CS 348: Computer Networks

- Security; 30th - 31st Oct 2012

Instructor: Sridhar Iyer
IIT Bombay

Network security

- Security Plan (RFC 2196)
  - Identify assets
  - Determine threats
  - Perform risk analysis
  - Implement security mechanisms
  - Monitor events, handle incidents

- Cost of protecting should be less than the cost of recovering

Security requirements

- Confidentiality:
  - No unauthorized disclosure
- Integrity:
  - No unauthorized modification
- Authentication:
  - Assurance of identity of originator
- Non-Repudiation:
  - Originator cannot deny sending the message

Security threats and levels

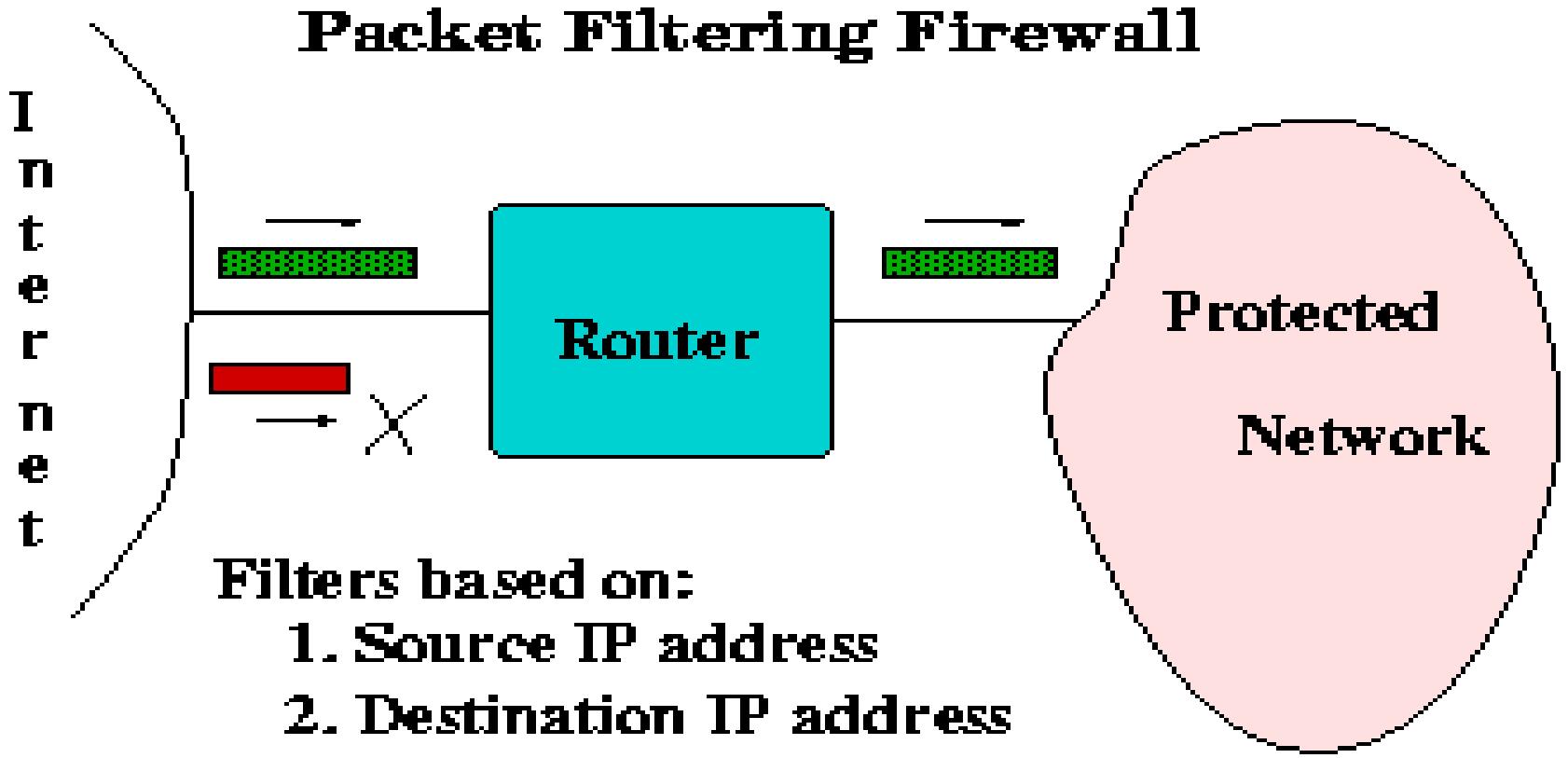
- Threats
  - Host: Unauthorized access
  - Transmission: sniffing, masquerading
- Host level
  - Authentication and access control
- Network level
  - Firewalls and proxies
- Application level
  - Encryption and signatures

Firewalls

- Control the flow of traffic between the Internet and internal networks and systems
- Like a guard post in the lobby of a building

- Single “choke point” is easier to control/defend from outside hackers (and inside spies!)

