

# Network Security

COS 460 & 540



## HOMELAND SECURITY ADVISORY SYSTEM

### DANGER DOG

SEVERE RISK OF  
TERRORIST ATTACKS



### CHICAGO DOG

HIGH RISK OF  
TERRORIST ATTACKS



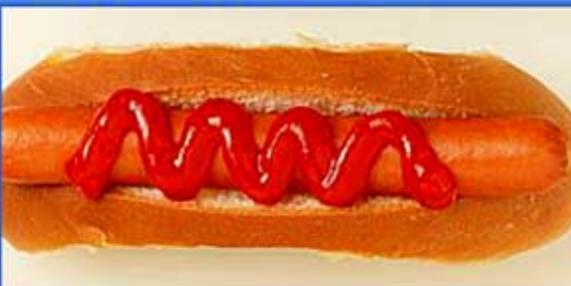
### KOSHER DOG

SIGNIFICANT RISK OF  
TERRORIST ATTACKS



### WIENER

GENERAL RISK OF  
TERRORIST ATTACKS



### TOFU PUP

LOW RISK OF  
TERRORIST ATTACKS



# Threat Types

- Data *integrity* and *confidentiality*
- Endpoint *trust* and *access*
- Denial of *service*
- Denial of *activity*

# Confidentiality

**Confidentiality** means a third-party cannot eavesdrop on the communication.



# Confidentiality



**Traffic Confidentiality** means no one can tell the conversation even happened.

# Confidentiality



## Encryption

# Data Integrity

“CRC and checksum values already protect my data.”

# Checksum & CRC

CRC and checksum do not tell who sent the data, only that it “looks right”



# Data Integrity

- Tampering
- Replay Attacks
- Originality
- Delay (timeliness)

# Tampering

“I only ordered  
one Tribble”



