

COMS3200

COMS3200 – Week 12 Network Security

Some slides are taken from:
"Computer Networking: A Top Down Approach Featuring
the Internet",
3rd edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2004.

School of Information Technology and Electrical Engineering
The University of Queensland

COMS3200

What about this email?

Subject: Citibank Identity Theft Solutions

Recently there have been a large number of identity theft attempts targeting Citibank customers. In order to safeguard your account, we require that you update your Citibank ATM/Debit card PIN.

This update is requested of you as a precautionary measure against fraud. Please note that we have no particular indications that your details have been compromised in any way.

This process is mandatory, and if not completed within the nearest time your account may be subject to temporary suspension.

To securely update your Citibank ATM/Debit card PIN please go to:
https://www.citibank.com/signin/citifi/scripts/login2/update_pin.jsp

Thank you for your prompt attention to this matter and thank you for using Citibank!

Regards,
Madeline Walter
Head of Citi® Identity Theft Solutions

2

COMS3200

Phishing – Social Engineering

- **"Phishing"** is a technique used to gain personal information for purposes of identity theft, using fraudulent e-mail messages that appear to come from legitimate businesses. These authentic-looking messages are designed to fool recipients into divulging personal data such as account numbers and passwords, credit card numbers and Social Security numbers.
- **"Social Engineering"** is a method to trick people into revealing passwords or other information that compromises a target system's security.
- Phishing and Social Engineering are just examples of many possible attacks on Computers and Networks

3

COMS3200

Outline

- Aspects of Security
- Encryption
 - Secret Key
 - Public Key
- Authentication Protocols
- Message Integrity - Digital signatures
- Key Distribution – Public Key Certificates
- Security Protocols
 - IPsec
 - TLS/SSL
- Firewalls

4

COMS3200

Learning objectives

After this week, you should:

- be able to describe **security aspects** and show which security mechanisms can provide these security aspects
- be able to describe and compare **encryption algorithms**
- be able to describe and compare **authentication algorithms**
- be able to describe mechanisms to achieve **message integrity** and **non-repudiation**
- be able to describe problems arising for **distribution of shared keys** (DES) or public keys (RSA) and show possible solutions
- be able to describe security protocols at the IP and TCP layers (**IPsec, TLS/SSL**) and firewalls

5

COMS3200

Aspects of 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

Non-repudiation: ensuring that users cannot deny the occurrence of particular events

- e.g. "I never authorised that purchase"
- Techniques: digital signatures

6

COMS3200

Technology for Security

- In the remainder of this lecture, we will look at technical methods such as Cryptography, Protocols and Firewalls to address these aspects of security
- Note:** These Technologies can only provide a partial solution to the problem of Security. A major percentage of successful attacks are due to human errors and negligence.

7

COMS3200

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

8

COMS3200

Who might Bob, Alice be?

- ... well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?

9

COMS3200

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)

10

COMS3200

Roadmap

- Aspects of Security
- Encryption**
 - Secret Key
 - Public Key
- Authentication Protocols
- Message Integrity - Digital signatures
- Key Distribution – Public Key Certificates
- Security Protocols
 - IPsec
 - TLS/SSL
- Firewalls

11

COMS3200

The language of cryptography

Cryptography: the art/science of hidden writing (Greek: Kryptos → hidden, Graphen → write)

symmetric key crypto: sender, receiver keys *identical*

public-key crypto: encryption key *public*, decryption key *secret* (private)

12

COMS3200

Encryption Algorithm

- Imagine you are Julius Caesar and you need to send the following secret message to one of your Generals via an un-trusted courier.

"attack at dawn"
- Obviously, you will not have computers but only pencil and paper available.
- Can you think of a simple encryption method?

13

COMS3200

Caesar Cipher

Cipher = Encryption Algorithm

- Each letter in the message is replaced by the letter shifted by 3 positions in the alphabet.
- A-> D, B-> E, C-> F, ...

abcdefghijklmnopqrstuvwxyz
 defghijklmnopqrstuvwxyzabc
- "attack at dawn" -> "dwwdfn dw gdzq"
- We could generalise this approach by shifting the letters by k positions instead of 3.
- Is this encryption method secure?

