Network Security

- Cryptography
- Symmetric-Key Algorithms
- Public-Key Algorithms
- Digital Signatures
- Management of Public Keys
- Communication Security
- Authentication Protocols
- Email Security
- Web Security
- Social Issues

Revised: August 2011
Network Security

Security concerns a variety of threats and defenses across all layers.

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Link</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>
Some different adversaries and security threats

- Different threats require different defenses

<table>
<thead>
<tr>
<th>Adversary</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>To have fun snooping on people’s email</td>
</tr>
<tr>
<td>Cracker</td>
<td>To test out someone’s security system; steal data</td>
</tr>
<tr>
<td>Sales rep</td>
<td>To claim to represent all of Europe, not just Andorra</td>
</tr>
<tr>
<td>Businessman</td>
<td>To discover a competitor’s strategic marketing plan</td>
</tr>
<tr>
<td>Ex-employee</td>
<td>To get revenge for being fired</td>
</tr>
<tr>
<td>Accountant</td>
<td>To embezzle money from a company</td>
</tr>
<tr>
<td>Stockbroker</td>
<td>To deny a promise made to a customer by email</td>
</tr>
<tr>
<td>Con man</td>
<td>To steal credit card numbers for sale</td>
</tr>
<tr>
<td>Spy</td>
<td>To learn an enemy’s military or industrial secrets</td>
</tr>
<tr>
<td>Terrorist</td>
<td>To steal germ warfare secrets</td>
</tr>
</tbody>
</table>
Cryptography

Cryptography is a fundamental building block for security mechanisms.

- Introduction
- Substitution ciphers
- Transposition ciphers
- One-time pads
- Fundamental cryptographic principles
Introduction

The encryption model (for a symmetric-key cipher)

- Kerckhoff’s principle: Algorithms (E, D) are public; only the keys (K) are secret
Substitution Ciphers

Substitution ciphers replace each group of letters in the message with another group of letters to disguise it.

plaintext:  a b c d e f g h i j k l m n o p q r s t u v w x y z
            Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

ciphertext: Simple single-letter substitution cipher
Transposition Ciphers

Transposition ciphers reorder letters to disguise them.

Simple column transposition cipher

**Plaintext**
- please transfer one million dollars to my Swiss bank account six two two

**Ciphertext**
- AFLLSKSOSELAWAIAITOOSSTCLNMOMANT
- ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Key gives column order
- Column 5 6 7 8

MEGABUCK
7 4 5 1 2 8 3 6
please set
transfer one million dollars to my Swiss bank account six two two
out o a b c d
One-Time Pads (1)

Simple scheme for perfect secrecy:
- XOR message with secret pad to encrypt, decrypt
- Pad is as long as the message and can’t be reused!
  - It is a “one-time” pad to guarantee secrecy

Different secret pad decrypts to the wrong plaintext
One-Time Pads (2)

Alice sending Bob a one-time pad with quantum crypto.

- Bob’s guesses yield bits; Trudy misses some
Fundamental Cryptographic Principles

1. Messages must contain some redundancy
   – All encrypted messages decrypt to something
   – Redundancy lets receiver recognize a valid message
   – But redundancy helps attackers break the design

2. Some method is needed to foil replay attacks
   – Without a way to check if messages are fresh then old messages can be copied and resent
   – For example, add a date stamp to messages
Symmetric-Key Algorithms

Encryption in which the parties share a secret key

- DES – Data Encryption Standard »
- AES – Advanced Encryption Standard »
- Cipher modes »
- Other ciphers »
- Cryptanalysis »
Symmetric-Key Algorithms (1)

Use the same secret key to encrypt and decrypt; block ciphers operate a block at a time

- Product cipher combines transpositions/substitutions

Permutation (transposition) box  
Substitution box  
Product with multiple P- and S-boxes
Data Encryption Standard (1)

DES encryption was widely used (but no longer secure)

DES steps

A single iteration

Contains transpositions & substitutions
Data Encryption Standard (2)

Triple encryption ("3DES") with two 56-bit keys

- Gives an adequate key strength of 112 bits
- Setting $K_1 = K_2$ allows for compatibility with DES

Triple DES encryption

Triple DES decryption
Advanced Encryption Standard (1)

