

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/cse473-19/>



1. Secret Key Encryption
2. Public Key Encryption
3. Hash Functions, Digital Signature, Digital Certificates
4. Secure Email

Not Covered:, SSL, IKE, WEP, IPSec, VPN, Firewalls, Intrusion Detection

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 \Rightarrow 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

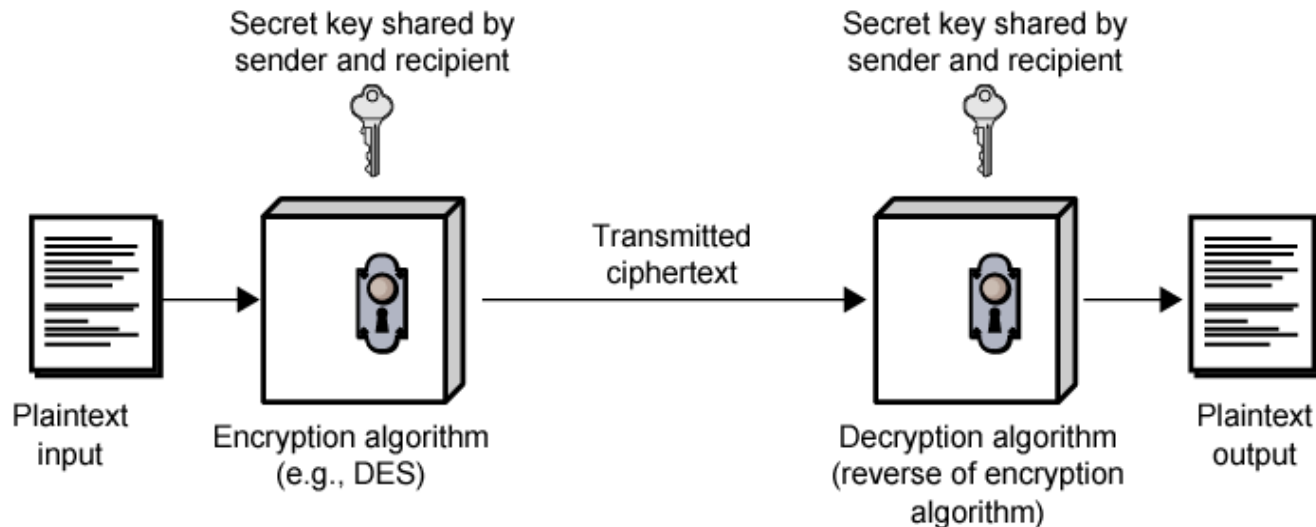
Secret Key Encryption: Overview

1. Concept: Secret Key Encryption
2. Method: Block Encryption
3. Improvement: Cipher Block Chaining (CBC)
4. Standards: DES, 3DES, AES



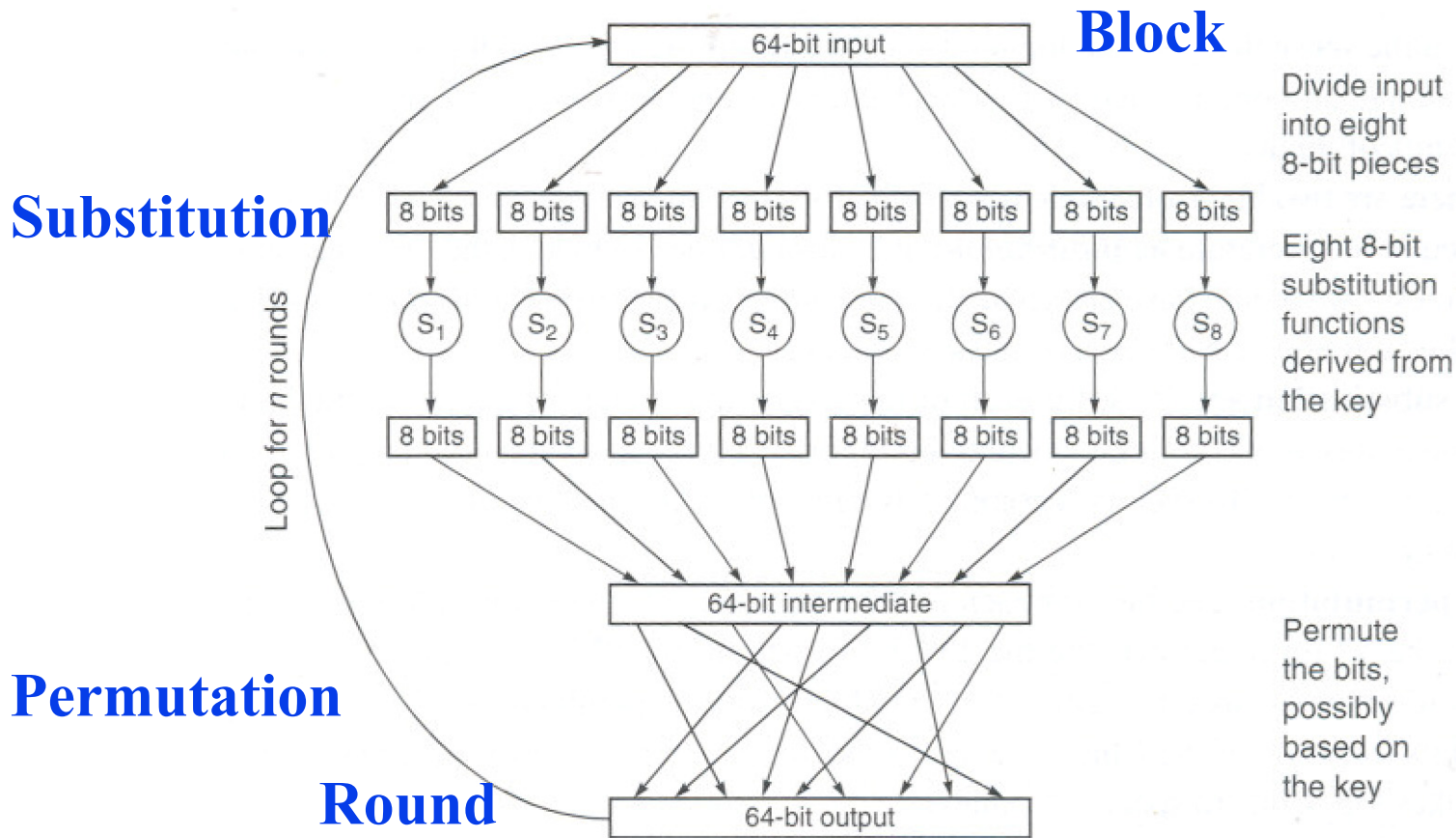
Secret Key Encryption

- ❑ 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)



