Security in Computer Networks

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@wustl.edu

Audio/Video recordings of this lecture are available on-line at:

http://www.cse.wustl.edu/~jain/cse473-09/

1. Secret Key Encryption
2. Public Key Encryption
3. Hash Functions
4. Digital Signature, Digital Certificates
5. IPSec, VPN, Firewalls, Intrusion Detection

Not Covered: Email Security, SSL, IKE, WEP

Note: This class lecture is based on Chapter 8 of the textbook (Kurose and Ross) and the figures provided by the authors.

Security Requirements

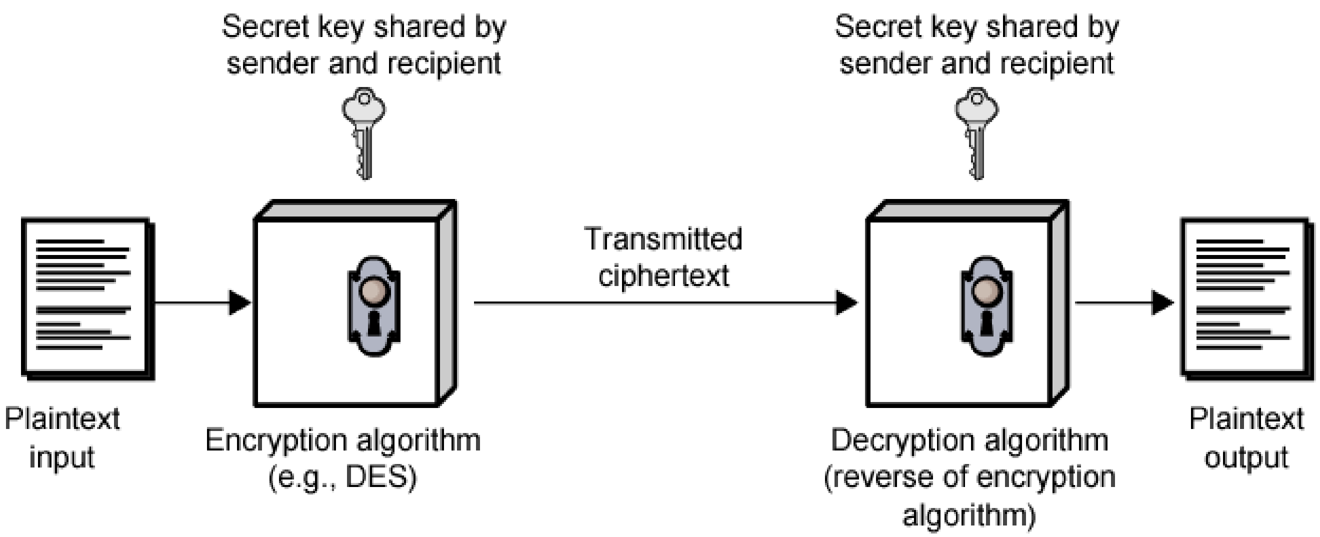
- **Integrity**: Received = sent?
- **Availability**: Legal users should be able to use.
  Ping continuously ⇒ No useful work gets done.
- **Confidentiality and Privacy**:
  No snooping or wiretapping
- **Authentication**: You are who you say you are.
  A student at Dartmouth posing as a professor canceled the exam.
- **Authorization** = Access Control
  Only authorized users get to the data
- **Non-repudiation**: Neither sender nor receiver can deny the existence of a message

Secret Key Encryption

1. Secret Key Encryption
2. Block Encryption
3. Cipher Block Chaining (CBC)
4. DES, 3DES, AES
5. Stream Cipher: RC4
6. Key Distribution

Secret Key Encryption

- Also known as symmetric key encryption
- Encrypted_Message = Encrypt(Key, Message)
- Message = Decrypt(Key, Encrypted_Message)
- Example: Encrypt = division
- 433 = 48 R 1 (using divisor of 9)



Secret Key: A Simple Example

- **Substitution**: Substituting one thing for another
- **Monoalphabetic**: substitute one letter for another

plaintext: abcdefghijklmnopqrstuvwxyz

ciphertext: mnbvcxzasdfghjklpoiuytrewq

E.g.:

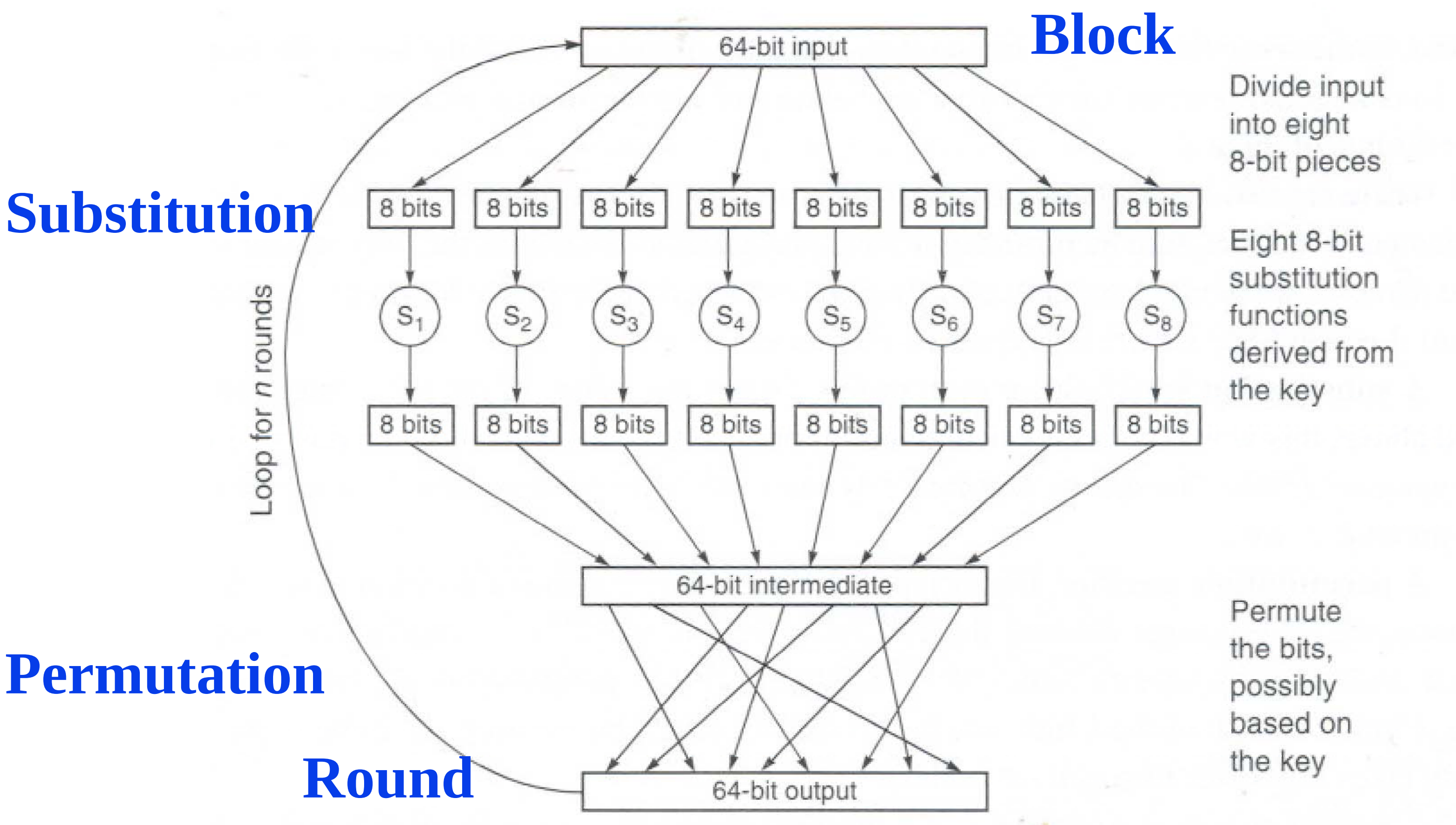
Plaintext: bob. i love you. alice

ciphertext: nkn. s gktc wky. mgsbc

- **Polyalphabetic**: Use multiple substitutions C1, C2, …
  Substitution selected depends upon the position
  ⇒Same letter coded differently in different position

Block Encryption

- Block Encryption

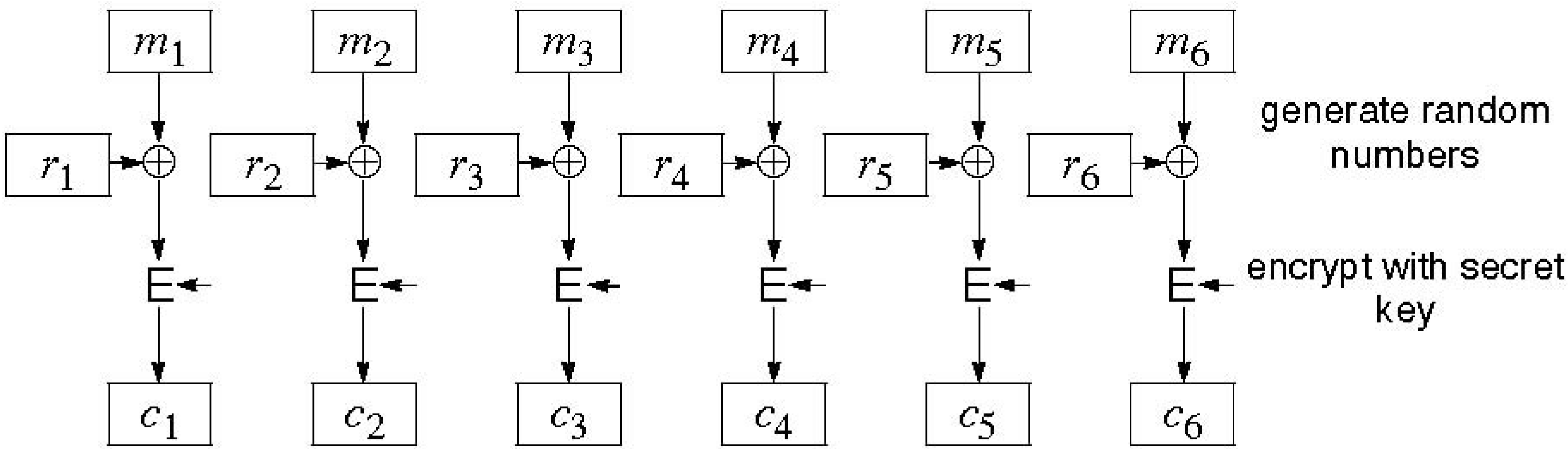


Block Encryption (Cont)

- Short block length $\Rightarrow$ tabular attack
- 64-bit block
- Transformations:
  - Substitution: replace k-bit input blocks with k-bit output blocks
  - Permutation: move input bits around. $1 \rightarrow 13$, $2 \rightarrow 61$, etc.
- Round: Substitution round followed by permutation round and so on. Diffusion + Confusion.

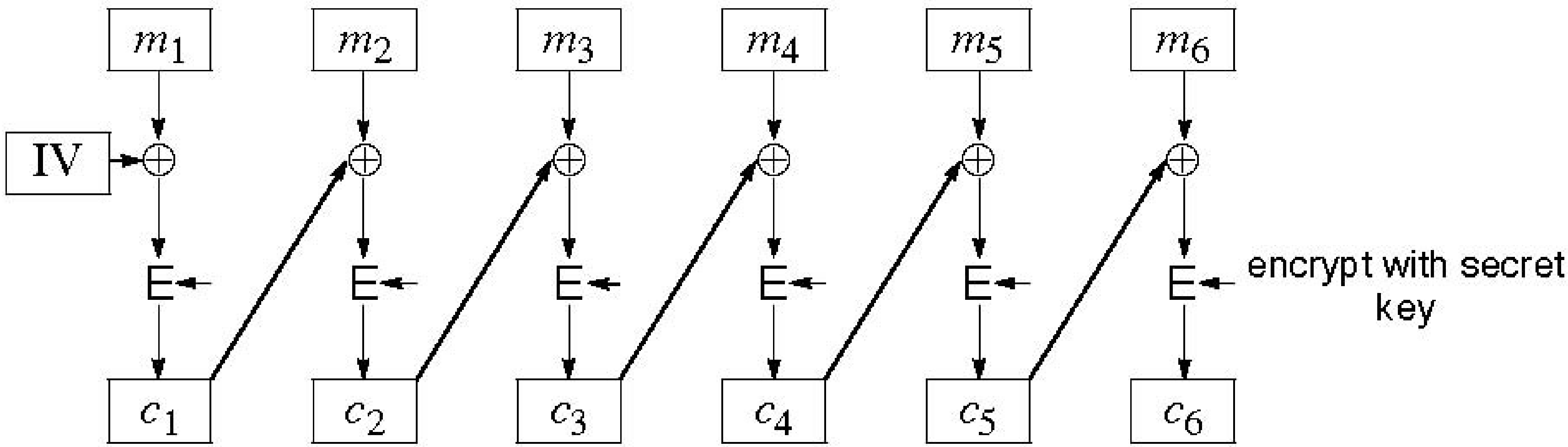
Cipher Block Chaining (CBC)

- Goal: Same message encoded differently
- Add a random number before encoding



CBC (Cont)

- Use C_i as random number for i+1



- Need Initial Value (IV)
- no IV ⇒ Same output for same message
  ⇒ one can guess changed blocks
- Example: Continue Holding, Start Bombing

