

# Introduction to Computer Security

Instructor:

Mahadevan Gomathisankaran

[mgomathi@unt.edu](mailto:mgomathi@unt.edu)

# Announcements

- Assignment 1 Graded (out of 100)

- Max: 95

- Min: 50

- Mean: 77.33

- Median: 80

- Q<sub>3</sub>

Key: INTROCOMP USEYABDFGHJKLQVWXZ

monoalphabetic ciphers use permuted alphabets as key

GEBEMUHRMNKCAF FAHRKDJ IJK HKDGICKO MUHRMNKCJ MJ TKL

- Q<sub>4</sub>

Key: SECURITY

- TCP

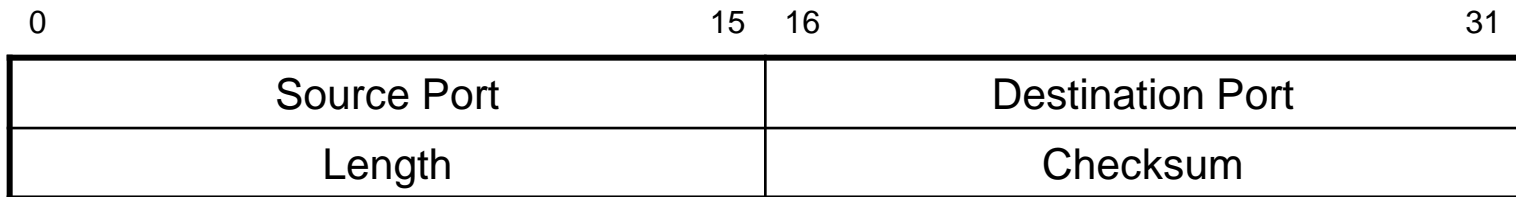
- TCP adds “sessions” or “connections” to the bare IP protocol

0	7	8	15	16	23	24	31
Source Port				Destination Port			
Sequence Number							
Acknowledgment Number							
Data Offset		Flags		Window			
Checksum				Urgent Pointer			
Options							
Data							

## Flags:

- |                                       |                       |                           |
|---------------------------------------|-----------------------|---------------------------|
| CWR: Congestion window reduced        | URG: Urgent ptr valid | RST: Reset flag           |
| ECN: Explicit congestion notification | ACK: ACK valid        | SYN: Synchronize seq #s   |
|                                       | PSH: Push function    | FIN: Finish of connection |

- UDP
  - UDP adds connection distinguishers (ports) to IP

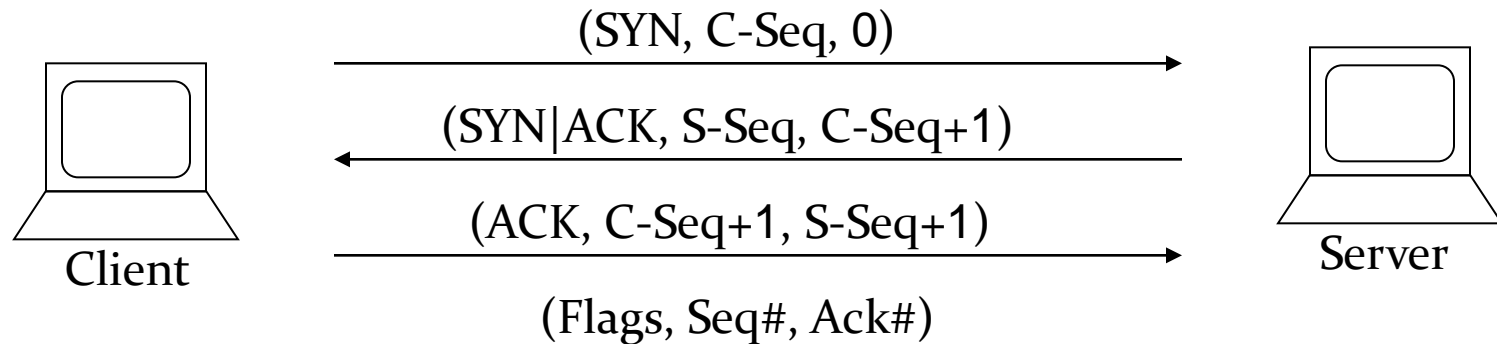


Some common UDP protocols:

- Domain Name Service (DNS) – port 53
- Network Time Protocol (NTP) – port 123
- “Discoverable” services (IPP, Rendezvous, ...)
- Streaming/multicast transmissions