Packet filtering



Default deny vs. Default allow

Sample filtering rules

- Permit incoming Telnet sessions only to a specific internal hosts
- Permit all outbound Telnet and FTP sessions
- Deny all incoming traffic from specific external networks

Sample /etc/hosts.allow

This file describes the names of the hosts which are allowed to use the local INET services, as decided by the '/usr/sbin/tcpd' server

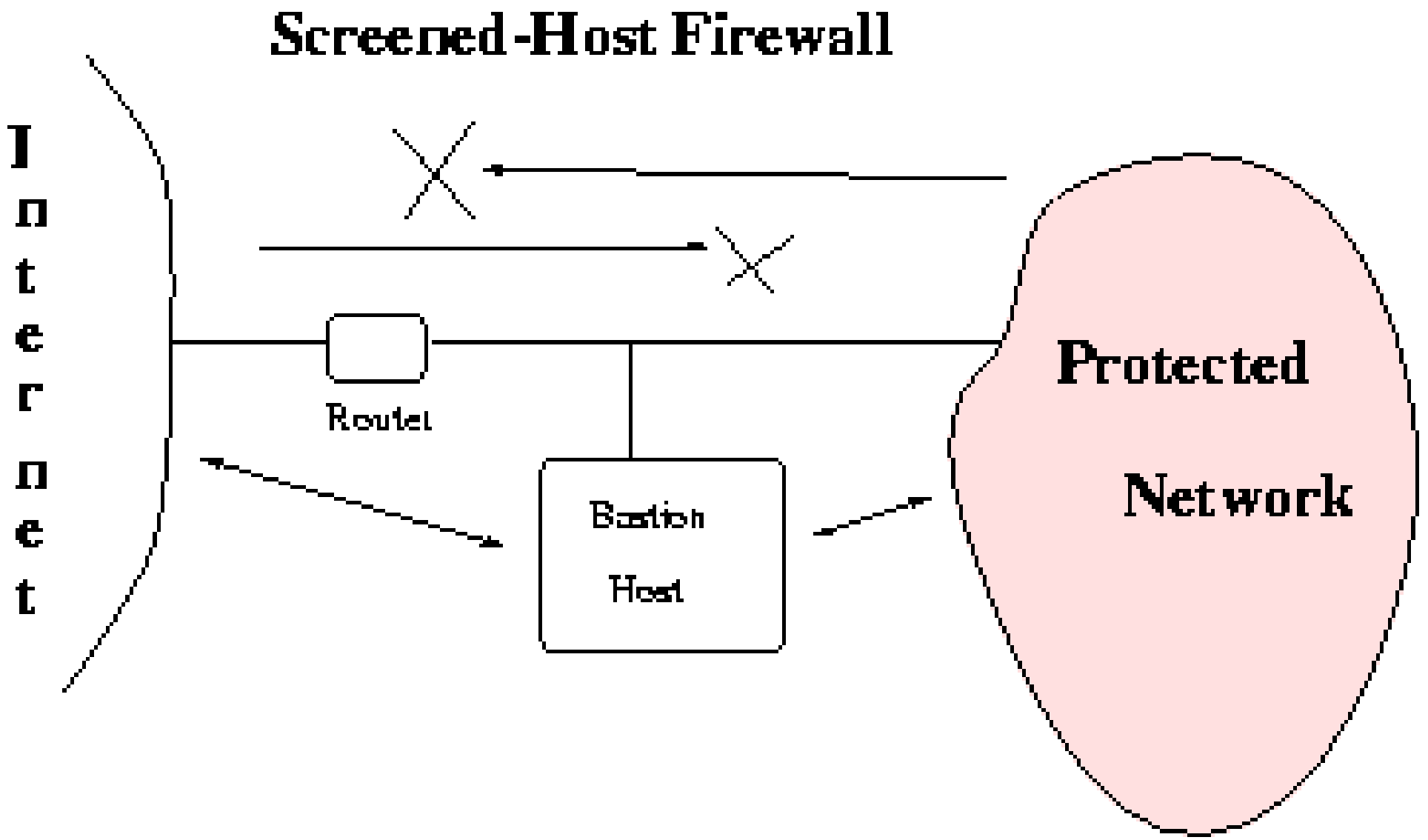
- ALL: 202.54.44.112/255.255.255.240
- ALL: 144.16.111.180
- ALL: 144.16.111.81
- ALL: 144.16.106.218

