

Principles of Network Security

Kai Shen

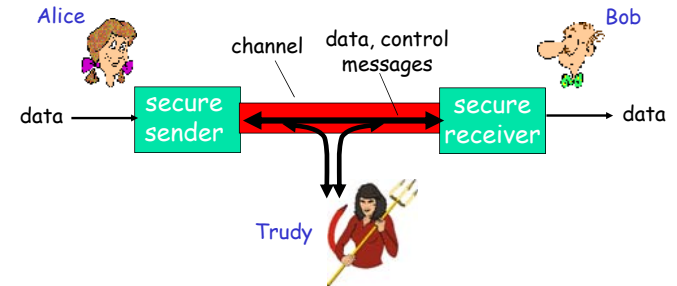
12/5/2011

CSC 257/457 - Fall 2011

1

The Network Security Model

- Bob and Alice want to communicate "securely".
- Trudy (the adversary) has access to the channel.



12/5/2011

CSC 257/457 - Fall 2011

2

Who might Bob and Alice be?

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

12/5/2011

CSC 257/457 - Fall 2011

3

What can an adversary do?

- Eavesdrop:** understand the content of messages
- Actively **changing** messages
- Impersonation:** fake (spoof) identity
- Denial of service:** prevent service from being used by others (e.g., by overloading resources)

12/5/2011

CSC 257/457 - Fall 2011

4

What is Network Security?

- Confidentiality:** only sender, intended receiver should "understand" message contents
- Authentication:** sender, receiver want to confirm identity of each other
- Message Integrity:** sender, receiver want to ensure message not altered (in transit, or afterwards)
- Access and Availability:** services must be accessible and available to (and only to) legitimate users.

12/5/2011
CSC 257/457 - Fall 2011
5

Principles of Network Security

- Confidentiality: cryptography
- Authentication
- Integrity
- Key distribution and certification

12/5/2011
CSC 257/457 - Fall 2011
6

The Language of Cryptography

First goal of cryptography: **confidentiality.**

```

    graph LR
      P1[plaintext] --> E[encryption algorithm]
      KA[Alice's encryption key KA] --> E
      E --> C[ciphertext]
      C --> D[decryption algorithm]
      KB[Bob's decryption key KB] --> D
      D --> P2[plaintext]
  
```

- **Symmetric key** crypto: encryption and decryption keys are identical. (both are **secret**)
- **Public key** crypto: encryption key is **public**, decryption key is **secret**.

12/5/2011
CSC 257/457 - Fall 2011
7

Symmetric Key Cryptography: Monoalphabetic Cipher

Monoalphabetic cipher: substitute one letter for another.

```

    plaintext:  abcdefghijklmnopqrstuvwxyz
                ↓                               ↓
    ciphertext: mnbvcxzasdfghjklpoiuytrewq
  
```

Example: Plaintext: bob. i love you. alice
 ciphertext: nkn. s gktc wky. mgsbc

Q1: How hard to break this simple cipher?

- brute force?
- other?

Q2: How to make it more difficult to break?

12/5/2011
CSC 257/457 - Fall 2011
8

Symmetric Key Cryptography: DES

- **DES: Data Encryption Standard**
 - US encryption standard [NIST 1993]
 - 56-bit symmetric key, 64-bit plaintext input
 - **encryption**: initial permutation \Rightarrow 16 "rounds", each using different 48 bits of key \Rightarrow final permutation
 - **decryption**: reverse operation using the same key
- How secure is DES?
 - DES Challenge (1999): 56-bit-key-encrypted phrase decrypted (brute force) in 22 hours 15 minutes
- Making DES more secure:
 - use three keys sequentially (3-DES)
 - use more bits

12/5/2011 CSC 257/457 - Fall 2011 9

AES: Advanced Encryption Standard

- newer (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 128-bit AES

12/5/2011 CSC 257/457 - Fall 2011 10

Public Key Cryptography

Symmetric key cryptography

- requires sender, receiver know shared secret key
- Q: how to agree on key in the first place? (particularly difficult if Trudy is eavesdropping on all communication)

Public key cryptography

- encryption key is different from decryption key
- encryption key is **public**, known to **everyone**, also called **public key**
- decryption key is **secret**, known only to **receiver**, also called **private key**

12/5/2011 CSC 257/457 - Fall 2011 11

Public Key Cryptography

12/5/2011 CSC 257/457 - Fall 2011 12

Public Key Cryptography: RSA

(Ron Rivest, Adi Shamir and Len Adleman)

- Choosing keys:
 - Choose two large prime numbers p, q . (e.g., 1024 bits each)
 - Compute $n = pq, z = (p-1)(q-1)$
 - Choose e (with $e < n$) that has no common factors with z .
 - Choose d such that $ed-1$ is exactly divisible by z .
 - Public key is (n,e) . Private key is (n,d) .
- To encrypt a message, $m (< n)$: do $c = m^e \bmod n$
- To decrypt a received ciphertext, c : do $m = c^d \bmod n$
- Reason: for any m (relatively prime with n)
 - $m^z \bmod n = 1$; therefore $m^{ed-1} \bmod n = 1$
- Another property: $(m^d \bmod n)^e \bmod n = m$
- RSA is much slower than the symmetric key cryptos.