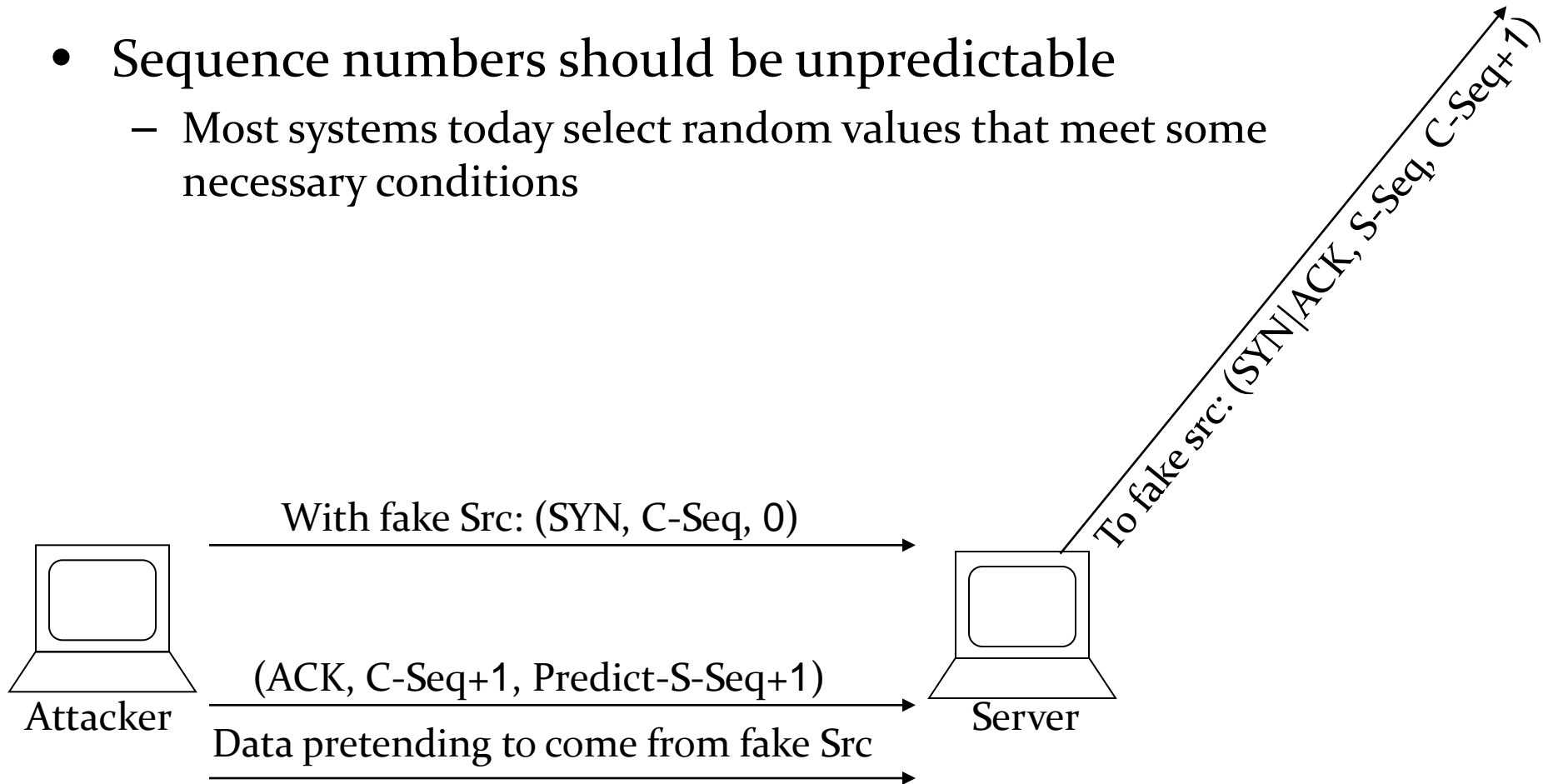
- Connection Establishment

- To establish connection, client must prove that it received the SYN|ACK packet
- SYN|ACK packet routed to system with source address from first SYN packet
  - Since based on routing, only secure back to the *subnet* of the source