Sample /etc/hosts.deny

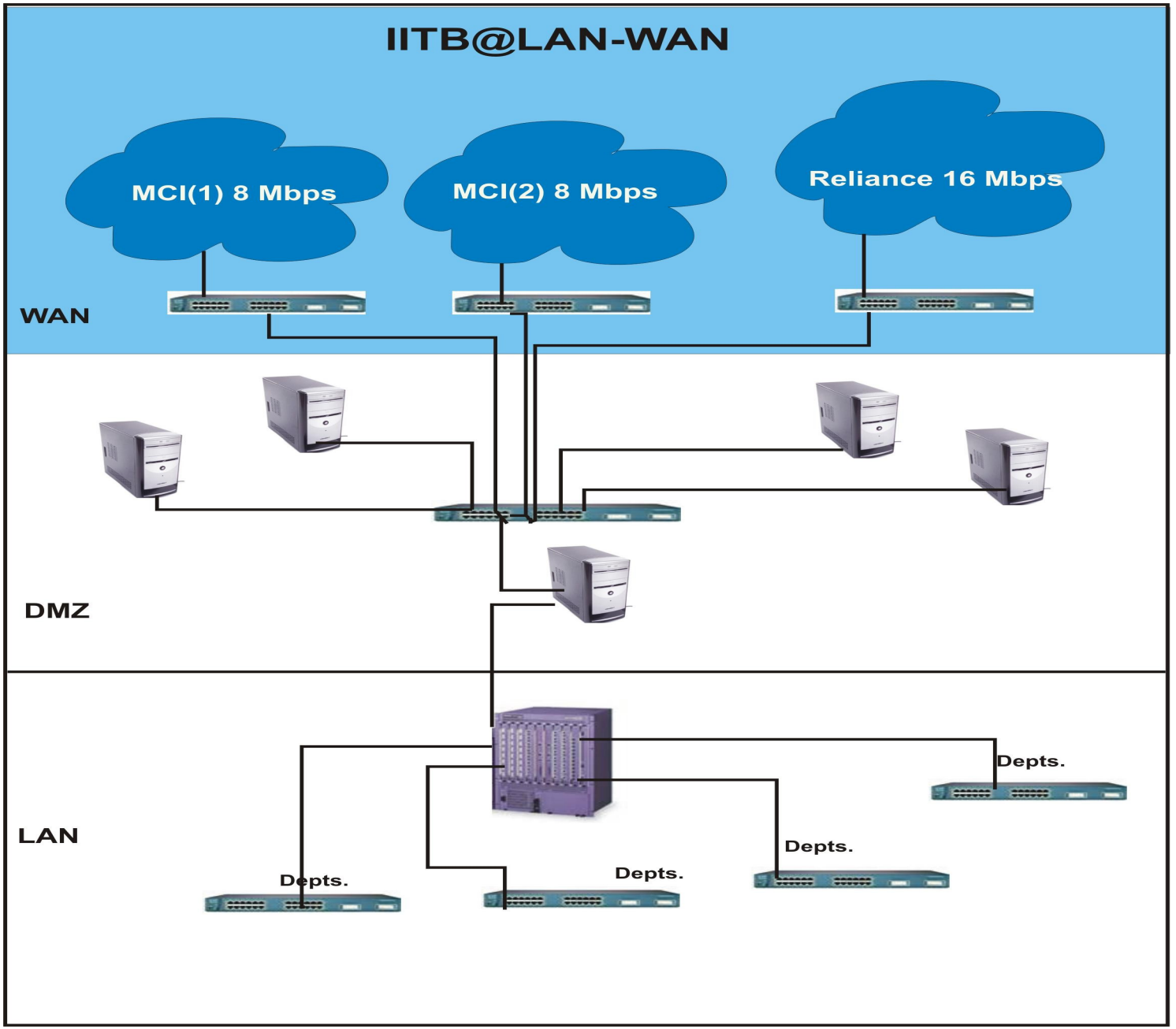
This file describes the names of the hosts which are *not* allowed to use the local INET services, as decided by the '/usr/sbin/tcpd' server

- ALL:ALL

Proxy servers



IITB configuration



Benefits of firewalls/proxy servers

- Internet security can be monitored
  - Audit and log Internet Usage
  - Network Address Translator (NAT) alleviates IP address shortage
  - Central point of contact (email,www,ftp)
  - Caching WWW proxy servers (squid)

Cryptography

Components:

- Plain text
- Encryption/Decryption Algorithms
- Encryption/Decryption Keys
- Cipher text

Encryption: Basic scheme

Text $\oplus$ Key $\rightarrow$ Encrypted Text (Cipher)

$$P \oplus K \rightarrow C$$

Encrypted Text $\oplus$ Key $\rightarrow$ Plain Text

$$C \oplus K \rightarrow P$$

- Plain Text : **We need 20,000 litres of diesel**
- Key : **qW3edkl*&B43@tn,,';[67~]}23#@!h3**
- Encrypted: **bv56*($#@vbgGGHT';[]=+_',.Gfuyrt**