Block Encryption

Block Encryption

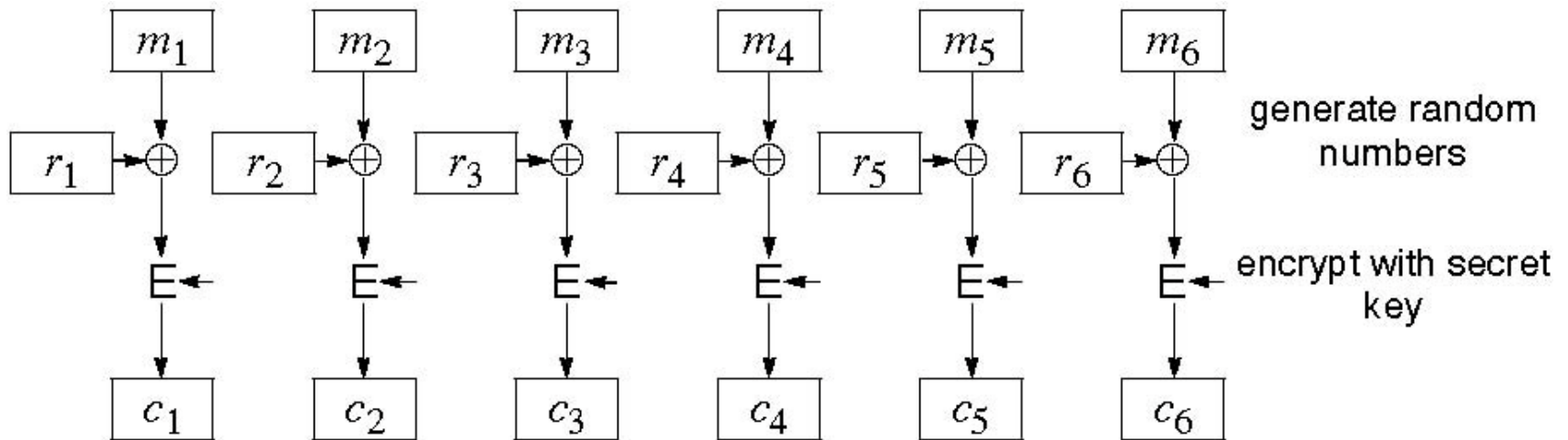


Block Encryption (Cont)

- ❑ 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.
 $1 \rightarrow 13, 2 \rightarrow 61, \text{ etc.}$
- ❑ Round: Substitution round followed by permutation round and so on. Diffusion + Confusion.

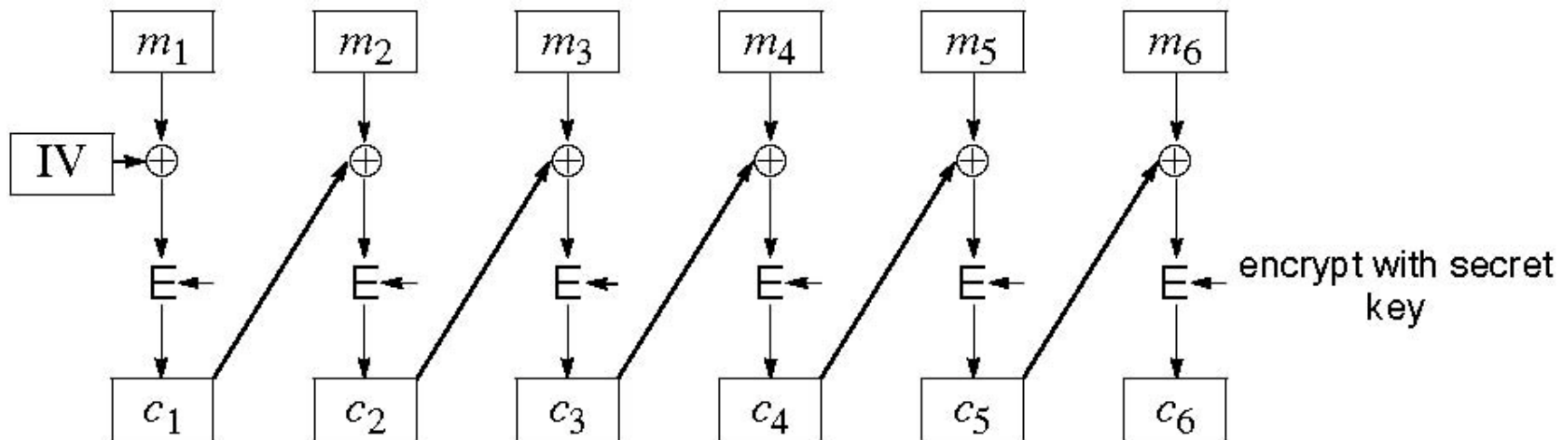
Cipher Block Chaining (CBC)

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



CBC (Cont)

- Use C_i as random number for $i+1$



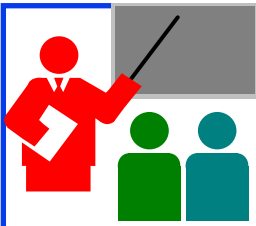
- Need Initial Value (IV)
- no IV \Rightarrow Same output for same message
 \Rightarrow one can guess changed blocks
- Example: Continue Holding, Start Bombing

Data Encryption Standard (DES)

- ❑ Published by NIST in 1977
- ❑ For commercial and *unclassified* government applications
- ❑ 8 octet (64 bit) key.
Each octet with 1 odd parity bit \Rightarrow 56-bit key
- ❑ Efficient hardware implementation
- ❑ Used in most financial transactions
- ❑ Computing power goes up 1 bit every 2 years
- ❑ 56-bit was secure in 1977 but is not secure today
- ❑ Now we use DES three times \Rightarrow Triple DES = 3DES

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



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.