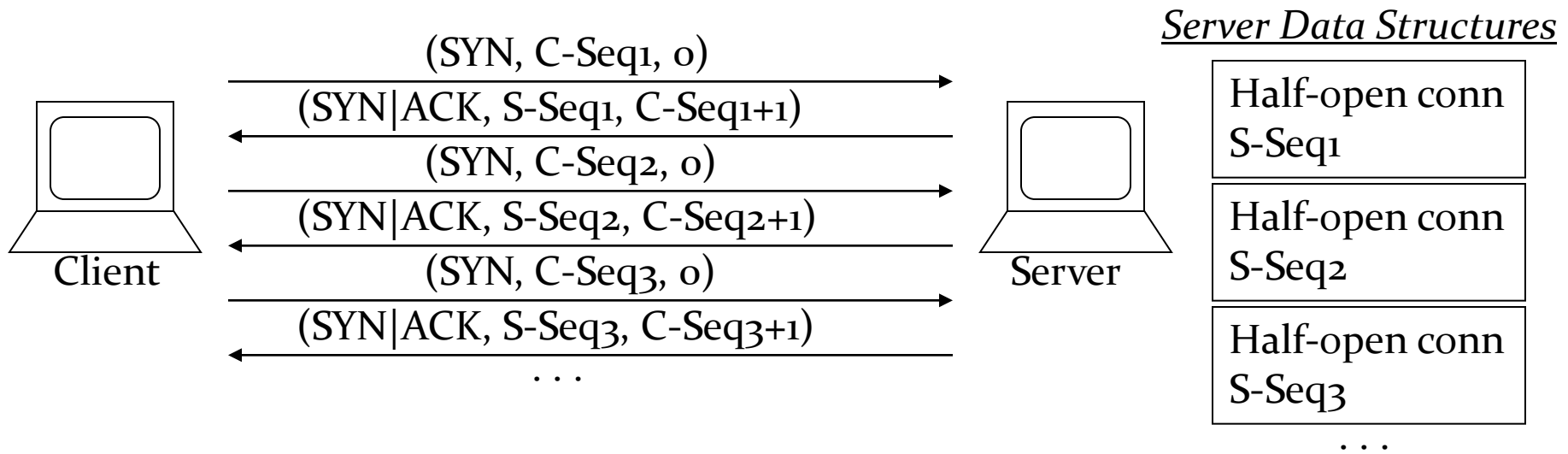


# TCP Handshake

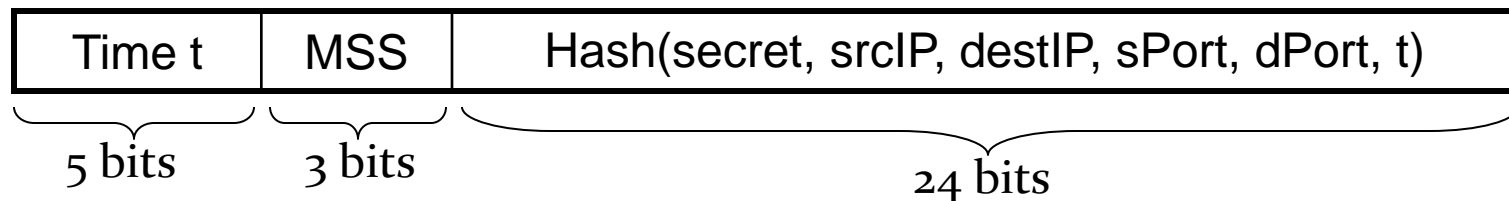
- Sequence numbers should be unpredictable
  - Most systems today select random values that meet some necessary conditions



- SYN Flooding
  - DoS isn't due to traffic volume but to resource exhaustion (memory) in the server O.S.
  - Early network stacks had a severely limited number of half-open structures available
  - Can spoof SRC address with non-existent host



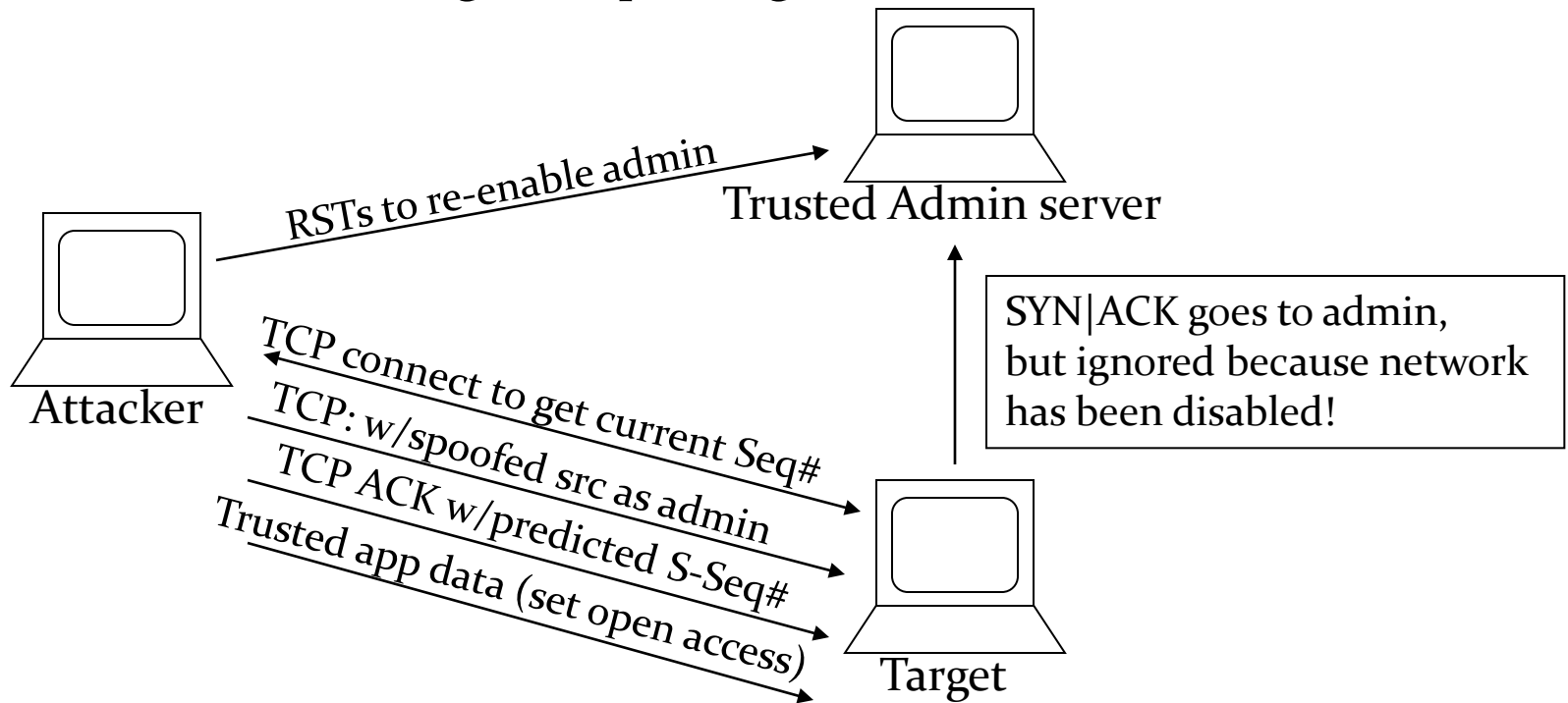
- SYN cookies
  - Basic idea: Use cryptography to avoid saving state
  - Specifically: Store info in Seq # to verify upon ACK