12/5/2011 CSC 257/457 - Fall 2011 13

Principles of Network Security

- Confidentiality: cryptography
- Authentication
- Integrity
- Key distribution and certification

12/5/2011 CSC 257/457 - Fall 2011 14

Authentication: version 1.0

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

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

Failure scenario??
Trudy can simply declare herself to be Alice

12/5/2011 CSC 257/457 - Fall 2011 15

Authentication: version 2.0

Protocol ap2.0: Alice says "I am Alice" and sends her secret password to "prove" it.

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

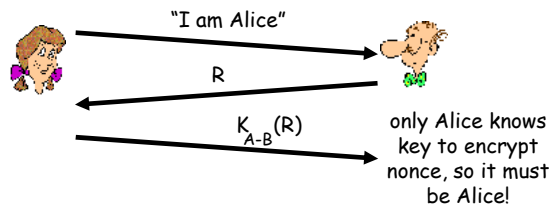
12/5/2011 CSC 257/457 - Fall 2011 16

Authentication: version 3.0

Goal: avoid playback attack

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

ap3.0: Bob sends Alice a **nonce**, R. Alice must return R, encrypted with shared secret key



12/5/2011

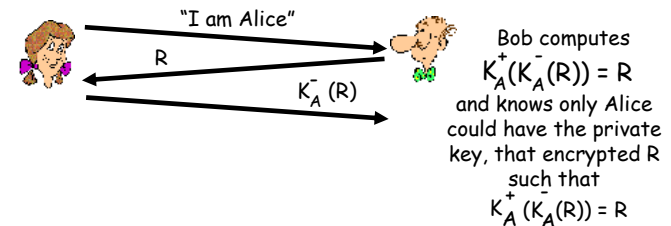
CSC 257/457 - Fall 2011

17

Authentication: version 4.0

ap3.0 requires shared symmetric key. Key distribution can be a problem.

ap4.0: use nonce, public key cryptography.



12/5/2011

CSC 257/457 - Fall 2011

18

Principles of Network Security

- Confidentiality: cryptography
- Authentication
- **Integrity**
- Key distribution and certification

12/5/2011

CSC 257/457 - Fall 2011

19

Integrity

- Digital Signatures:
 - cryptographic technique to ensure document integrity.
 - analogous to hand-written signatures.
- Sender (Bob) digitally signs document, establishing he is document owner/creator.
- The recipient (Alice) receives the document and the digital signatures.
- The recipient can be sure that the document is
 - **verifiable:** Bob signed the document.
 - **nonforgeable:** the document hasn't been changed since Bob signed it.

12/5/2011

CSC 257/457 - Fall 2011

20

Digital Signatures

- Bob signs m by encrypting with his private key, creating a digital signature $K_B^-(m)$

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 $K_B^-(m)$

- Suppose Alice receives msg m and its digital signature $K_B^-(m)$
- Alice applies Bob's public key K_B^+ to $K_B^-(m)$ then checks whether $K_B^+(K_B^-(m)) = m$.
- If so, whoever signed m must have used Bob's private key.

Problem: computationally expensive to public-key-encrypt long messages.

12/5/2011 CSC 257/457 - Fall 2011 21

Signed Message Digest

Bob sends digitally signed (small) message digest:

Alice verifies signature and integrity of digitally signed message:

Bob's private key K_B^-

large message m

H: Hash function

$H(m)$

digital signature (encrypt)

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

Bob's private key K_B^-

large message m

H: Hash function

$H(m)$

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

Bob's public key K_B^+

digital signature (decrypt)

$H(m)$

equal ?

12/5/2011 CSC 257/457 - Fall 2011 22

Message Digests

- Apply a hash function H to m , get a much smaller message digest $H(m)$.
- Public-key-encrypt the message digest to generate the digital signature $K_B^-(H(m))$.

Good/bad hash functions?

- Hint:** given a hash function, it is possible for many messages sharing the same digest.