14

COMS3200

Substitution Cipher

Substitution Cipher: substituting one thing for another

- Mono alphabetic cipher: substitute one letter for another
- (Caesar cipher is an example of a mono alphabetic substitution cipher)

plaintext: abcdefghijklmnopqrstuvwxyz
 ↓ ↓
 ciphertext: mnbvcxzasdfghjklpoiuytrewq

E.g.: Plaintext: attack at dawn
 Ciphertext: muumbf mu vmrj

Q: How hard is it to break this simple cipher?

- brute force (how hard?)
- other?

15

COMS3200

Breaking the Mono alphabetic Substitution Cipher

- Brute force: try every possible mapping (key)
 - Number of mappings = number of permutations of 26 letters = $26!$
 $\approx 4 * 10^{26}$
 - Even if we try one mapping per nsec, it will take 10^{10} years to try all possibilities
- Better approach: use statistics
 - Relative frequency of letters in the English language
 - Relative frequency of letter combinations (the, it, ...)
 - (It's a bit like solving cross-word puzzles)

letter	frequency (%)	letter	frequency (%)
a	8.2	n	6.7
b	1.5	o	7.5
c	2.8	p	1.9
d	4.3	q	0.1
e	12.7	r	6.0
f	2.2	s	6.3
g	2.0	t	9.1
h	6.1	u	2.8
i	7.0	v	1.0
j	0.2	w	2.4
k	0.8	x	0.2
l	4.0	y	2.0
m	2.4	z	0.1

16

COMS3200

Transposition Cipher

- Instead of substituting letters, Transposition Ciphers reorder them
- If you see a block of ciphertext, how can you guess that it is encrypted with a Transposition Cipher?

M E G A B U C K 7 4 5 1 2 8 3 6 p l e a s e t r a n s f e r o n e m i l l i o n d o l l a r s t o m y s w i s s b a n k a c c o u n t s i x t w o t w o a b c d	← key Plaintext pleasetransferonemilliondollarsto myswissbankaccountsixtwo Ciphertext AFLLSKSOSELAWAIATOSSCTCLNMOMANT ESILYNTWRNNTSOWDPAEDOBUEIRICXB
--	--

17

COMS3200

Transposition and Substitution

- Transposition Ciphers and Substitution Ciphers as described in the previous slides are not being used anymore.
- However, the principles of Transposition and Substitution form the basis of all modern ciphers.

18

Symmetric key cryptography

symmetric key crypto: Bob and Alice share same (symmetric) key: K_{A-B}

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher
- Qs:** how do Bob and Alice agree on a key value?
 - Via a secure channel. (Chicken-and-egg problem)
- If there are N people who want to communicate secretly with each other, how many keys do we need?

19

Symmetric key crypto: DES

DES: Data Encryption Standard

- US encryption standard [NIST 1993] for unclassified (commercial) information
- 56-bit symmetric key, 64-bit plaintext input (block cipher)
- How secure is DES?
 - no known "backdoor" decryption approach, brute force is the best we can do
 - However, with today's supercomputers a DES key can be broken in less than an hour by brute force
- making DES more secure:
 - use three keys sequentially (3-DES, "triple DES") on blocks of data

20

Symmetric key crypto: DES

DES operation

initial permutation
16 identical "rounds" of function application, each using different 48 bits of key
final permutation

- Swapping of L and R is transposition
- Function f performs substitution

21

AES: Advanced Encryption Standard

- New (Nov. 2001) symmetric-key NIST standard, replacing DES
 - Open selection process. Researchers could submit proposals. Public scrutiny of algorithm. Voting on best algorithm. Winner: "Rijndael" by Belgian cryptographers Daemen and Rijmen.
 - Further info and implementations available at: <http://www.esat.kuleuven.ac.be/~rijmen/rijndael/>
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1h on DES, takes 149 trillion years for AES

22

Security of a Cipher

- A cipher is considered "strong" if there is no significantly faster method of breaking it than brute force
- If this is the case, the key length determines security
- However, there is no guarantee or proof that somebody will not be able to find a better attack
- Is there a perfect/unbreakable cipher?