Homework 8A

- [6 points] Consider 3-bit block cipher in the Table below

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

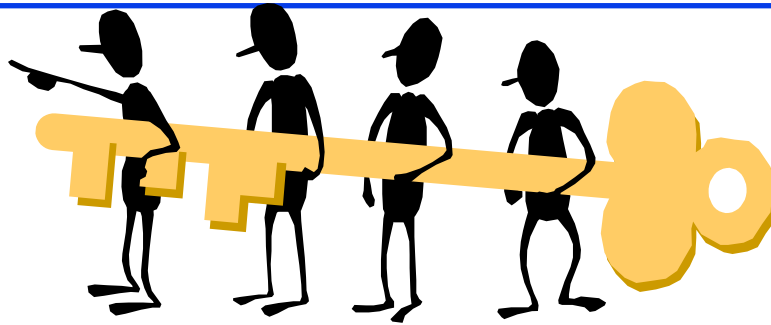
- Suppose the plaintext is 100100100.
 - Initially assume that CBC is not used. What is the resulting ciphertext?
 - Suppose Trudy sniffs the cipher text. Assuming she knows that a 3-bit block cipher without CBC is being employed (but doesn't know the specific cipher), what can she surmise?
 - Now suppose that CBC is used with IV-111. What is the resulting ciphertext?



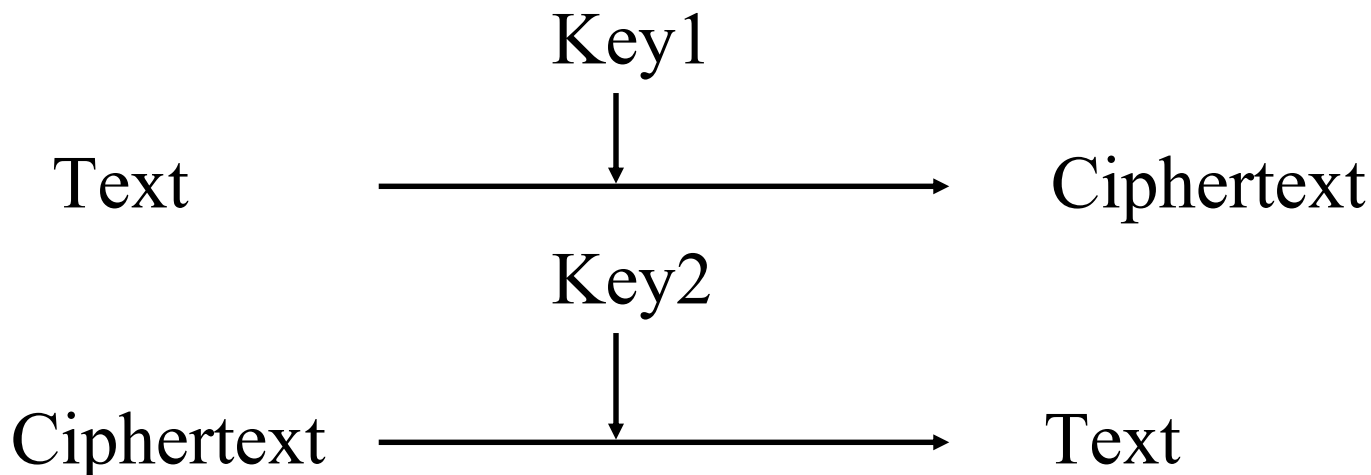
Public Key Encryption

1. Public Key Encryption
2. Modular Arithmetic
3. RSA Public Key Encryption

Public Key Encryption

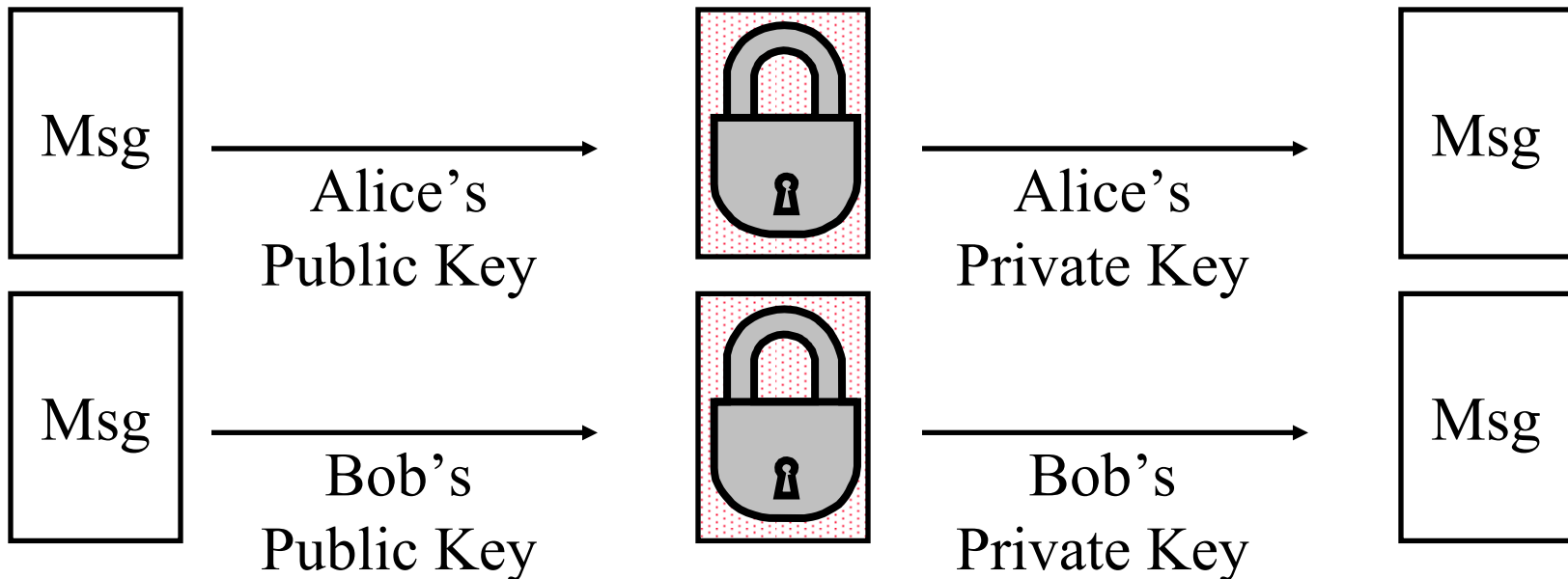


- ❑ Invented in 1975 by Diffie and Hellman
- ❑ $\text{Encrypted_Message} = \text{Encrypt}(\text{Key1}, \text{Message})$
- ❑ $\text{Message} = \text{Decrypt}(\text{Key2}, \text{Encrypted_Message})$



Public Key (Cont)

- ❑ One key is private and the other is public
- ❑ $\text{Message} = \text{Decrypt}(\text{Public_Key}, \text{Encrypt}(\text{Private_Key}, \text{Message}))$
- ❑ $\text{Message} = \text{Decrypt}(\text{Private_Key}, \text{Encrypt}(\text{Public_Key}, \text{Message}))$