DES and 3DES

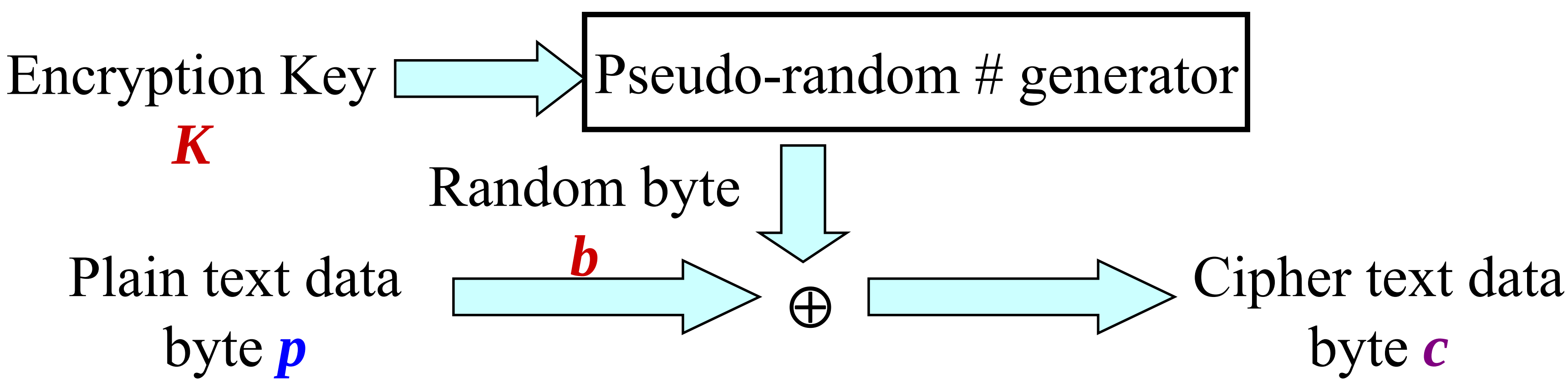
- Data Encryption Standard (DES)
  - 64 bit plain text blocks, 56 bit key
  - Broken in 1998 by Electronic Frontier Foundation
- Triple DES (3DES)
  - Uses 2 or 3 keys and 3 executions of DES
  - Effective key length 112 or 168 bit
  - Block size (64 bit) too small $\Rightarrow$ Slow

Advanced Encryption Standard (AES)

- Designed in 1997-2001 by National Institute of Standards and Technology (NIST)
- Federal information processing standard (FIPS 197)
- Symmetric block cipher, Block length 128 bits
- Key lengths 128, 192, and 256 bits

Ron's Cipher 4 (RC4)

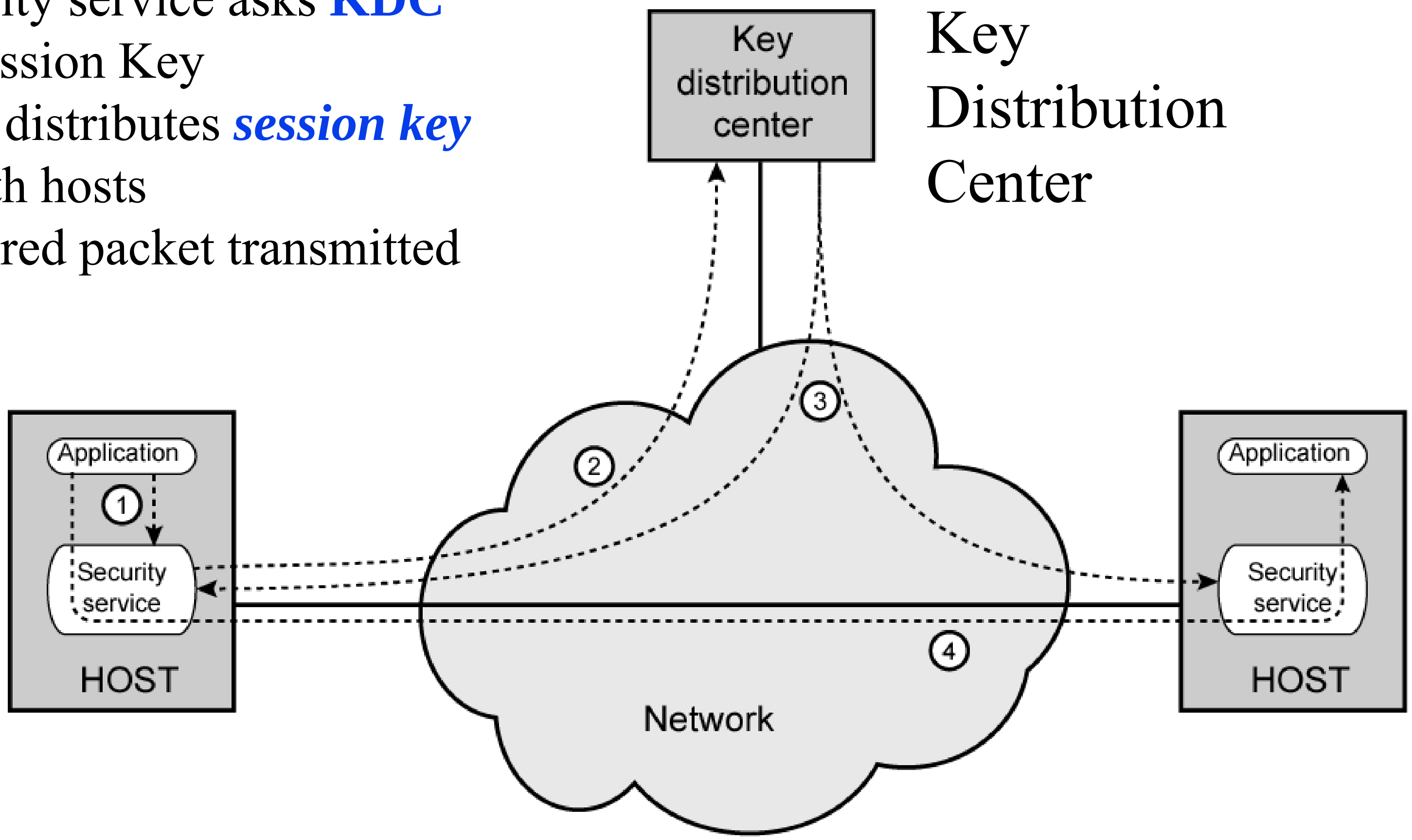
- Developed by Ron Rivest in 1987. Trade secret. Leaked 1994.
- Stream Cipher
  - A pseudo-random stream is generated using a given key and xor'ed with the input
- Pseudo-random stream is called **One-Time pad**
- Key can be 1 to 256 octet
- See the C code in the textbook [KPS].



Key Distribution

1. Application requests connection
2. Security service asks **KDC** for session Key
3. KDC distributes ***session key*** to both hosts
4. Buffered packet transmitted

Key Distribution Center



KDC shares a secret key with each Host.

Secret Key Encryption: Review

1. Secret key encryption requires a shared secret key
2. Block encryption, e.g., DES, 3DES, AES break into fixed size blocks and encrypt
3. CBC is one of many modes are used to ensure that the same plain text results in different cipher text.
4. Stream Cipher, e.g., RC4, generate a random stream and xor to the data
5. Key distribution center can be used to exchange session keys

Home Exercises

- Try but do not submit
- Review questions R1, R2, R6
- Problems P1, P2, P3, P4, P5, P6

Homework 8A

- Problem P6: Consider 3-bit block cipher in Table 8.1.

Plain	000	001	010	011	100	101	110	111
Cipher	110	111	101	100	011	010	000	001

- Suppose the plaintext is 100100100.

(a) Initially assume that CBC is not used. What is the resulting ciphertext?

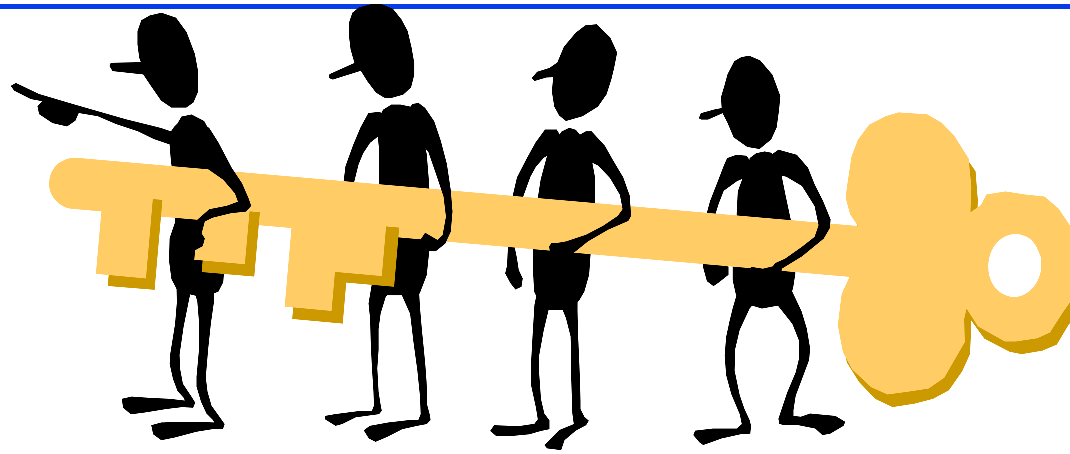
(b) Suppose Trudy sniffs the cipher text. Assuming she knows that a 3-bit block cipher without CBC is being employed (but doesn't know the specific cipher), what can she surmise?

(c) Now suppose that CBC is used with IV-111. What is the resulting ciphertext?

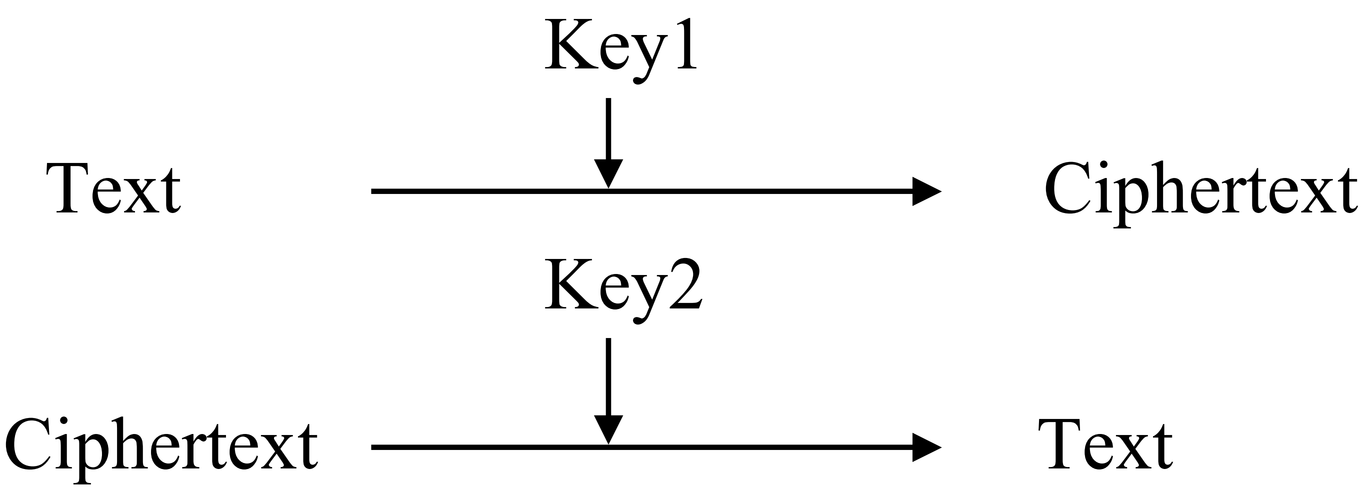
Public Key Encryption

1. Public Key Encryption
2. Modular Arithmetic
3. RSA Public Key Encryption
4. Confidentiality
5. Diffie-Hellman Key Agreement
6. Hash Functions: MD5, SHA-1
7. Message Authentication Code (MAC)
8. Digital Signature
9. Digital Certificates

Public Key Encryption

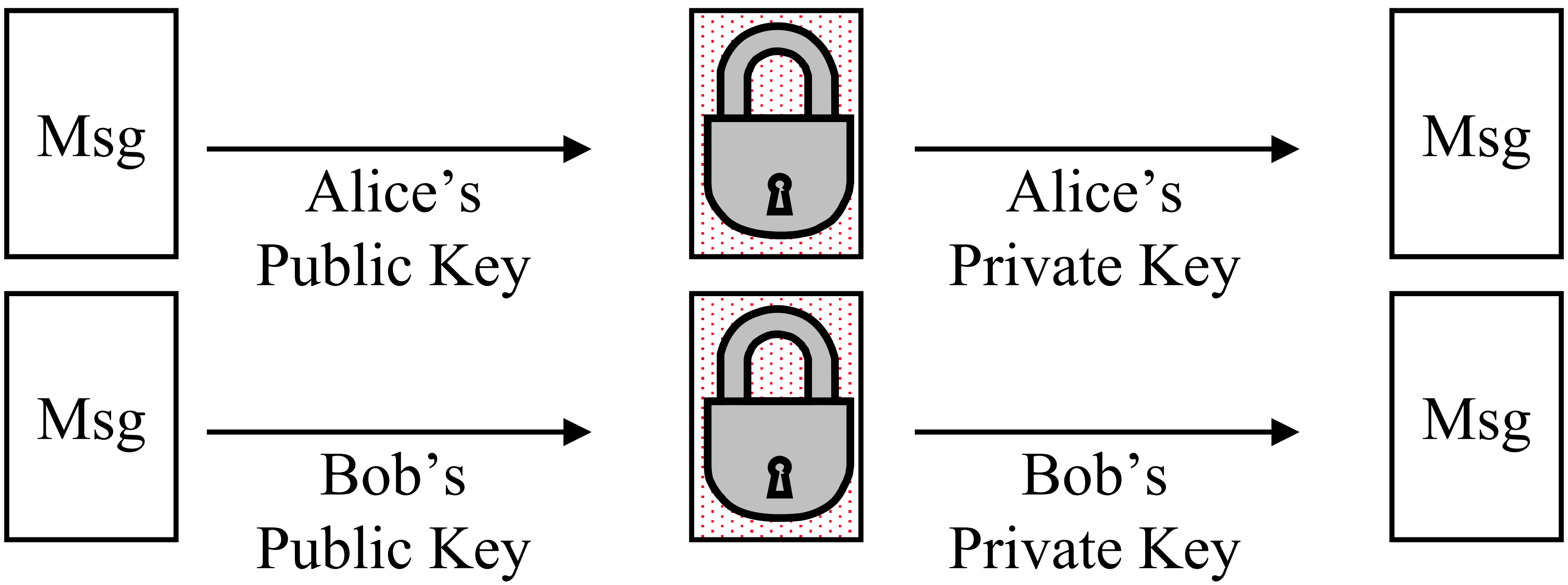


- Invented in 1975 by Diffie and Hellman
- Encrypted_Message = Encrypt(Key1, Message)
- Message = Decrypt(Key2, Encrypted_Message)