Encryption operator: XOR

$P \oplus K = C$

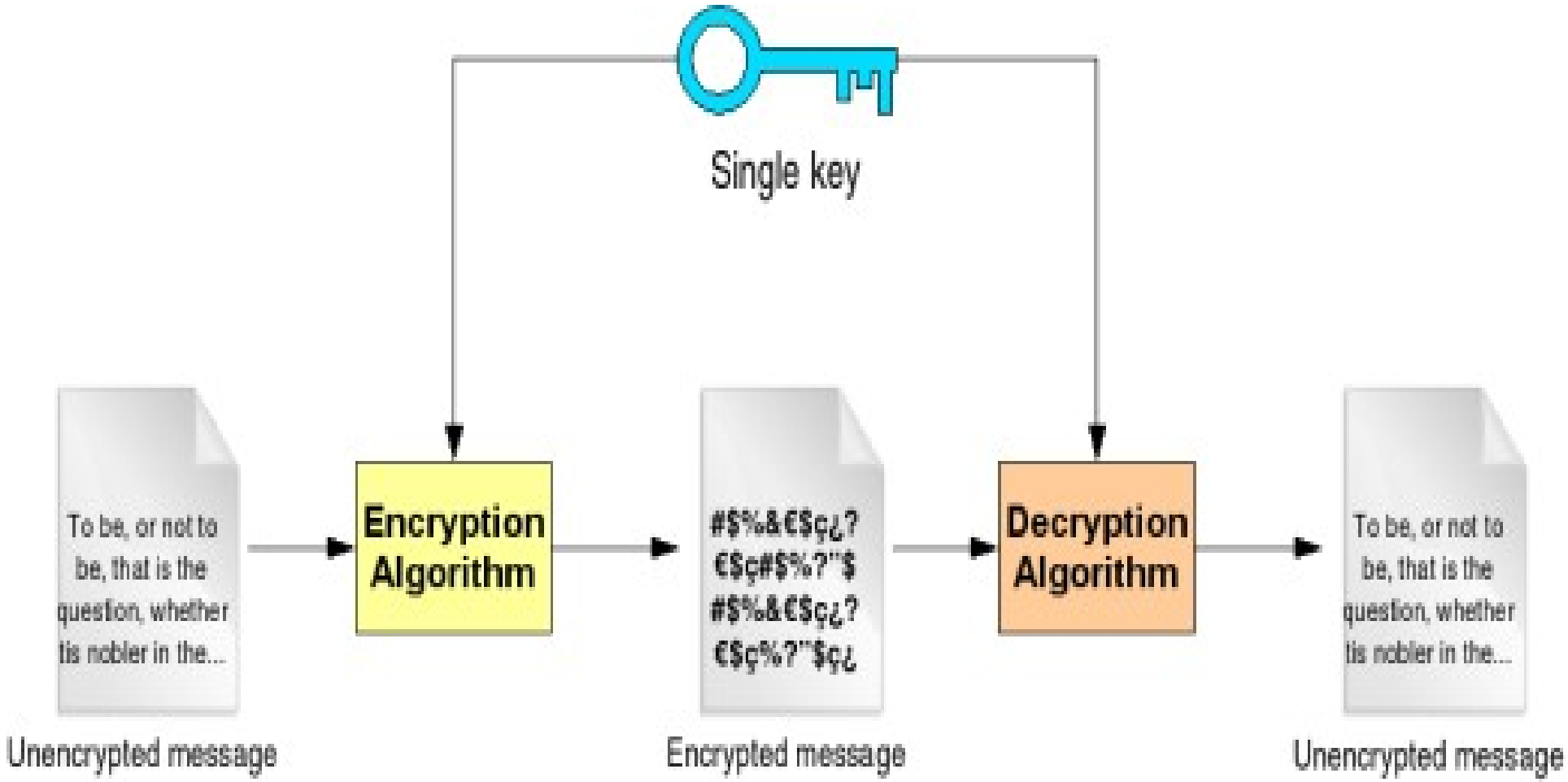
$C \oplus K = P$

P : 110101	C: 011110
$\oplus$ K : 101011	$\oplus$ K: 101011
= C : 011110	= P: 110101

Some cryptography techniques

- Symmetric/Private Key: (DES)
  - secure environment for key exchange
- Asymmetric/Public Key: (RSA)
  - private-public key pair
- Hash Algorithms: (MD5, SHA)
  - message integrity
- Digital Signatures:
  - integrity and authentication