23

The unbreakable cipher: One-time Pad

- A one-time Pad is theoretically unbreakable, no matter how much time and computing resources an attacker has.
- How does it work?
 - For a plaintext message of n bits choose n random bits (the one-time pad, only to be used once)
 - Each bit of the message is XORed with the corresponding bit of the random bit string \rightarrow ciphertext

Message: 1001000100010010010
Pad: 00110100001110100100 XOR
Ciphertext: 1010010101100110110

- Why is it unbreakable? What about brute force?
- Why is it not very practical?

24

One-time Pad

- Given a ciphertext of n bits, a brute force attack would try out all possible pads of length n
- This would result in all possible n bit strings as potential messages
- All are equally likely. The ciphertext contains no information (in an information theoretical sense) about the message.
- What's the catch?
 - For every n bits of message, we need n bits of key → not practical for most applications
 - Compare with AES where a 256-bit key can be used to encrypt Terrabytes of data

25


Public Key Cryptography

symmetric key crypto

- requires sender, receiver to know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

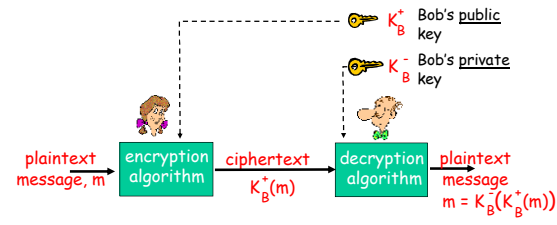
public key cryptography


- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- public* encryption key known to *all*
- private* decryption key known only to receiver
- Makes key management a lot easier!



26

Public Key Cryptography



Mechanical Analogue: 

27

Public key encryption algorithms

Requirements:

- need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that $K_B^-(K_B^+(m)) = m$
- given public key K_B^+ , it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adleman algorithm
Most widely used public key algorithm

28

RSA

- How it works:
 - The plaintext message is split up into blocks of constant size and converted into Integers M

Encryption: $C = M^e \text{ mod } n$
 Decryption: $M = C^d \text{ mod } n$

(*mod n* means taking only the remainder when divided by n)

C: ciphertext
 M: plaintext
 e: public key
 d: private key
 n: a value that everybody needs to agree on, public

29

RSA

Requirements:

- For any message M , Decryption of Encrypted message must result in original message:

$$C^d \text{ mod } n = (M^e)^d \text{ mod } n = M^{ed} \text{ mod } n = M$$
- Given e and n , it should be impossible to find the private key d

- Requirements hold for a specific choice of n , e , and d
- Q: How do we choose n , e , d so that requirements are met?
- A: Answer is provided by a branch of mathematics called **Number Theory** (Details beyond scope of this course)

30

COMS3200

RSA: Choosing Parameters

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 $e < n$) that has no common factors with z . (e, z are "relatively prime").
4. Choose d such that $ed-1$ is exactly divisible by z . (in other words: $ed \bmod z = 1$). (Extended Euclid's Algorithm)
5. Public key is (n, e) . Private key is (n, d) .

$$\underbrace{(n, e)}_{K_B^+}$$

$$\underbrace{(n, d)}_{K_B^-}$$

31

COMS3200

RSA example

Bob chooses $p=5, q=7$. Then $n=35, z=(5-1)(7-1)=24$.
 $e=5$ (so e, z relatively prime).
 $d=29$ (so $ed-1$ exactly divisible by z).

	<u>letter</u>	<u>m</u>	<u>m^e</u>	<u>c = m^e mod n</u>
encrypt:	l	12	1524832	17
decrypt:	<u>c</u>	<u>c^d</u>	<u>m = c^d mod n</u>	<u>letter</u>
	17	481968572106750915091411825223071697	12	l

It works!!!

32

COMS3200

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key (encryption)}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key (digital signature)}}$$

Result is the same!