# Replay Attack

## Replay Attack



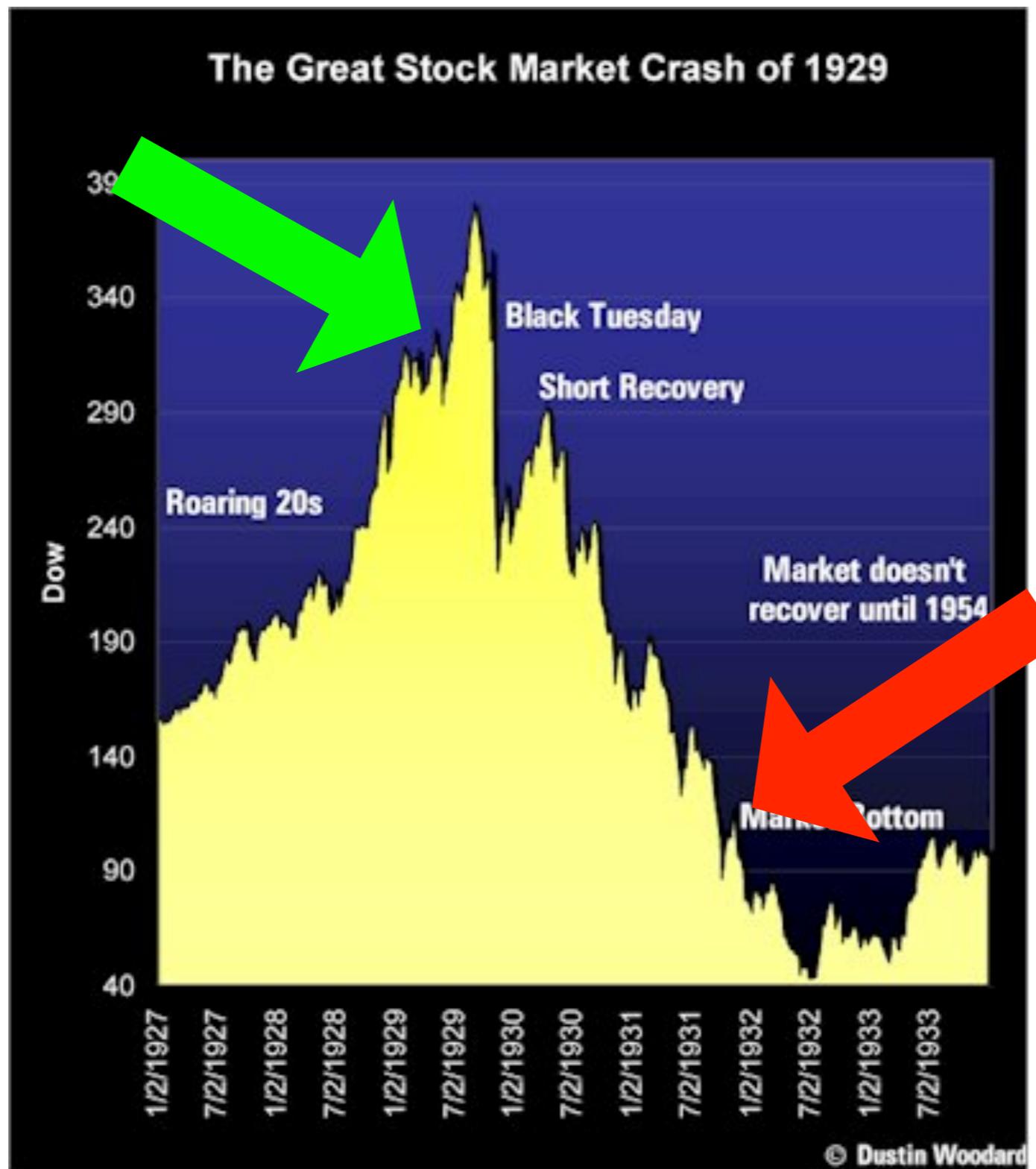
Username: fred

Password: lisa

8



# Delay



# Denial of Service



# Denial of Activity



“These aren’t the droids you’re looking for.”

# Threat Types

- Data *integrity* and *confidentiality*
- Endpoint *trust* and *access*
- Denial of *service*
- Denial of *activity*

# Security Tools

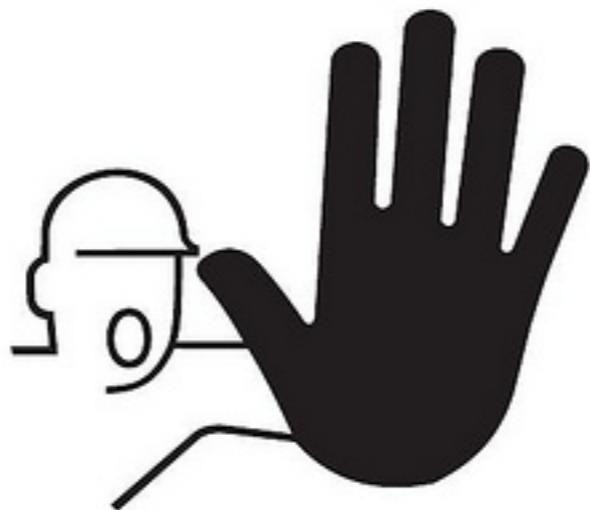
- Authentication & Authorization (OS)
- Cryptography
- Firewalls

# Authentication



# Access Control (Authorization)

**NOTICE**



**AUTHORIZED  
PERSONNEL  