Symmetric key encryption



Source: http://gdp.globus.org/gt4-tutorial/multiplehtml/ch09s02.html

Diffie-Hellman Key exchange

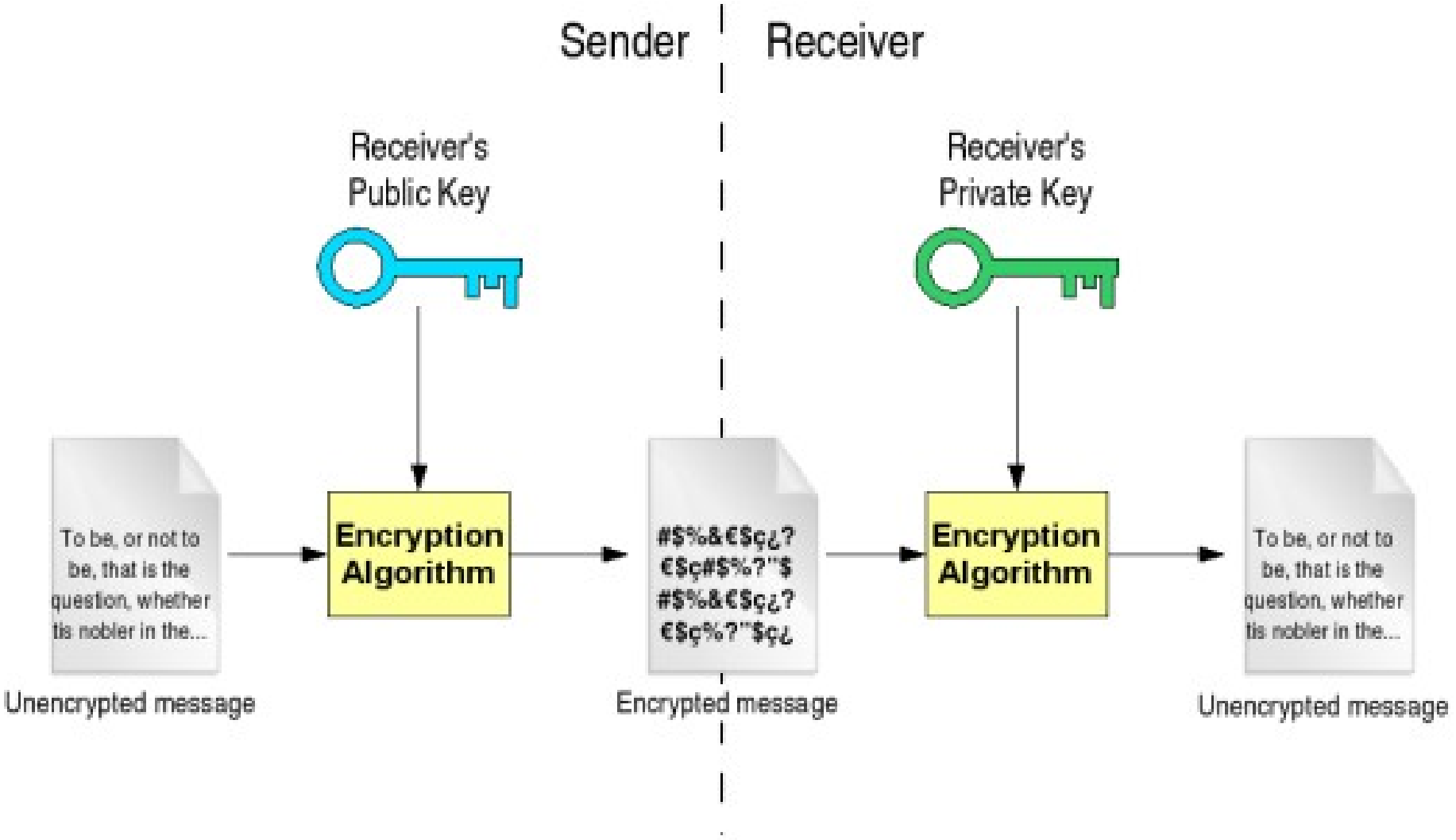
- Establish a shared secret key in public
- A and B agree on two large primes n & g. (n-1)/2 must be prime
- A picks a large (say 512 bit number) x and keeps it secret. B picks y
- A sends to B: $X = g^x \bmod n$
- B sends to A: $Y = g^y \bmod n$

D-H key exchange

- A computes Y^x mod n
- B computes X^y mod n
- **Secret Key is now g^{xy} mod n**

- Suppose C knows n & g. If C could get x and y, C would know the key
- Given only g^x mod n, C **cannot** find x
- no algorithm for computing discrete logarithms modulo a large prime known

Asymmetric key encryption



Source: http://gdp.globus.org/gt4-tutorial/multiplehtml/ch09s03.html

Public-Key cryptography

- Every user is assigned two keys
  - a **private** key that is known only to user
  - a **public** key that is known to everyone

- Cryptosystem has following properties
  - D(E(P)) = P
  - Exceedingly difficult to deduce D from E
  - E cannot be broken by a “chosen plaintext” attack

Public-Key cryptography

- E_{KU} Encryption using pUblic key
  - KU
- D_{KR} Decryption using pRivate key
  - KR

- P Plain Text
- $D_{KR}(E_{KU}(P)) = P$

Public-key vs. Symmetric key

- Public-key algorithms are slow
- Symmetric algorithms are typically at least 1000 times faster

