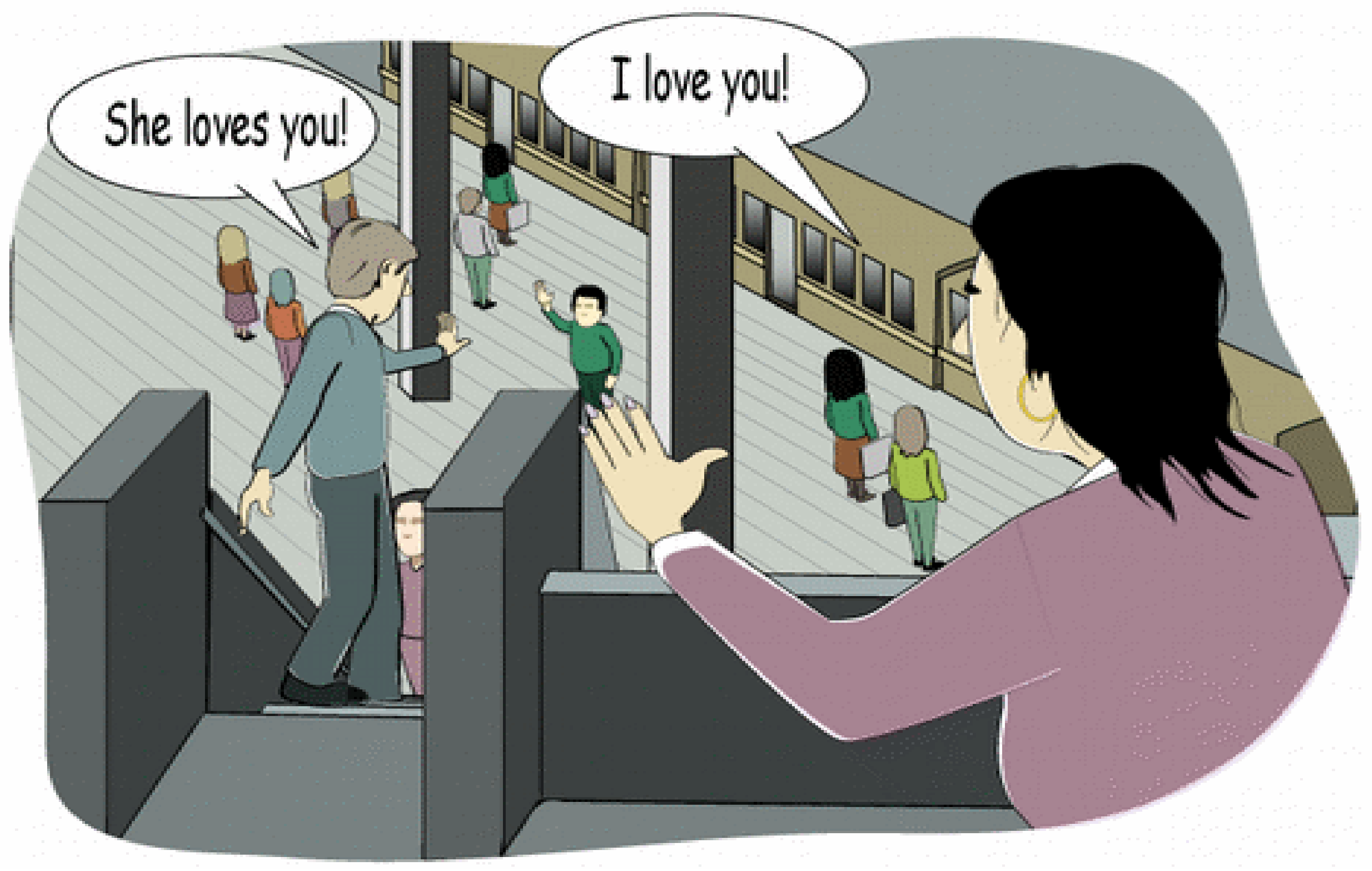
Cybersecurity

TLS

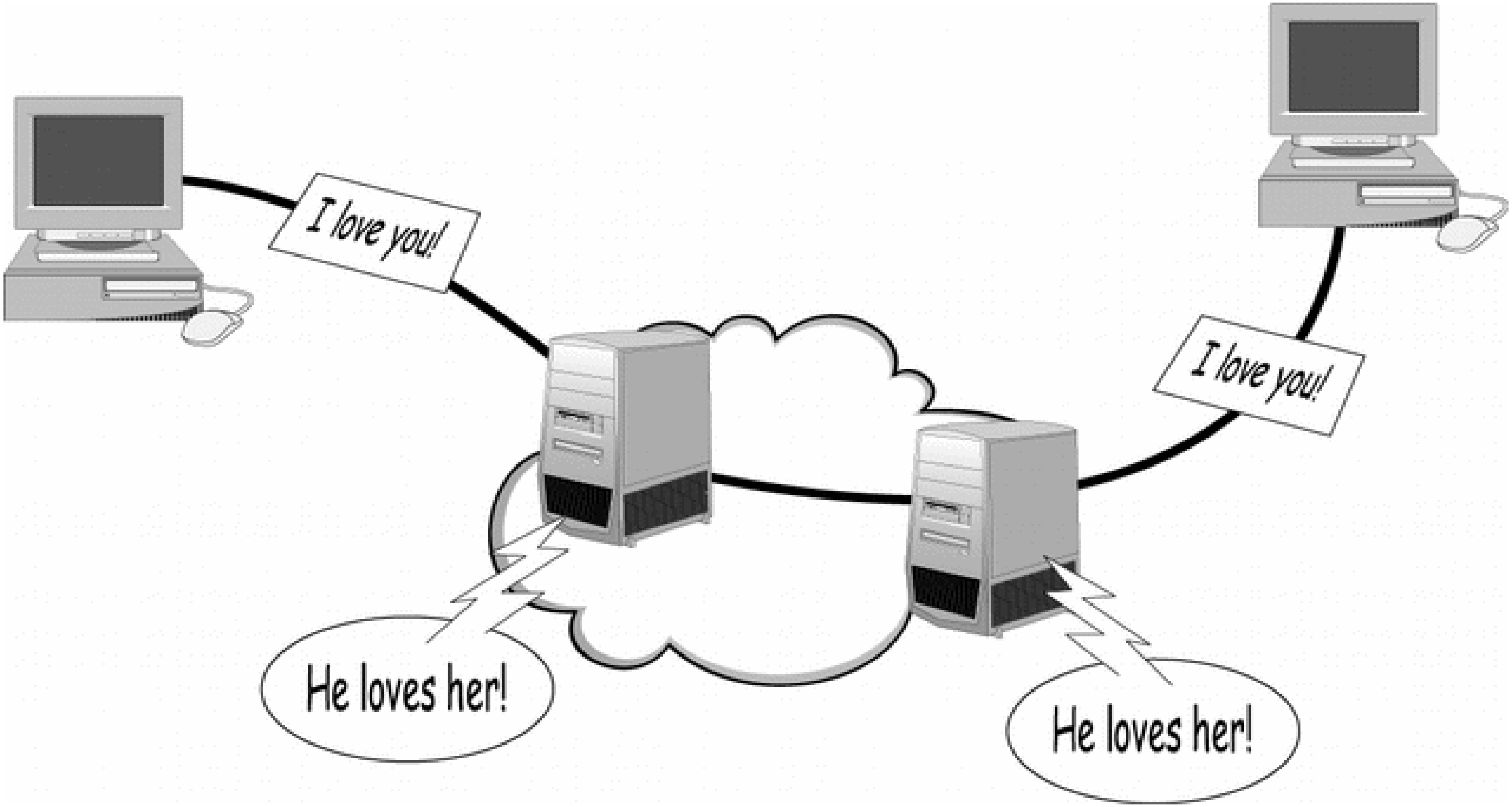
Kasun De Zoysa

Department of Communication and Media Technologies
University of Colombo School of Computing
University of Colombo
Sri Lanka