Public Key (Cont)

- One key is private and the other is public
- Message = Decrypt(Public_Key,
  Encrypt(Private_Key, Message))
- Message = Decrypt(Private_Key,
  Encrypt(Public_Key, Message))



Public Key Encryption Method

- RSA: Encrypted_Message = m^3 mod 187
- Message = Encrypted_Message^{107} mod 187
- Key1 = <3,187>, Key2 = <107,187>
- Message = 5
- Encrypted Message = $5^3 = 125$
- Message = 125^{107} mod 187 = 5
  $= 125^{(64+32+8+2+1)}$ mod 187
  $= \{(125^{64}$ mod 187$)(125^{32}$ mod 187$)\ldots$
  $(125^2$ mod 187$)(125$ mod 187$)\}$ mod 187

Modular Arithmetic

- $xy \bmod m = (x \bmod m)\,(y \bmod m) \bmod m$
- $x^4 \bmod m = (x^2 \bmod m)(x^2 \bmod m) \bmod m$
- $x^{ij} \bmod m = (x^i \bmod m)^j \bmod m$
- $125 \bmod 187 = 125$
- $125^2 \bmod 187 = 15625 \bmod 187 = 104$
- $125^4 \bmod 187 = (125^2 \bmod 187)^2 \bmod 187$
  $= 104^2 \bmod 187 = 10816 \bmod 187 = 157$
- $128^8 \bmod 187 = 157^2 \bmod 187 = 152$
- $128^{16} \bmod 187 = 152^2 \bmod 187 = 103$
- $128^{32} \bmod 187 = 103^2 \bmod 187 = 137$
- $128^{64} \bmod 187 = 137^2 \bmod 187 = 69$
- $128^{64+32+8+2+1} \bmod 187 = 69 \times 137 \times 152 \times 104 \times 125 \bmod 187$
  $= 18679128000 \bmod 187 = 5$

RSA Public Key Encryption

- Ron Rivest, Adi Shamir, and Len Adleman at MIT 1978
- Both plain text M and cipher text C are integers between 0 and n-1.
- Key 1 = {e, n},
  Key 2 = {d, n}
- $C = M^e \bmod n$
  $M = C^d \bmod n$
- How to construct keys:
  - Select two large primes: p, q, $p \neq q$
  - $n = p \times q$
  - Calculate $z = (p-1)(q-1)$
  - Select e, such that lcd(z, e) = 1; $0 < e < z$
  - Calculate d such that de mod z = 1

RSA Algorithm: Example

- Select two large primes: p, q, $p \neq q$
  $p = 17$, $q = 11$
- $n = p \times q = 17 \times 11 = 187$
- Calculate $z = (p-1)(q-1) = 16x10 = 160$
- Select e, such that $lcd(z, e) = 1$; $0 < e < z$
  say, $e = 7$
- Calculate d such that $de \bmod z = 1$
  - $160k+1 = 161, 321, 481, 641$
  - Check which of these is divisible by 7
  - 161 is divisible by 7 giving $d = 161/7 = 23$
- Key 1 = {7, 187}, Key 2 = {23, 187}

Homework 8B

Problem P8: Consider RSA with p=5, q=11

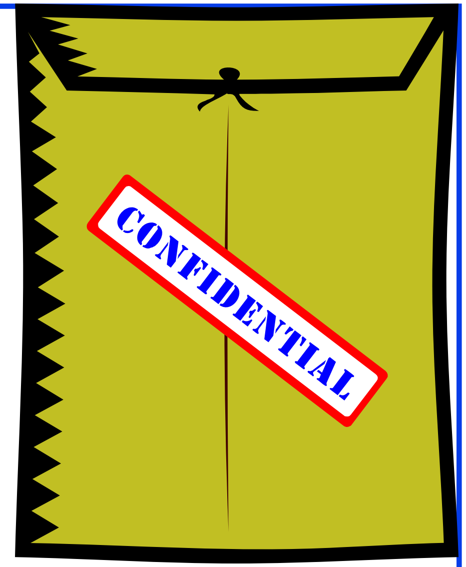
A. what are n and z

B. let e be 3. Why is this an acceptable choice for e?

C. Find d such that de=1(mod z) and d<160

D. Encrypt the message m=8 using the key (n,e). Let c be the corresponding cipher text. Show all work.

Confidentiality



- User 1 to User 2:
- Encrypted_Message
  = Encrypt(Public_Key2,
  Encrypt(Private_Key1, Message))
- Message = Decrypt(Public_Key1,
  Decrypt(Private_Key2, Encrypted_Message)
  ⇒ Authentic and Private

Your Public Key

My Private Key

Message

Diffie-Hellman Key Agreement

- Allows two party to agree on a secret key using a public channel
- A selects p=large prime, and g=a number less than p
- A selects a random # S_A, B selects another random # S_B

$$\begin{array}{ccc} \boxed{\begin{array}{c} S_A, g, p \\ T_A = g^{S_A} \bmod p \\ \\ T_{AB} = T_B{}^{S_A} \bmod p \end{array}} & \begin{array}{c} \xrightarrow{g,\ p,\ T_A} \\ \xleftarrow{T_B} \end{array} & \boxed{\begin{array}{c} S_B \\ T_B = g^{S_B} \bmod p \\ \\ T_{AB} = T_A{}^{S_B} \bmod p \end{array}} \end{array}$$

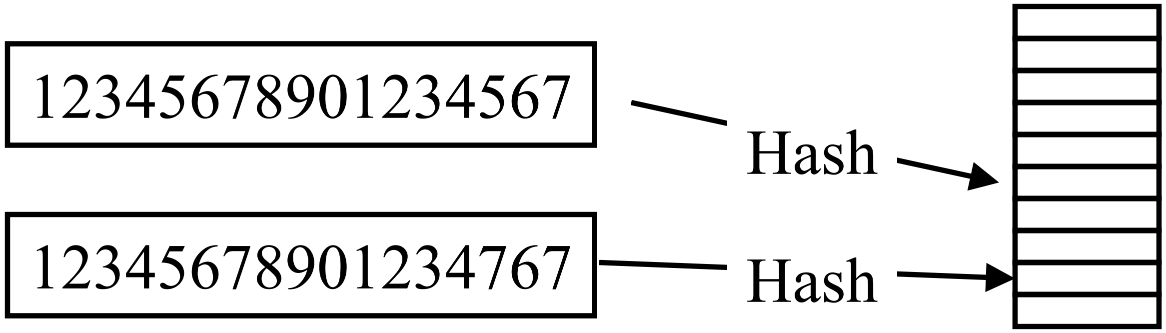
$$T_{AB} = g^{S_A\, S_B} \bmod p$$

- Eavesdropper can see T_A, g, p but cannot compute S_A
- Computing S_A requires discrete logarithm - a difficult problem

Diffie-Hellman (Cont)

- Example: g=5, p=19
  - A selects 6 and sends $5^6 \bmod 19 = 7$
  - B selects 7 and sends $5^7 \bmod 19 = 16$
  - A computes $K = 16^6 \bmod 19 = 7$
  - B computes $K = 7^7 \bmod 19 = 7$
- Preferably (p-1)/2 should also be a prime.
- Such primes are called safe prime.

Hash Functions



Example: CRC can be used as a hash
(not recommended for security applications)

Requirements:

1. Applicable to any size message
2. Fixed length output
3. Easy to compute
4. Difficult to Invert ⇒ Can't find *x* given H(*x*) ⇒ One-way
5. Difficult to find y, such that H(*x*) = H(*y*) ⇒ Can't change msg
6. Difficult to find *any* pair (*x*, *y*) such that H(*x*) = H(*y*)
   ⇒ Strong hash

MD5 Hash

- 128-bit hash using 512 bit blocks using 32-bit operations
- Invented by Ron Rivest in 1991
- Described in RFC 1321
- Commonly used to check the integrity of files (easy to fudge message and the checksum)
- Also used to store passwords

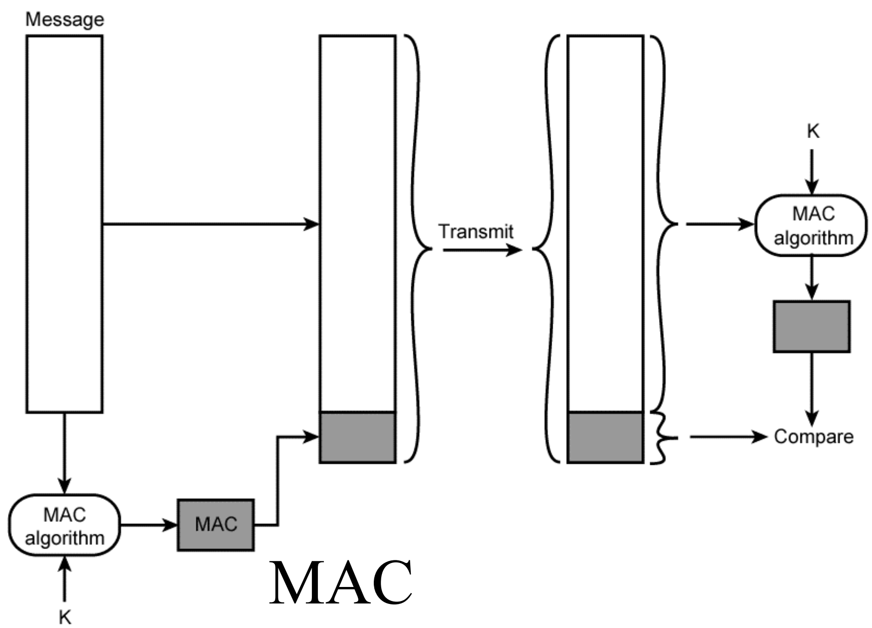
SHA-1 Algorithm

- 160 bit hash using 512 bit blocks and 32 bit operations
- Five passes (4 in MD5 and 3 in MD4)
- Maximum message size is 2^{64} bit

Message Authentication Code (MAC)

- Authentic Message = Contents unchanged + Source Verified
- May also want to ensure that the time of the message is correct
- Encrypt({Message, CRC, Time Stamp}, Source's secret key)
- Message + Encrypt(Hash, Source's secret key)
- Message + Encrypt(Hash, Source's private key)

Message



MAC

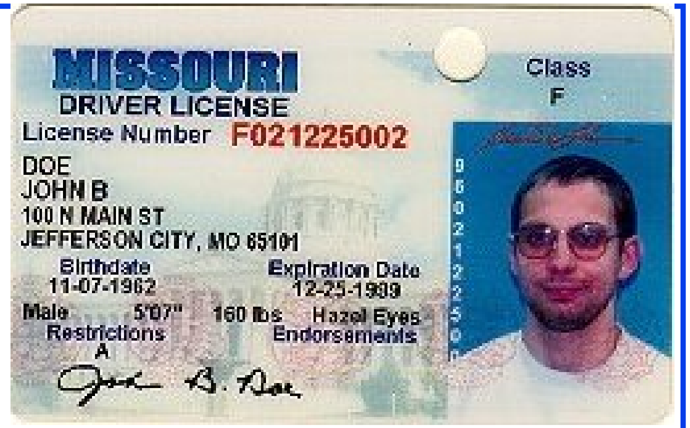
Digital Signature

- Message Digest = Hash(Message)
- Signature = Encrypt(Private_Key, Hash)
- Hash(Message) = Decrypt(Public_Key, Signature) $\Rightarrow$ Authentic
- Also known as Message *authentication* code (MAC)

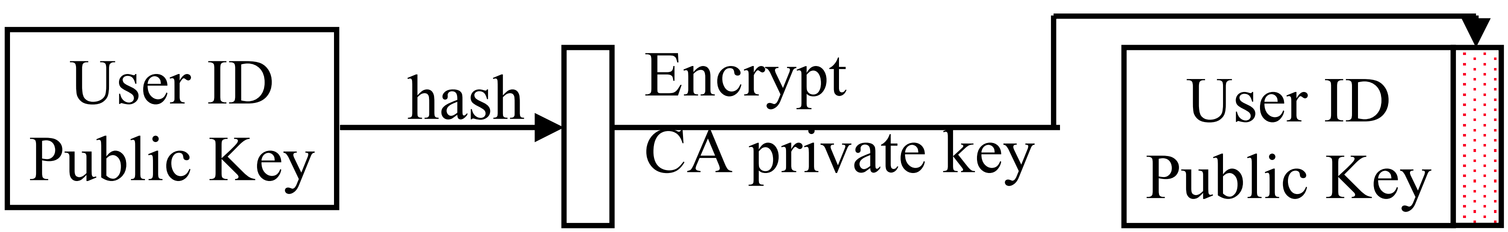
$$\text{Text} \xrightarrow{\text{Hash}} \text{Digest} \xrightarrow{\text{Private Key}} \text{Signature}$$

$$\text{Signature} \xrightarrow{\text{Public Key}} \text{Digest} \xleftarrow{\text{Hash}} \text{Text}$$