ONLY**

# Cryptography Tools

- Ciphers
- Hashes
- Key Distribution

# Ciphers

- plaintext - the message
- cipher - function that *encrypts*
- ciphertext - the encoded message

# Cipher

Hello  
World



**Axeeh**  
**Phkew**

# Ciphers

- Symmetric-key
- Asymmetric-key

# Symmetric Key

- “Key” shared by all parties
- “Key” used for encryption and decryption
- How do you share the key?
- How do you recover from a lost/stolen key

# Exclusive OR

message 0 | | | 0 0 | | |

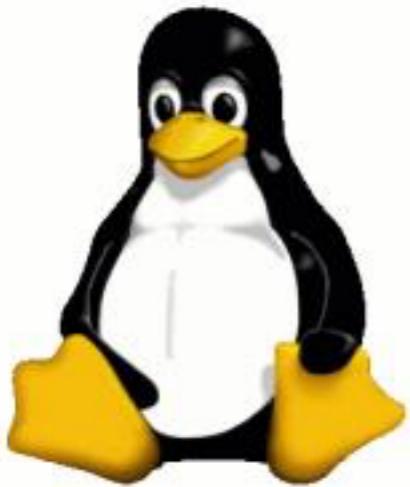
key					0	0	0	0
-----	--	--	--	--	---	---	---	---