AES is the successor to DES:
- Symmetric block cipher, key lengths up to 256 bits
- Openly designed by public competition (1997-2000)
- Available for use by everyone
- Built as software (e.g., C) or hardware (e.g., x86)
- Winner was Rijndael cipher
- Now a widely used standard
Advanced Encryption Standard (2)

AES uses 10 rounds for 128-bit block and 128-bit key

- Each round uses a key derived from 128-bit key
- Each round has a mix of substitutions and rotations
- All steps are reversible to allow for decryption
Cipher Modes (1)

Cipher modes set how long messages are encrypted

- Encrypting each block independently, called ECB (Electronic Code Book) mode, is vulnerable to shifts

<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
<th>Bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adams, Leslie</td>
<td>Clerk</td>
<td>$10</td>
</tr>
<tr>
<td>Black, Robin</td>
<td>Boss</td>
<td>$500,000</td>
</tr>
<tr>
<td>Collins, Kim</td>
<td>Manager</td>
<td>$100,000</td>
</tr>
<tr>
<td>Davis, Bobbie</td>
<td>Janitor</td>
<td>$5</td>
</tr>
</tbody>
</table>

With ECB mode, switching encrypted blocks gives a different but valid message

Leslie gets a large bonus!
Cipher Modes (2)

CBC (Cipher Block Chaining) is a widely used mode
- Chains blocks together with XOR to prevent shifts
- Has a random IV (Initial Value) for different output

CBC mode encryption

CBC mode decryption
There are many other modes with pros / cons, e.g., cipher feedback mode is similar to CBC mode but can operate a byte (rather than a whole block) at a time.
A stream cipher uses the key and IV to generate a stream that is a one-time pad; can’t reuse (key, IV) pair

Doesn’t amplify transmission errors like CBC mode
Cipher Modes (5)

Counter mode (encrypt a counter and XOR it with each message block) allows random access for decryption.

Encryption above; repeat the operation to decrypt.
Other Ciphers

Some common symmetric-key cryptographic algorithms

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Author</th>
<th>Key length</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blowfish</td>
<td>Bruce Schneier</td>
<td>1–448 bits</td>
<td>Old and slow</td>
</tr>
<tr>
<td>DES</td>
<td>IBM</td>
<td>56 bits</td>
<td>Too weak to use now</td>
</tr>
<tr>
<td>IDEA</td>
<td>Massey and Xuejia</td>
<td>128 bits</td>
<td>Good, but patented</td>
</tr>
<tr>
<td>RC4</td>
<td>Ronald Rivest</td>
<td>1–2048 bits</td>
<td>Caution: some keys are weak</td>
</tr>
<tr>
<td>RC5</td>
<td>Ronald Rivest</td>
<td>128–256 bits</td>
<td>Good, but patented</td>
</tr>
<tr>
<td>Rijndael</td>
<td>Daemen and Rijmen</td>
<td>128–256 bits</td>
<td>Best choice</td>
</tr>
<tr>
<td>Serpent</td>
<td>Anderson, Biham, Knudsen</td>
<td>128–256 bits</td>
<td>Very strong</td>
</tr>
<tr>
<td>Triple DES</td>
<td>IBM</td>
<td>168 bits</td>
<td>Second best choice</td>
</tr>
<tr>
<td>Twofish</td>
<td>Bruce Schneier</td>
<td>128–256 bits</td>
<td>Very strong; widely used</td>
</tr>
</tbody>
</table>
Public-Key Algorithms

Encryption in which each party publishes a public part of their key and keep secret a private part of it

- RSA (by Rivest, Shamir, Adleman)
Public-Key Algorithms (1)

Downsides of keys for symmetric-key designs:
- Key must be secret, yet be distributed to both parties
- For N users there are $N^2$ pairwise keys to manage

Public key schemes split the key into public and private parts that are mathematically related:
- Private part is not distributed; easy to keep secret
- Only one public key per user needs to be managed

Security depends on the chosen mathematical property
- Much slower than symmetric-key, e.g., 1000X
- So use it to set up per-session symmetric keys
RSA (1)

RSA is a widely used public-key encryption method whose security is based on the difficulty of factoring large numbers

Key generation:

- Choose two large primes, p and q
- Compute \( n = p \times q \) and \( z = (p - 1) \times (q - 1) \).
- Choose \( d \) to be relatively prime to \( z \)
- Find \( e \) such that \( e \times d = 1 \mod z \)
- Public key is \((e, n)\), and private key is \((d, n)\)

Encryption (of k bit message, for numbers up to \( n \)):
- Cipher = Plain\(^e\) (mod \( n \))

Decryption:
- Plain = Cipher\(^d\) (mod \( n \))
RSA (2)

Small-scale example of RSA encryption

- For p=3, q=11 → n=33, z=20 → d=7, e=3

<table>
<thead>
<tr>
<th>Plaintext (P)</th>
<th>Ciphertext (C)</th>
<th>After decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Symbolic</td>
<td>Numeric</td>
</tr>
<tr>
<td>S</td>
<td>19</td>
<td>6859</td>
</tr>
<tr>
<td>U</td>
<td>21</td>
<td>9261</td>
</tr>
<tr>
<td>Z</td>
<td>26</td>
<td>17576</td>
</tr>
<tr>
<td>A</td>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>14</td>
<td>2744</td>
</tr>
<tr>
<td>N</td>
<td>14</td>
<td>2744</td>
</tr>
<tr>
<td>E</td>
<td>05</td>
<td>125</td>
</tr>
</tbody>
</table>

Sender's computation

Encryption: $C = P^3 \mod 33$

Receiver's computation

Decryption: $P = C^7 \mod 33$
Digital Signatures

Let's receiver verify the message is authentic

- Symmetric-Key signatures »
- Public-Key signatures »
- Message digests »
- The birthday attack »
Digital Signatures (1)

Requirements for a signature:

- Receiver can verify claimed identity of sender.
- Sender cannot later repudiate contents of message.
- Receiver cannot have concocted message himself.
Symmetric-key Signatures

Alice and Bob each trust and share a key with Big Brother; Big Brother doesn’t trust anyone

- A=Alice, B=Bob, P=message, $R_A$=random, t=time

Only Alice can send this encrypted message to BB

Only BB can send this encrypted message to Bob
Public-Key Signatures

No Big Brother and assumes encryption and decryption are inverses that can be applied in either order

- But relies on private key kept and secret

![Diagram of public-key signatures]
Message Digests (1)

Message Digest (MD) converts arbitrary-size message (P) into a fixed-size identifier MD(P) with properties:

- Given P, easy to compute MD(P).
- Given MD(P), effectively impossible to find P.
- Given P no one can find P′ so that MD(P′) = MD(P).
- Changing 1 bit of P produces very different MD.

Message digests (also called cryptographic hash) can “stand for” messages in protocols, e.g.,
authentication
- Example: SHA-1 160-bit hash, widely used
- Example: MD5 128-bit hash – now known broken
Message Digests (2)

Public-key signature for message authenticity but not confidentiality with a message digest

- Alice signs message digest
- Message sent in the clear
- Alice signs message digest

Diagram:

- Alice
- Bob
- $P, D_A (MD (P))$
Message Digests (3)

In more detail: example of using SHA-1 message digest and RSA public key for signing nonsecret messages
Message Digests (4)

SHA-1 digests the message 512 bits at a time to build a 160-bit hash as five 32-bit components.
Birthday Attack

How hard is it to find a message $P'$ that has the same message digest as $P$?

- Such a collision will allow $P'$ to be substituted for $P$!

Analysis:

- $N$ bit hash has $2^N$ possible values
- Expect to test $2^N$ messages given $P$ to find $P'$
- But expect only $2^{N/2}$ messages to find a collision
- This is the birthday attack
Management of Public Keys

We need a trusted way to distribute public keys

- Certificates
- X.509, the certificate standard
- Public Key infrastructures
Management of Public Keys (1)

Trudy can subvert encryption if she can fake Bob’s public key; Alice and Bob will not necessarily know

Trudy replaces $E_B$ with $E_T$ and acts as a “man in the middle”
Certificates

CA (Certification Authority) issues signed statements about public keys; users trust CA and it can be offline.