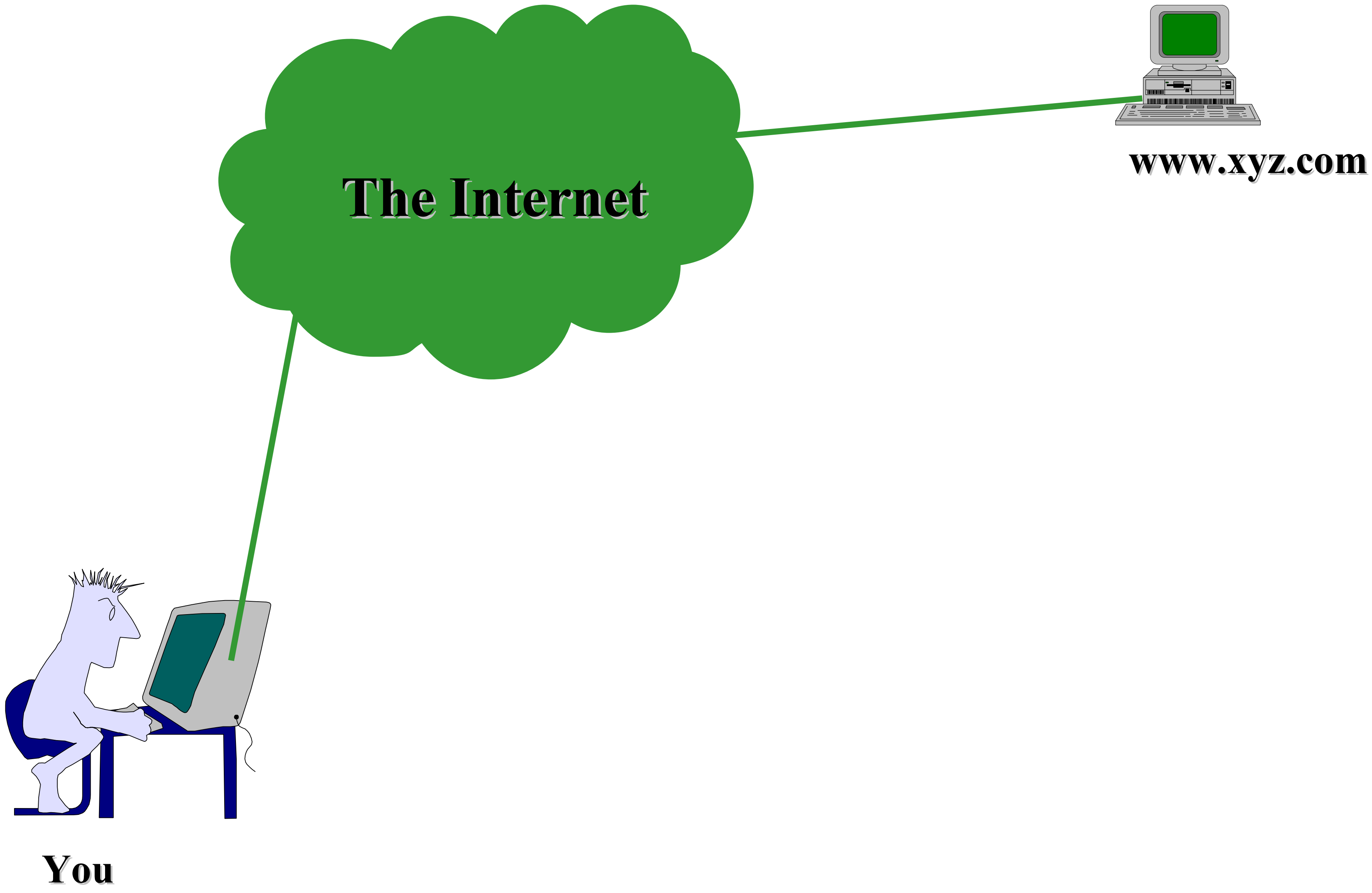
How the Internet Works -1



How the Internet Works -2



Security Requirements and User Needs