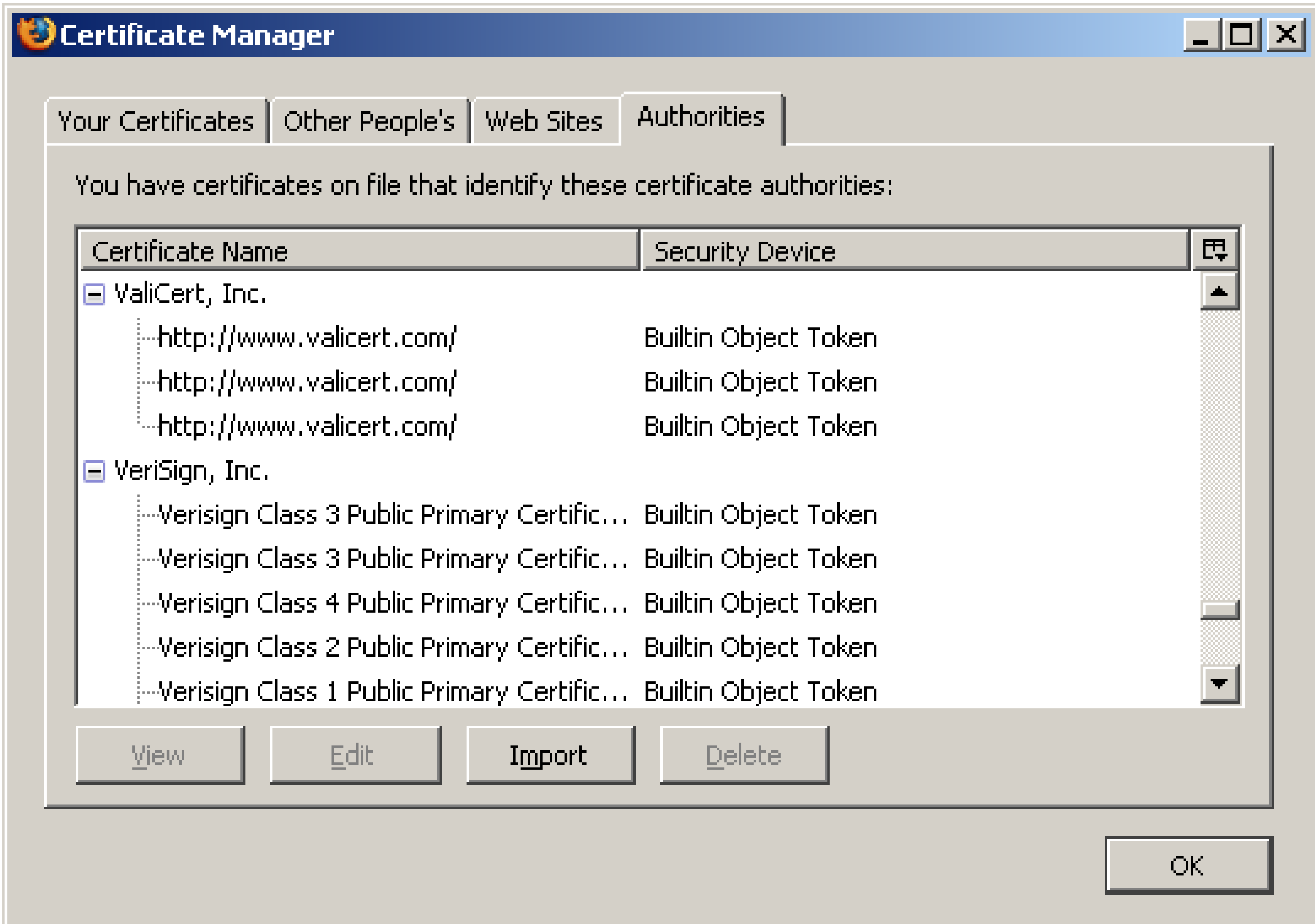
Digital Certificates



- Like driver license or passport
- Digitally signed by Certificate authority (CA) - a trusted organization
- Public keys are distributed with certificates
- CA uses its private key to sign the certificate
  ⇒ Hierarchy of trusted authorities
- X.509 Certificate includes: Name, organization, effective date, expiration date, public key, issuer's CA name, Issuer's CA signature

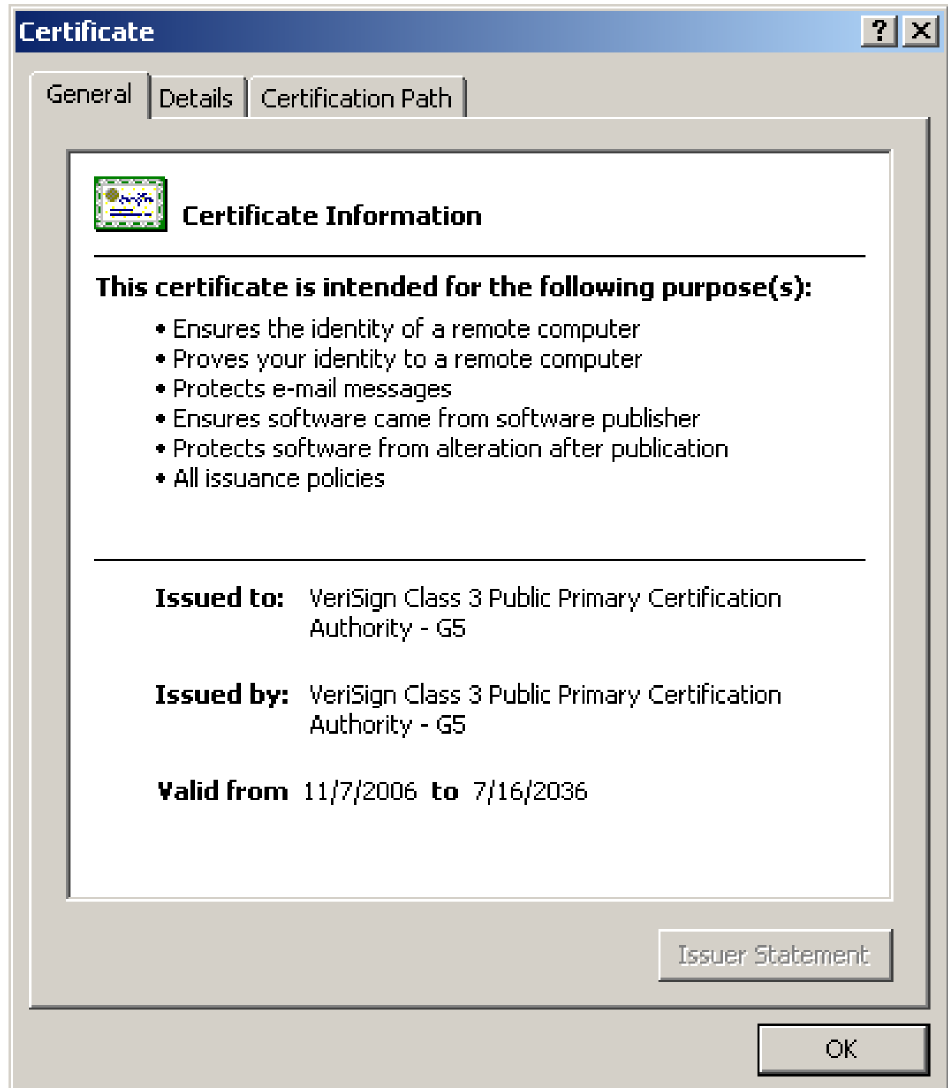


Oligarchy Example



Sample X.509 Certificate

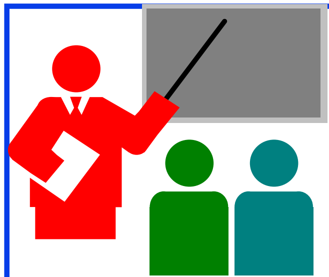
Internet Explorer



X.509 Sample (Cont)

Field	Value
Version	V3
Serial number	18 da d1 9e 26 7d e8 bb 4a 21…
Signature algorithm	sha1RSA
Issuer	VeriSign Class 3 Public Primary …
Valid from	Tuesday, November 07, 2006 …
Valid to	Wednesday, July 16, 2036 6:…
Subject	VeriSign Class 3 Public Primary …
Public key	RSA (2048 Bits)

Version	V3
Serial number	18 da d1 9e 26 7d e8 bb 4a 21…
Signature algorithm	sha1RSA
Issuer	VeriSign Class 3 Public Primary …
Valid from	Tuesday, November 07, 2006 …
Valid to	Wednesday, July 16, 2036 6:…
Subject	VeriSign Class 3 Public Primary …
Public key	RSA (2048 Bits)



Public Key Encryption: Review

1. Public Key Encryption uses two keys: Public and Private
2. RSA method is based on difficulty of factorization
3. Diffie-Hellman Key Agreement allows agreeing on a shared secret in public
4. Hashes are one-way functions such that it difficult to find another input with the same hash like MD5, SHA-1
5. Message Authentication Code (MAC) ensures message integrity and source authentication using hash functions
   Digital Signature consists of encrypting the hash of a message using private key
6. Digital certificates are signed by root certification authorities and contain public keys

Review Exercises

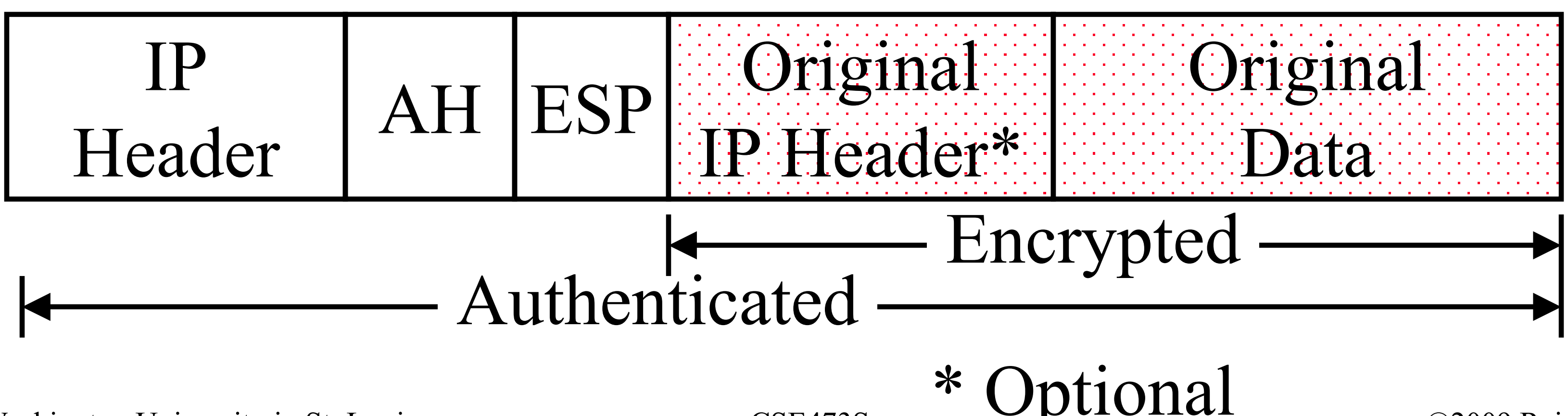
- Try but do not submit
- Review exercises:R7, R9, R10, R11, R12, R13, R14, R15
- Problems: P7, P9, P10, P11

IPSec, VPN, Firewalls

1. IPSec
   - Tunnel vs. Transport Mode
   - Authentication Header
   - Encapsulating Security Payload (ESP)
2. Virtual Private Networks
3. Firewalls
4. Application Gateways: Proxy Servers
5. Intrusion Detection Systems

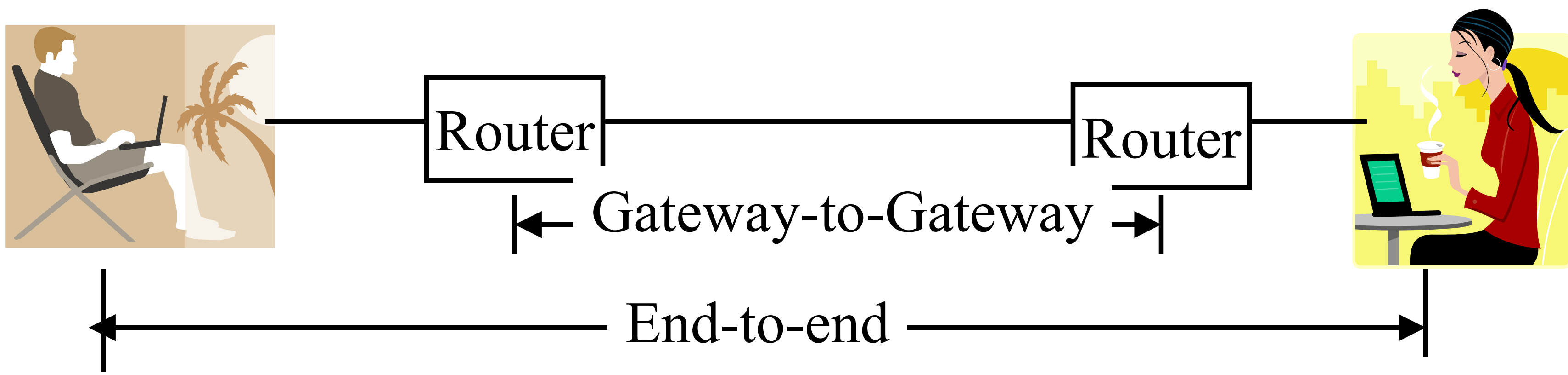
IPSec

- Secure IP: A series of proposals from IETF
- Separate Authentication and privacy
- Authentication Header (AH) ensures data *integrity* and *data origin authentication*
- Encapsulating Security Protocol (ESP) ensures *confidentiality, data origin authentication, connectionless integrity, and anti-replay service*