Public Key Encryption Method

- ❑ Rivest, Shamir, and Adelson (RSA) method
- ❑ Example: Key1 = $\langle 3, 187 \rangle$, Key2 = $\langle 107, 187 \rangle$
- ❑ Encrypted_Message = $m^3 \bmod 187$
- ❑ Message = Encrypted_Message¹⁰⁷ mod 187
- ❑ Message = 5
- ❑ Encrypted Message = $5^3 = 125 \bmod 187 = 125$
- ❑ Message = $125^{107} \bmod 187 = 5$
= $125^{(64+32+8+2+1)} \bmod 187$
= $\{(125^{64} \bmod 187)(125^{32} \bmod 187) \dots$
 $(125^2 \bmod 187)(125 \bmod 187)\} \bmod 187$

Modular Arithmetic

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

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 $\gcd(z, e) = 1; 0 < e < z$
 - Calculate d such that $de \bmod z = 1$

RSA Algorithm: Example

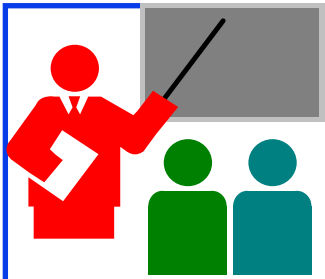
- ❑ Select two large primes: $p, q, p \neq q$
 $p = 17, q = 11$
- ❑ $n = p \times q = 17 \times 11 = 187$
- ❑ Calculate $z = (p-1)(q-1) = 16 \times 10 = 160$
- ❑ Select e , such that $\gcd(z, e) = 1; 0 < e < z$
say, $e = 7$
- ❑ Calculate d such that $de \bmod z = 1$
 - $160k+1 = 161, 321, 481, 641$
 - Check which of these is divisible by 7
 - 161 is divisible by 7 giving $d = 161/7 = 23$
- ❑ Key 1 = $\{7, 187\}$, Key 2 = $\{23, 187\}$



Confidentiality and Non-Repudiation

- ❑ 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





Public Key Encryption: Review

1. Public Key Encryption uses two keys: Public and Private
2. Either key can be used to encrypt. Other key will decrypt.
3. RSA public key method is based on difficulty of factorization

Homework 8B

[6 points] 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 $de=1 \pmod{z}$ and $d < 160$

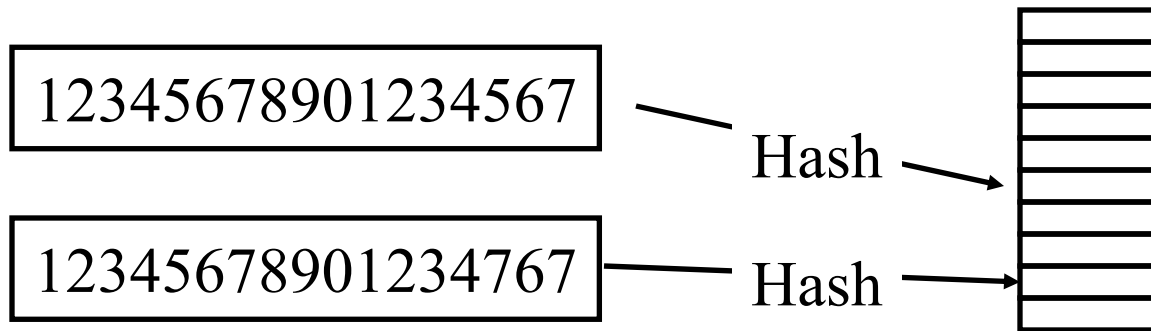
D. Encrypt the message $m=8$ using the key (n,e) . Let c be the corresponding cipher text. Show all work including decryption.



Hash, Signatures, Certificates

1. Hash Functions
2. MD5 Hash
3. SHA-1 Algorithm
4. Message Authentication Code (MAC)
5. Digital Signature
6. Digital Certificates
7. End Point Authentication

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 \Rightarrow Can't find x given $H(x) \Rightarrow$ One-way
5. Difficult to find y , such that $H(x) = H(y) \Rightarrow$ Can't change msg
6. Difficult to find *any* pair (x, y) such that $H(x) = H(y)$
 \Rightarrow Strong hash

MD5 Hash

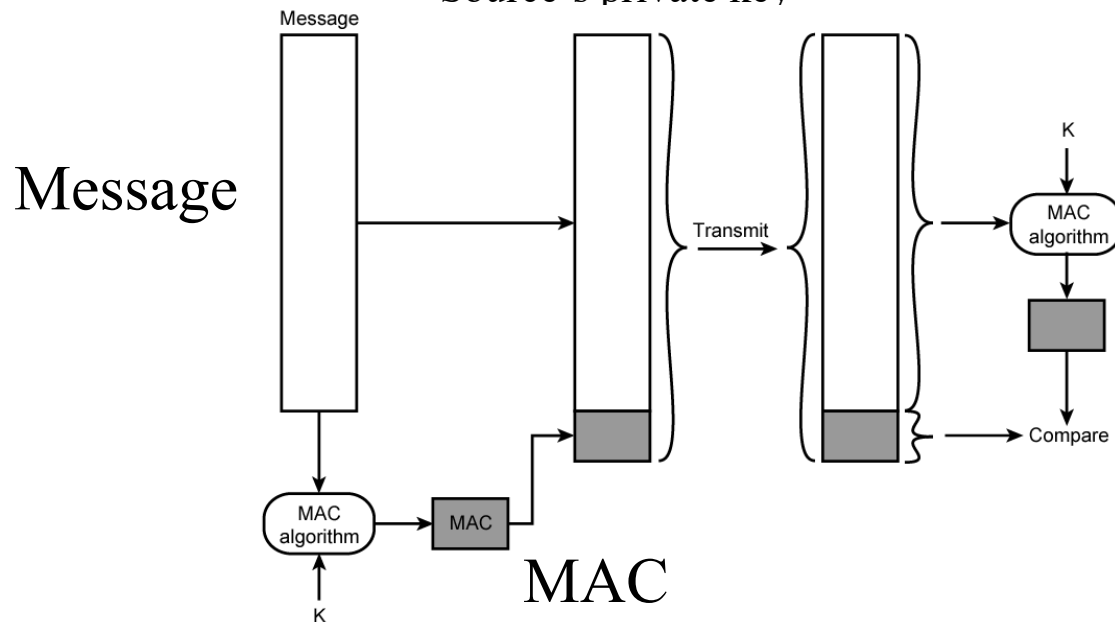
- ❑ 128-bit hash using 512 bit blocks using 32-bit operations
- ❑ Invented by Ron Rivest in 1991
- ❑ Described in RFC 1321
- ❑ Commonly used to check the integrity of files (easy to fudge message and the checksum)
- ❑ Also used to store passwords