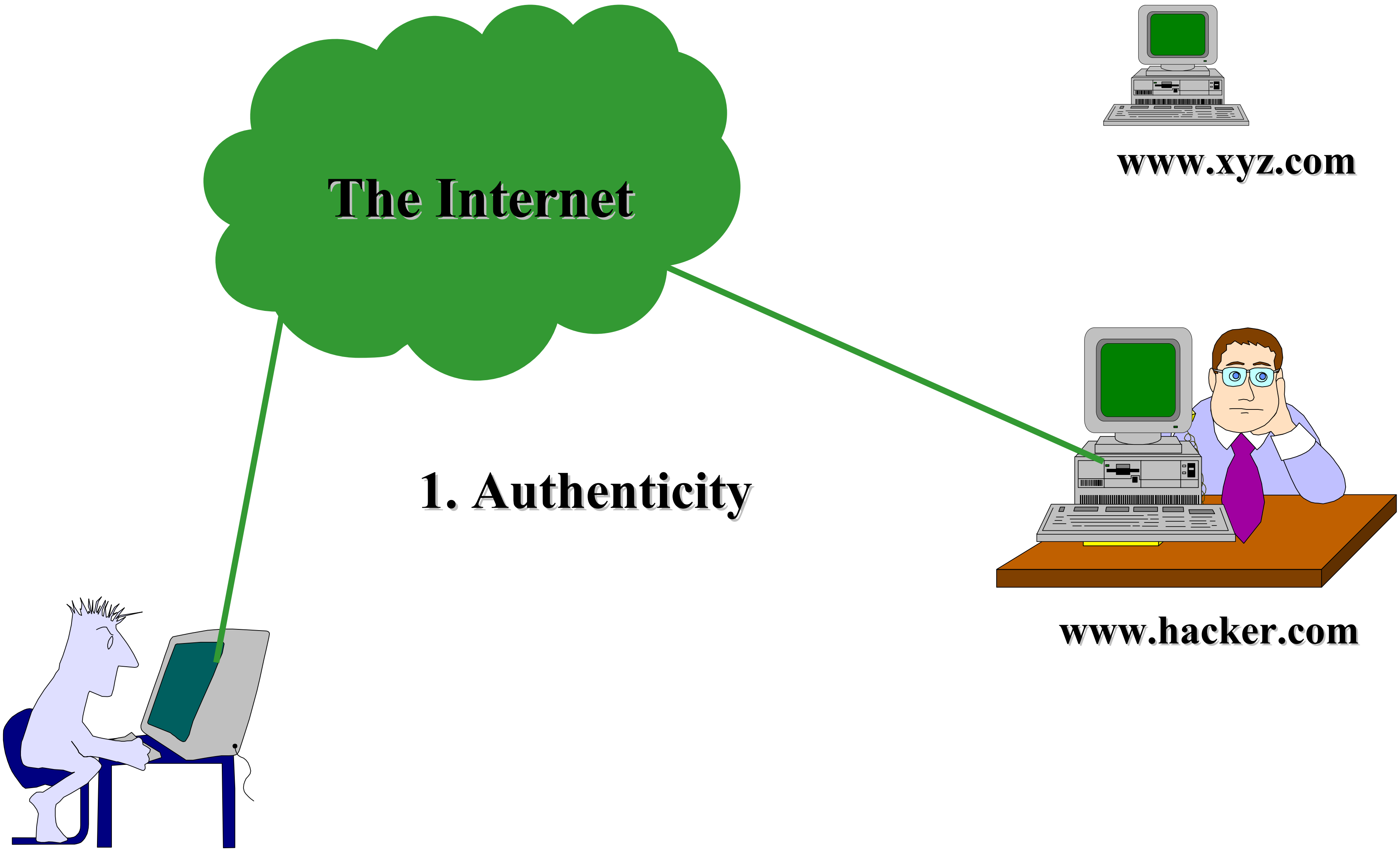


The Internet

www.xyz.com

You

Security Requirements and User Needs



