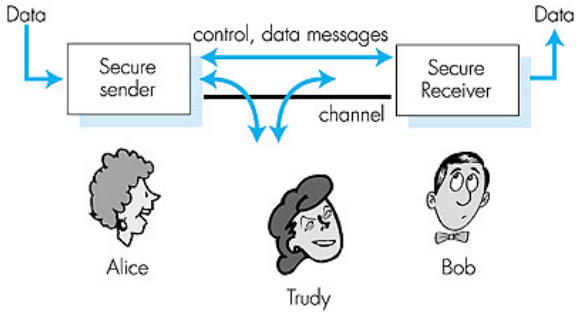
Chapter 7: Network security

Foundations:

- what is security?
- cryptography
- authentication
- message integrity
- key distribution and certification

Friends and enemies: Alice, Bob, Trudy



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

What is network security?

Secrecy: only sender, intended receiver should "understand" msg contents

- sender encrypts msg
- receiver decrypts msg

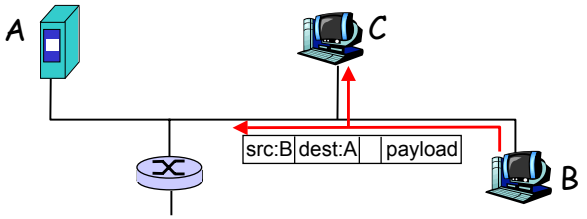
Authentication: sender, receiver want to confirm identity of each other

Message Integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

Internet security threats

Packet sniffing:

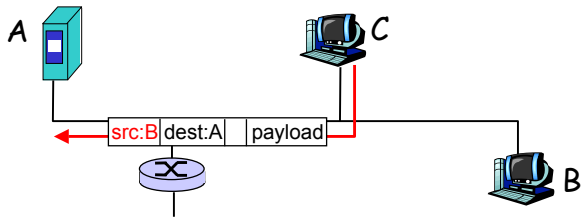
- broadcast media
- *promiscuous* NIC reads all packets passing by
- can read all unencrypted data (e.g. passwords)
- e.g.: C sniffs B's packets



Internet security threats

IP Spoofing:

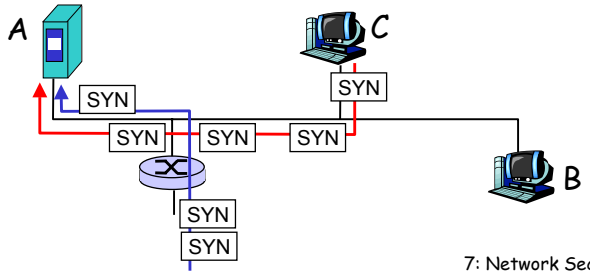
- can generate "raw" IP packets directly from application, putting any value into IP source address field
- receiver can't tell if source is spoofed
- e.g.: C pretends to be B



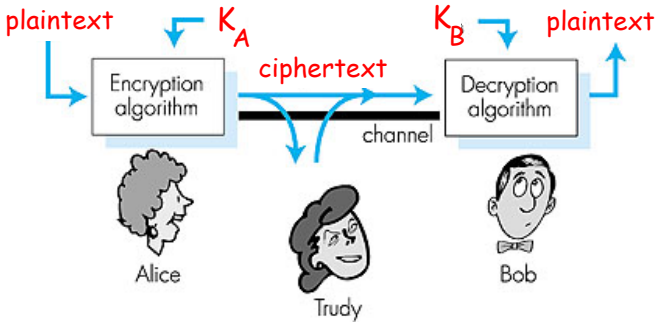
Internet security threats

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



The language of cryptography



symmetric key crypto: sender, receiver keys identical

public-key crypto: encrypt key *public*, decrypt key *secret*

Symmetric key cryptography

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

```
plaintext:  abcdefghijklmnopqrstuvwxyz
            ↓                       ↓
ciphertext: mnbvcxzasdfghjklpoiuytrewq
```

E.g.:

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

Q: How hard 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

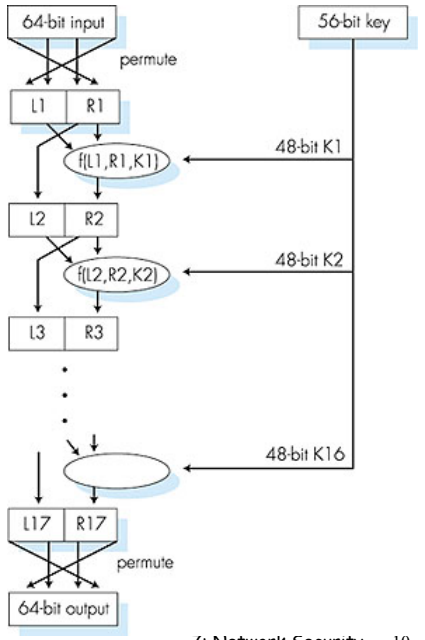
Symmetric key crypto: DES

DES operation

initial permutation

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

final permutation



Public Key Cryptography

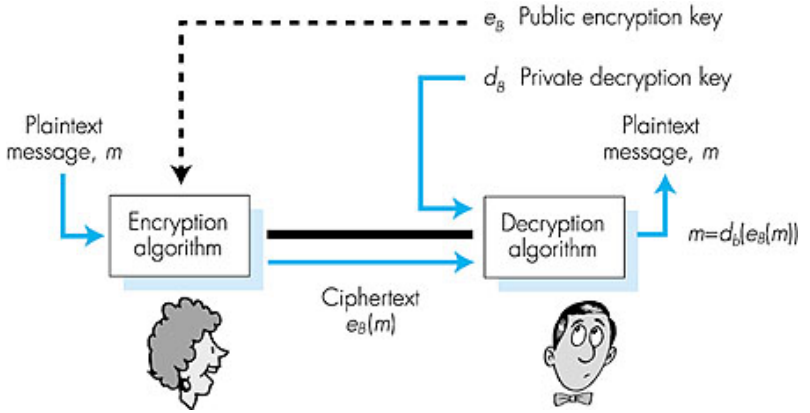
symmetric key crypto

- requires sender, receiver 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
- encryption key *public* (known to *all*)
- decryption key private (known only to receiver)

Public key cryptography



Public key encryption algorithms

Two inter-related requirements:

1. need $d_B(\cdot)$ and $e_B(\cdot)$ such that
   $d_B(e_B(m)) = m$
2. need public and private keys for $d_B(\cdot)$ and $e_B(\cdot)$

RSA: Rivest, Shamir, Adelson algorithm

RSA: Choosing keys

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* mod *z* = *1*).
5. *Public* key is *(n,e)*. *Private* key is *(n,d)*.

RSA: Encryption, decryption

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

Magic happens! $m = (m^e \bmod n)^d \bmod n$

RSA example:

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

	letter	m	m^e	$c = m^e \bmod n$
encrypt:	l	12	1524832	17

	c	c^d	$m = c^d \bmod n$	letter
decrypt:	17	481968572106750915091411825223072000	12	l

RSA: Why: $m = (m^e \bmod n)^d \bmod n$

Number theory result: If p,q prime, $n = pq$, then

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(using number theory result above)

$$= m^1 \bmod n$$

(since we chose ed to be divisible by $(p-1)(q-1)$ with remainder 1)

$$= m$$