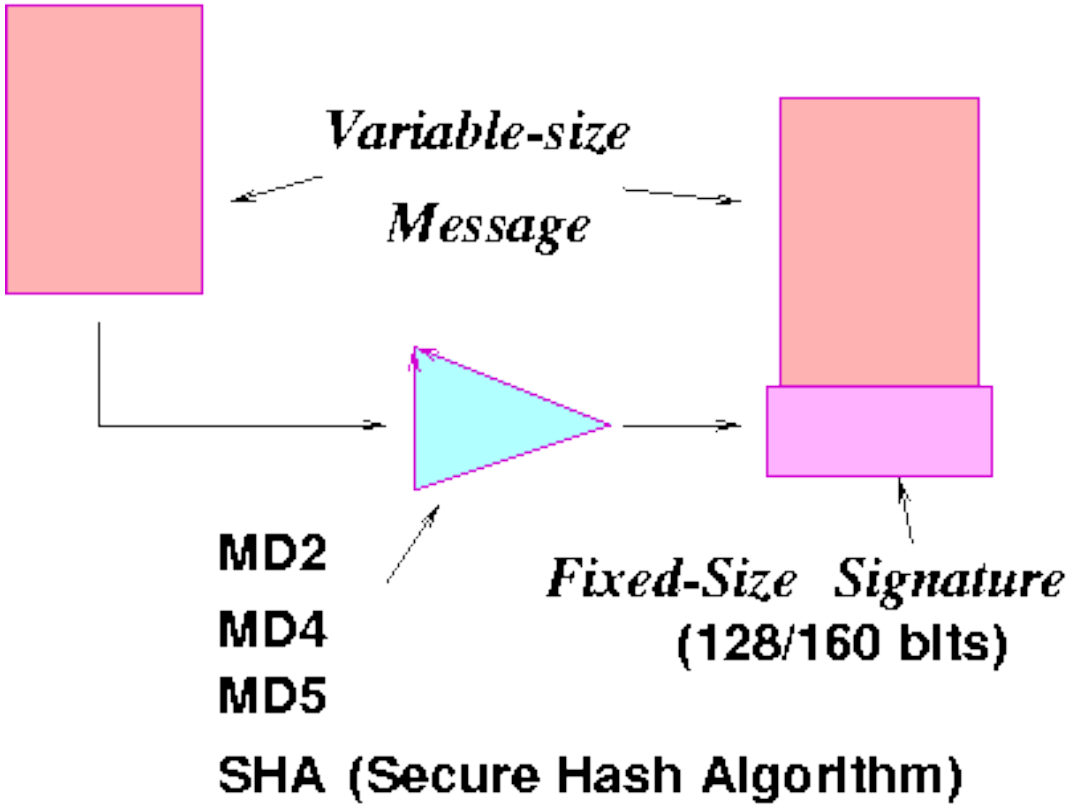
- In practice a public-key system is used to secure and distribute session keys - hybrid cryptosystem

- Also see:
  - http://en.wikipedia.org/wiki/Public-key_cryptography

HASH Algorithms

(Message Digests, Fingerprints)