The Internet

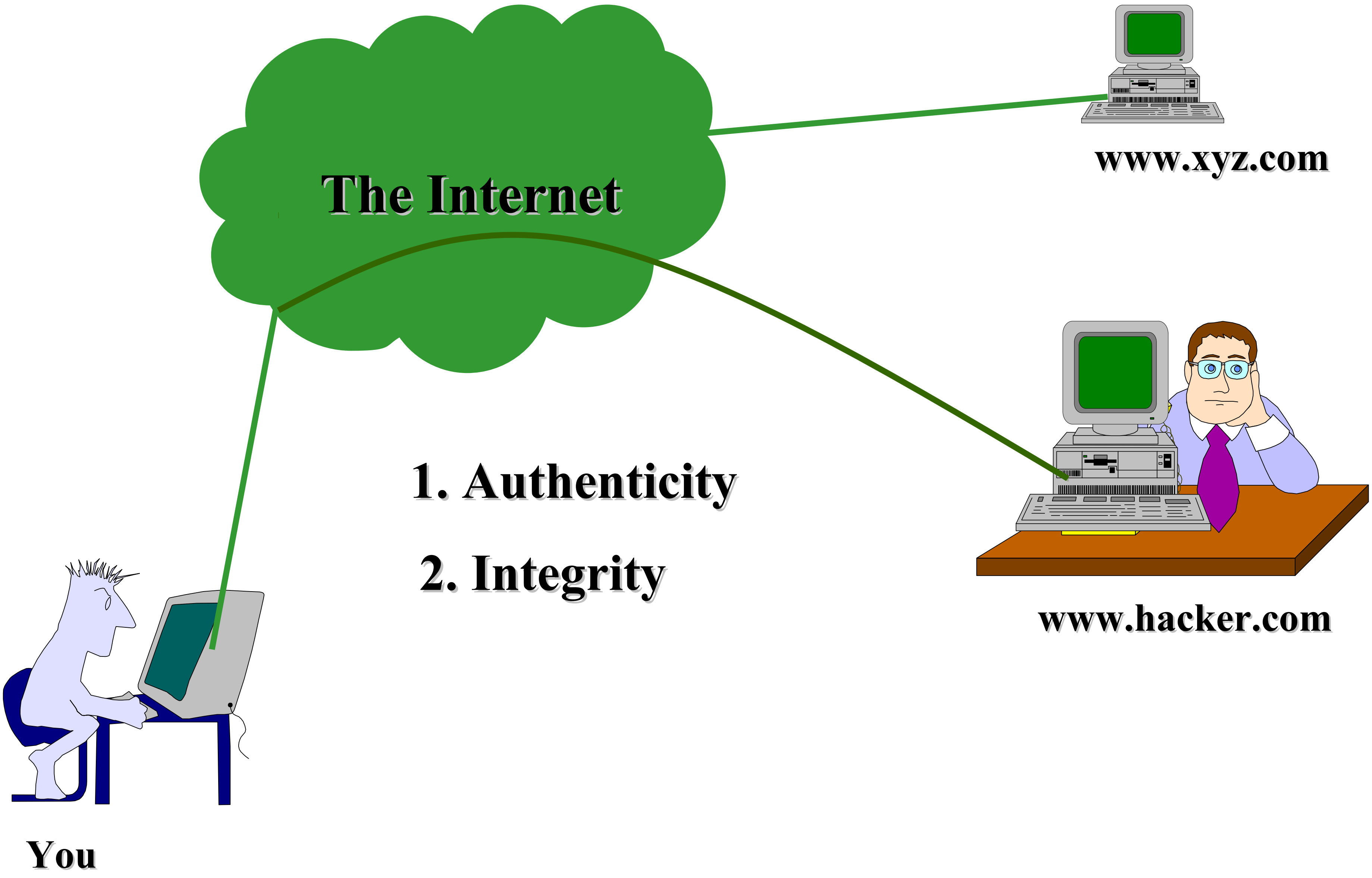
www.xyz.com

1. Authenticity

www.hacker.com

You