<table>
<thead>
<tr>
<th>I hereby certify that the public key</th>
</tr>
</thead>
<tbody>
<tr>
<td>19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A</td>
</tr>
<tr>
<td>belongs to</td>
</tr>
<tr>
<td>Robert John Smith</td>
</tr>
<tr>
<td>12345 University Avenue</td>
</tr>
<tr>
<td>Berkeley, CA 94702</td>
</tr>
<tr>
<td>Birthday: July 4, 1958</td>
</tr>
<tr>
<td>Email: <a href="mailto:bob@superduperernet.com">bob@superduperernet.com</a></td>
</tr>
</tbody>
</table>

SHA-1 hash of the above certificate signed with the CA’s private key

A possible certificate
**X.509**

X.509 is the standard for widely used certificates

- Ex: used with SSL for secure Web browsing

<table>
<thead>
<tr>
<th>Field</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Which version of X.509</td>
</tr>
<tr>
<td>Serial number</td>
<td>This number plus the CA’s name uniquely identifies the certificate</td>
</tr>
<tr>
<td>Signature algorithm</td>
<td>The algorithm used to sign the certificate</td>
</tr>
<tr>
<td>Issuer</td>
<td>X.500 name of the CA</td>
</tr>
<tr>
<td>Validity period</td>
<td>The starting and ending times of the validity period</td>
</tr>
<tr>
<td>Subject name</td>
<td>The entity whose key is being certified</td>
</tr>
<tr>
<td>Public key</td>
<td>The subject’s public key and the ID of the algorithm using it</td>
</tr>
<tr>
<td>Issuer ID</td>
<td>An optional ID uniquely identifying the certificate’s issuer</td>
</tr>
<tr>
<td>Subject ID</td>
<td>An optional ID uniquely identifying the certificate’s subject</td>
</tr>
<tr>
<td>Extensions</td>
<td>Many extensions have been defined</td>
</tr>
<tr>
<td>Signature</td>
<td>The certificate’s signature (signed by the CA’s private key)</td>
</tr>
</tbody>
</table>

Basic fields in X.509 certificates
Public Key Infrastructures (PKIs)

PKI is a system for managing public keys using CAs

- Scales with hierarchy, may have multiple roots
- Also need CRLs (Certificate Revocation Lists)

Hierarchical PKI

Trust anchor → Root → RA 1 → CA 1
RA 1 is approved. Its public key is 47383AE349...
Root's signature

RA 2 is approved. Its public key is 47383AE349...
Root's signature

RA 2 → RA 3 → CA 2
RA 1’s signature

RA 2 → RA 3 → CA 3

RA 2 → RA 3 → RA 4 → CA 4

RA 2’s signature

RA 2 → RA 3 → RA 4 → RA 5 → CA 5

RA 2’s signature

Chain of certificates for CA 5
Communication Security

Applications of security to network protocols

- IPsec (IP security) »
- Firewalls »
- Virtual private networks »
- Wireless security »
IPsec (1)

IPsec adds confidentiality and authentication to IP
- Secret keys are set up for packets between endpoints called security associations
- Adds AH header; inserted after IP in transport mode

AH (Authentication Header) provides integrity and anti-replay
IPsec (2)

ESP (Encapsulating Security Payload) provides secrecy and integrity; expands on AH

- Adds ESP header and trailer; inserted after IP header in transport or before in tunnel mode

**Transport mode**

- IP header
- ESP header
- TCP header
- Payload + padding
- Authentication (HMAC)

**Tunnel mode**

- New IP header
- ESP header
- Old IP header
- TCP header
- Payload + padding
- Authentication (HMAC)

Whole IP packet is wrapped
Firewalls

A firewall protects an internal network by filtering packets.

- Can have stateful rules about what packets to pass:
  - E.g., no incoming packets to port 80 (Web) or 25 (SMTP)
- DMZ helps to separate internal from external traffic.
Virtual Private Networks (1)

VPNs (Virtual Private Networks) join disconnected islands of a logical network into a single virtual network

- Islands are joined by tunnels over the Internet

VPN joining London, Paris, Home, and Travel
VPN traffic travels over the Internet but VPN hosts are separated from the Internet

- Need a gateway to send traffic in/out of VPN

Topology as seen from inside the VPN
Wireless Security (1)

Wireless signals are broadcast to all nearby receivers
- Important to use encryption to secure the network
- This is an issue for 802.11, Bluetooth, 3G, ...