33

COMS3200

Security of RSA

- Requirement 2: Should be impossible to deduce secret key from public key
- RSA can be broken if an attacker manages to find p and q from n . (factoring n) (Knowledge of p and q allows an attacker to calculate z and therefore d)
- It is assumed that factoring large numbers is hard (no polynomial algorithm exists) and RSA's security relies on that. However, there is no proof for that.
- With the currently best factoring algorithm, it would take 10^7 MIPS-years to factor a 1'000-bit integer.

34

COMS3200

Public Key Cryptography

- Makes key distribution or key management a lot easier
 - No secure channel needed to distribute keys
 - For N people, only need N keys needed, compared to $N(N-1)/2$ in secret key crypto
- Computationally expensive
 - Suitable only for short messages
 - Often used to support key distribution for secret key encryption
- Open problem, authenticity of public keys
 - If Bob gets Alice's public key, how can he be sure that it is indeed Alice's and not Trudy's?

35

COMS3200

Roadmap

- Aspects of Security
- Encryption
 - Secret Key
 - Public Key
- Authentication Protocols
- Message Integrity - Digital signatures
- Key Distribution – Public Key Certificates
- Security Protocols
 - IPsec
 - TLS/SSL
- Firewalls

36

Authentication

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

Protocol ap1.0: Alice says "I am Alice"

Failure scenario??

37

Authentication

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

Protocol ap1.0: Alice says "I am Alice"

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

38

Authentication: another try

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address

Failure scenario??

39

Authentication: another try

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address

Trudy can create a packet "spoofing" Alice's address

40

Authentication: another try

Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.

Failure scenario??

41

Authentication: another try

Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.

playback attack: Trudy records Alice's packet and later plays it back to Bob

42

Authentication: yet another try

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

Alice's IP addr | encrypted password | "I'm Alice"

Alice's IP addr | OK

Failure scenario??

43

Authentication: another try

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

Alice's IP addr | encrypted password | "I'm Alice"

Alice's IP addr | OK

record and playback still works!

44

Authentication: yet another try

Goal: avoid playback attack

Nonce: number (R) used only *once -in-a-lifetime*

ap4.0: to prove Alice "live", Bob sends Alice **nonce**, R. Alice must return R, encrypted with shared secret key

"I am Alice"

R

$K_{A-B}(R)$

Alice is live, and only Alice knows key to encrypt nonce, so it must be Alice!

Failures, drawbacks?

45

Authentication: ap5.0

ap4.0 requires shared symmetric key

- can we authenticate using public key techniques?

ap5.0: use nonce, public key cryptography

"I am Alice"

R

$K_A^-(R)$

"send me your public key"

K_A^+

Bob computes $K_A^+(K_A^-(R)) = R$ and knows only Alice could have the private key, that encrypted R such that $K_A^+(K_A^-(R)) = R$

Can anybody see the security hole?

46

ap5.0: security hole

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

I am Alice

I am Alice

R

$K_A^-(R)$

Send me your public key

K_T^+

Trudy gets $K_T^+(m)$

$m = K_A^-(K_T^+(m))$ sends m to Alice encrypted with Alice's public key

$m = K_A^+(K_A^+(m))$

47

ap5.0: security hole

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

Difficult to detect:

- Bob receives everything that Alice sends, and vice versa. (e.g., so Bob, Alice can meet one week later and recall conversation)
- problem is that Trudy receives all messages as well!

Main Problem: Authenticity of public keys. How do we know for sure which public key belongs to whom?

48

Roadmap

- Aspects of Security
- Encryption
 - Secret Key
 - Public Key
- Authentication Protocols
- Message Integrity - Digital signatures
- Key Distribution – Public Key Certificates
- Security Protocols
 - IPsec
 - TLS/SSL
- Firewalls

49

Digital Signatures

Cryptographic technique analogous to hand-written 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

50

Digital Signatures

Simple digital signature for message m :

- Bob signs m by encrypting with his private key K_B^- , creating "signed" message, $K_B^-(m)$
- Different order of using private and public key!!