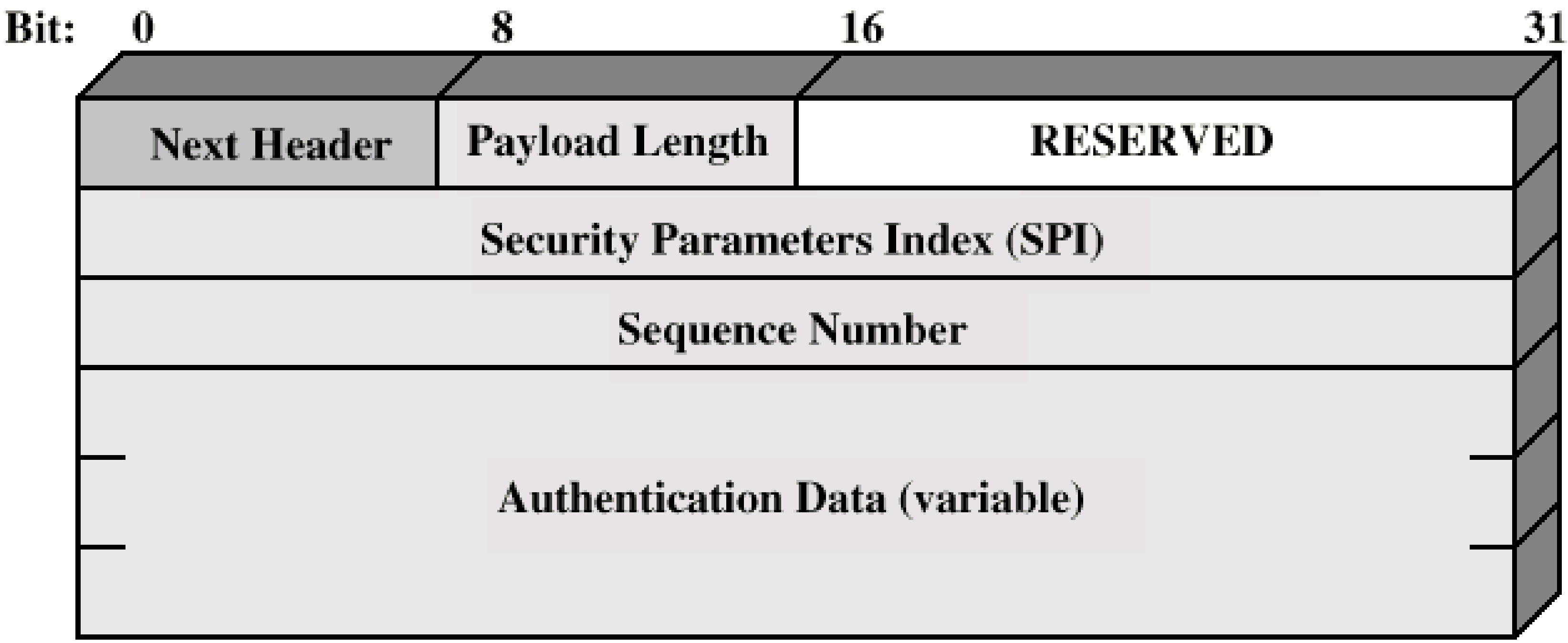


Tunnel vs. Transport Mode

- Gateway-to-gateway vs. end-to-end

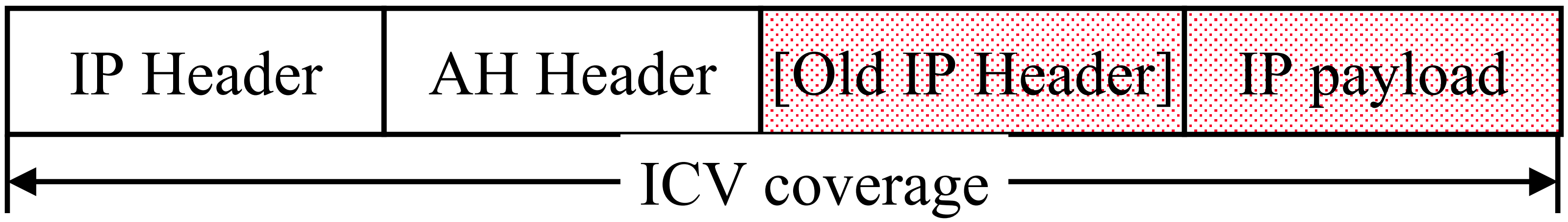


Authentication Header



- Next Header = TCP=6, UDP=17, IP=4, AH=51
  ⇒ Designed by IPv6 fans
- Payload Length = Length of ***AH*** in 32-bit words − 2 (for IPv4)
  =Length of AH in 64-bit words -1 (for IPv6)
- SPI = Identifies Security association (0=Local use, 1-255 reserved)
- Authentication data = Integrity Check Value

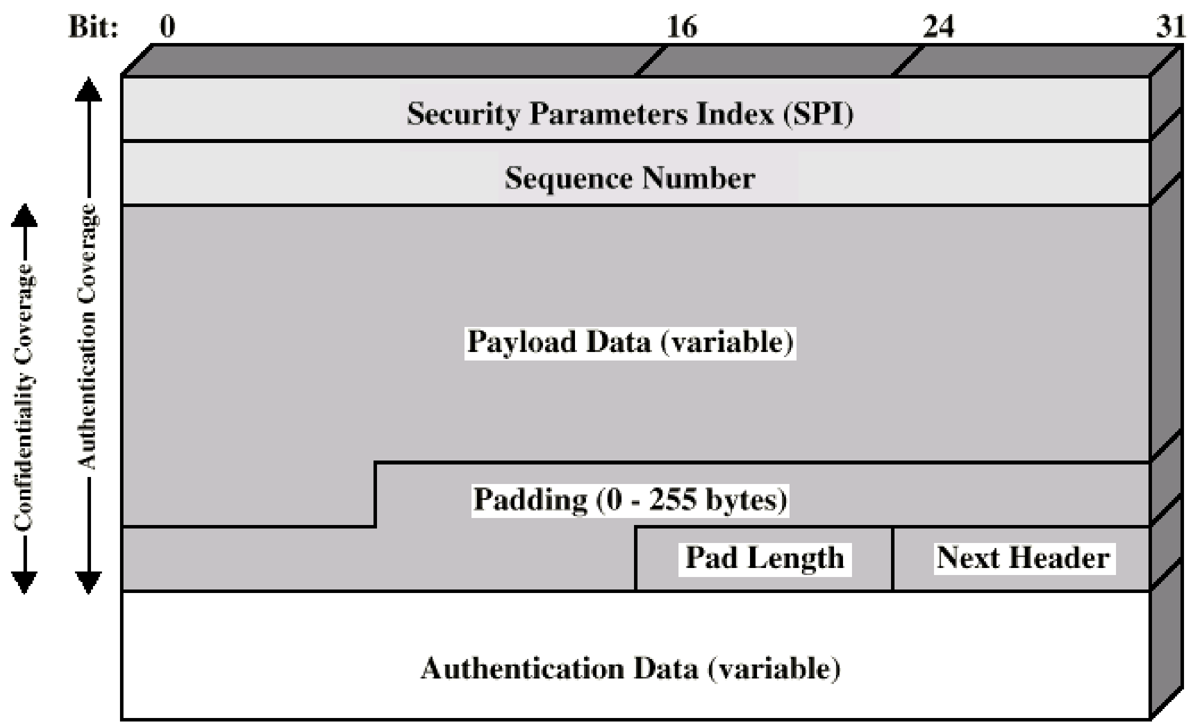
AH ICV Computation



The AH ICV is computed over:

- IP header fields that are either *immutable* in transit or that are *predictable* in value upon arrival at the endpoint for the AH SA, e.g., source address (immutable), destination address with source routing (mutable but predictable)
- The AH header (Next Header, Payload Len, Reserved, SPI, Sequence Number, and the Authentication Data (which is set to zero for this computation), and explicit padding bytes (if any))
- The upper level protocol data, which is assumed to be immutable in transit

ESP Packet



- Payload data: IP, TCP, UDP packet
- Pad Length in bytes
- Next Header: Type of payload (TCP, UDP, …)
- Authentication Data: Integrity Check Value over ESP packet

Encapsulating Security Payload (ESP)

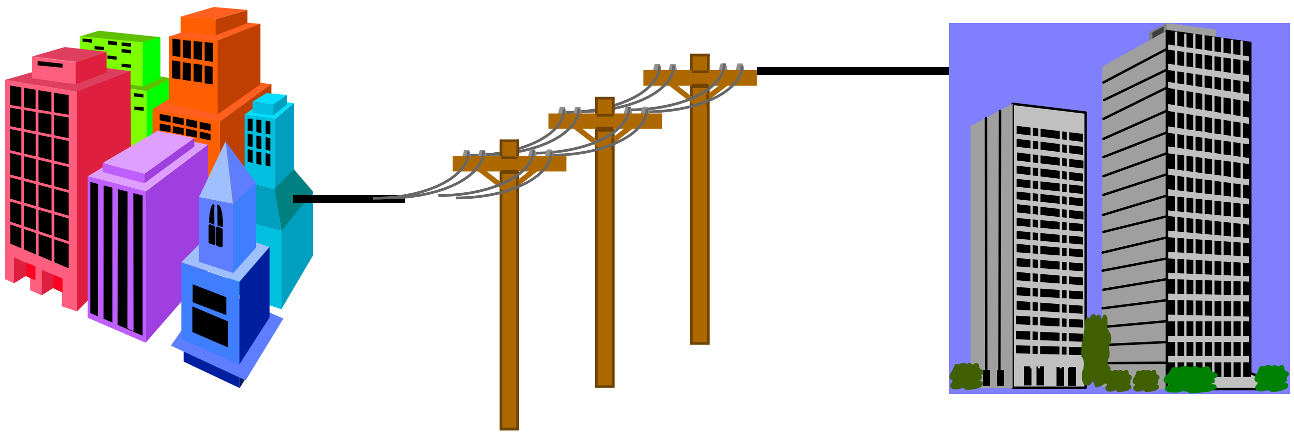
- Provides encryption and/or integrity
  $\Rightarrow$ Confidentiality=ESP, Integrity=AH or ESP, Confidentiality+Integrity=ESP, ESP+AH
- Null encryption algorithm $\Rightarrow$ No confidentiality
- IV and authentication data sizes available from SA database

Homework 8C

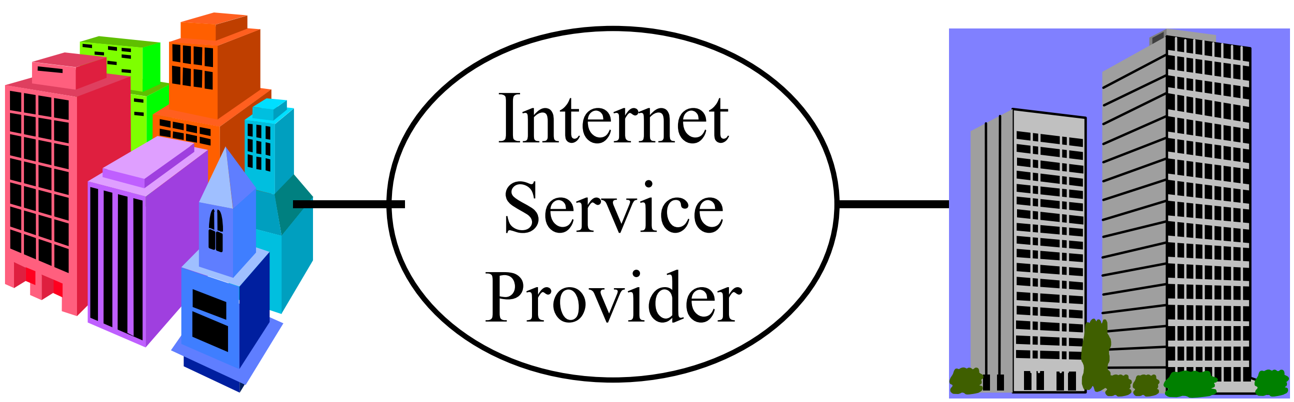
For each of the fields in IPv4 header, indicate whether the field is immutable, mutable but predictable, or mutable (zeroed prior to ICV calculation).

What is a VPN?

