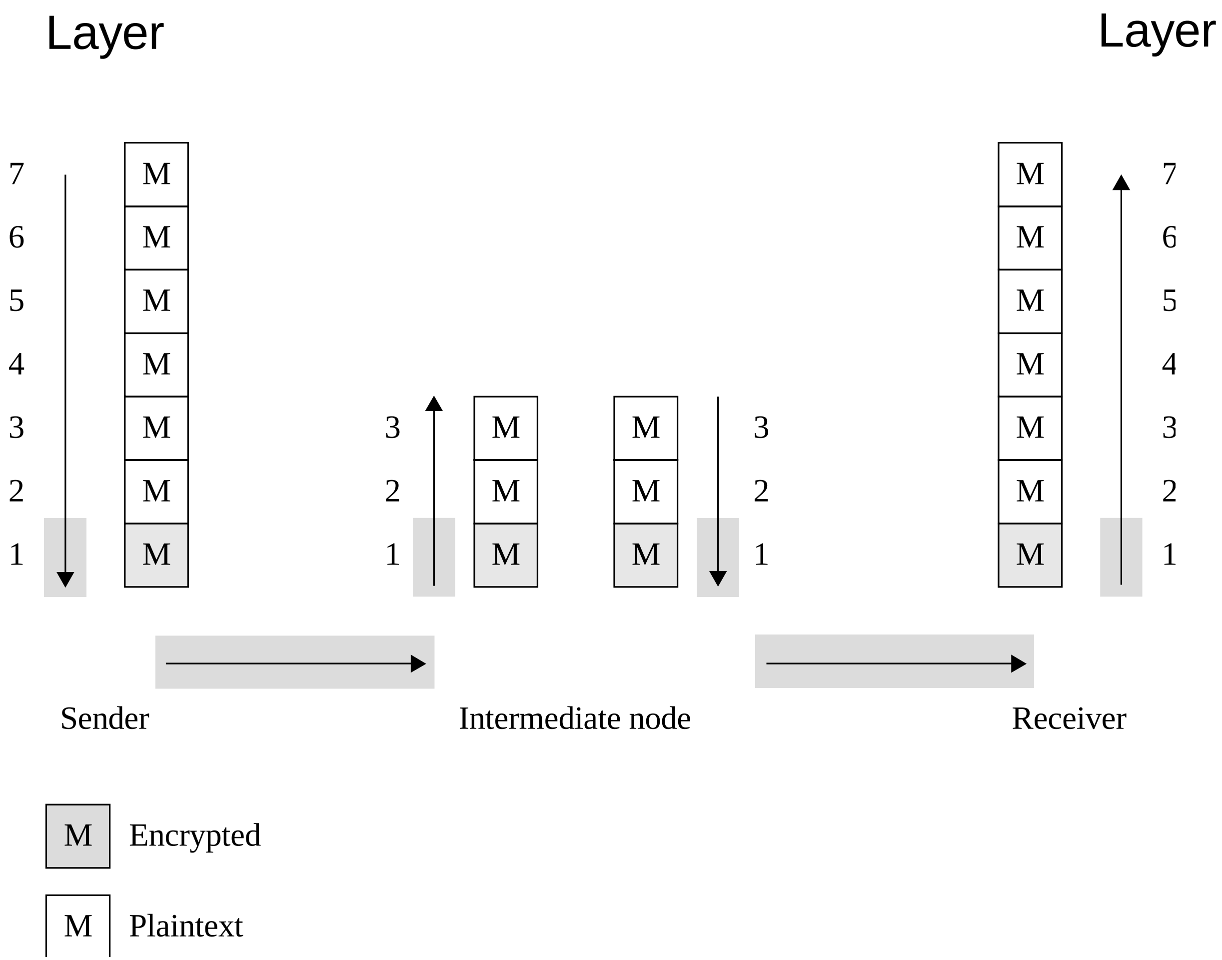
Network encryption concepts and tools

Network Encryption

- Protect only what's encrypted
  - Not every piece of transmission is protected
- Encryption is no more secure than its key management
  - Game over when key is deduced (e.g., weak key)
  - Key distribution is very important (Diffie-Hellman, RSA, …)
- Not silver bullet
  - Flawed system with encryption is still flawed
- Encryption can be done between two hosts (link encryption) and two applications (end-to-end encryption)