Remember RSA's property: $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$

Bob's message, m
Dear Alice
Oh, how I have missed you. I think of you all the time! ... (blah blah blah)
Bob

Bob's private key K_B^-

Public key encryption algorithm

Bob's message, m , signed (encrypted) with his private key $K_B^-(m)$

51

Digital Signatures (more)

- Suppose Alice receives msg m and 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 $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key.

Alice thus verifies that:

- 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 .

52

Message Digests

large message m

H: Hash Function

$H(m)$

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 $x = H(m)$
- cryptographically secure hash functions

53

Digital signature = signed message digest

Bob sends digitally signed message:

large message m

H: Hash function

$H(m)$

Bob's private key K_B^-

digital signature (encrypt)

encrypted msg digest $K_B^-(H(m))$

Alice verifies signature and integrity of digitally signed message:

encrypted msg digest $K_B^-(H(m))$

Bob's public key K_B^+

digital signature (decrypt)

large message m

H: Hash function

$H(m)$

$H(m)$

equal ?

54

Hash Function Algorithms

- Cryptographic one-way hash functions
 - MD5 hash function still used (RFC 1321) in spite of some security problems
 - 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 widely used
 - US standard [NIST, FIPS PUB 180-1]
 - 160-bit message digest

55

Hash-based Message Authentication Code

- Message Authentication Code based on cryptographic hash (e.g. MD5) and symmetric keys
 - Requires symmetric keys
 - Much faster than public key cryptography
- Alice wants to send message m to Bob. She adds a MAC calculated as the hash of m and the key k
- Bob, knowing the key k , can verify the MAC and concludes that only somebody knowing k could have calculated $H(m,k)$. Therefore:
 - The message must be from Alice
 - Nobody could have altered m (integrity)

m : message
 k : secret key shared between Alice and Bob
 $H()$: Cryptographic hash function (e.g. MD5)

56

Roadmap

- Aspects of Security
- Encryption
 - Secret Key
 - Public Key
- Authentication Protocols
- Message Integrity - Digital signatures
- Key Distribution – Public Key Certificates
- Security Protocols
 - IPsec
 - TLS/SSL
- Firewalls

57

Key Distribution - 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, diskette), how does she know it is Bob's public key, not Trudy's?

Solution:

- trusted certification authority (CA)

58

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.

59

Key Distribution Center (KDC)

Q: How does KDC allow Bob, Alice to determine shared symmetric secret key to communicate with each other?

(Simplified protocol)

Alice and Bob communicate: using $R1$ as session key for shared symmetric encryption

Well-known KDC System: Kerberos

60

COMS3200

Recap

- Symmetric key encryption
 - Secret key K_{A-B} shared between Bob and Alice
 - Problem: How to distribute secret keys
 - Need secure channel
 - Or Trusted third party, KDC

61

COMS3200

Recap (II)

- Public Key Cryptography
 - Bob has public key K_B^+ and private Key K_B^-
 - Confidentiality
 - Encryption: $c = K_B^+(m)$
 - Decryption: $m = K_B^-(c)$
 - Digital Signature
 - Signing m : $K_B^-(m)$
 - Verifying signature on m : $K_B^+(m)$
 - Key distribution is easy, no secret channel required
 - Problem: need to establish authenticity of public key

62

COMS3200

Problem of Authentication of Public Keys

- A way for Trudy to subvert public key encryption

Get Bob's Home Page

Bob's fake Home page with Trudy's public key K_T^+

$K_T^+(m)$

- Now Trudy can decrypt secret message m from Alice intended for Bob

63

COMS3200

Public Key Distribution

- Distribution is not a problem. Public keys can be sent with message, can be published on web site etc.
- **Problem:** How to ensure that public keys are authentic?
- **Solution:** Certificates
 - special type of digitally signed document:

"I certify that the public key in this document belongs to the entity named in this document, signed X."
 - Who is X?

64

COMS3200

Public Key Distribution

- Certification Authority (CA)
 - Trusted administrative entity that issues certificates. E.g. VeriSign, Thawte ..
 - Bob can go to CA and present proof of his identity (passport etc.) and CA will issue a Certificate
 - Alice needs the CA's public key to verify Bob's certificate
 - How does Alice get the CA's public key (And how can she be sure it's authentic?)