- Time: Increments every 64 seconds
- MSS = Maximum Segment Size (must be remembered!)
- Not perfect: Limited MSS options, TCP Options
- Router solutions (protect hosts without modifying hosts)
  - Rate limiting/shaping, Cisco router “TCP Intercept” feature, ...

# TCP Handshake

- Combination of Flooding and Spoofing



## Lessons learned:

In network stack: Seq#'s must be unpredictable!

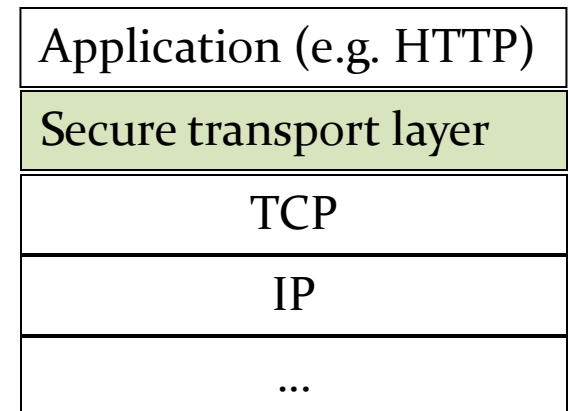
In network setup: Should filter out local srcIPs coming from outside

In application: IP-based trust is a very bad idea!