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
- ❑ $\text{Encrypt}_{\text{secret key}} \{\text{Message, CRC, Time Stamp}\}$
- ❑ Message + $\text{Encrypt}_{\text{secret key}}(\text{Hash})$
Or, Message + $\text{Encrypt}_{\text{Source's private key}}(\text{Hash})$



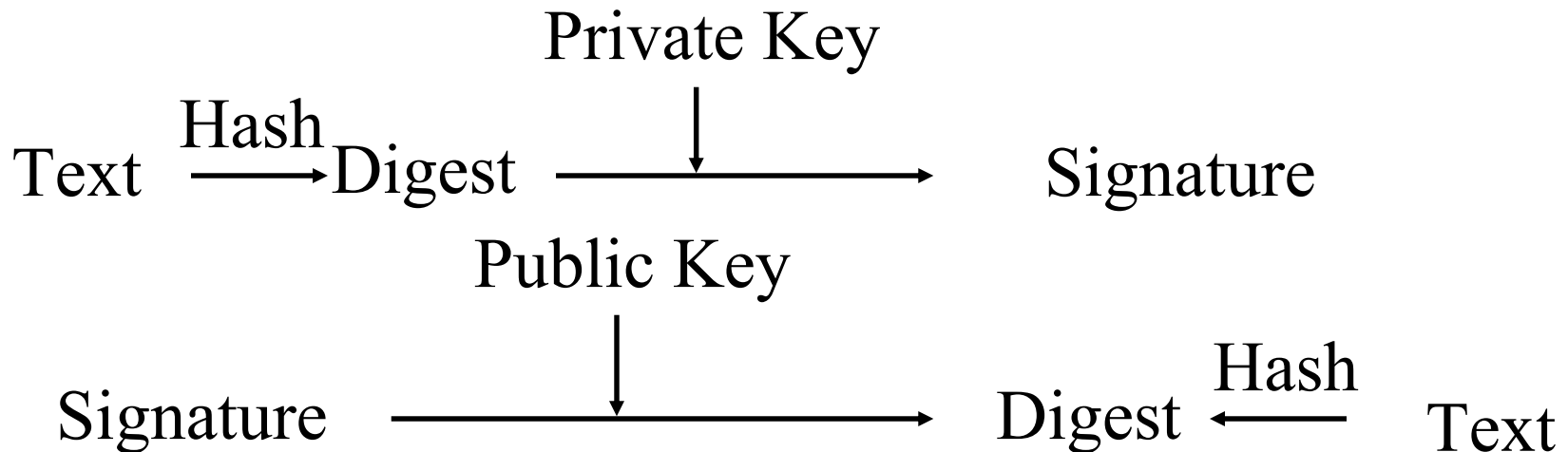
HMAC Overview

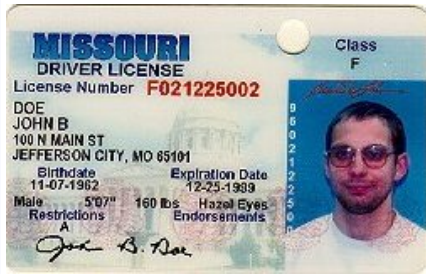
- ❑ Keyed Hash \Rightarrow includes a key along with message
- ❑ HMAC is a general design. Can use any hash function
 \Rightarrow HMAC-MD5, HMAC-AES
- ❑ Uses hash functions without modifications
- ❑ Has well understood cryptographic analysis of authentication mechanism strength



Digital Signature

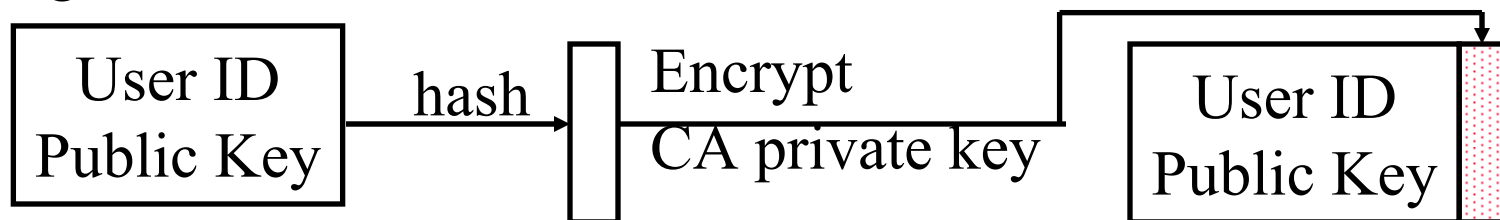
- ❑ Message Digest = Hash(Message)
- ❑ Signature = Encrypt(Private_Key, Hash)
- ❑ Hash(Message) = Decrypt(Public_Key, Signature)
⇒ Authentic
- ❑ Also known as Message *authentication* code (MAC)