cipherT | 0 0 | 0 | | |

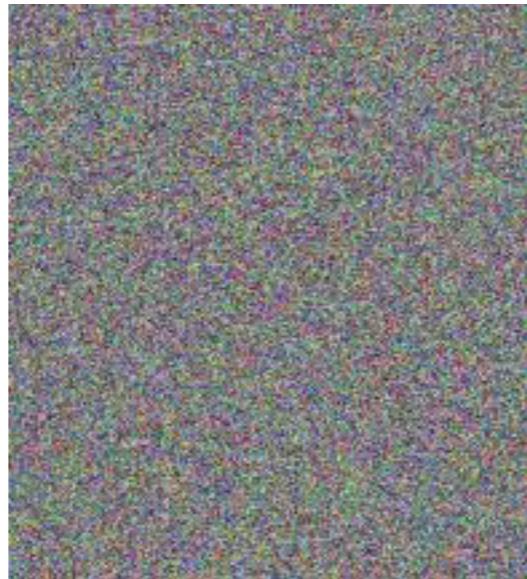
key					0	0	0	0
-----	--	--	--	--	---	---	---	---

message 0 | | | 0 0 | | |

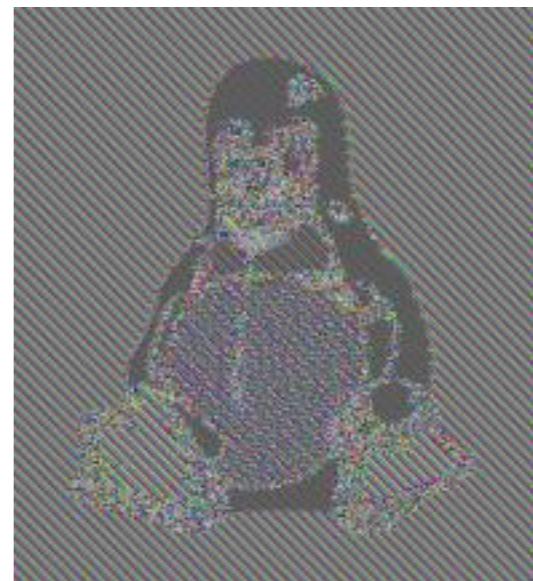
# The Problem



**Original**



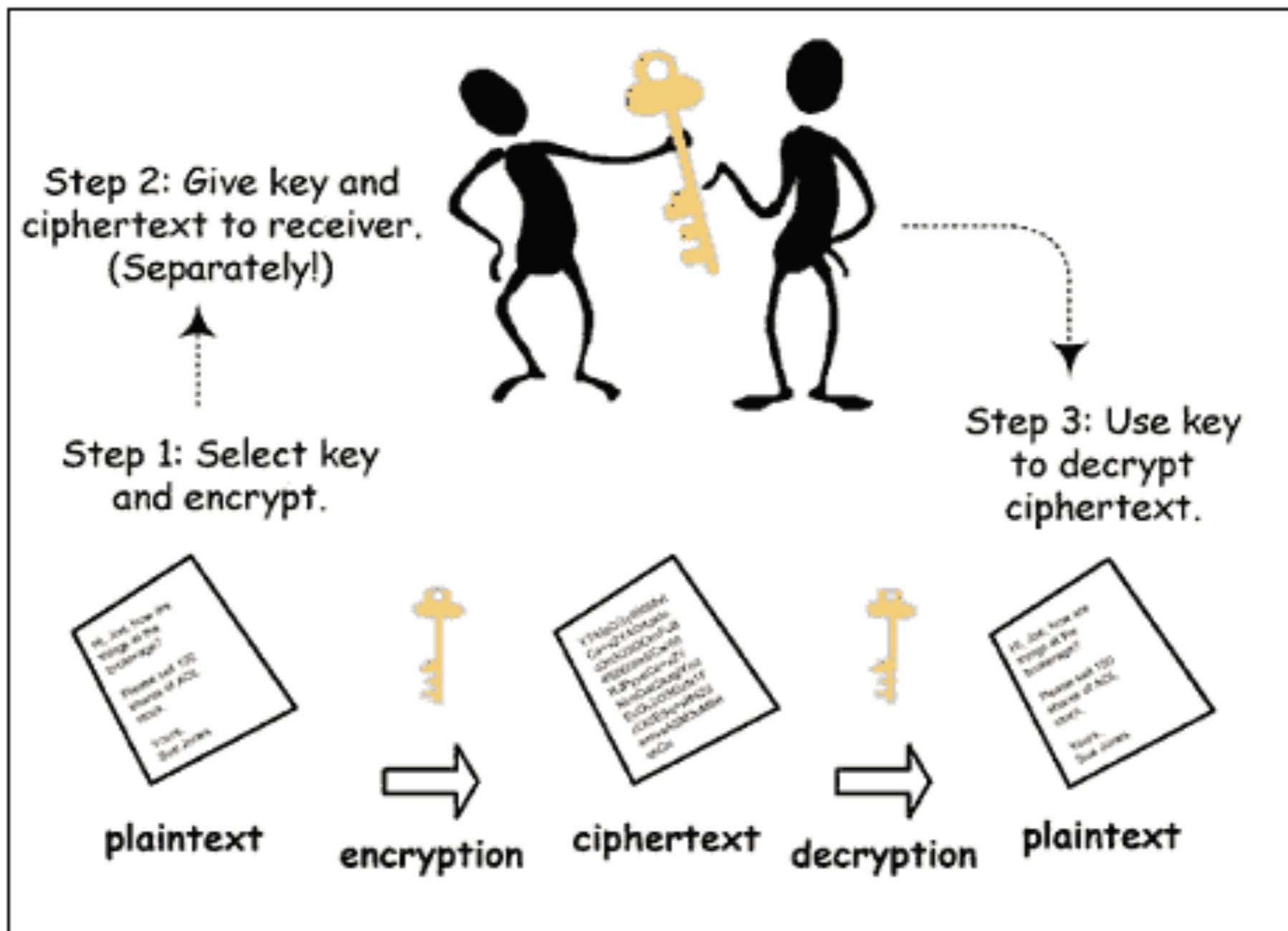
**What we “want”**



**Reality**

[https://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation)

# Symmetric Key



# Key: A=T

Hello  
World



**Axeeh**  
**Phkew**

# Symmetric Key

- Example: Data Encryption Standard (DES)