Common design:
1. Clients have a password set up for access
2. Clients authenticate to infrastructure and set up a session key
3. Session key is then used to encrypt packets
Wireless Security (2)

802.11i session key setup handshake (step 2)

- Client and AP share a master key (password)
- MIC (Message Integrity Check) is like a signature
- $K_X(M)$ means a message $M$ encrypted with key $K_X$
Authentication verifies the identity of a remote party

- Shared Secret Key
- Diffie-Hellman Key Exchange
- Key Distribution Center
- Kerberos
- Public-Key Cryptography
Shared Secret Key (1)

Authenticating with a challenge-response (first attempt)

- Alice (A) and Bob (B) share a key $K_{AB}$

1. $F$

2. Challenge

3. Response

4. $K_{AB}(R_B)$

5. $K_{AB}(R_A)$

Bob knows it’s Alice

Alice knows it’s Bob
A shortened two-way authentication (second attempt)

But it is vulnerable to reflection attack.
Trudy impersonates Alice to Bob with reflection attack

Bob thinks he is talking to Alice now
Shared Secret Key (4)

First attempt is also vulnerable to reflection attack!

– Trudy impersonates Bob to Alice after Alice initiates

Alice thinks she is talking to Bob again

Alice thinks she is talking to Bob
Moral: *Designing a correct authentication protocol is harder than it looks; errors are often subtle.*

General design rules for authentication:

1. Have initiator prove who she is before responder
2. Initiator, responder use different keys
3. Draw challenges from different sets
4. Make protocol resistant to attacks involving second parallel session
An authentication protocol that is not vulnerable

- HMAC (Hashed Message Authentication Code) is an authenticator, like a signature
Diffie-Hellman Key Exchange (1)

Lets two parties establish a shared secret

- Eavesdropper can’t compute secret $g^{xy} \mod n$ without knowing $x$ or $y$
Diffie-Hellman Key Exchange (2)

But it is vulnerable to a man-in-the-middle attack
- Need to confirm identities, not just share a secret

1. Alice picks $x$ and sends $n, g, g^x \mod n$ to Trudy.
2. Trudy picks $z$ and sends $n, g, g^z \mod n$ to Bob.
3. Alice receives $g^z \mod n$ from Trudy.
4. Bob receives $g^y \mod n$ from Trudy.

$g^{xz} \mod n$ is shared between Alice and Trudy.
$g^{xz} \mod n$ is shared between Trudy and Bob.
KDC – Key Distribution Center (1)

Trusted KDC removes need for many shared secrets

- Alice and Bob share a secret only with KDC ($K_A, K_B$)
- End up with $K_S$, a shared secret session key
- First attempt below is vulnerable to replay attack in which Trudy captures and later replays messages

![Diagram of KDC process]

- Alice has session key $K_S$
- Trudy can send this later to impersonate Alice to Bob
- Bob has session key $K_S$
Key Distribution Center (2)

The Needham-Schroeder authentication protocol
  – Not vulnerable to replays; doesn’t use timestamps

Alice knows it’s Bob

Bob knows it’s Alice
Key Distribution Center (3)

The Otway-Rees authentication protocol (simplified)

— Slightly stronger than previous; Trudy can’t replay even if she obtains previous secret $K_S$
Kerberos

Kerberos V5 is a widely used protocol (e.g., Windows).

1. Alice asks for a secret shared with Bob.
2. Alice asks the Ticket Granting Server (TGS) for a session key $K_S$.
3. Alice gets the shared key $K_{AB}$.
4. Alice knows it's Bob.
5. Bob gets the key $K_{AB}$.
6. Bob knows it's Alice.
Public-Key Cryptography

Mutual authentication using public-key cryptography

- Alice and Bob get each other’s public keys \((E_A, E_B)\) from a trusted directory; shared \(K_S\) is the result
Email Security

Use of security for authenticated, confidential email

- PGP—Pretty Good Privacy »
PGP—Pretty Good Privacy (1)

PGP uses public- and symmetric-key cryptography for email secrecy and signatures; it also manages keys.