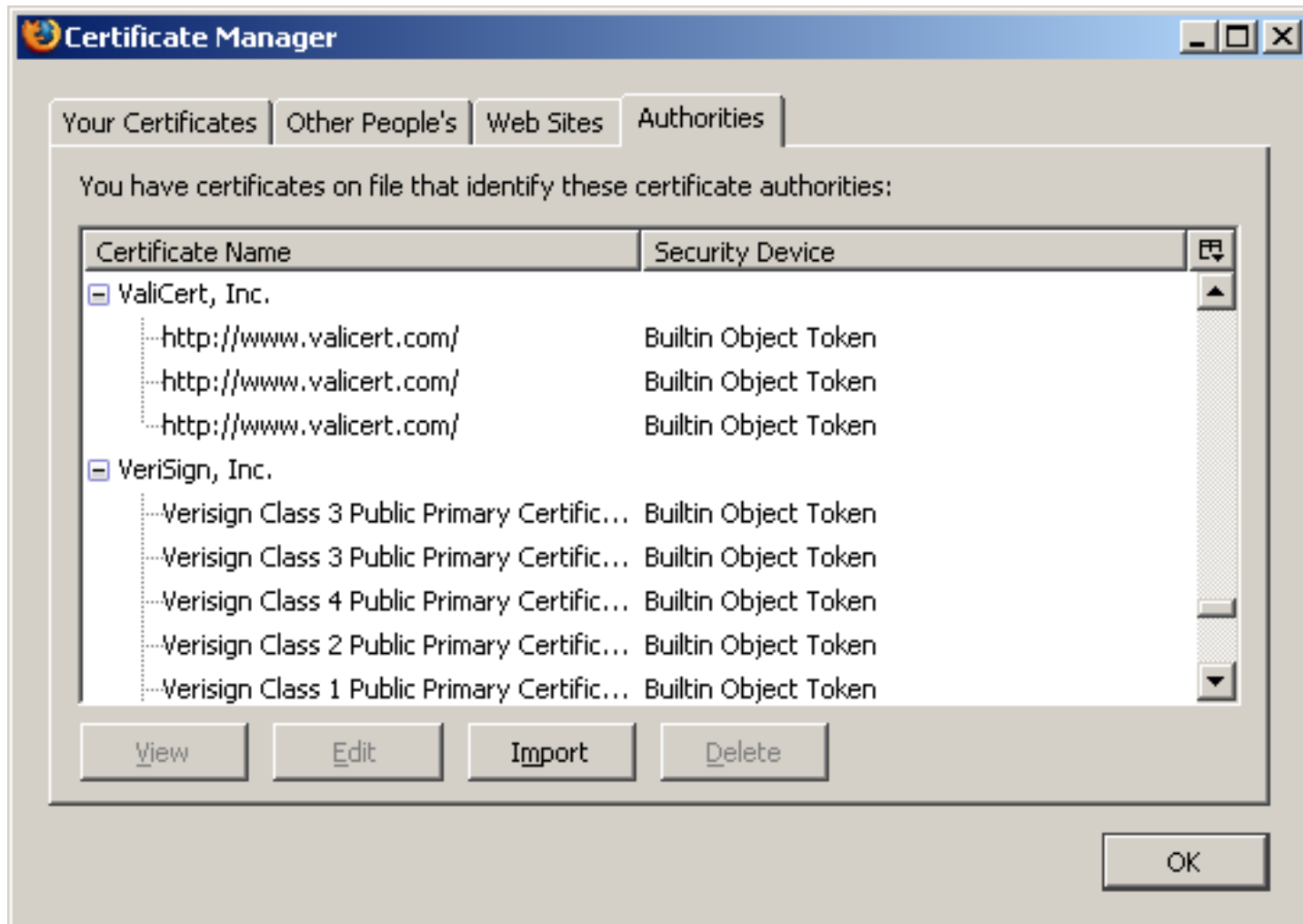


Digital Certificates

- ❑ 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



Ref: Windows: <http://smallbusiness.chron.com/see-security-certificates-stored-computer-54732.html>

MAC: <https://superuser.com/questions/992167/where-are-digital-certificates-physically-stored-on-a-mac-os-x-machine>

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse473-19/>

















©2019 Raj Jain

Sample X.509 Certificate

Internet Explorer

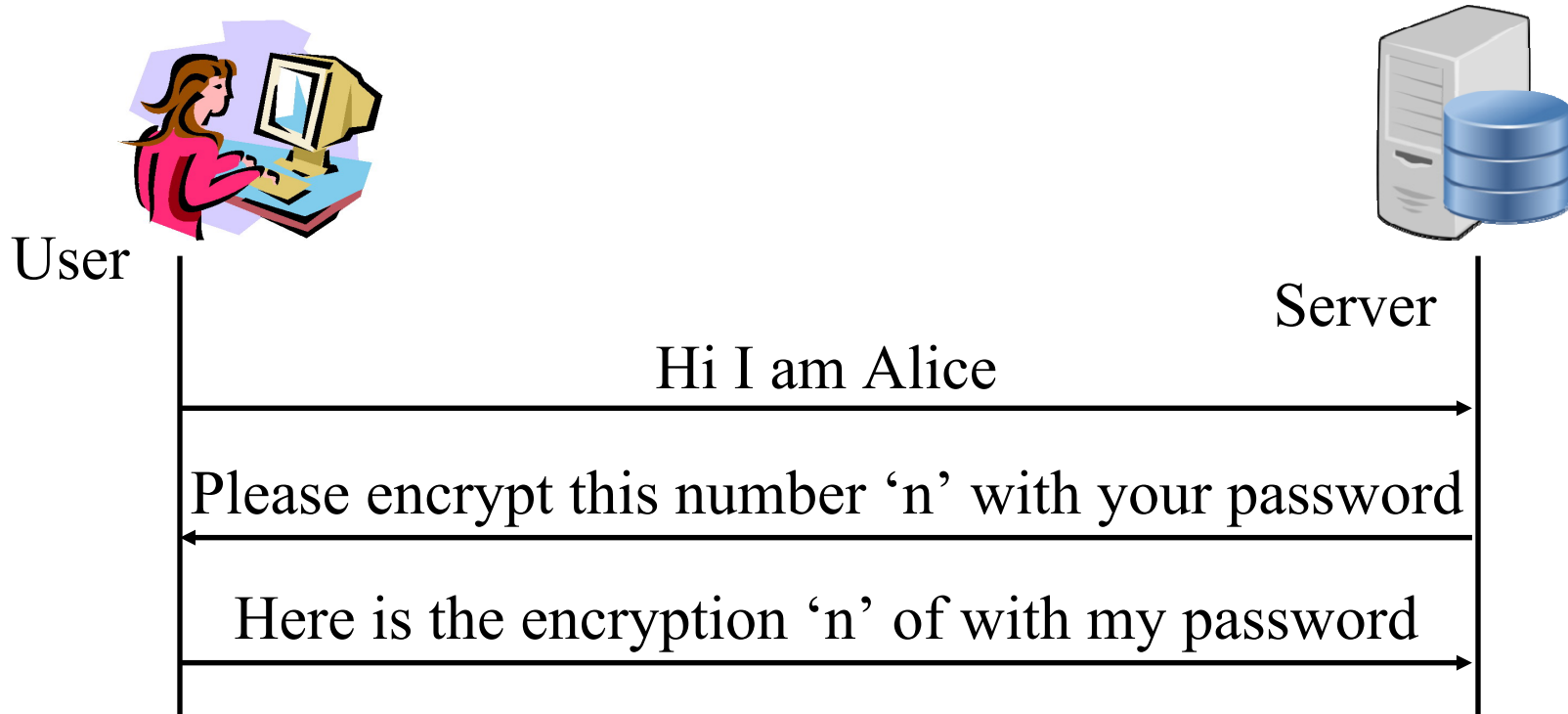


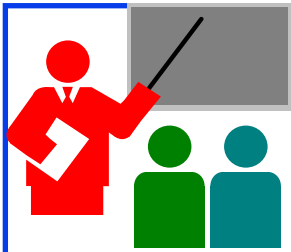
X.509 Sample (Cont)

