Chapter 8: Network Security

Chapter goals:
- understand principles of network security:
  - cryptography and its many uses beyond confidentiality
  - authentication
  - message integrity
  - key distribution
- security in practice:
  - firewalls
  - security in application, transport, network, link layers

Chapter 8 roadmap

8.1 What is network security?
8.2 Principles of cryptography
8.3 Authentication
8.4 Integrity
8.5 Key Distribution and certification
8.6 Access control: firewalls
8.7 Attacks and counter measures
8.8 Security in many layers

What is network security?

Confidentiality: only sender, intended receiver should "understand" message contents
- sender encrypts message
- receiver decrypts message
Authentication: sender, receiver want to confirm identity of each other
Message Integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
Access and Availability: services must be accessible and available to users

Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob, Alice (lovers) want to communicate "securely"
- Trudy (intruder) may intercept, delete, add messages

Who might Bob, Alice be?

- well, real-life Bobs and Alices!
- Web browsers/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?
There are bad guys (and girls) out there!

Q: What can a "bad guy" do?
A: a lot!
- eavesdrop: intercept messages
- actively insert messages into connection
- impersonation: can fake (spoof) source address in packet (or any field in packet)
- hijacking: "take over" ongoing connection by removing sender or receiver, inserting himself in place
- denial of service: prevent service from being used by others (e.g., by overloading resources)
more on this later ....

Chapter 8 roadmap

8.1 What is network security?
8.2 Principles of cryptography
8.3 Authentication
8.4 Integrity
8.5 Key Distribution and certification
8.6 Access control: firewalls
8.7 Attacks and counter-measures
8.8 Security in many layers

The language of cryptography

symmetric key crypto: sender, receiver keys identical
public key crypto: encryption key public, decryption key secret (private)

Symmetric key cryptography

Substitution cipher: substituting one thing for another
- monoalphabetic cipher: substitute one letter for another
- plaintext: abcd efgh jklmnopqrstuvwxyz
- ciphertext: mbvcssa xdfgh klptuyrewq

Q: How hard is it to break this simple cipher?
- brute force (how hard?)
- other?

Symmetric key crypto: DES

DES: Data Encryption Standard
- US encryption standard (NIST 1993)
- 56-bit symmetric key, 64-bit plaintext input
- How secure is DES?
  - DES challenge: 56-bit key encrypted phrase ("Strong cryptography makes the world a safer place") decrypted (brute force) in 4 months
  - no known "backdoor" decryption approach
- making DES more secure:
  - use three keys sequentially (3-DES) on each datum
  - use cipher-block chaining
Symmetric key crypto: DES

- DES operation
- Initial permutation
- 16 identical "rounds" of function application, each using different 48 bits of key
- Final permutation

AES: Advanced Encryption Standard

- New (Nov. 2001) symmetric-key NIST standard, replacing DES
- Processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- Brute force decryption (try each key)
- Taking 1 sec on DES, takes 149 trillion years for AES

Public Key Cryptography

- Public key cryptography
- Requires sender, receiver know shared secret key
- How to agree on key in first place (particularly if never "met")

Public key encryption algorithms

Requirements:

1. Need $K_B^*(z)$ and $K_B^*(m)$ such that $K_B(K_B^*(m)) = m$
2. Given public key $K_B^*$, it should be impossible to compute private key $K_B$

RSA: Rivest, Shamir, Adelson algorithm

1. Choose two large prime numbers $p$, $q$
   (e.g., 1024 bits each)
2. Compute $n = pq$, $z = (p-1)(q-1)$
3. Choose $e$ (with encr) that has no common factors with $z$ (i.e., are "relatively prime")
4. Choose $d$ such that $ed\mod z = 1$
5. Public key is $(n,e)$, private key is $(n,d)$
RSA: Encryption, decryption

0. Given \((e, n)\) and \((n, d)\) as computed above
1. To encrypt bit pattern, \(m\), compute
   \(c = m^e \mod n\) (i.e., remainder when \(m^e\) is divided by \(n\))
2. To decrypt received bit pattern, \(c\), compute
   \(m = c^d \mod n\) (i.e., remainder when \(c^d\) is divided by \(n\))

Magic happens: \(m = (m^e \mod n)^d \mod n\)

RSA example:

Bob chooses \(p=5, q=7\). Then \(n=35, \phi=24\).
\(e=5\) (as \(e, n\) relatively prime).
\(d=23\) (as \(ed-1\) exactly divisible by \(z\)).