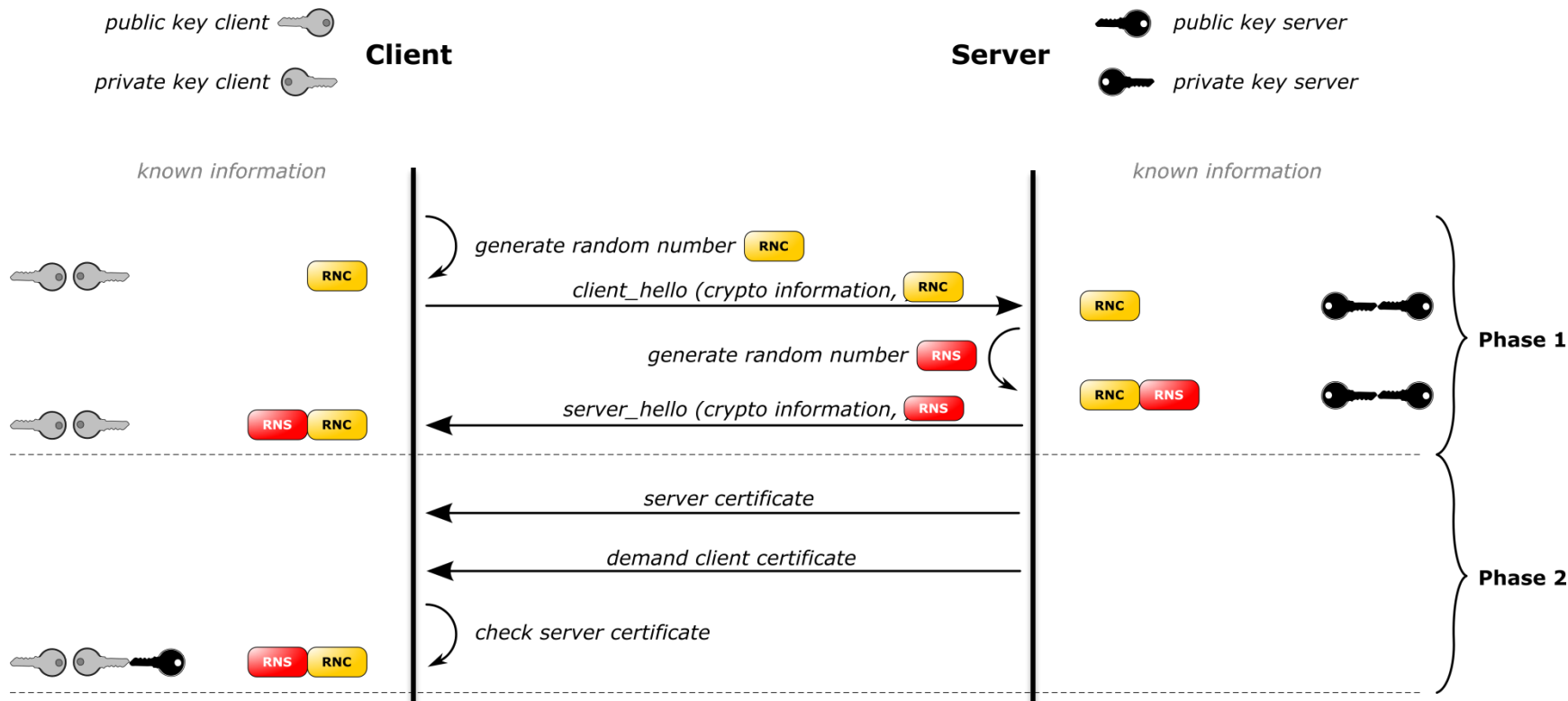
# Transport Layer Security

- Originally designed to protect web browser to web server
  - Invented by Netscape
  - Generic TCP protection
  - Authentication: Supports server and client certificates
  - Confidentiality: Symmetric encryption after key establishment
  - Integrity: All packets protected with a MAC
- Later versions (SSL v2.1) referred to as TLS
  - TLS incorporated within application-layer protocols now in addition to in a sub-application layer
    - Example 1: IMAP (mail) can be either a separate SSL protected service/port (imaps: port 993) or negotiated after plaintext startup in standard IMAP (port 143)
    - Example 2: LDAP with similar options (ldap is port 389, ldaps is port 636)

- SSL provides privacy, authentication and integrity of web transactions.
- This is a protocol that is located between application layer (e.g. HTTP) and transport protocol (e.g. TCP)
- By running SSL on top of TCP, all standard features of TCP as reliability, flow control and congestion control are provided to the application layer.
- As an example, when HTTP is used this way, it is called HTTPS.