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)

End Point Authentication

- ❑ Passwords can not be exchanged in clear
- ❑ Nonce = random number used only once
- ❑ Also done using certificates





Hashes, Signatures, Certificates

1. Hashes are one-way functions such that it difficult to find another input with the same hash like MD5, SHA-1
2. Message Authentication Code (MAC) ensures message integrity and source authentication using hash functions
3. Digital Signature consists of encrypting the hash of a message using private key
4. Digital certificates are signed by root certification authorities and contain public keys

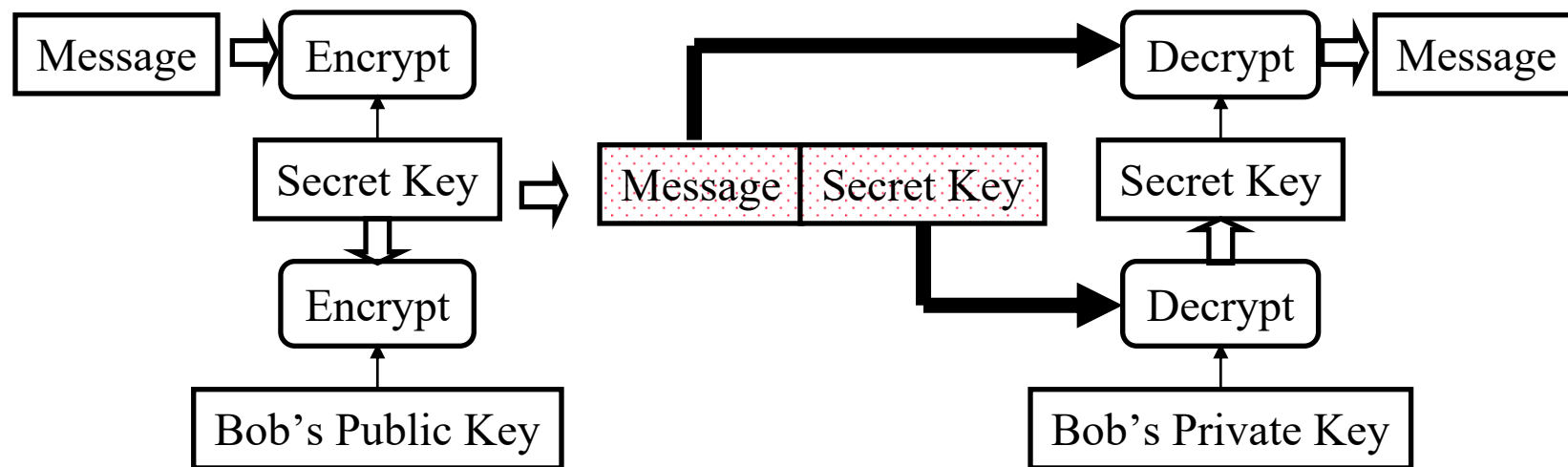


Secure Email

1. Secure E-Mail
2. Signed Secure E-Mail
3. Pretty Good Privacy (PGP)

Secure E-Mail

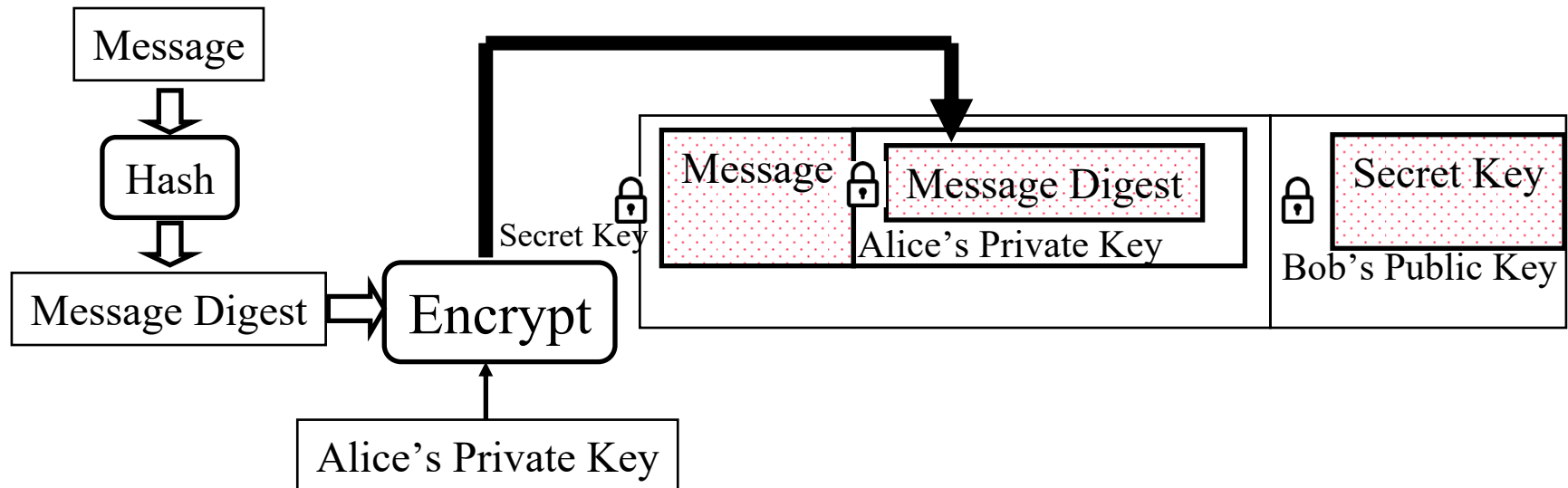
- Alice wants to send confidential e-mail, m , to Bob.



- **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

Signed Secure E-Mail

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



- ❑ Alice uses three keys: her private key, Bob's public key, newly created secret key
- ❑ Bob uses his private key to recover the secret key
- ❑ Bob uses Alice's public key to verify that the message came from Alice and was not changed.

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]

Lab 8

[20 points] You will receive a “signed” email from the TA. Reply to this email with a “encrypted and signed” email to TA.

If outlook says “*There is a problem with the signature on the TA’s message*” then click on the signature icon on the top right of the message and accept CAcert’s certificate. The warning will go away.

- ❑ You can reply to the TA’s email with a signed encrypted message. Content of the reply is not important.
- ❑ Before sending the reply, on the outlook message window, Select View → Options → (More Options →) Security Settings
Select encryption and signature
Now send the message.
- ❑ **Outlook is required** for both Windows and Mac
Only Firefox browser has been tested with both Windows and Mac. Other browsers may or may not work.