<table>
<thead>
<tr>
<th>Letter</th>
<th>(m)</th>
<th>(m^e)</th>
<th>(c = m^e \mod n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1624832</td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>17</td>
<td>1</td>
</tr>
</tbody>
</table>

decrypt: \(c^d \mod n\) letter

RSA: Why is that \(m = (m^e \mod n)^d \mod n\)?

Useful number theory result: If \(p, q\) prime and \(n = pq\), then:
\[x^\phi \mod n = x \mod (p-1)(q-1) \mod n\]

\[(m^e \mod n)^d \mod n = m^d \mod (p-1)(q-1) \mod n\]
(Using number theory result above)
\[m^d \mod n = m \mod n\]
(since we chose \(ed\) to be divisible by \(p-1)(q-2)\) with remainder 1

RSA: another important property

The following property will be very useful later:

\[K_B(K_A(m)) = m = K_B(K_A(m))\]

use public key use private key
first, followed first, followed
by private key by public key

Result is the same!

Chapter 8 roadmap

- 8.1 What is network security?
- 8.2 Principles of cryptography
- 8.3 Authentication
- 8.4 Integrity
- 8.5 Key Distribution and certification
- 8.6 Access control: firewalls
- 8.7 Attacks and counter measures
- 8.8 Security in many layers

Authentication

Goal: Bob wants Alice to "prove" her identity to him

Protocol step 1: Alice says "I am Alice"

Failure scenario??
**Authentication**

Goal: Bob wants Alice to "prove" her identity to him.

Protocol ap10. Alice says "I am Alice".

In a network, Bob can not "see" Alice, so Trudy simply declares herself to be Alice.

**Authentication: another try**

Protocol ap20. Alice says "I am Alice" in an IP packet containing her source IP address.

Failure scenario??

**Authentication: another try**

Protocol ap30. Alice can create a packet "spoofing" Alice's address.

Failure scenario??

**Authentication: another try**

Protocol ap30. Alice says "I am Alice" and sends her secret password to "prove" it.

Failure scenario??

**Authentication: yet another try**

Protocol ap31. Alice says "I am Alice" and sends her encrypted secret password to "prove" it.

Failure scenario??
Authentication: another try

Protocol ap3.1: Alice says "I am Alice" and sends her encrypted secret password to "prove" it.

Authentication: ap5.0

ap4.0 requires shared symmetric key
- can we authenticate using public key techniques?
- ap5.0: use nonce, public key cryptography

Authentication: yet another try

Goal: avoid playback attack
Nonce: number (R) used only once - in-a-lifetime

ap5.0: security hole

Man (woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)

Chapter 8 roadmap

8.1 What is network security?
8.2 Principles of cryptography
8.3 Authentication
8.4 Message integrity
8.5 Key Distribution and certification
8.6 Access control: firewalls
8.7 Attacks and counter measures
8.8 Security in many layers
Digital Signatures

Cryptographic technique analogous to handwritten signatures.
- Sender (Bob) digitally signs document, establishing he is document owner/creator.
- Verifiable, nonforgeable: recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document.

Digital Signatures (more)
- Suppose Alice receives msg m, digital signature K_B(m).
- Alice verifies m signed by Bob by applying Bob’s public key K_B to K_B(m) then checks K_B(K_B(m)) = m.
- If K_B(K_B(m)) = m, then whoever signed m must have used Bob’s private key.
- Alice thus verifies:
  - Bob signed m.
  - No one else signed m.
  - Bob signed m and not m.
- Non-repudiation:
  - Alice can take m, and signature K_B(m) to court and prove that Bob signed m.

Internet checksum: poor crypto hash function

Internet checksum has some properties of hash function:
- Produces fixed length digest (16-bit sum) of message.
- Is many-to-one.