Security Requirements and User Needs



The Internet

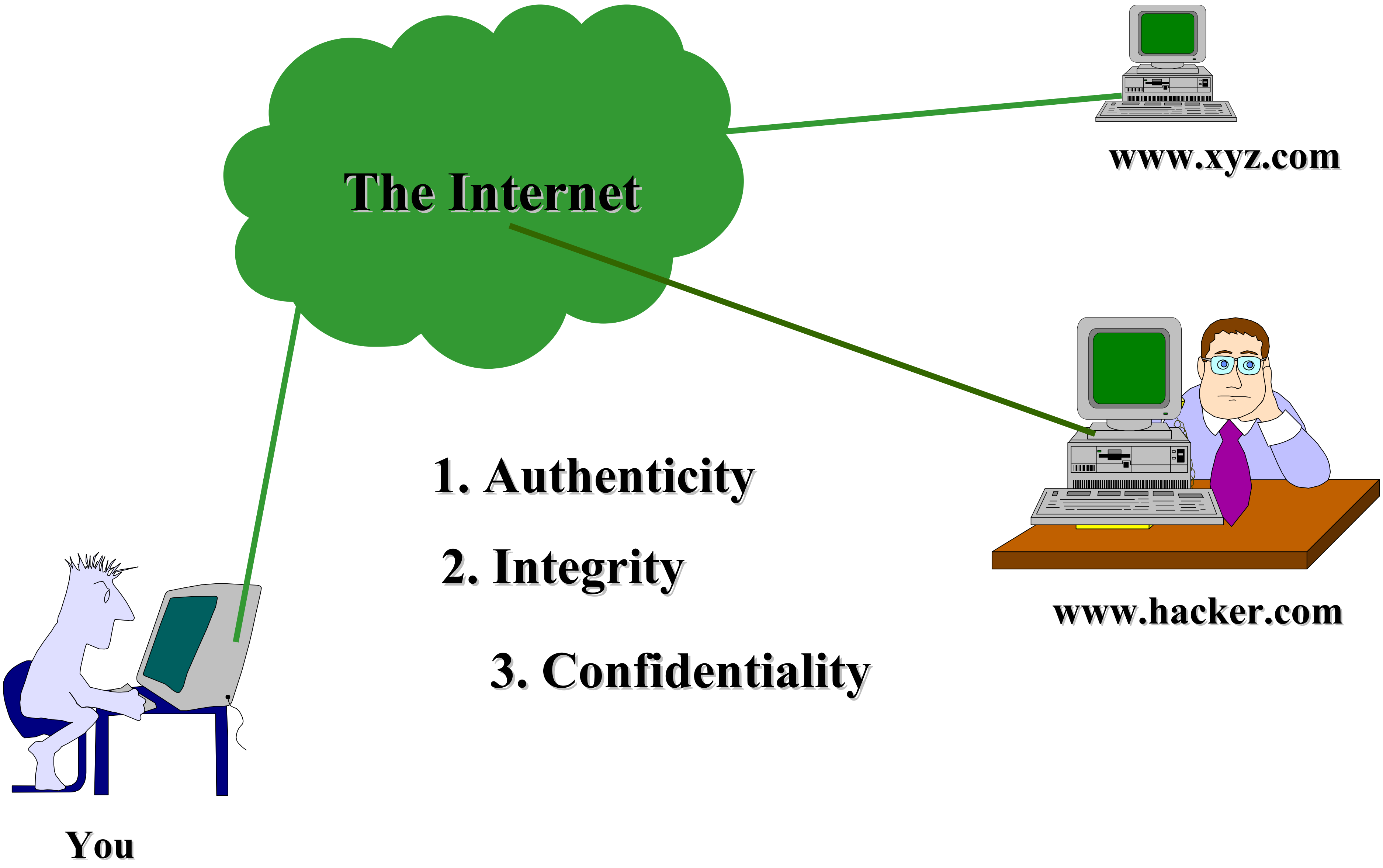
www.xyz.com

www.hacker.com