  - 56 bit key
  - also, Triple DES = DES × 3
  - is running same encryption 3x better?
- Advanced Encryption Standard (AES)
  - 128, 192, 256 bits in key

# Asymmetric Key

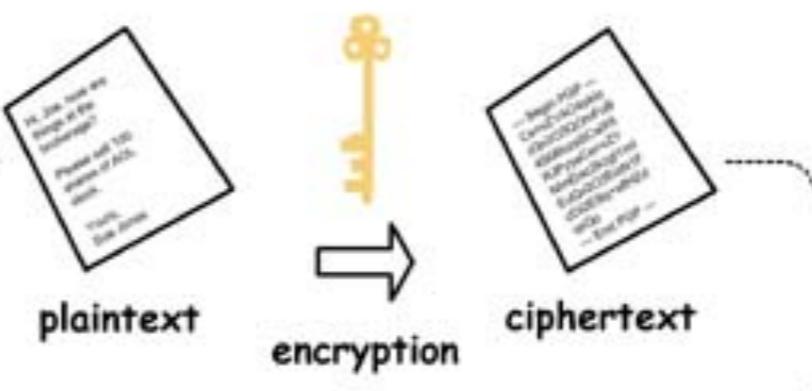
- Also known as Public Key Ciphers
- Two “keys”
  - Encryption (private)
  - Decryption (public)
- Share decryption key freely

# Asymmetric Key

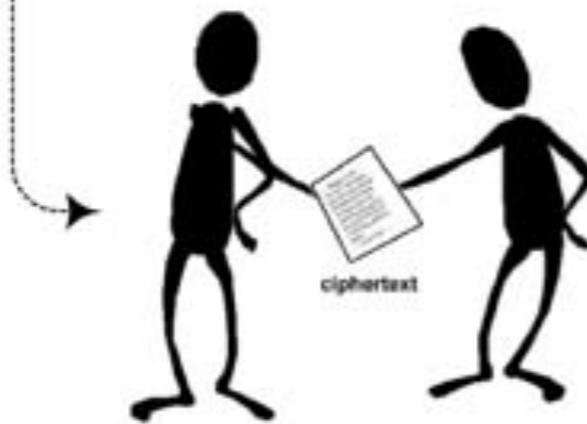
Step 1: Give your public key to sender.



