

Principles of Network Security

Kai Shen

Dept. of Computer Science, University of Rochester

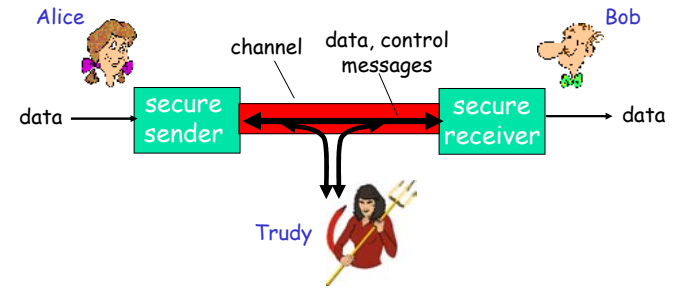
12/3/2007

CSC 257/457 - Fall 2007

1

The Network Security Model

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



12/3/2007

CSC 257/457 - Fall 2007

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!

12/3/2007

CSC 257/457 - Fall 2007

3

What can an adversary do?

- **eavesdrop**: understand the content of messages
- actively **changing** messages
- **impersonation**: can fake (spoof) source address in packet (or any field in packet)
- **denial of service**: prevent service from being used by others (e.g., by overloading resources)

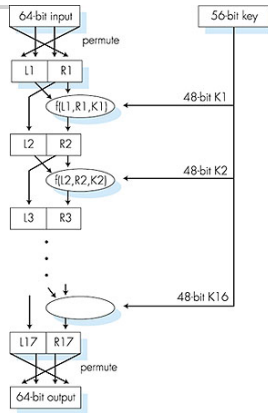
12/3/2007

CSC 257/457 - Fall 2007

4

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



12/3/2007

CSC 257/457 - Fall 2007

9

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 128-bit AES

12/3/2007

CSC 257/457 - Fall 2007

10

Public Key Cryptography

symmetric key cryptography

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

public key cryptography

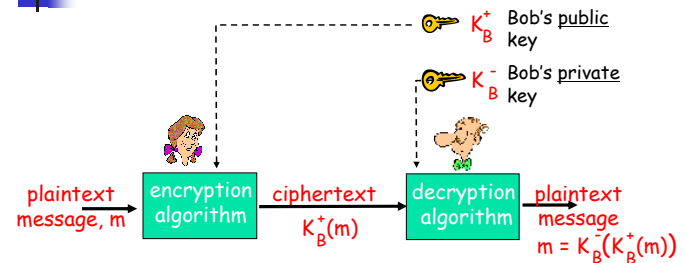
- encryption key is different from decryption key
- encryption key is **public**, known to **everyone**, also called **public key**
- decryption key is **secret**, known only to **receiver**, also called **private key**

12/3/2007

CSC 257/457 - Fall 2007

11

Public Key Cryptography



Principle for choosing the public/private key pair:
One should not be able to derive the private key from the public key.

12/3/2007

CSC 257/457 - Fall 2007

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.

12/3/2007

CSC 257/457 - Fall 2007

13

Outline

- What is network security?
- Confidentiality: cryptography
- Authentication
- Integrity

12/3/2007

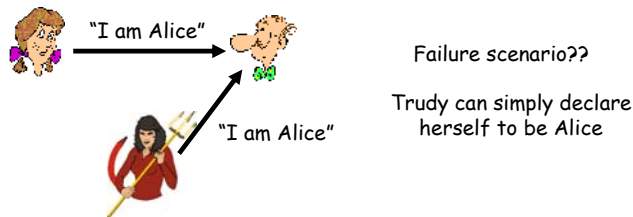
CSC 257/457 - Fall 2007

14

Authentication: version 1.0

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

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



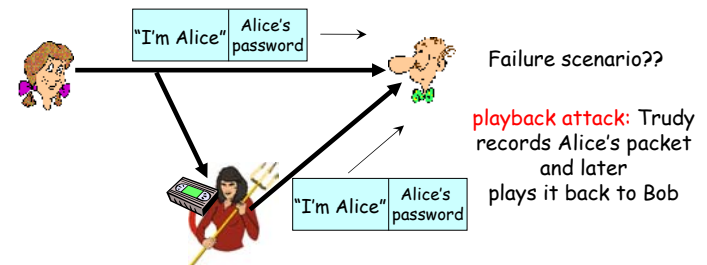
12/3/2007

CSC 257/457 - Fall 2007

15

Authentication: version 2.0

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



12/3/2007

CSC 257/457 - Fall 2007

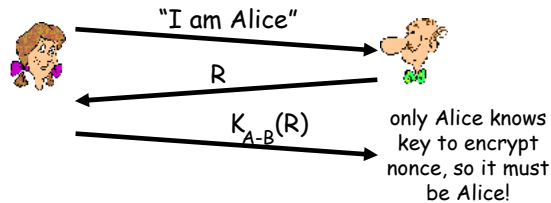
16

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



12/3/2007

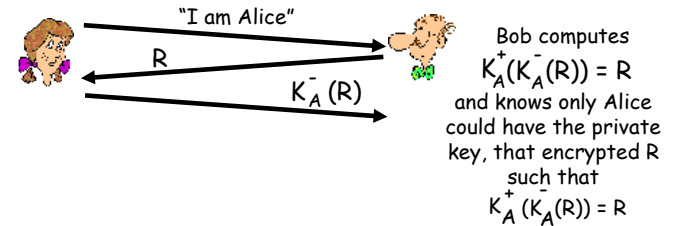
CSC 257/457 - Fall 2007

17

Authentication: version 4.0

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

ap4.0: use nonce, public key cryptography.



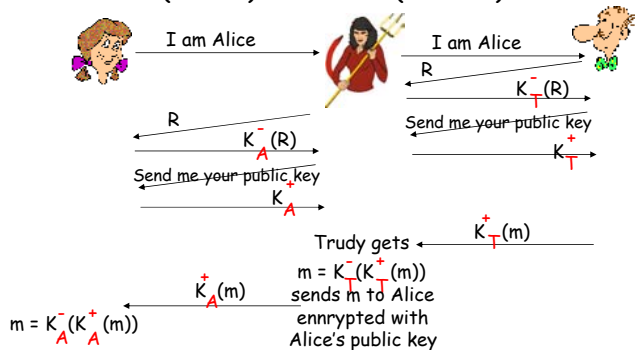
12/3/2007

CSC 257/457 - Fall 2007

18

Security hole when public keys are not well known

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



12/3/2007

CSC 257/457 - Fall 2007

19

Outline

- What is network security?
- Confidentiality: cryptography
- Authentication
- Integrity

12/3/2007

CSC 257/457 - Fall 2007

20

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.

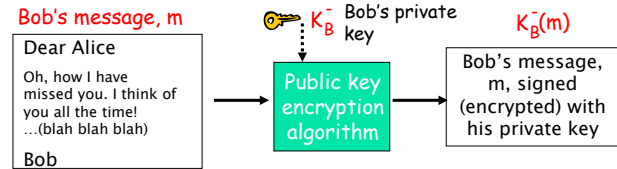
12/3/2007

CSC 257/457 - Fall 2007

21

Digital Signatures

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



- 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.

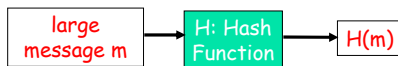
12/3/2007

CSC 257/457 - Fall 2007

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 $K_B^-(H(m))$.



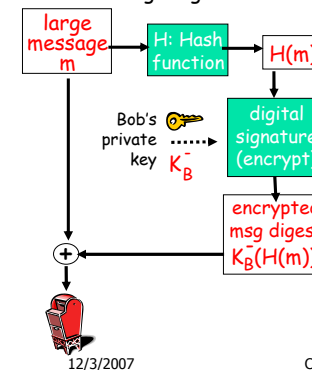
12/3/2007

CSC 257/457 - Fall 2007

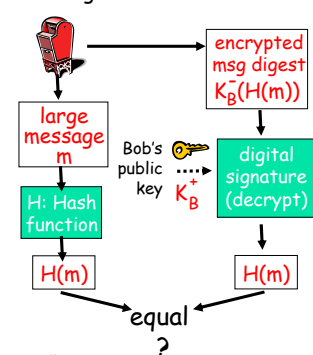
23

Digital signature = signed message digest

Bob sends digitally signed message digest:



Alice verifies signature and integrity of digitally signed message:

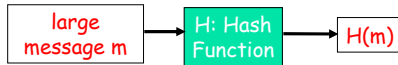


CSC 257/457 - Fall 2007

24

Message Digests: good/bad hash function

- 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 $K_B(H(m))$.



- Note:** it is possible for many messages sharing the same digest.

12/3/2007

CSC 257/457 - Fall 2007

25

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		B2 C1 D2 AC

different messages but identical checksums!

Hash function property: given message digest x for message m , computationally infeasible to find another message m' such that $x = H(m')$.

12/3/2007

CSC 257/457 - Fall 2007

26

Good Hash Functions for Generating Message Digests

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

12/3/2007

CSC 257/457 - Fall 2007

27

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).

12/3/2007

CSC 257/457 - Fall 2007

28