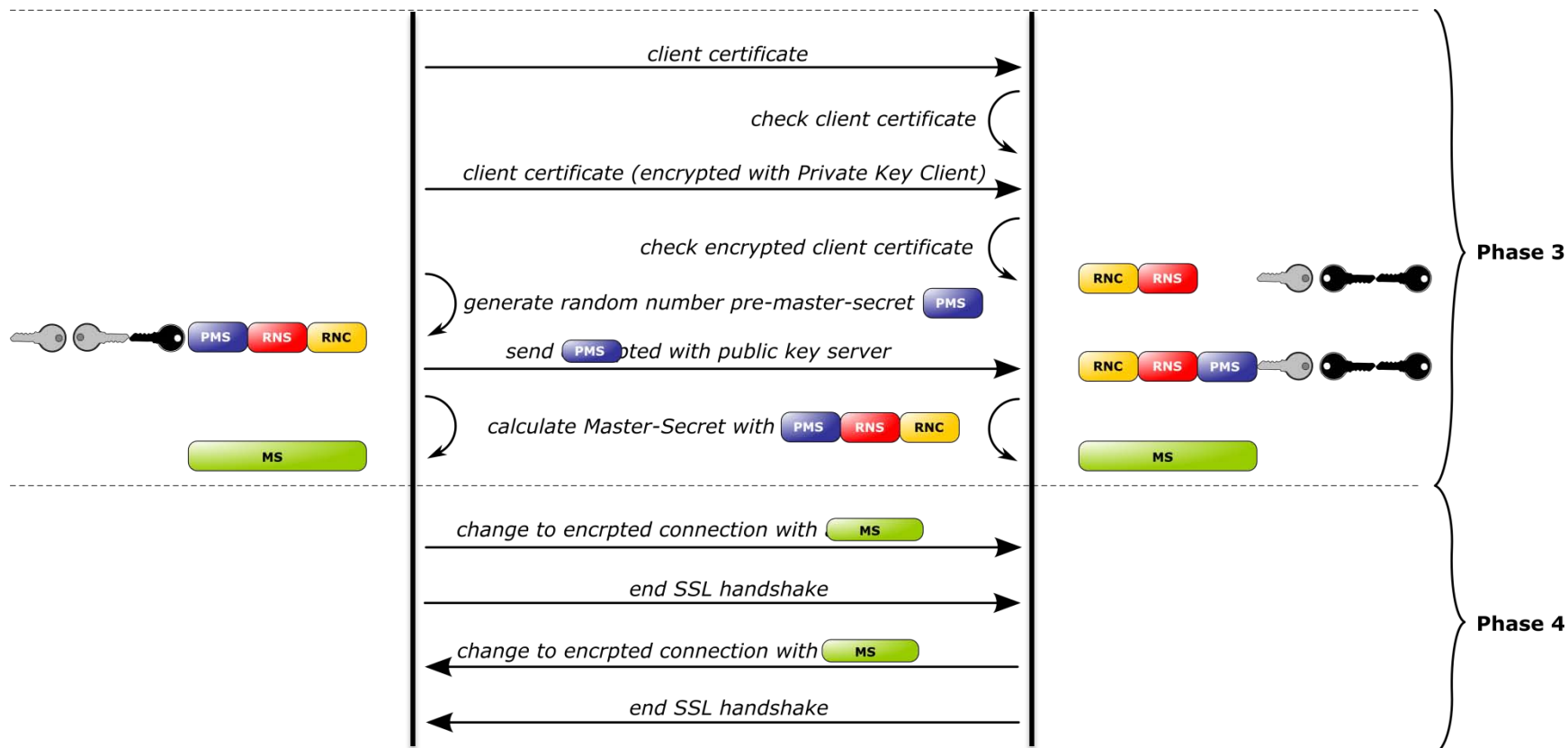


# TLS/SSL



Source: Wikipedia

# TLS/SLL

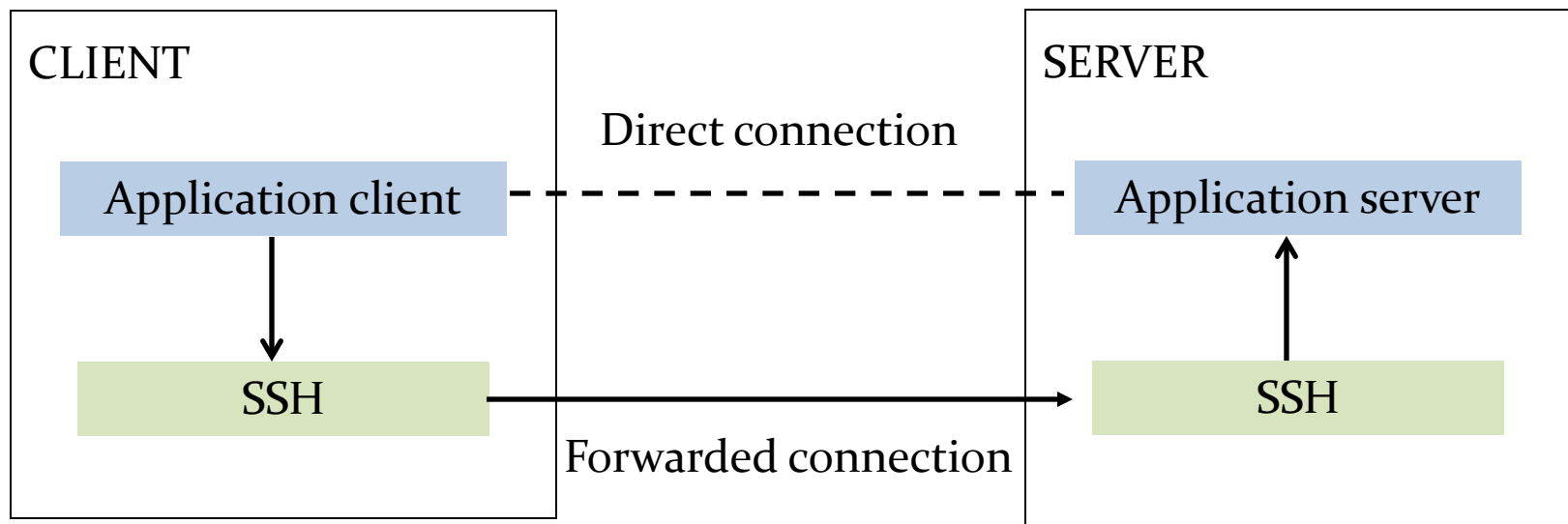


- SSH: Secure Shell
  - Designed to replace telnet for secure logins
  - Provides encryption and public key host verification
- Provides
  - client/server authentication
  - message integrity
  - Confidentiality
- Consists of 3 protocols:
  - SSH-TRANS: a transport layer protocol
  - SSH-AUTH: an authentication protocol
  - SSH-CONN: a connection protocol
- Port forwarding:
  - Arbitrary TCP sessions encapsulated in an established SSH connection

- Provides an encrypted channel between client and server
- Runs on a TCP connection
- Client authenticates server using RSA (a public key algorithm)
- Once SSH-TRANS channel exists, client logs onto the server, i.e. authenticates himself to the server.
  - Client sends his password to server as encrypted through SSH-TRANS channel.
  - Client uses public key encryption to send his password
  - Host-based auth.: Client host authenticates himself to server when they first connect. Afterwards, the same user is automatically authenticated b/c he belongs to a set of trusted hosts

- SSH has proven so useful as securing remote login that it also supports other insecure TCP applications, such as X Windows and IMAP mail readers.
- These applications are run over a secure SSH tunnel.
- This capability is called port forwarding, which uses SSH-CONN protocol.

- When message arrives at SSH port on the server, SSH decrypts the contents and forwards the data to the actual port at which the server is listening.