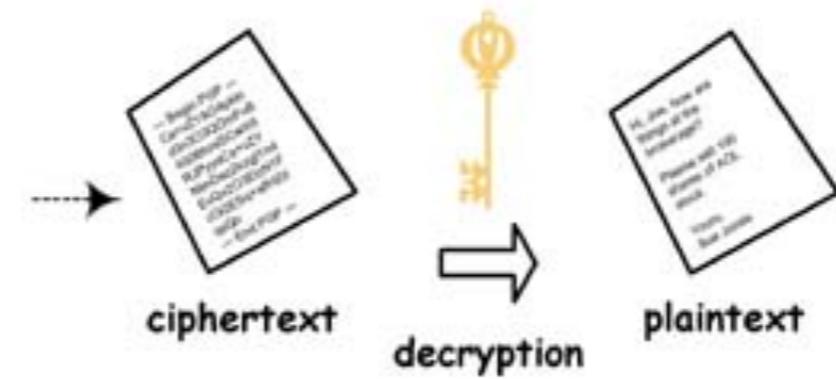
Step 2: Sender uses your public key to encrypt the plaintext.



Step 3: Sender gives the ciphertext to you.



Step 4: Use your private key (and passphrase) to decrypt the ciphertext.



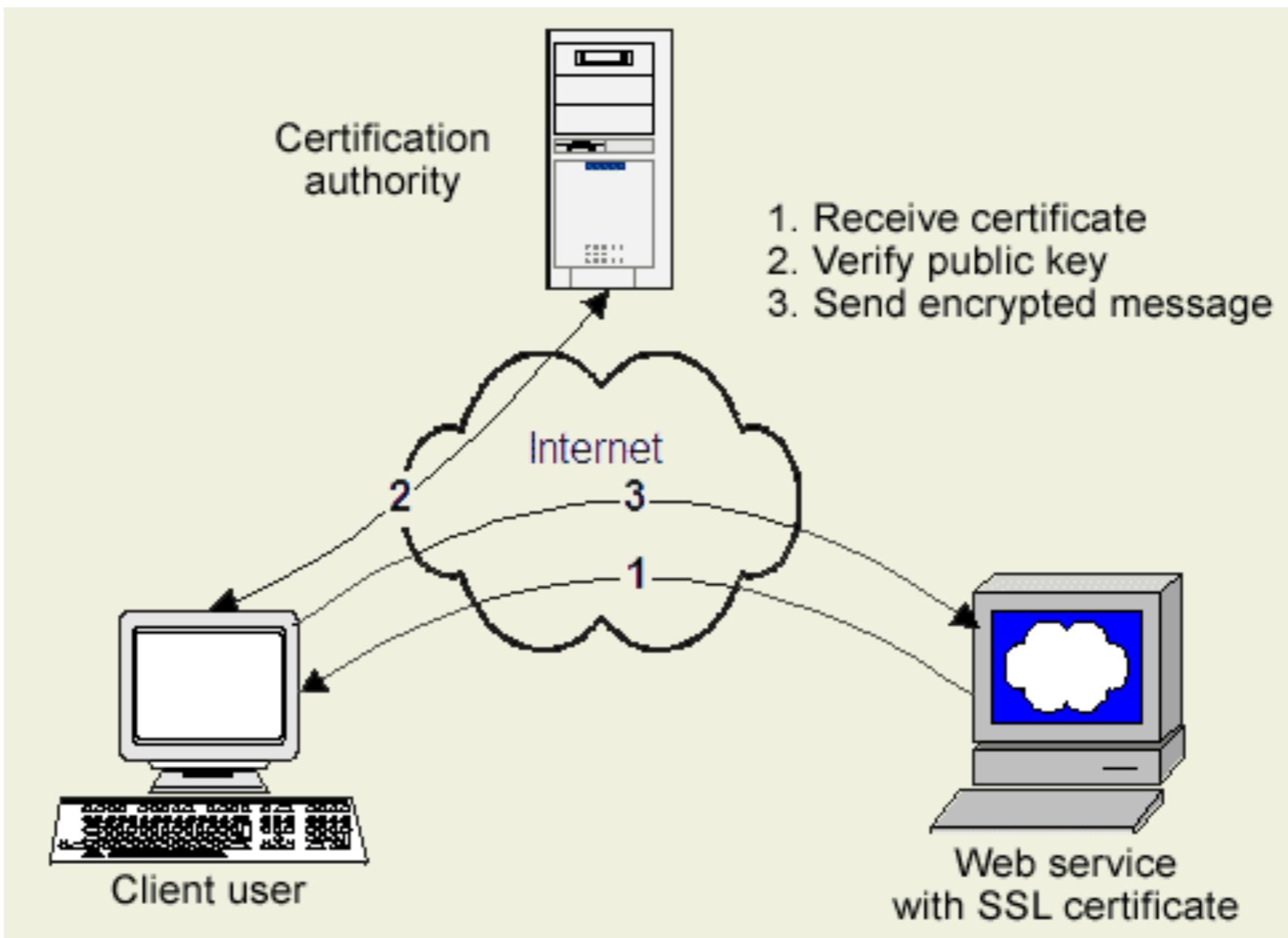
# Asymmetric Keys

- What's the difference between public and private keys?
- Use public key to encrypt so only private key can decrypt.