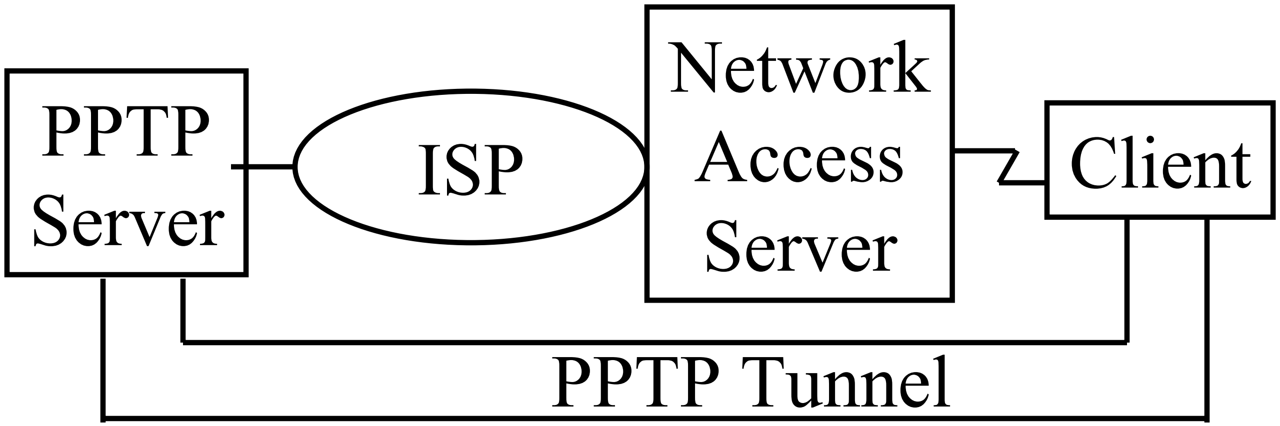
- Private Network: Uses leased lines



- *Virtual* Private Network: Uses public Internet

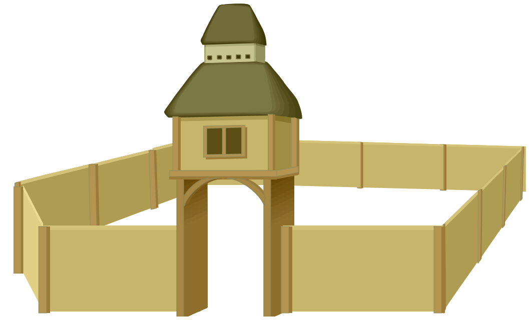


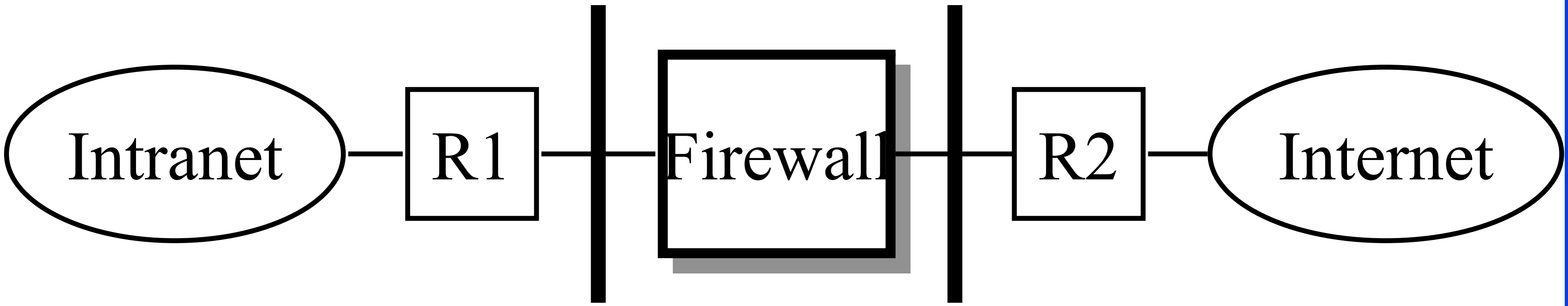
PPTP



- PPTP = Point-to-point Tunneling Protocol
- Developed jointly by Microsoft, Ascend, USR, 3Com and ECI Telematics
- PPTP server for NT4 and clients for NT/95/98

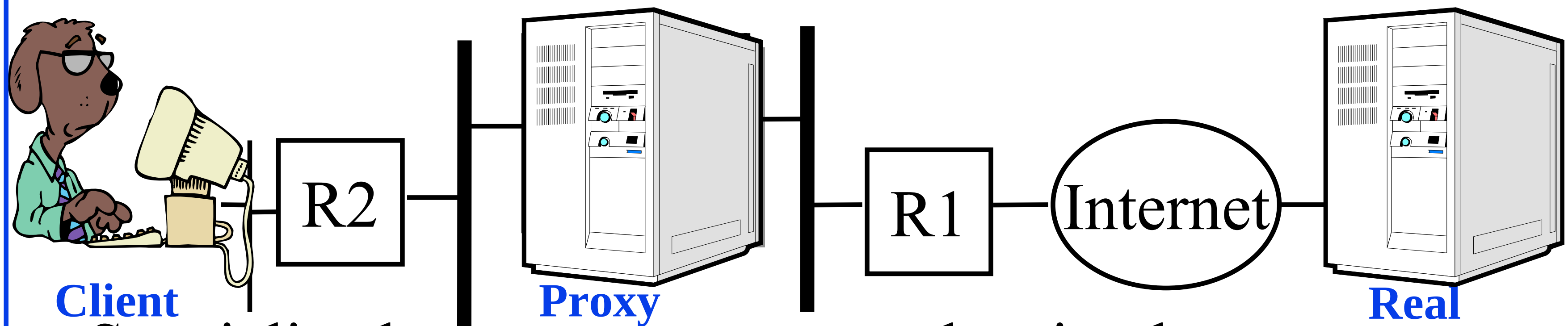
Firewall





- Enforce rules on what internal hosts/applications can be accessed from outside and vice versa
- One point of entry. Easier to manage security.
- Discard based on IP+TCP header. Mainly port #.
- Firewall-Friendly applications: Use port 80.

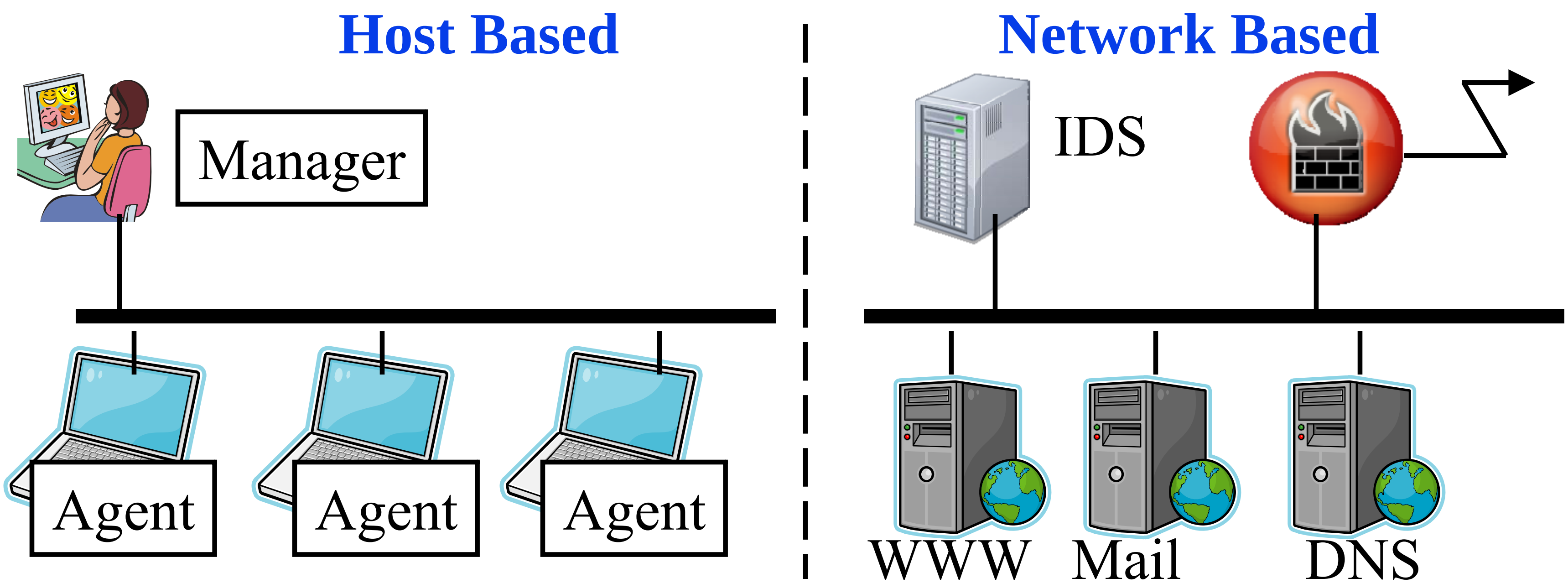
Application Gateways: Proxy Servers



- Specialized server programs on bastion host
- Take user's request and forward them to real servers
- Take server's responses and forward them to users
- Enforce site security policy ⇒ Refuse some requests.
- Also known as application-level gateways
- With special "Proxy client" programs, proxy servers are almost transparent

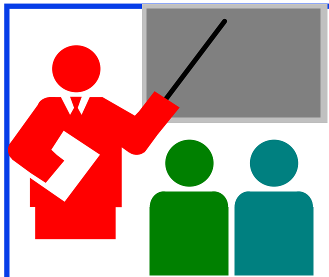
Types of IDS

- IDS Sensor: SW/HW to collect and analyze network traffic
- Host IDS: Runs on each server or host
- Network IDS: Monitors traffic on the network
  Network IDS may be part of routers or firewalls



Signature Based IDS

- 5-tuple packet filtering (SA/DA/L4 protocol/ports)
- Use Ternary Content Addressable Memories (TCAMs)
- Deep packet inspection requires pattern string matching algorithms (Aho-Corasik algorithm and enhancements)
- Regular expression signatures



IPSec, VPN, Firewalls: Review

1. IPSec has two modes: end-to-end (Transport mode) or router-to-router (tunnel mode)
2. Authentication Header (AH) ensures data integrity and data origin authentication
3. Encapsulating Security Protocol (ESP) ensures confidentiality, data origin authentication, connectionless integrity, and anti-replay service
4. Virtual Private Networks provide encryption over public networks
5. Firewalls filter traffic based on port numbers
6. Proxy Servers provide application specific protection
7. Intrusion Detection Systems inspect incoming traffic for specific attack signatures

Review Exercises

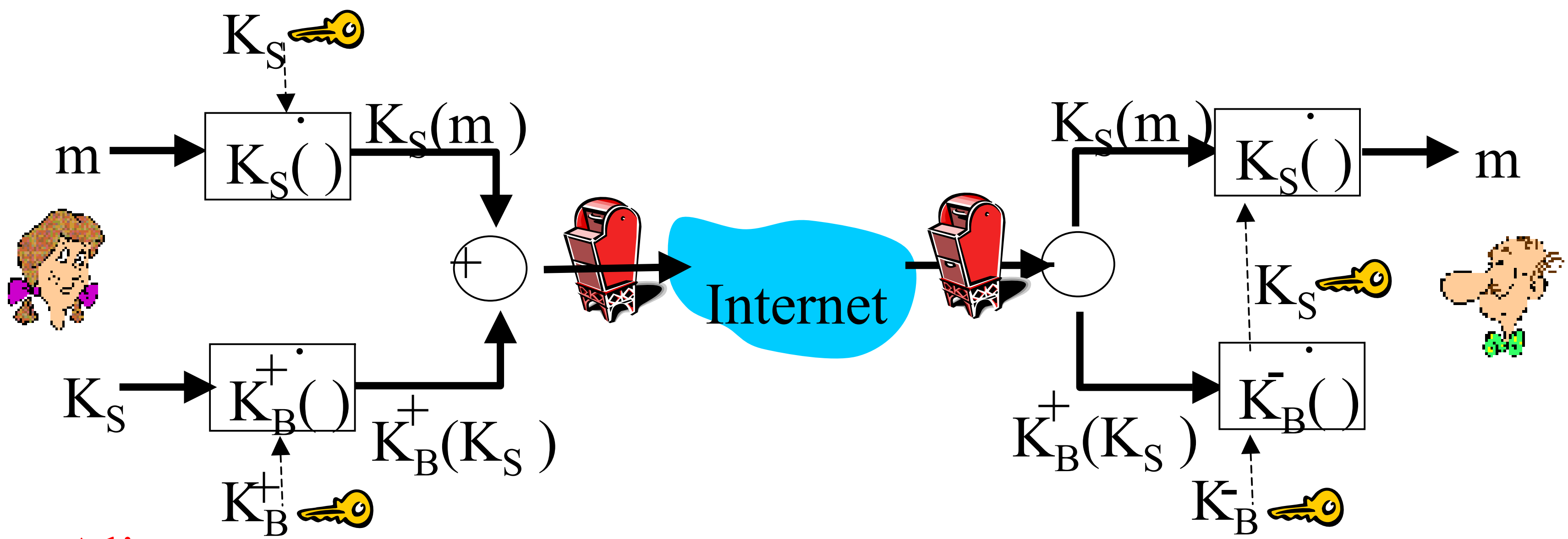
- Try but do not submit
- Review Questions: R24, R25, R29, R30, R33

Secure Email, SSL, IKE,WEP

- Secure E-Mail
- Pretty Good Privacy (PGP)
- SSL
- Internet Key Exchange (IKE)
- Wired Equivalent Privacy (WEP)

Secure E-Mail

- Alice wants to send confidential e-mail, m, to Bob.