- Voice traffic
  - Secure, encrypted phones
    - STU-III, etc.
    - VoIP issues become more important with spread of VoIP
- Web traffic:
  - Not protected at the application layer
  - https is regular HTTP over SSL/TLS
- E-mail
  - Store-and-forward communication
    - Many people can see unprotected e-mails
    - Trivial to forge
  - Most e-mail on the Internet is forged
    - Up to 90% of all e-mail is spam/phishing

- Security goals for e-mail:
  - Message confidentiality
  - Message integrity
  - Sender authenticity
  - Non-repudiation
- Two main types of e-mail security
  - S/MIME based
    - Standards-based, using X.509 certificates
    - Hierarchical certificate structure fine for companies
  - PGP (Pretty Good Privacy) based
    - Grew out of early 1990's program (by Phil Zimmerman)
    - Non-hierarchical trust model (“web of trust”)
    - “Free” (open source and patent-free) cousin: GPG

# Secure E-mail /X.509

*"Trust Anchor" or "Root CA"*

Subject: Verisign
Verisign Public Key
Issuer: Verisign (trusted)

*Signs*

Subject: UNT CA
UNT Public Key
Issuer: Verisign

*Signs*

Subject: UNT CSE CA
UNT CSE Public Key
Issuer: UNT CA

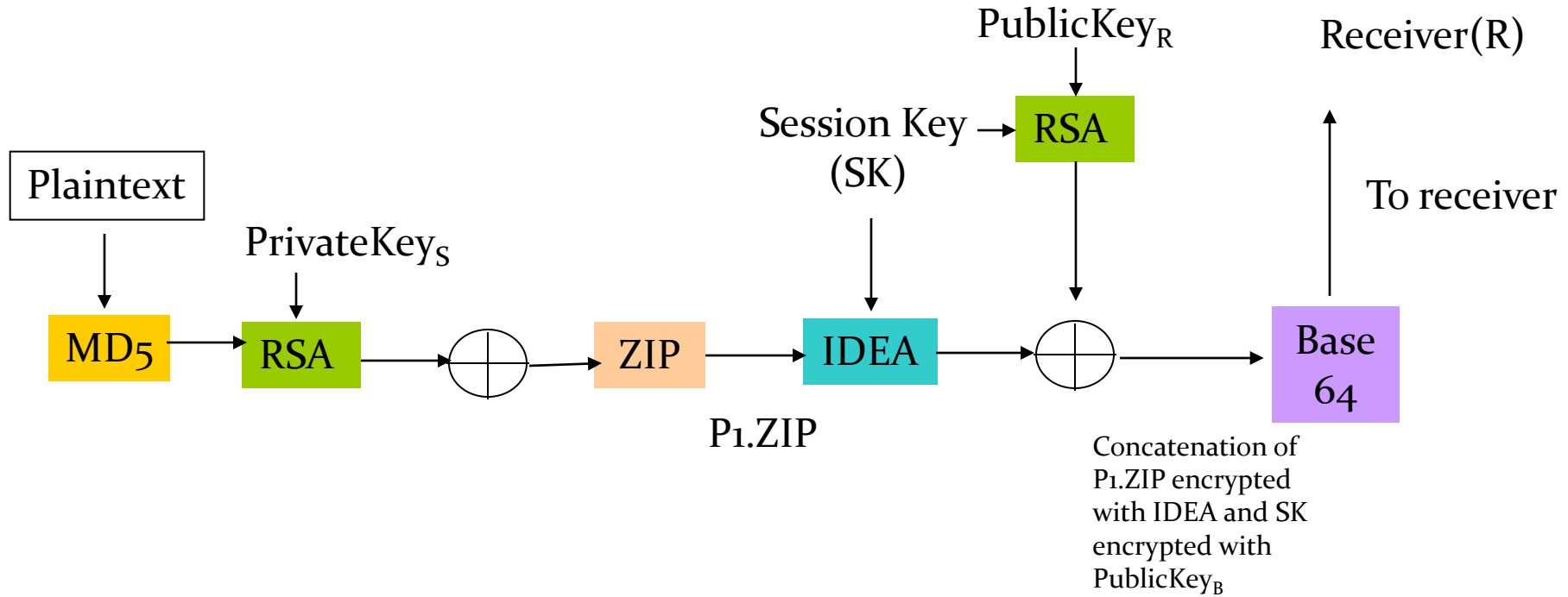
*Signs*

Subject: CSCE4550/5550
Instructor's Public Key
Issuer: UNT CSE CA

Good for companies, military, or other organizations with hierarchical structure.