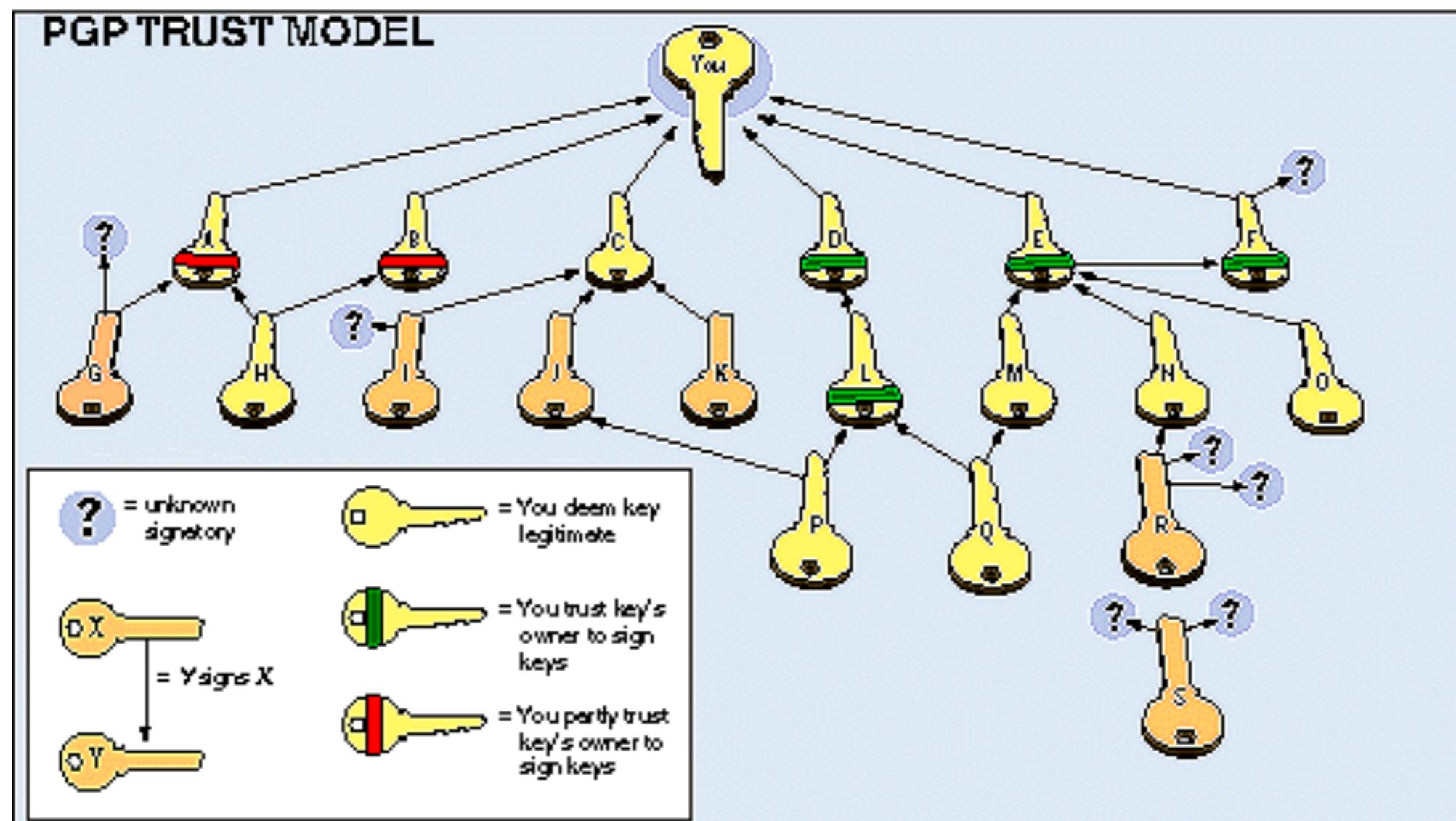
# Public-Key Distribution

We need a method to reliably get public-keys so we can reliably decrypt messages.

# Certificate Authority



# Web of Trust



# Asymmetric Keys

- Other Benefits
  - Authentication of sender
  - Limit reception to one receiver
- Problems
  - Public key distribution (validation of)
  - Complexity = slow

# Authenticators

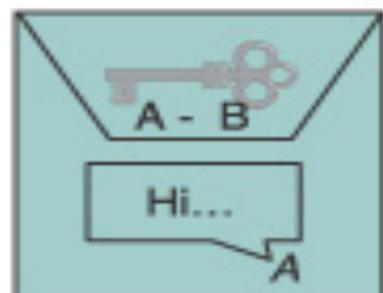
- “digest” or “hash” of message
- similar to Checksum or CRC - one way
- High probability sent and received messages are the same.
- Digital Signature = digest encrypted with public-key algorithm

# Example: PGP

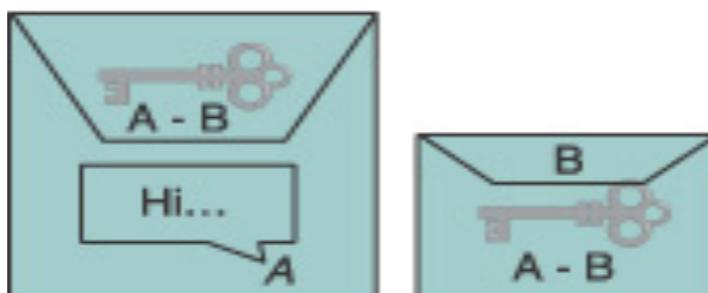
Hi... = The plaintext message



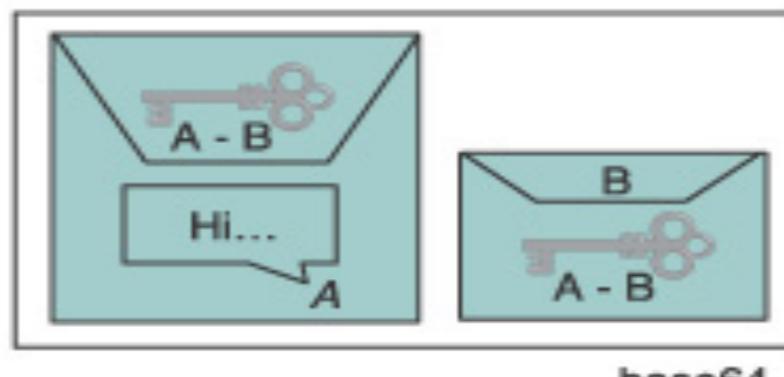
- 1) Digitally sign using Alice's private key