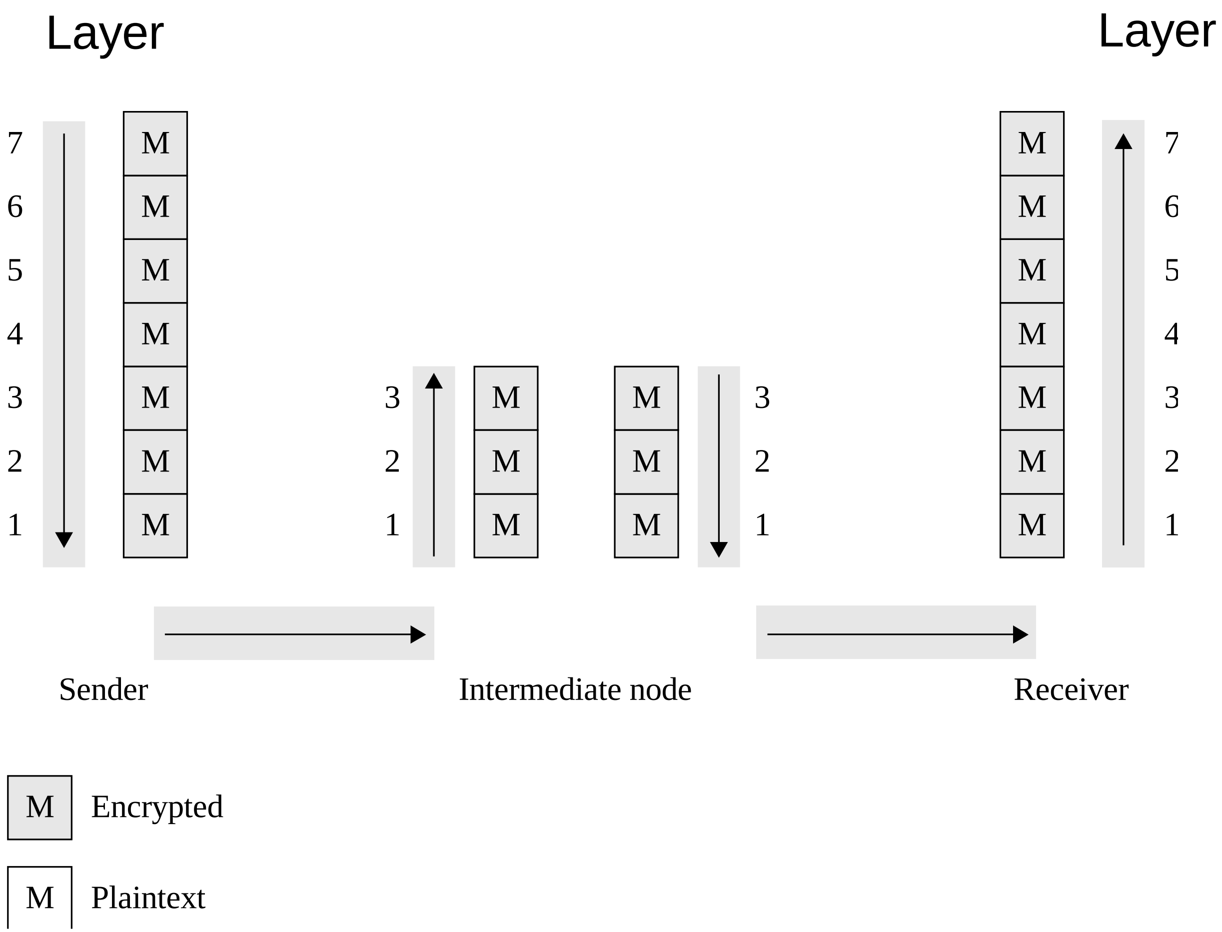
Link Encryption

- Encryption/ Decryption occurs at layer 1 or 2 by software/hardware
- Intermediate host learns everything
- Use case
  - All hosts are trusted
  - Transmission line is most vulnerable
  - E.g., WPA



End-to-End Encryption

- Encryption/ Decryption occurs at layer 7 by application, sometimes 4-6
- Intermediate host doesn't learn payload
- Use case
  - Some intermediate hosts are untrusted
  - Lower layers fail to preserve security
  - E.g., HTTPS

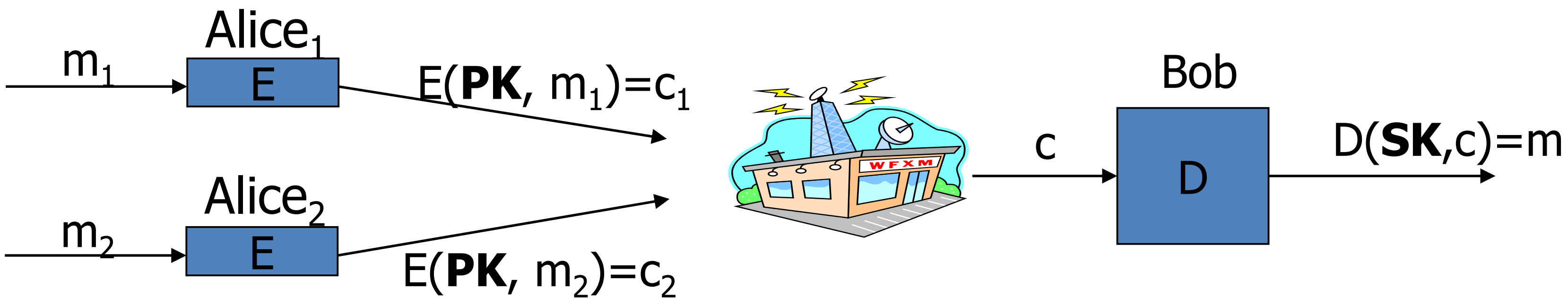


SSL and TLS (End-to-End)