65

COMS3200

Certificates

- A possible certificate

I hereby certify that the public key
 19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
 belongs to
 Robert John Smith
 12345 University Avenue
 Berkeley, CA 94702
 Birthday: July 4, 1958
 Email: bob@superdupernet.com

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

T-808 66

X.509 standard for certificates

- The basic fields of an X.509 certificate

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

T-810 67

Public Key Distribution

- Chain of Trust
 - if X certifies that a certain public key belongs to Y, and Y certifies that another public key belongs to Z, then there exists a chain of certificates from X to Z
 - someone that wants to verify Z's public key has to know X's public key and follow the chain
- Certificate Revocation List (CRL)

68

Public-Key Infrastructures

- (a) A hierarchical PKI, (b) A chain of certificates

```

    graph TD
      Root[Root] --> RA1[RA 1]
      Root --> RA2[RA 2]
      RA1 --> CA1[CA 1]
      RA1 --> CA2[CA 2]
      RA1 --> CA3[CA 3]
      RA2 --> CA4[CA 4]
      RA2 --> CA5[CA 5]
  
```

- Bootstrap problem: how do we get the public key of the Root CA?
 - E.g. Web browsers have public keys of trusted CAs preinstalled in the form of self-signed certificates.

T-811 69

Certificates in Internet Explorer

- To view the contents of a public key certificate in Internet Explorer, go to a secure web site, i.e. one using TLS/SSL (more on this in a couple of slides...).
 - E.g. <https://www.westpac.com.au>
- Double-click on the padlock at the top right-hand corner of Browser window and view the certificate.
- To view pre-installed root certificates in IE
 - Tools → Internet Options → content → Certificates → Trusted Root Certification Authorities

70

Roadmap

- Aspects of Security
- Encryption
 - Secret Key
 - Public Key
- Authentication Protocols
- Message Integrity - Digital signatures
- Key Distribution – Public Key Certificates
- Security Protocols
 - IPsec
 - TLS/SSL
- Firewalls

71

IPsec Network Layer Security

- Operates at Network layer
 - Provides security for all higher layer protocols (TCP, UDP, ICMP etc.)
 - Transparent to applications
 - Optional in IPv4, standard in IPv6
 - Based on symmetric key cryptography (performance)
 - Can support many different Ciphers
 - Requires distribution of secret keys. Key management protocols: ISAKMP, IKE (not discussed here)
- Network-layer secrecy:
 - sending host encrypts the data in IP datagram
- Network-layer authentication and data integrity
 - destination host can authenticate source IP address and verify integrity of data

72

IPsec: Two Protocols

- **authentication header (AH) protocol**
 - Provides authentication and integrity
- **encapsulation security payload (ESP) protocol**
 - Provides secrecy and authentication (o)
- **Security Association (SA)**
 - For both AH and ESP, source and destination perform a handshake:
 - create network-layer logical channel called a security association (SA) and negotiate shared secret key
 - Each SA unidirectional
 - Uniquely determined by:
 - security protocol (AH or ESP)
 - source IP address
 - 32-bit connection ID (SPI: Security Parameter Index)

73

Authentication Header (AH) Protocol

- provides source authentication, data integrity, no confidentiality
- AH header inserted between IP header, data field.
- protocol field: 51
- intermediate routers process datagrams as usual

AH header includes:

- connection identifier
- authentication data: source-signed message digest calculated over original IP datagram (HMAC)
 - Payload and static part of IP header
- next header field: specifies type of data (e.g., TCP, UDP, ICMP)

74

ESP Protocol

- provides secrecy, host authentication and data integrity.
- data, ESP trailer encrypted.
- next header field is in ESP trailer.
- ESP authentication field is similar to AH authentication field.
- Protocol = 50.