- 2) Encrypt using a newly generated one-time session key



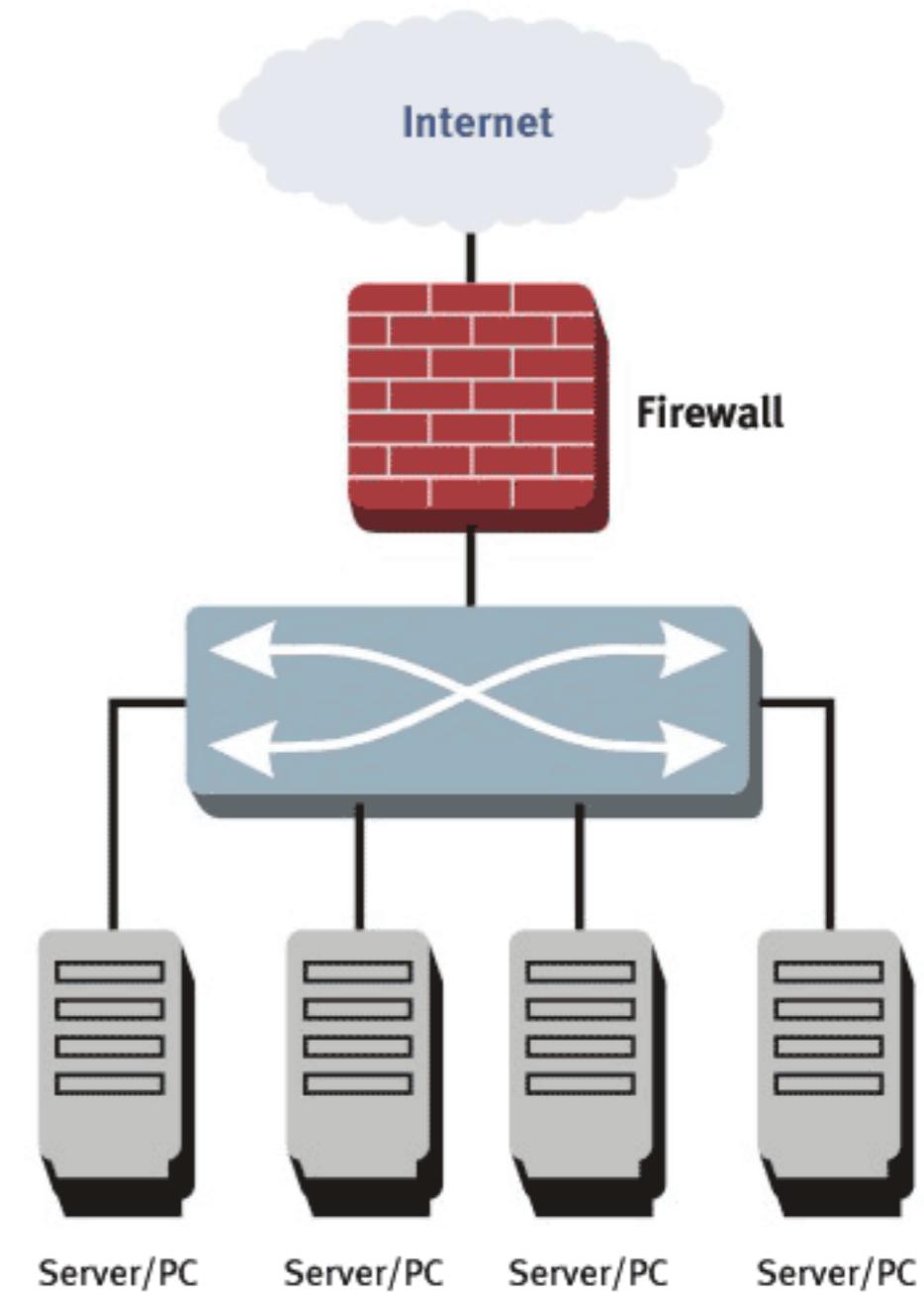
- 3) Encrypt the session key using Bob's public key, and append that



- 4) Use base64 encoding to obtain an ASCII-compatible representation

# Firewalls

- “Zones” of trust
  - Internal
  - DMZ
  - Internet
- Network Address Tran



# Firewalls

- Protects from undesirable traffic
- Can be deployed by one user
  - no endpoint agreement needed
- Once “inside” all bets are off
- No protection from “trojans”

# Network Security

- Threat types
- Security Tools
  - Cryptography
  - Firewalls

End  
Network Security