- Secure Sockets Layer (SSL) was designed in the 1990s to protect communication between a web browser and server
  - SSL 1.0 (private), SSL 2.0 (1995), SSL 3.0 (1996)
- In a 1999 upgrade to SSL (SSL 3.1), it was renamed Transport Layer Security (TLS)
  - TLS 1.0, TLS 1.1 (2006), TLS 1.2 (2008), TLS 1.3 (2018)
- SSL and TLS are used interchangeably
- SSL is implemented at OSI layer 4 (transport) and provides
  - Server authentication
  - Client authentication (optional)
  - Encrypted communication

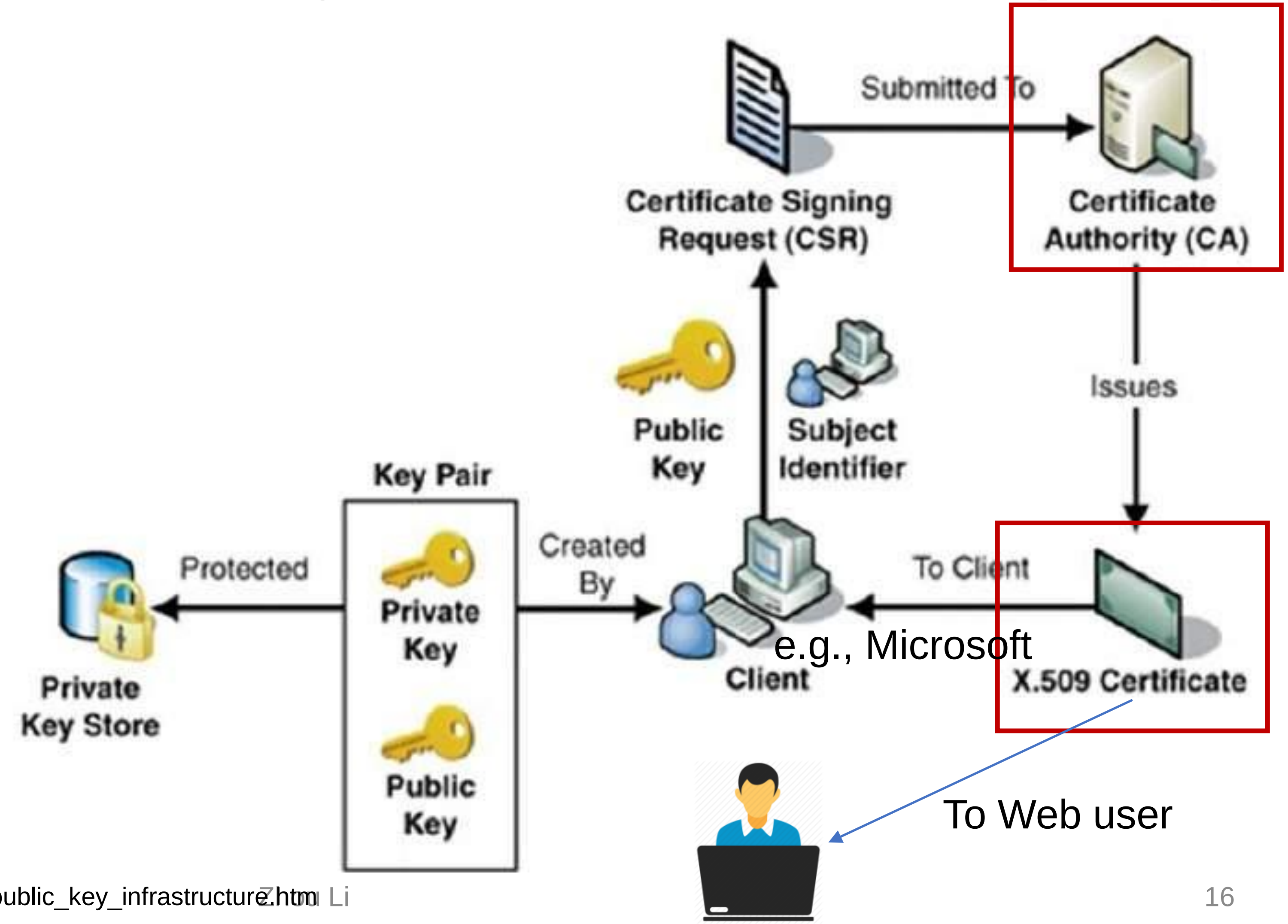
Flashback: Public Key Cryptography

- Proposed by Whitfield Diffie & Martin Hellman at 1976
- Instead of two users sharing one secret key, each user has two keys: one *public* and one *private*
- Messages encrypted using the user's public key can only be decrypted using the user's private key, and vice versa



Flashback: Public Key Infrastructure (PKI)

- Tackle's the problem of certificate (public key and identity) creation and distribution



TLS Handshake

- Before a client and server can communicate securely, several things need to be set up first:
  - Encryption algorithm and key
  - MAC algorithm
  - Algorithm for key exchange

- These cryptographic parameters need to be agreed upon by the client and server