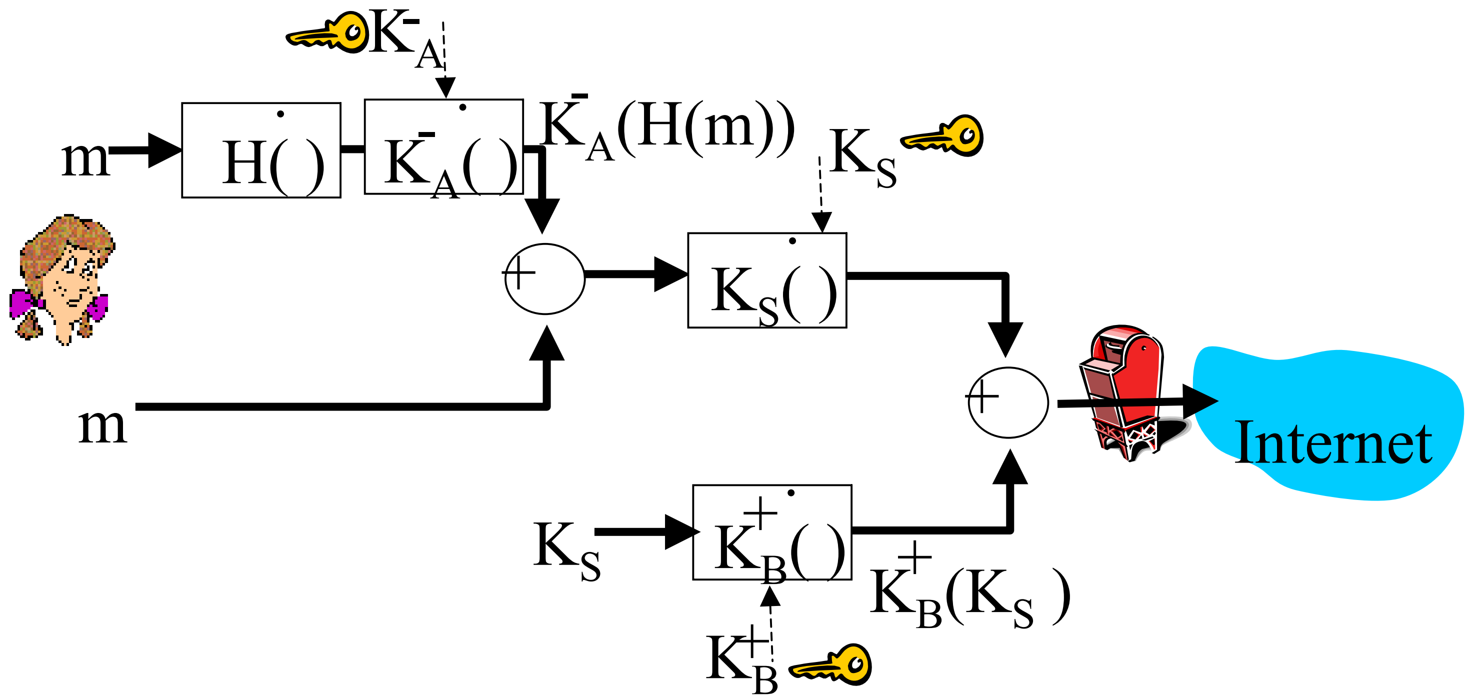
Alice:

- Generates random *secret* key, K_S.
- Encrypts message with K_S (for efficiency)
- Also encrypts K_S with Bob's public key.
- Sends both $K_S(m)$ and $K_B(K_S)$ to Bob.

- Bob uses his private key to recover K_s

Secure E-Mail (Cont)

- Alice wants to provide secrecy, sender authentication, message integrity.



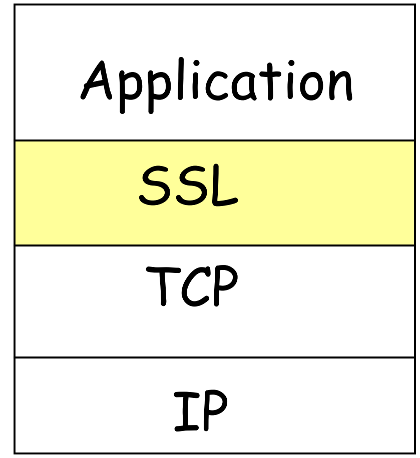
- Alice uses three keys: her private key, Bob's public key, newly created secret key

Pretty Good Privacy (PGP)

- Used RSA and IDEA (RSA patent in US until 2000)
- V2.6.2 became legal for use within US and can be downloaded from MIT
- A patent-free version using public algorithm has also been developed
- Code published as an OCRable book
- Initially used web of trust- certificates issued by people
- Certificates can be registered on public sites, e.g., MIT
- hushmail.com is an example of PGP mail service
- OpenPGP standard [RFC 4880]
- GNU Privacy Guard, an alternative to PGP, follows OpenPGP
- Ref: Wikipedia, http://en.wikipedia.org/wiki/Pretty_Good_Privacy

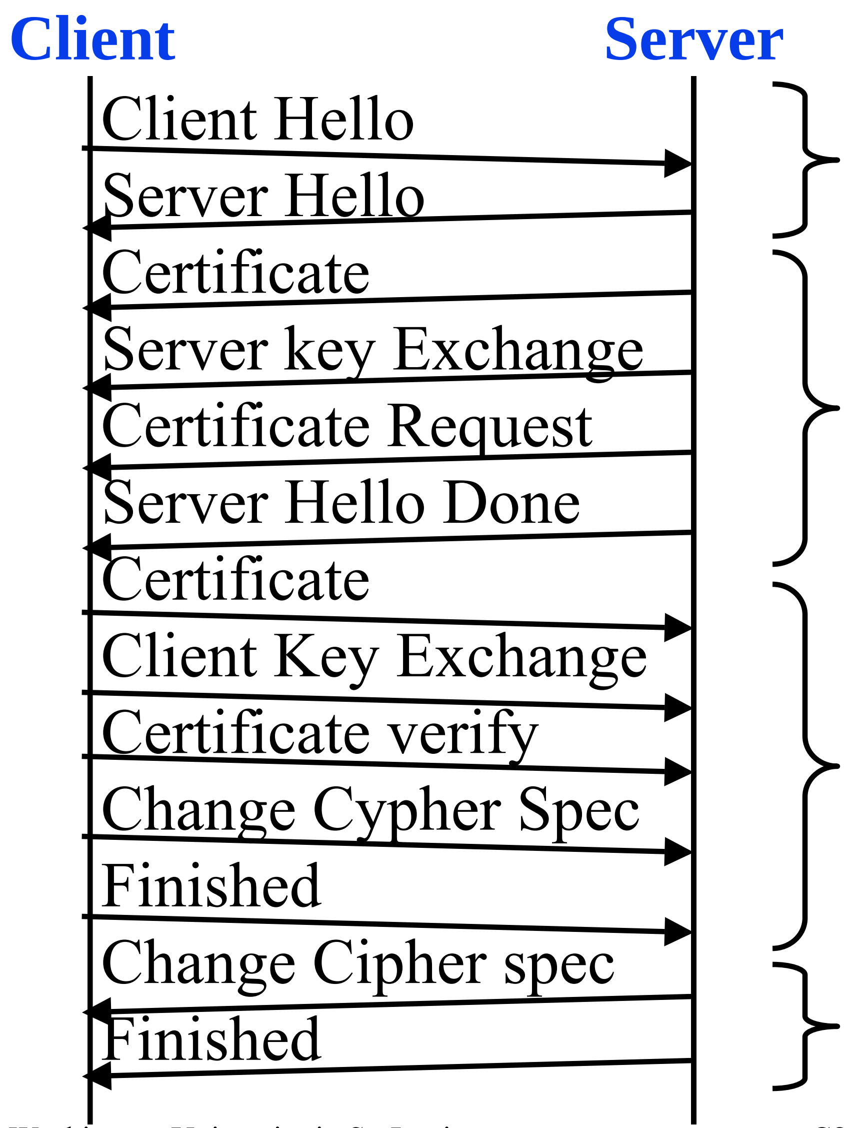
SSL

- Secure Socket Layer (SSL)
  Reliable end-to-end secure service over TCP
- Transport Layer Security (TLS) [RFC 5246]
- Embedded in specific packages,
  E.g., Netscape and Microsoft Explorer and most Web servers
- Session = Multiple end-to-end TCP connections
- Four Protocols:
  - Handshake protocol: Negotiate security parameters
  - Record protocol: Provide end-to-end encryption
  - Change cipher spec protocol: Updates cipher suite
  - Alert protocol: Warnings and fatal errors to peer



Application with SSL

Handshake Protocol



Phase 1: Exchange Protocol version, session ID, cipher suite, compression method and initial random numbers

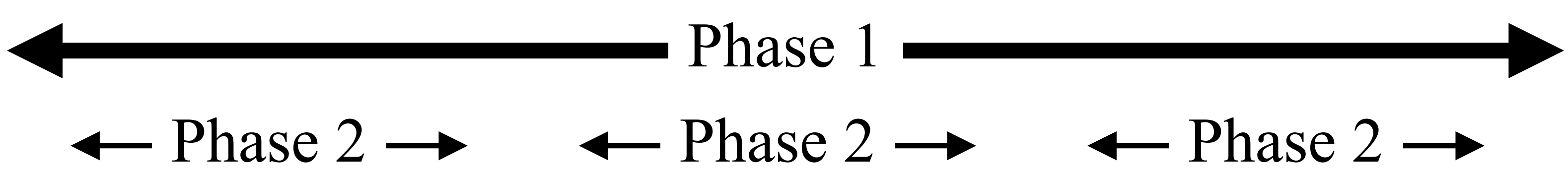
Phase 2: Certificate

Phase 3: Certificate

Phase 4: Change to new parameters

IKE Phases

- Crypto negotiation for IPsec
- Two phases
  - Phase 1: Mutual authentication and session keys = IKE SA
  - Phase 2: Use results of phase 1 to create multiple associations between the same entities = ESP or AH SA
- IKE SA is bi-directional
- AH and ESP SAs are unidirectional



IKE Modes and Authentication Methods

- **IKE Main Mode**: Allows ability to *hide end-point identifiers* and to select crypto algorithms ⇒ requires 6 messages
- **IKE Aggressive Mode**: End-points ID not hidden ⇒ Requires only three messages
- **IKE Authentication Methods**
  1. Original Public Key Encryption (separately encrypt each field with other sides public key)
  2. Revised Public Key Encryption (Encrypt session key with public key. Use session key to encrypt the rest)
  3. Public key signature
  4. Pre-shared secret key

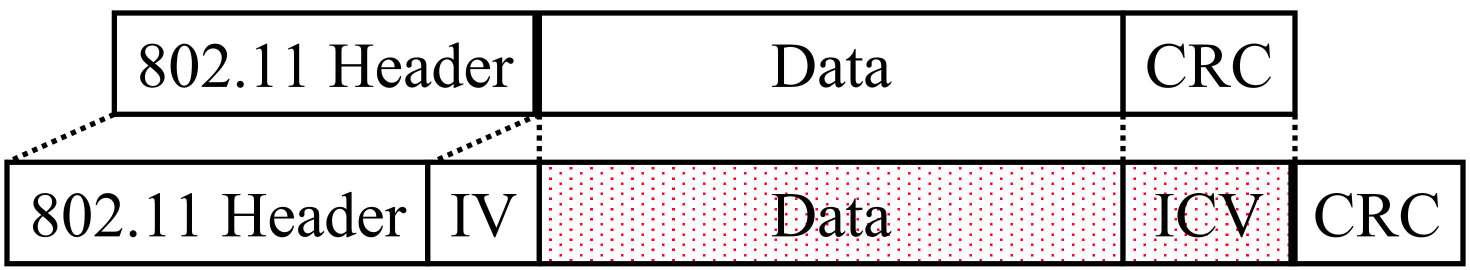
4 Methods × 2 Modes = 8 variants of Phase 1

Wired Equivalent Privacy (WEP)

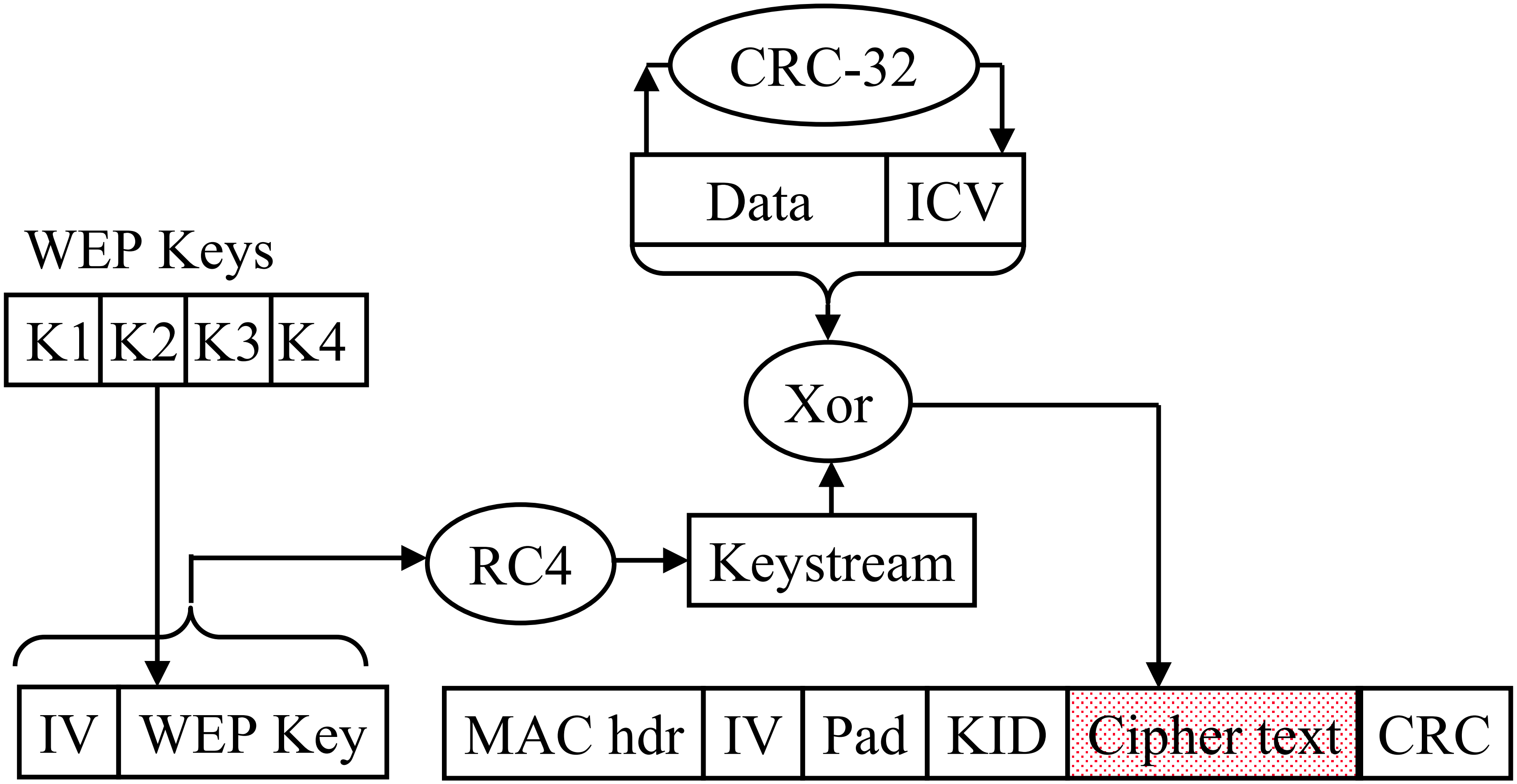
- WEP ⇒ Privacy similar to a wired network
  ⇒ Intellectual property not exposed to casual browser
  ⇒ Not protect from hacker
- First encryption standard for wireless. Defined in 802.11b
- Provides authentication and encryption
- Shared Key Authentication
  ⇒ Single key is shared by all users and access points
- Manual key distribution
- If an adapter or AP is lost, all devices must be re-keyed