But given message with given hash value, it is easy to find another message with same hash value:

<table>
<thead>
<tr>
<th>message</th>
<th>ASCII format</th>
<th>message</th>
<th>ASCII format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1</td>
<td>69 47 55 31</td>
<td>0 0 1</td>
<td>69 47 55 31</td>
</tr>
<tr>
<td>0 0 9</td>
<td>30 30 28 21</td>
<td>0 0 1</td>
<td>30 30 28 21</td>
</tr>
<tr>
<td>9 9 0</td>
<td>39 42 02 43</td>
<td>9 9 0</td>
<td>39 42 02 43</td>
</tr>
</tbody>
</table>

---

Digital Signatures

Simple digital signature for message m:
- Bob signs m by encrypting with his private key K_B, creating “signed” message K_B(m).
- Alice verifies signature by applying Bob’s public key K_B to the “signed” message K_B(m).

Message Digests

Computationally expensive to public key-encrypt long messages.
Goal: fixed length, easy-to-compute digital “fingerprint”
- Apply hash function H to m, get fixed size message digest, H(m).

Hash function properties:
- Many-to-1.
- Produces fixed-size msg digest (Fingerprint).
- Given message digest x, computationally infeasible to find m such that x = H(m).

Digital signature = signed message digest

Bob sends digitally signed message.
- Alice verifies signature and integrity of digitally signed message.

<table>
<thead>
<tr>
<th>message</th>
<th>ASCII format</th>
<th>message</th>
<th>ASCII format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1</td>
<td>69 47 55 31</td>
<td>0 0 1</td>
<td>69 47 55 31</td>
</tr>
<tr>
<td>0 0 9</td>
<td>30 30 28 21</td>
<td>0 0 1</td>
<td>30 30 28 21</td>
</tr>
<tr>
<td>9 9 0</td>
<td>39 42 02 43</td>
<td>9 9 0</td>
<td>39 42 02 43</td>
</tr>
</tbody>
</table>
Hash Function Algorithms
- MD5 hash function widely used (RFC 1321)
  - Computes 128-bit message digest in 4-step process.
  - Arbitrary 128-bit string x appears difficult to construct msg m whose MD5 hash is equal to x.
- SHA-1 is also used.
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit message digest

Chapter 8 roadmap
- 8.1 What is network security?
- 8.2 Principles of cryptography
- 8.3 Authentication
- 8.4 Integrity
- 8.5 Key distribution and certification
- 8.6 Access control: firewalls
- 8.7 Attacks and countermeasures
- 8.8 Security in many layers

Trusted Intermediaries
Symmetric key problem:
- How do two entities establish shared secret key over network?
Solution:
- Trusted key distribution center (KDC) acting as intermediary between entities.

Public key problem:
- When Alice obtains Bob's public key (from web site, e-mail directory), how does she know it is Bob's public key, not Trudy's?
Solution:
- Trusted certification authority (CA)

Key Distribution Center (KDC)
- Alice, Bob need shared symmetric key.
- KDC server shares different secret key with each registered user (many users).
- Alice, Bob know own symmetric keys, $K_{A,KDC}$, $K_{B,KDC}$, for communicating with KDC.

Certification Authorities
- Certification authority (CA) binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides "proof of identity" to CA.
  - CA creates certificate binding E to its public key.
  - Certificate containing E's public key digitally signed by CA.
  - CA says "This is E's public key.

Key Distribution Center (KDC)
- How does KDC allow Bob, Alice to determine shared symmetric secret key to communicate with each other?

Alice and Bob communicate using RI as session key for shared symmetric encryption.

Alice and Bob communicate using RI as session key for shared symmetric encryption.
**Certification Authorities**

- When Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere)
  - apply CA's public key to Bob's certificate, get Bob's public key

**A certificate contains:**
- Serial number (unique to issuer)
- Info about certificate owner (including algorithm and key value itself, not shown)
- Info about certificate issuer
- Valid dates
- Digital signature by issuer

**Chapter 8 roadmap**

8.1 What is network security?
8.2 Principles of cryptography
8.3 Authentication
8.4 Integrity
8.5 Key Distribution and certification
8.6 Access control: firewalls
8.7 Attacks and counter measures
8.8 Security in many layers