Levels of public-key strengths:

- Casual (384 bits): Can be broken easily today.
- Commercial (512 bits): Breakable by three-letter organizations.
- Military (1024 bits): Not breakable by anyone on earth.
- Alien (2048 bits): Unbreakable by anyone on other planets.
PGP—Pretty Good Privacy (2)

Signing and encrypting a message from Alice to Bob

For speed, message symmetric-key IDEA encrypted with $K_M$; $K_M$ is RSA public-key encrypted with $K_B$.

$K_M$: One-time message key for IDEA

$\times$: Concatenation

Bob's public RSA key, $E_B$

Original plaintext message from Alice

Authentication

Confidentiality

ASCII text to the network
PGP—Pretty Good Privacy (3)

Three parts of a PGP message and their encryption:

- Message key part
- Signature part
- Message part

PGP also manages public keys for a user:
- Private key ring has user’s public/private keys
- Public key ring has correspondent’s public keys
Web Security

Applications of security to the Web

- Secure naming
- SSL—Secure Sockets Layer

Many other issues with downloaded code
Secure Naming (1)

DNS names are included as part of URLs – so spoofing DNS resolution causes Alice contact

Trudy sends spoofed reply

1. Give me Bob’s IP address
2. 36.1.2.3 (Bob’s IP address)
3. GET index.html
4. Bob’s home page

1. Give me Bob’s IP address
2. 42.9.9.9 (Trudy’s IP address)
3. GET index.html
4. Trudy’s fake of Bob’s home page
Secure Naming (2)

How Trudy spoofs the DNS for *bob.com* in more detail

1. Look up foobar.trudy-the-intruder.com (to force it into the ISP's cache)
2. Look up www.trudy-the-intruder.com (to get the ISP's next sequence number)
3. Request for www.trudy-the-intruder.com (Carrying the ISP's next sequence number, n)
4. Quick like a bunny, look up bob.com (to force the ISP to query the com server in step 5)
5. Legitimate query for bob.com with seq = n+1
6. Trudy's forged answer: Bob is 42.9.9.9, seq = n+1
7. Real answer (rejected, too late)
Secure Naming (3)

DNSsec (DNS security) adds strong authenticity to DNS

- Responses are signed with public keys
- Public keys are included; client starts with top-level
- Also optional anti-spoofing to tie request/response

<table>
<thead>
<tr>
<th>Domain name</th>
<th>Time to live</th>
<th>Class</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bob.com.</td>
<td>86400</td>
<td>IN</td>
<td>A</td>
<td>36.1.2.3</td>
</tr>
<tr>
<td>bob.com.</td>
<td>86400</td>
<td>IN</td>
<td>KEY</td>
<td>3682793A7B73F731029CE2737D...</td>
</tr>
<tr>
<td>bob.com.</td>
<td>86400</td>
<td>IN</td>
<td>SIG</td>
<td>86947503A8B848F5272E53930C...</td>
</tr>
</tbody>
</table>

Resource Record set for bob.com.
Has Bob’s public key (KEY), and is signed by .com server (SIG)
SSL—Secure Sockets Layer (1)

SSL provides an authenticated, secret connection between two sockets; uses public keys with X.509

- TLS (Transport Layer Security) is the IETF version

SSL runs on top of TCP and below the application

HTTPS means HTTP over SSL

SSL in the protocol stack
SSL—Secure Sockets Layer (2)

Phases in SSL V3 connection establishment (simplified)

- Only the client (Alice) authenticates the server (Bob)
SSL—Secure Sockets Layer (3)

Data transmission using SSL. Authentication and encryption for a connection use the session key.

Message from browser

- Fragmentation
- Compression
- MAC added
- Encryption

Part 1

Part 2

Message authentication code

Header added
Social Issues

Networks give rise to many social issues

- Privacy »
- Freedom of speech »
Privacy

Anonymous remailers hide the identity of the sender

- Unlike PGP, which only hides message contents

Alice looks up keys E₁, E₂, E₃ separately

Bob gets a very anonymous mail
Freedom of Speech

Steganography hides messages on unrelated content
– Can help avoid censorship or protect ownership

“Three zebras and a tree”

“Three zebras and a tree,” with five plays by Shakespeare”

Text hidden in low-order bits
End

Chapter 8