12/5/2011 CSC 257/457 - Fall 2011 23

Internet Checksum: Poor Hash Function for Generating Message Digests

Given a message and its Internet checksum, it is easy to find another message with same checksum.

| message | ASCII format | message | ASCII format |
|---------|--------------|----------------|--------------------|
| I O U 1 | 49 4F 55 31 | I O U <u>9</u> | 49 4F 55 <u>39</u> |
| 0 0 . 9 | 30 30 2E 39 | 0 0 . <u>1</u> | 30 30 2E <u>31</u> |
| 9 B O B | 39 42 D2 42 | 9 B O B | 39 42 D2 42 |

B2 C1 D2 AC — different messages — B2 C1 D2 AC
but identical checksums!

Hash function property: given digest x for message m , computationally infeasible to find another message m' that shares the same digest.

12/5/2011 CSC 257/457 - Fall 2011 24

Good Hash Functions for Generating Message Digests

- **MD5**
 - computes 128-bit message digest in 4-step process.
 - appears difficult to construct message m whose MD5 hash is equal to x .
- **SHA-1**
 - [NIST, FIPS PUB 180-1]
 - 160-bit message digest

12/5/2011

CSC 257/457 - Fall 2011

25

Principles of Network Security

- Confidentiality: cryptography
- Authentication
- Integrity
- Key distribution and certification

12/5/2011

CSC 257/457 - Fall 2011

26

Key Distribution and Certification

Symmetric key distribution problem:

- How do Alice and Bob establish shared secret key over network without Trudy's knowledge?

Public key distribution 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?

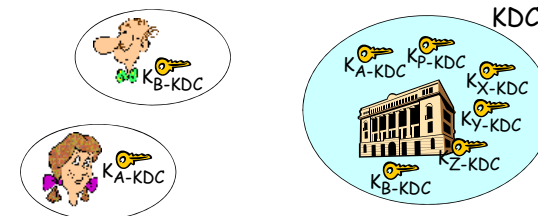
12/5/2011

CSC 257/457 - Fall 2011

27

Secret Key Distribution: Key Distribution Center (KDC)

- **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.



12/5/2011

CSC 257/457 - Fall 2011

28

Key Distribution using KDC

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

Alice knows $R1$

Bob knows to use $R1$ to communicate with Alice

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

12/5/2011 CSC 257/457 - Fall 2011 29

Security vulnerability with public key distribution

A case example for public key-based authentication.

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

What if Bob doesn't know Alice's public key ahead of time?

12/5/2011 CSC 257/457 - Fall 2011 30

Security vulnerability when public keys are not well known

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

Trudy gets $K_T^+(m)$

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

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

12/5/2011 CSC 257/457 - Fall 2011 31

Public Key Distribution: Certification Authorities

- **Certification authority (CA):** trustable by everyone; every one knows its public key.
- 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 is CA-signed document saying "E's public key is ..."

Bob's public key K_B^+

Bob's identifying information

CA private key K_{CA}^-

digital signature (encrypt)

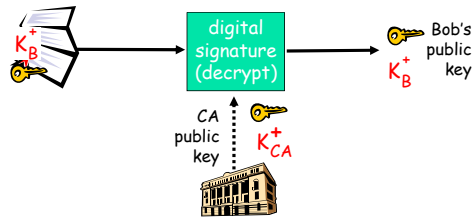
certificate for Bob's public key, signed by CA

12/5/2011 CSC 257/457 - Fall 2011 32

Certification Authorities (cont.)

When Alice wants to verify Bob's public key:

- gets Bob's certificate (Bob or elsewhere).
- apply CA's public key to Bob's certificate, verify Bob's public key.



12/5/2011

CSC 257/457 - Fall 2011

33

Certification Chain

- Public key certificate signed by a certification authority
- Peer certification:
 - If A knows B personally, they can verify each other's public keys using offline means and sign them;
- Certificate chain leading to a certification authority
 - CA signs A's public key certificate
 - A signs B's public key certificate
 - B signs C's public key certificate
 ⇒ Do we then trust C's certificate?

12/5/2011

CSC 257/457 - Fall 2011

34

Summary: Principles of Network Security

Cryptography:

- symmetric keys: protocols? weakness?
- public keys: protocol? weakness?

Confidentiality:

- only sender, intended receiver should "understand" message contents

Authentication:

- sender, receiver want to confirm identity of each other

Message Integrity:

- sender, receiver want to ensure message not altered (in transit, or afterwards)

Key distribution?

12/5/2011

CSC 257/457 - Fall 2011

35

Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).

12/5/2011

CSC 257/457 - Fall 2011

36