**Firewalls**

Firewall isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others

**Packet Filtering**

- Internal network connected to Internet via router firewall
- Router filters packet by packet, decision to forward/drop packet based on:
  - IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

**Firewalls: Why**

- Prevent denial of service attacks:
  - SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections
  - Prevent illegal modification/access of internal data:
    - E.g., attacker replaces CA's homepage with something else
    - Allow only authorized access to inside network (set of authenticated users/nets)
- Two types of firewalls:
  - Application level
  - Packet-filtering
Packet Filtering

- Example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23
- All incoming and outgoing UDP flows and telnet connections are blocked.
- Example 2: block inbound TCP segments with ACK=0
- Prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

Application gateways

- Filters packets on application data as well as on IP/TCP/UDP fields
- Example: allow select internal users to telnet outside.
- 1. Require all telnet users to telnet through gateway
- 2. For authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections.
- 3. Router filter blocks all telnet connections not originating from gateway.

Limitations of firewalls and gateways

- IP spoofing, router can't know if data "really" comes from claimed source
- If multiple apps use same IP addr, each has own app gateway
- Client software must know how to contact gateway
- e.g., must enter IP address of proxy in Web browser
- Firewalls often use all or nothing policy for UDP
- Tradeoff: degree of communication with outside world, level of security
- Many highly-protected sites still suffer from attacks.

Chapter 8 roadmap

8.1 What is network security?
8.2 Principles of cryptography
8.3 Authentication
8.4 Integrity
8.5 Key Distribution and certification
8.6 Access control: firewalls
8.7 Attacks and countermeasures
8.8 Security in many layers

Internet security threats

Mapping:
- Before attacking: "case the joint" - find out what services are implemented on network
- Use ping to determine what hosts have addresses on network
- Port-scanning: try to establish TCP connection to each port in sequence (see what happens)
- Nmap (http://www.insecure.org/nmap/) mapper: "network exploration and security auditing"

Countermeasures?

Internet security threats

Mapping: countermeasures
- Record traffic entering network
- Look for suspicious activity (IP addresses, ports being scanned sequentially)
**Internet security threats**

**Packet sniffing:**
- broadcast media
- promiscuous NIC reads all packets passing by
- can read all unencrypted data (e.g. passwords)
- e.g.: C sniffs B's packets

![Diagram](image)

**Countermeasures?**

---

**Internet security threats**

**IP Spoofing:**
- can generate "raw" IP packets directly from application, putting any value into IP source address field
- receiver can't tell if source is spoofed
- e.g.: C pretends to be B

![Diagram](image)

**Countermeasures?**

---

**Internet security threats**

**Denial of service (DOS):**
- flood of maliciously generated packets "swamp" receiver
- Distributed DOS (DDoS): multiple coordinated sources swamp receiver
- e.g.: C and remote host SYN-attack A

![Diagram](image)

**Countermeasures?**

---

**Internet security threats**

**Packet sniffing: countermeasures**
- all hosts in organization run software that checks periodically if host interface in promiscuous mode.
- one host per segment of broadcast media
  (switched Ethernet at hub)

![Diagram](image)
Chapter 8 roadmap

8.1 What is network security?
8.2 Principles of cryptography
8.3 Authentication
8.4 Integrity
8.5 Key distribution and certification
8.6 Access control: firewalls
8.7 Attacks and countermeasures
8.8 Security in many layers
8.8.1 Secure email
8.8.2 Secure sockets
8.8.3 IPE
8.8.4 Security in 802.11

Secure e-mail

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

\[ K_a \Rightarrow K(m) \Downarrow \text{Alice} \]

Bob:

- uses his private key to decrypt and recover \( K_a \)
- uses \( K_a \) to decrypt \( K(m) \) to recover \( m \)

Pretty good privacy (PGP)

Internet e-mail encryption scheme, de-facto standard
- uses symmetric key cryptography, public key encryption, hash function, and digital signature as described
- provides secrecy, sender authentication, integrity
- inventor Phil Zimmermann, was target of 3-year federal investigation

A PGP signed message:

- \( \text{--BEGIN PGP SIGNED MESSAGE--} \)
- Bob: My husband is out of town tonight. Passionately yours, Alice
- \( \text{--END PGP SIGNED MESSAGE--} \)
Secure sockets layer (SSL)

- Transport layer security to any TCP-based app using SSL services
- Used between Web browsers, servers for e-commerce (https)
- Security services:
  - Server authentication
  - Data encryption
  - Client authentication (optional)

SSL (continued)

- Encrypted SSL session
  - Browser generates symmetric session key, encrypts it with server's public key, sends encrypted key to server.
  - Using private key, server decrypts session key.
  - Server knows session key.
  - All data sent to TCP socket (by client or server) encrypted with session key.