WEP Details

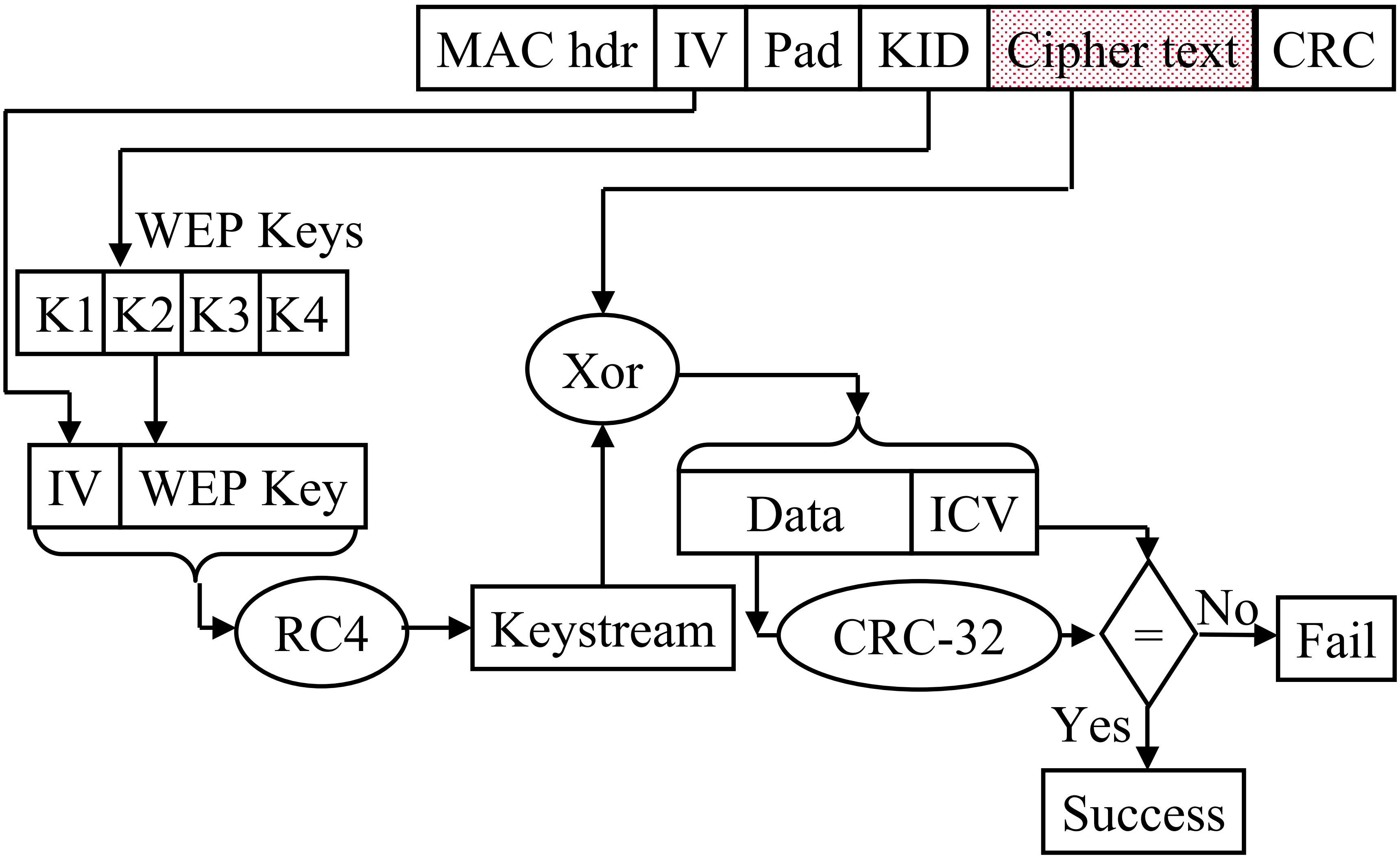
- Each device has 4 static WEP keys
- 2-bit key ID sent w Initialization Vector (IV) in clear in each packet
- Per-Packet encryption key =24-bit IV + one of pre-shared key
- Encryption Algorithm: RC4
  - Standard: 24 + 40 = 64-bit RC4 Key
  - Enhanced: 24 + 104 = 128 bit RC4 key
- WEP allows IV to be reused
- CRC-32 = Integrity Check Value (ICV)
- Data and ICV are encrypted under per-packet encryption key



WEP Encapsulation

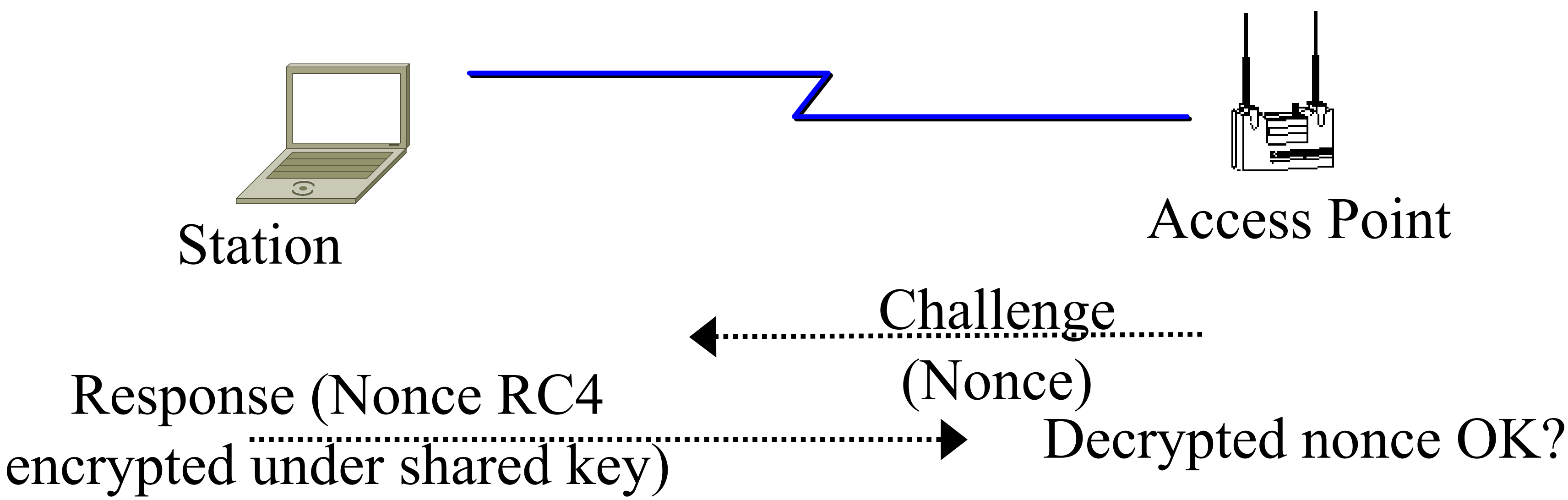


WEP Decapsulation



WEP Authentication

- Authentication is a via Challenge response using RC4 with the shared secret key.

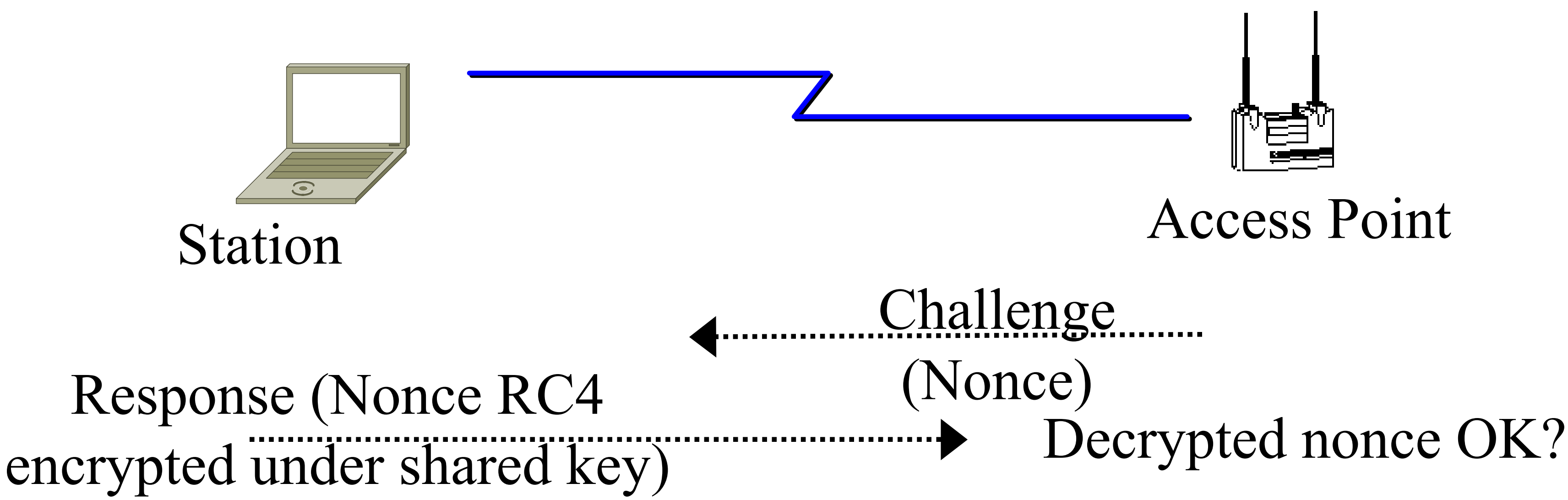


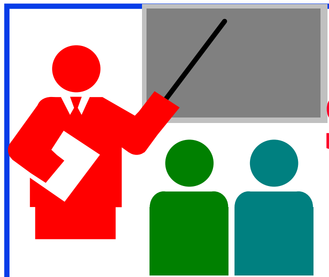
WEP Review

- Four 40-bit or 104-bit Keys are manually programmed in each subscriber station and AP
- A 24-bit IV and WEP key is used to form a 64b or 128b RC4 key
- A keystream is generated using the RC4 key
- A 32-bit CRC is added as "Integrity check value" (ICV) to the packet
- Plain text and keystream is xor'ed. A 32-bit CRC is added in clear.

Problems with WEP Authentication

- Record one challenge/response
- Both plain text and encrypted text are available to attacker
- XOR the two to get the keystream
- Use that keystream and IV to encrypt any subsequent challenges





Secure Email, SSL, IKE,WEP: Review

- Secure E-Mail requires using certificates to
- Pretty Good Privacy (PGP) uses
- SSL is TCP layer security and allows authentication, crypto negotiation, and key generation
- Internet Key Exchange (IKE) allows stations to negotiate encryption methods and generate keys for two phases
- If IV is reused, RC4 uses the same pad and encryption is defeated

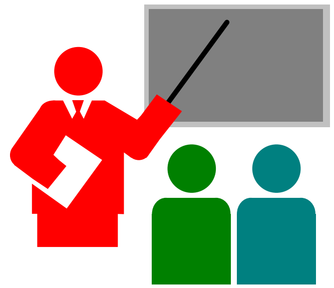
Review Exercises

- Try but do not submit
- Review Questions: R22, R23, R26, R27, R28,
- Problems: P10, P20, P21, P23

Homework 8D

- Submit answer to problem P24: Pseudo-WEP

Summary



- Network security requires confidentiality, integrity, availability, authentication, and non-repudiation
- Encryption can use one secret key or two keys (public and private)
- Public key is very compute intensive and is generally used to send secret key
- Digital certificate system is used to certify the public key
- IPSec with IKE provides integrity, data origin authentication, confidentiality, and anti-replay
- SSL provides security at transport layer
- WEP used initially in IEEE 802.11 was very weak.