Lab 8 (Cont)

- ❑ To sign your email with a private key you need your digital certificate. To send an encrypted email you need TA's public key.
- ❑ TA's public key is attached with his/her email.
- ❑ The steps to obtain a free certificate and use it for email depend upon your email software and your operating system.
- ❑ Instructions for Outlook on Windows 10 using Firefox are as included next. If you do not have windows, you can do it using remote desktop to a Wash U windows computer.
- ❑ Instructions for Mac are similar, except that you need to download your certificate as a PEM file from the CAcert.org website and use that. Further details for Mac are at:
<https://knowledge.digicert.com/solution/SO6722.html>

Lab 8 (Cont)

1. Getting your Certificate:

- ❑ In Firefox, go to CAcert.org, click the "Join" link under the "Join CAcert.org" title on the right side of the page, then fill out the form.
- ❑ Once you have fully registered, login, and click "+ Client Certificates", then "New" on the right side of the screen. Select your email, then accept the agreement and click "next."
- ❑ A screen with a label "Keysize:" then a dropdown menu, and a button saying "Generate key pair within browser" will appear.
- ❑ In the dropdown menu, select "High Grade", then click the "Generate key pair within browser" button.

Lab 8 (Cont)

- ❑ When it is done generating the certificate, click "Install the certificate into your browser". Now in Firefox, click the three horizontal lines directly under the X, and then select "Options".
- ❑ On the left side of the screen, select "Privacy and Security", then scroll all the way down to the bottom, and click the "View Certificates" button.
- ❑ In the popup window, click the "Your Certificates" tab. Select the certificate that you just generated (it should say CAcert WoT User" under "Certificate Name"), and click backup.
- ❑ Save it somewhere where you will remember it, then give it a name and save it as a ".p12" file for Windows or ".pem" file for Mac. It will require that you give the file a password, so give it something that you will remember (you will need this later).

Lab 8 (Cont)

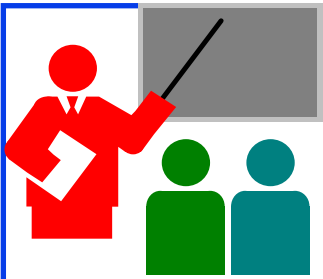
2. Installing your Certificate in Outlook:

- ❑ Now open the Outlook App (not the website). Click the "File" tab, then "Trust Center" in the popup, then "Trust Center Settings".
- ❑ Open the "Email Security" tab, then click the "Import/Export" button under the "Digital IDs (Certificates)" title.
- ❑ With the "Import" radio button selected, click the "Browse..." button, and find your certificate.
- ❑ Enter the password, and click "OK".
- ❑ Now, you can digitally sign an email by selecting the "Options" tab in the composing a message window, and clicking the "Sign" button.

Lab 8 (Cont)

3. Sending Encrypted Emails:

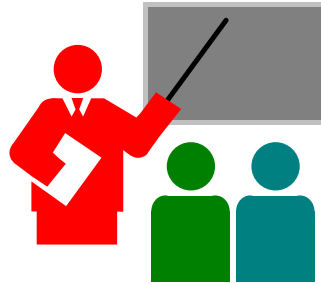
- ❑ The recipient may see "There is a problem with the signature" when they receive the signed message for the first time. This is because they may not have included CAcert as a trusted Certificate Authority. To fix this they need to click on the signature icon on the right-top of the message and accept CAcert's certificate. After this the problem message will go away.
- ❑ The recipient can also get a CAcert certificate and send a signed message to you. When you open that message, the recipient's public key is automatically installed in your outlook.
- ❑ After both of you have each other's public key, you can send encrypted emails to each other. You can send such messages by selecting the dropdown menu on the "Encrypt" button (right next to the "Sign" button), and selecting "Encrypt with S/MIME".



Secure Email: Review

1. Email provide confidentiality using a secret key
2. Public key and Certificates are used to:
 1. Sign the message
 2. To send the secret key

Summary



1. Network security requires confidentiality, integrity, availability, authentication, and non-repudiation
2. Encryption can use one secret key or two keys (public and private)
3. Public key is very compute intensive and is generally used to send secret key
4. Digital certificate system is used to certify the public key
5. Secure email uses confidentiality using a secret key, uses certificates and public keys to sign the email and to send the secret key

Acronyms

- ❑ 3DES Triple DES
- ❑ AES Advanced Encryption Standard
- ❑ CA Certificate authority
- ❑ CBC Cipher Block Chaining (CBC)
- ❑ CRC Cyclic Redundancy Check
- ❑ DES Data Encryption Standard (DES)
- ❑ FIPS Federal Information Processing standard
- ❑ HMAC
- ❑ ID Identifier
- ❑ IDEA
- ❑ IKE Internet Key Exchange
- ❑ IPsec Secure IP
- ❑ IV Initialization Vector
- ❑ MAC Message Authentication Code
- ❑ MD4 Message Digest 4
- ❑ MD5 Message Digest 5

Acronyms (Cont)

- ❑ NIST National Institute of Science and Technology
- ❑ OCR Optical Character Recognition
- ❑ OpenPGP Open PGP
- ❑ PGP Pretty Good Privacy
- ❑ RFC Request for Comment
- ❑ RSA Rivest, Shamir, Adleman
- ❑ SHA Secure Hash
- ❑ SSL Secure Socket Layer
- ❑ TA Teaching Assistant
- ❑ US United States
- ❑ VPN Virtual Private Network
- ❑ WEP Wired Equivalent Privacy
- ❑ XOR Exclusive OR

Scan This to Download These Slides



Raj Jain

<http://rajjain.com>

http://www.cse.wustl.edu/~jain/cse473-19/i_8sec.htm

Related Modules



CSE 567: The Art of Computer Systems Performance Analysis

https://www.youtube.com/playlist?list=PLjGG94etKypJEKjNAa1n_1X0bWWNyZcof

CSE473S: Introduction to Computer Networks (Fall 2011),

https://www.youtube.com/playlist?list=PLjGG94etKypJWOSPMh8Azcg5e_10TiDw



CSE 570: Recent Advances in Networking (Spring 2013)

<https://www.youtube.com/playlist?list=PLjGG94etKypLHyBN8mOgwJLHD2FFIMGq5>

CSE571S: Network Security (Spring 2011),

<https://www.youtube.com/playlist?list=PLjGG94etKypKvzfVtutHcPFJXumyyg93u>



Video Podcasts of Prof. Raj Jain's Lectures,

<https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw>