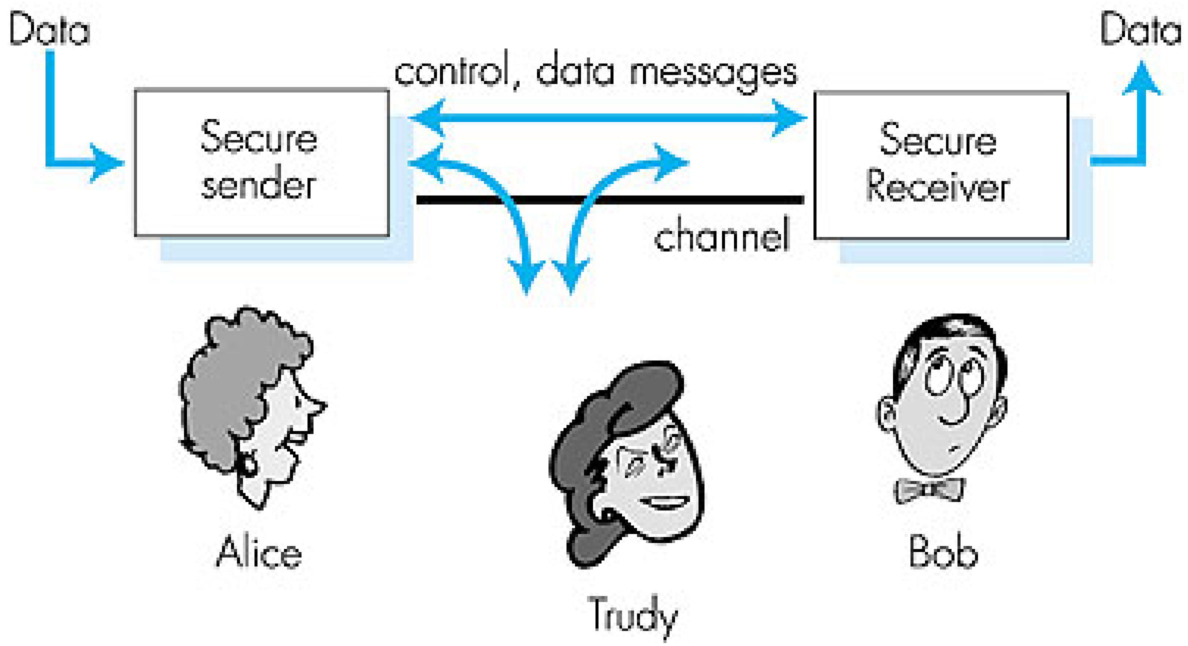
Lecture 8
Network Security Basics

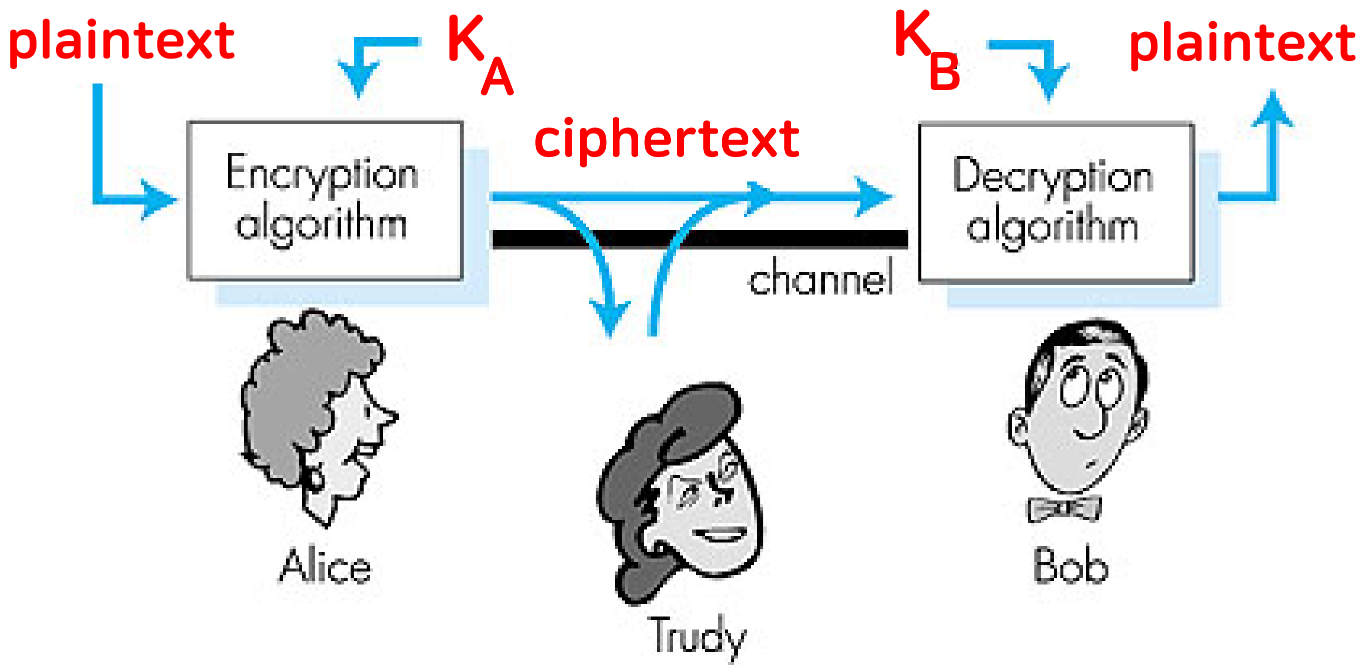
Symmetric Key Cryptography
Asymmetric Key Cryptography
    Public Key Cryptography
    RSA

Friends and Enemies: Alice, Bob, Trudy



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

The language of cryptography



Symmetric key crypto: sender, receiver keys identical

Asymmetric key crypto: sender key ≠ receiver key

(ex) 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

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

Symmetric key cryptography

DES: Data Encryption Standard

- US encryption standard [NIST]
- 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

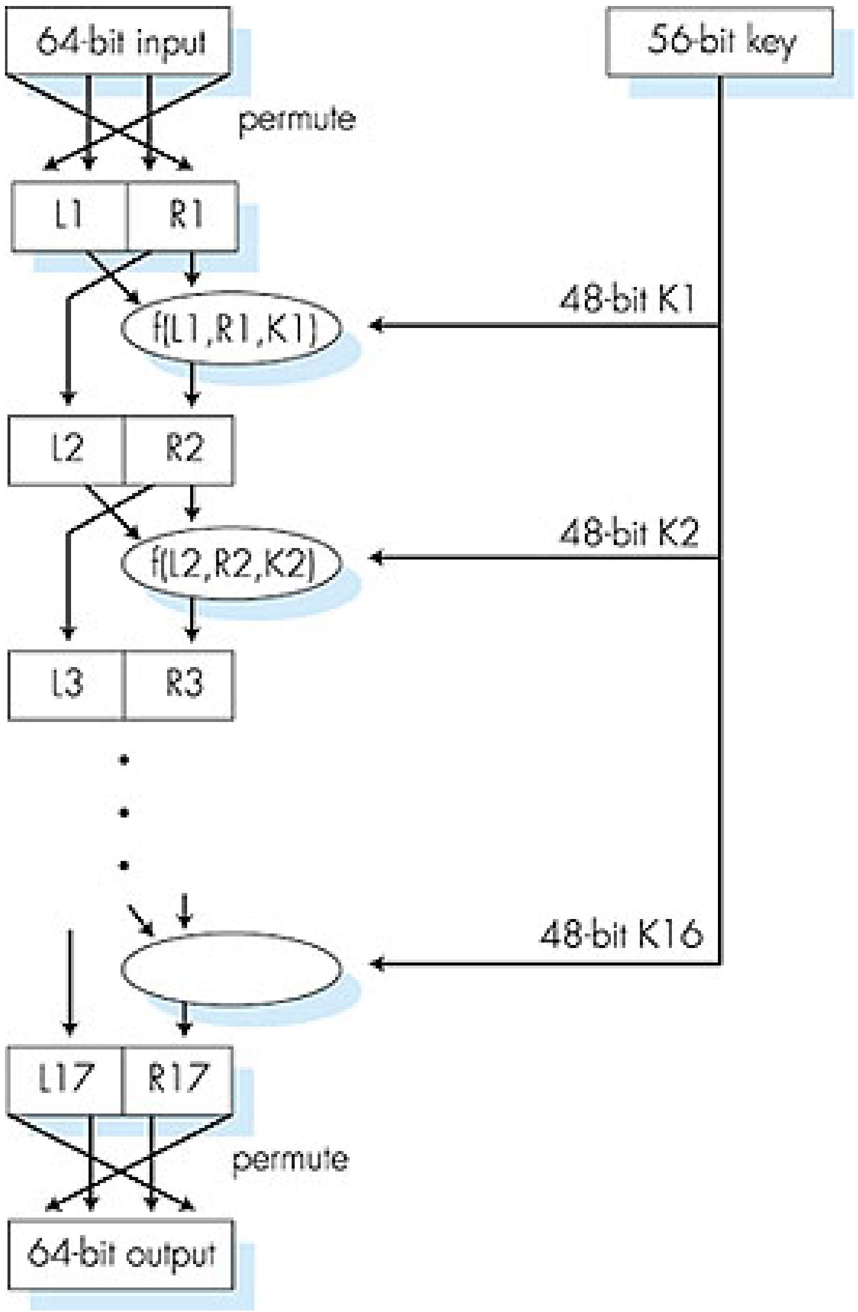
➔ AES: Advanced Encryption Standard [NIST]

Symmetric key crypto : DES

DES operation

1. initial permutation
2. 16 identical "rounds" of function application, each using different 48 bits of key
3. final permutation

➔ **AES** (Advanced Encryption Standard)



Public Key Cryptography

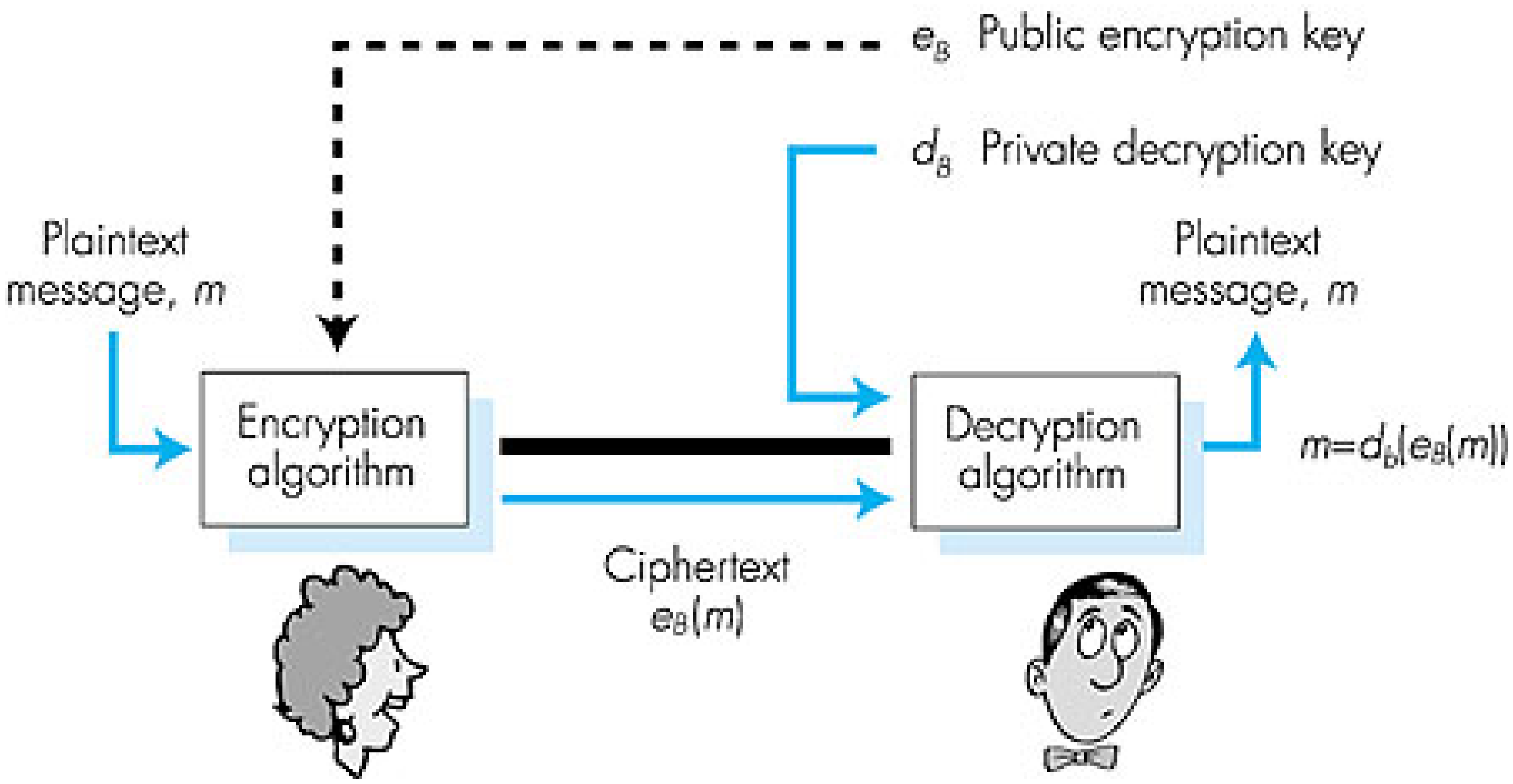
symmetric key crypto

- 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: $ed \bmod z = 1$).

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 and z* are relatively prime).

$d = 29$ (so $ed - 1$ exactly divisible by z).

encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	248832	17

decrypt:

c	c^d	$m = c^d \bmod n$	letter
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)

$$= m$$