- Trust model is less hierarchical than X.509
- I can sign keys and distribute them
  - Anyone who trusts me can use me as a CA!
  - Difference between “trusted” and “valid” keys
- Problem: How do you get public keys?
  - Note: In PGP public keys are always certificates
- Solution: Keyservers – databases of keys
  - You can submit your own keys
  - You can look up keys by name or e-mail address
  - Support integrated into many e-mail programs
- Keyservers can be accessed in many ways
  - LDAP
  - HTTP
  - E-mail

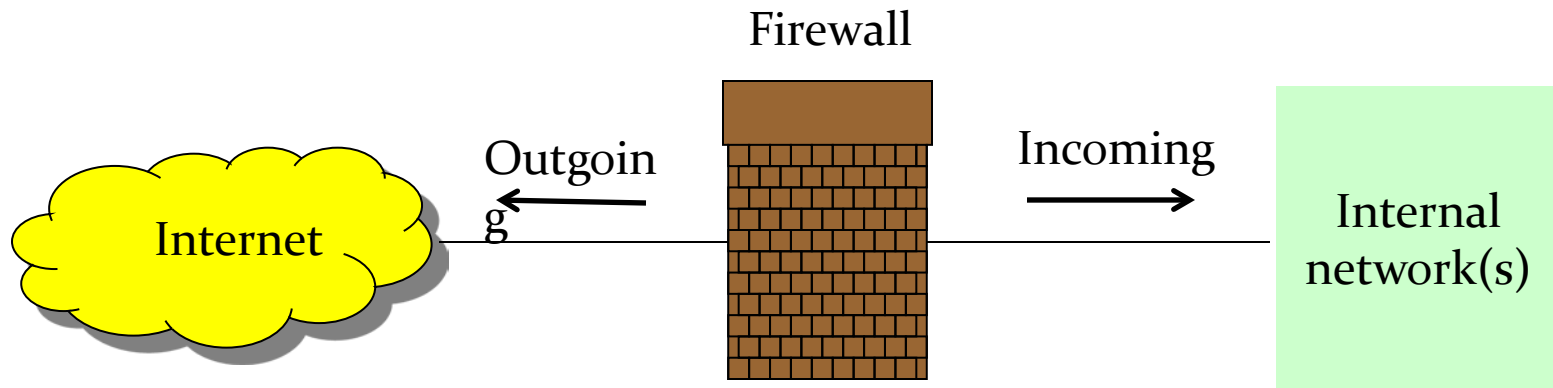
# Secure E-mail /PGP



- Secure Electronic Transactions
  - E-commerce money transfers/credit cards
  - “SET”: Protocol for secure electronic transactions
    - Developed by credit card companies
    - Protects customer, merchant, and bank from fraud
    - Unfortunately hasn’t taken off...
  - Protocol
    1. The customer obtains a credit card account with a bank that supports electronic payment and SET
    2. The customer receives an X.509v3 digital certificate signed by the bank.
    3. Merchants have their own certificates
    4. The customer places an order
    5. The merchant sends a copy of its certificate so that the customer can verify that it’s a valid store
    6. The order is sent to merchant and payment request is sent to bank are sent
    7. The merchant requests payment authorization
    8. The merchant confirms the order
    9. The merchant ships the goods or provides the service to the customer
    10. The merchant requests payment

- Secure RPC (Remote Procedure Calls)
  - RPCs are a way of obtaining service from another machine
  - Must consider confidentiality and authentication
  - Secure RPC standards (e.g., RFC 2203 for RPCSEC\_GSS)
- Web services
  - SOAP and .net
  - Similar to RPC in goals and considerations

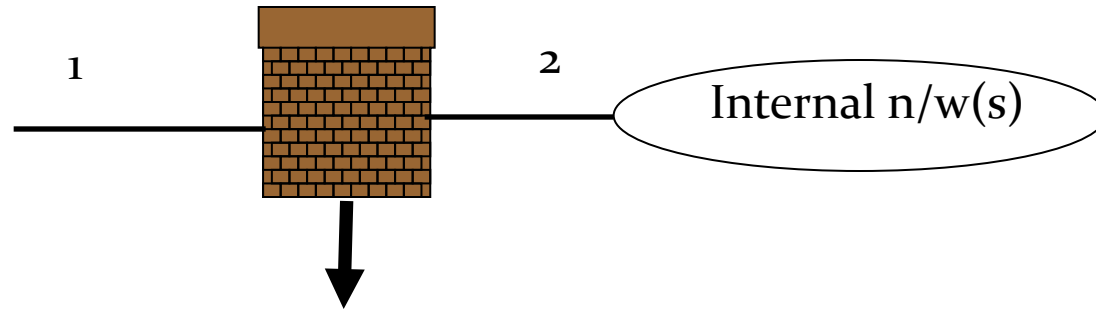
- Designed to forward some packets and filter (not forward) others.
  - Packet Filter
  - Application Gateway
  - Circuit Gateway
  - Proxy Server



# Packet Filter

- Firewall can forward or block packets based on n/w and transport layer headers:
  - Source/Dest. IP addresses
  - Source/Dest. port addresses
  - Protocol type (TCP or UDP)
- Decisions made independently on a packet-by-packet basis
- Good for blocking ports (“no incoming HTTP”) or blocking IP addresses/ranges (“blacklists”)
- Simple and fast – included in many routers

# Packet Filter



Interface	Source IP	Source Port	Destination IP	Destination Port
1	127.13.41.24	*	*	*
1	*	*	*	23
1	*	*	196.37.21.6	*
2	*	80	*	*

# Proxy Firewall

- Provides filtering based on application layer
- Doesn't forward packets at all – works at application layer (ex: Web proxies)
- Allows content filtering as well as security