For message Integrity

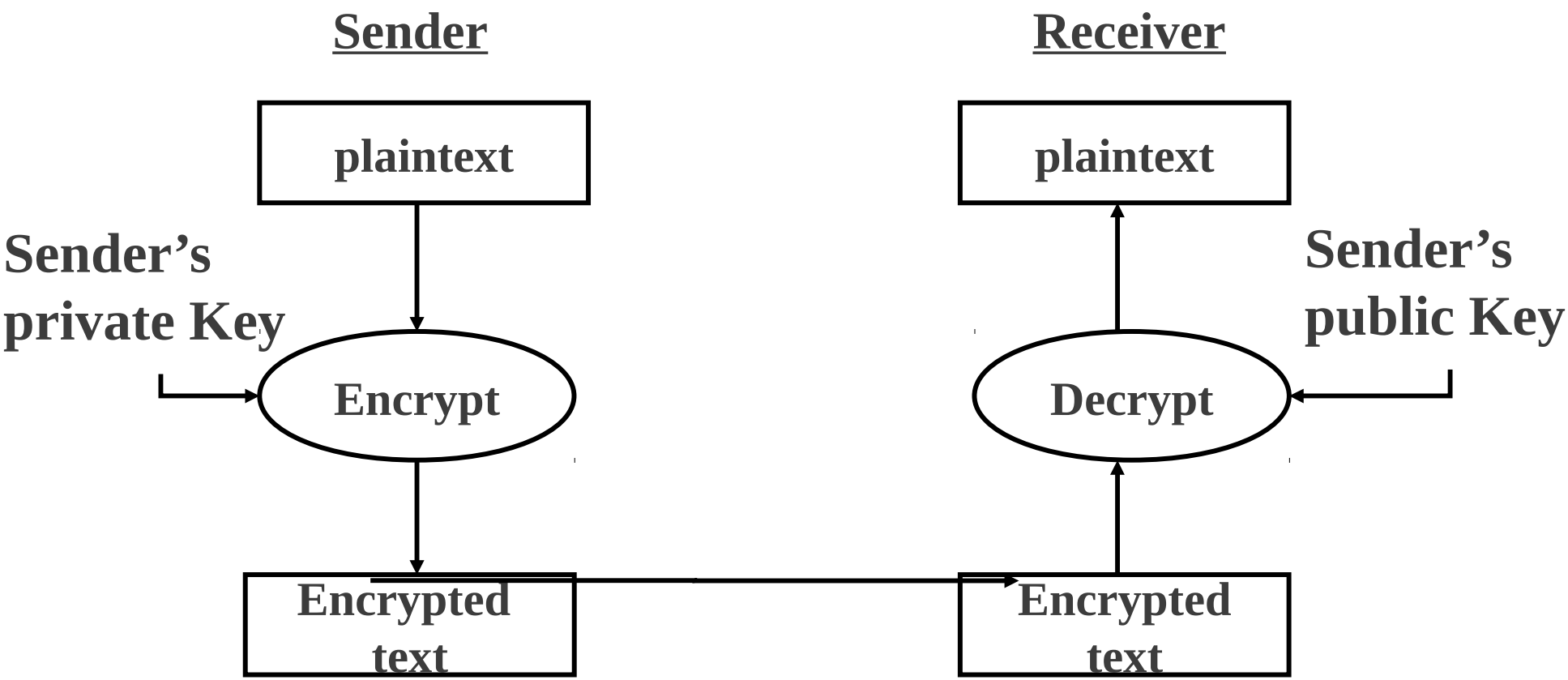


But, can't you easily compute and attach a new signature when you modify/forge a message?

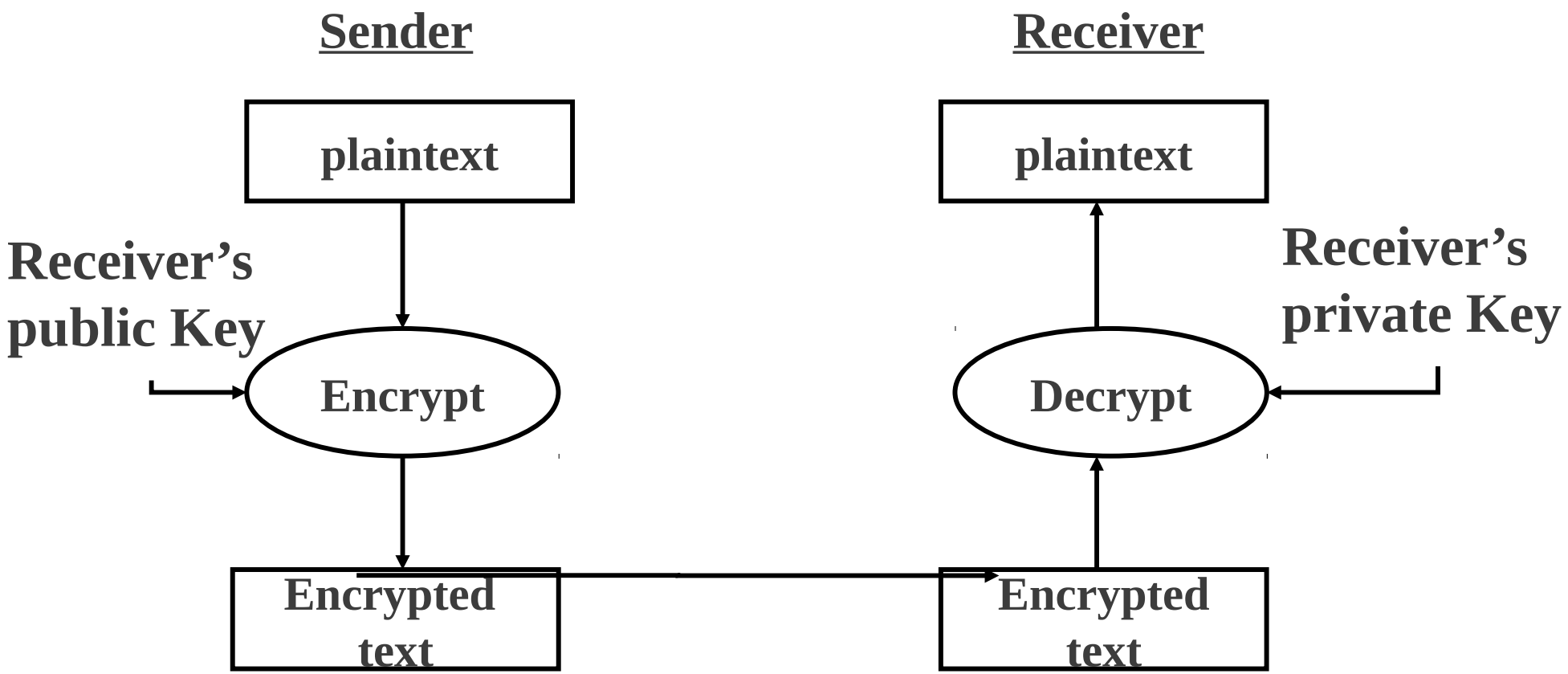
Public-key techniques to the rescue!

Signature should depend on the person signing!