IPsec: Network Layer Security

- Network layer security:
  - Sends host encrypts the data in IP datagram
  - TCP and UDP segments, ICMP, and SNMP messages
- Network layer authentication:
  - Destination host uses authentication source IP address
- Two-principle protocols:
  - Authentication header (AH) protocol
  - Encapsulating security protocol (ESP) protocol
- For both AH and ESP, source, destination handshake
- Create network-layer logical channel called a security association (SA)
- Each SA unique
- Security protocol (AH or ESP)
- Source IP address
- 32-bit connection ID

Authentication Header (AH) Protocol

- Provides source authentication, data integrity, no confidentiality
- AH header inserted between IP header, data field
- 32-bit protocol field 51
- Intermediate routers process datagrams as usual

ESP Protocol

- Provides security, host authentication, data integrity
- Data, ESP trailer encrypted
- Next header field in ESP trailer
- ESP authentication field is similar to AH authentication field
- Protocol = 50

IEEE 802.11 Security

- War driving: drive around Bay area, see what 802.11 networks available?
- More than 9000 accessible from public roadways
- 85% use no encryption/authentication
- Packet sniffing and various attacks easy
- Securing 802.11
  - Encryption, authentication
  - First attempt at 802.11 security: Wired Equivalent Privacy (WEP); a failure
  - Current attempt: 802.11i
Wired Equivalent Privacy (WEP):
- authentication as in protocol 802.11
- host requests authentication from access point
- access point sends 128 bit nonce
- host encrypts nonce using shared symmetric key
- access point decrypts nonce, authenticates host
- no key distribution mechanism
- authentication: knowing the shared key is enough

WEP data encryption
- Host/AP share 40 bit symmetric key (semi-permanent)
- Host appends 24-bit initialization vector (IV) to create 64-bit key
- 64 bit key used to encrypt nth byte, \( d_n \) in frame
- \( k^M \) used to encrypt nth byte, \( d_n \), in frame
- \( c_n = d_n \oplus k^M_n \)
- IV and encrypted bytes, \( c_n \), sent in frame

802.11 WEP encryption

Breaking 802.11 WEP encryption
Security hole:
- 24-bit IV, one IV per frame, \( \rightarrow \) IVs eventually reused
- IV transmitted in plaintext \( \rightarrow \) IV reuse detected

Attack:
- Trudy causes Alice to encrypt known plaintext \( d_1, d_2, d_3, d_4, \ldots \)
- Trudy sees \( c_1 = d_1 \oplus k^M_1 \)
- Trudy knows \( c_1, d_1 \) so can compute \( k^M_1 \)
- Trudy knows encrypting key sequence \( k^M_1, k^M_2, k^M_3, \ldots \)
- Next time IV is used, Trudy can decrypt

802.11i: improved security
- numerous (stronger) forms of encryption possible
- provides key distribution
- uses authentication server separate from access point

802.11i: four phases of operation

- Discovery of security capabilities
- STA and AP mutually authenticate together generate Master Key (MK), AP serves as "pass through"
- STA derives Temporal Key (TK) used for message encryption, integrity
EAP: extensible authentication protocol

- EAP: end-end client (mobile) to authentication server protocol
- EAP sent over separate "links"
  - mobile-to-AP (EAP over LAN)
  - AP to authentication server (RADIUS over UDP)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>EAP</th>
<th>EAP over LAN (802.1X)</th>
<th>RADIUS</th>
<th>IEEE 802.11</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Network Security (summary)

Basic techniques:
- cryptography (symmetric and public)
- authentication
- message integrity
- key distribution
- used in many different security scenarios
- secure email
- secure transport (SSL)
- IPsec
- 802.11