75

IPsec: 2 Modes of operation

- **Transport mode**
 - IPsec header is inserted after IP header
 - Only payload of original packet is protected
- **Tunnel mode**
 - Entire IP packet is protected and encapsulated in new IP packet

76

IPsec Transport Mode

- *End-to-end security* between the hosts
- Security within site networks as well
- Requires hosts to implement IPsec

77

IPsec Tunnel Mode

- Tunnel is established between IPsec servers/gateways
- Hosts operate in their usual way
 - Tunnel mode IPsec is *transparent* to the end hosts
- No security within the site networks

78

IPsec/VPN

- AH and ESP can be used both in Transport and Tunnel Mode
- Typical application of IPsec Tunnel mode: VPN (Virtual Private Network)

79

SSL/TLS

- **SSL** (Secure Socket Layer)
 - Designed by Netscape
- **TLS** (Transport Layer Security)
 - IETF adopted SSL and called it TLS
- Provides security on top of TCP
 - Authentication, Integrity, Encryption
 - Mostly used to secure web traffic (HTTP), but can be used for any TCP traffic
 - Application needs to be SSL/TLS aware

Application (HTTP)
Security (SSL)
Transport (TCP)
Network (IP)
Data link (PPP)
Physical (modem, ADSL, cable TV)

T-854 80

SSL/TLS

- Handshake is performed to set up secure connection
 - Exchange of public key certificates for authentication
 - Server → client (mandatory)
 - Client → Server (optional)
 - Negotiation of algorithms/parameters/keys to be used
- All application data is protected
 - Authentication
 - Data Integrity
 - Encryption

81

SSL – Connection Establishment

T-855 82

SSL – Data Transmission

T-856 83

Roadmap

- Aspects of Security
- Encryption
 - Secret Key
 - Public Key
- Authentication Protocols
- Message Integrity - Digital signatures
- Key Distribution – Public Key Certificates
- Security Protocols
 - IPsec
 - TLS/SSL
- **Firewalls**

84

COMS3200

Firewalls

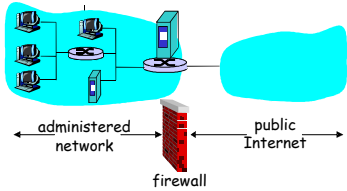
- Extra Note on Website:
"Firewalls fend off invasions from the Net", S. Lodin, C. Schuba
- Tanenbaum: 8.6.2

85

COMS3200

Firewalls

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



86

COMS3200

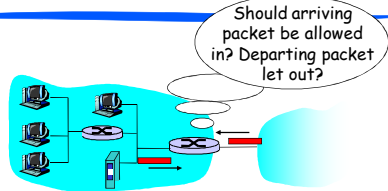
Firewalls: Why

- 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 CIA's homepage with something else
- allow only authorized access to inside network (set of authenticated users/hosts)
- two types of firewalls:
 - application-level
 - packet-filtering

87

COMS3200

Packet Filtering



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

88

COMS3200

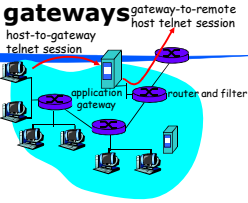
Packet Filtering

- **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.

89

COMS3200

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.

90

COMS3200

Limitations of firewalls and gateways

- **IP spoofing:** router can't know if data "really" comes from claimed source
- if multiple app's. need special treatment, each has own app. gateway.
- client software must know how to contact gateway.
 - e.g., must set IP address of proxy in Web browser
- filters 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.

91

COMS3200

Summary

- Aspects of Security
- Cryptography
 - Encryption (shared key, public key)
 - Authentication
 - Data Integrity
 - Key Distribution
- Security in communication protocols
 - IPsec
 - SSL/TLS
- Firewalls

92

COMS3200

Readings

- Tanenbaum: 8(intro), 8.1, 8.2 (8.2.1, 8.2.2), 8.3, 8.4 (8.4.1, 8.4.2, 8.4.3), 8.5, 8.6 (8.6.1, 8.6.2, 8.6.3), 8.7 (8.7.1, 8.7.5), 8.9.3
(4th ed: same sections)
- Next week:
 - **Tuesday - Revision, including exam information**
 - **No lecture Friday**

93