Authentication, No confidentiality



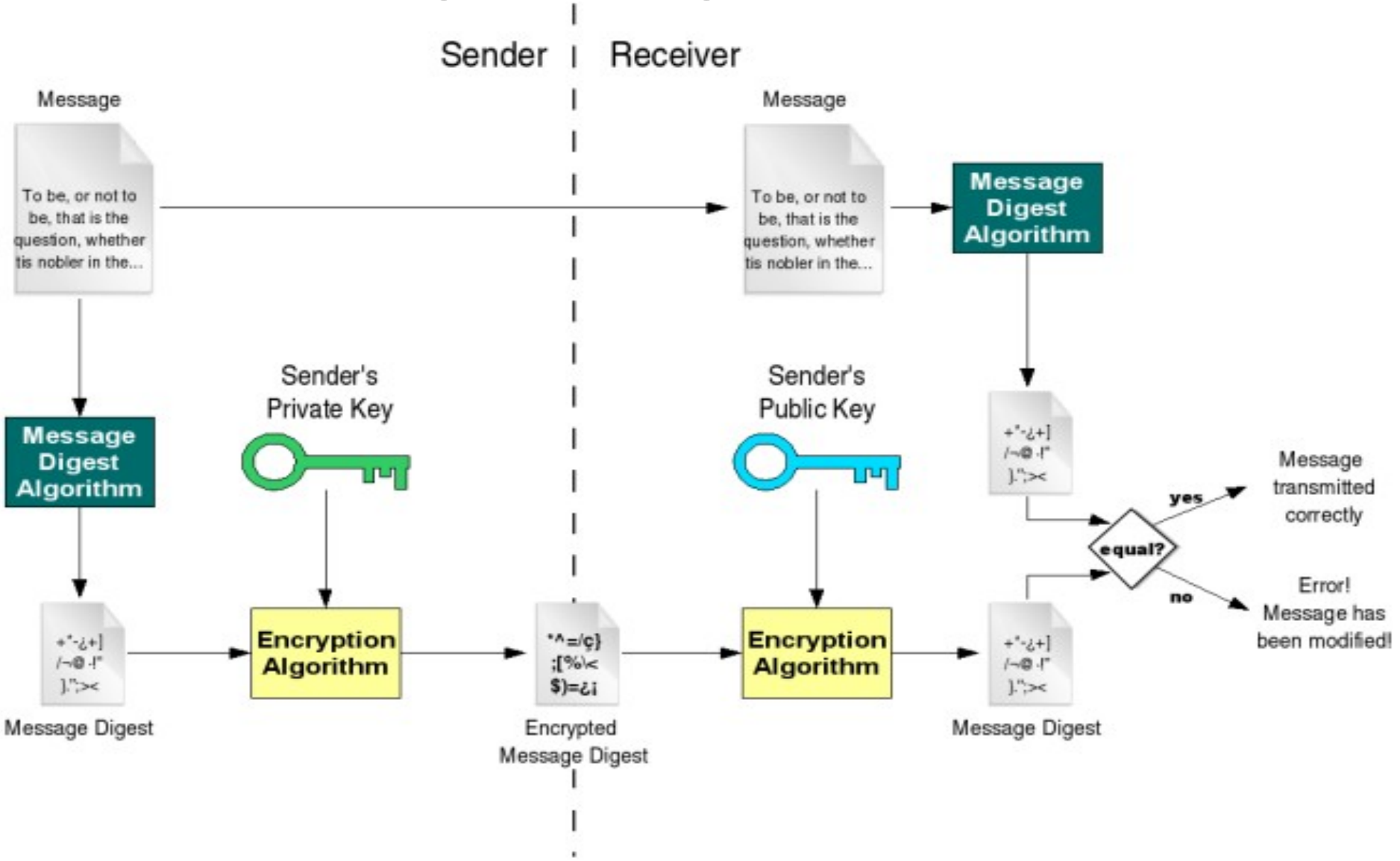
Confidentiality, No Authentication



Digital signatures

- Two approaches can be combined for both confidentiality and authentication
- Sender encrypts message with his/her private key for authentication
- Resulting cipher text is encrypted again using receiver’s public key, for confidentiality
- Receiver first decrypts with private key, then decrypts with senders’ private key

Digital Signatures



Source: http://gdp.globus.org/gt4-tutorial/multiplehtml/ch09s03.html

Encryption in practice

- In practice, for performance reasons
  - all data traffic is encrypted using secret key (symmetric) cryptography
  - public key cryptography is used for the authentication protocols themselves
  - and for establishing a session key

- session keys minimize the traffic which contains the users' secret/public keys and reduces cipher text available to intruder

Security mechanisms

- Confidentiality:
  - Encryption (usually symmetric)
- Integrity:
  - Message digests
- Authentication:
  - Signatures (asymmetric encryption)
- Non-Repudiation:
  - Certificates, Signatures

Secure Shell (SSH)

- Provides secure encrypted communication between two hosts over an insecure network
- Uses public key authentication

- User creates public/private pair
- Server knows the user's public key, and only the user has the private key