You

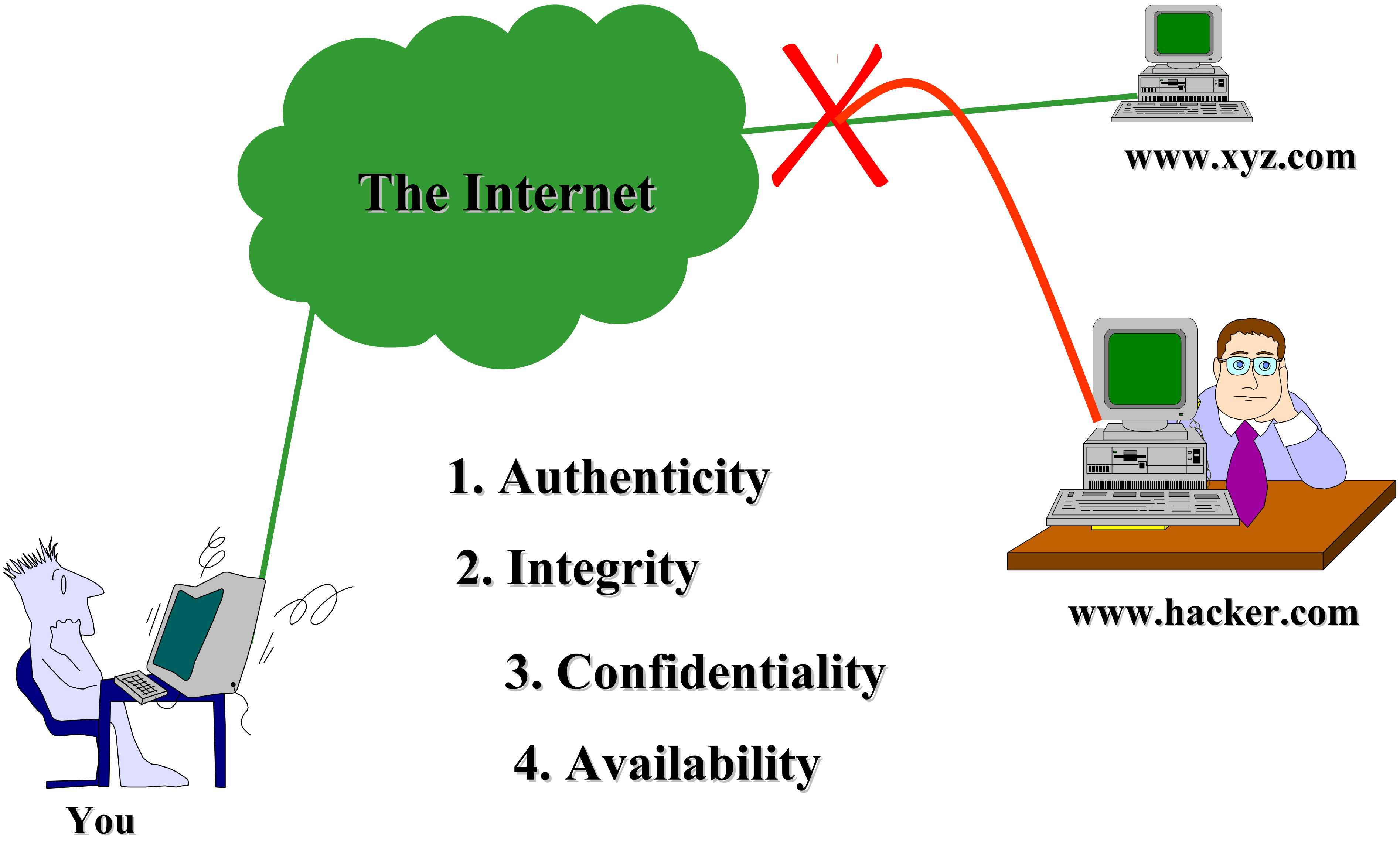
1. Authenticity
2. Integrity

Security Requirements and User Needs



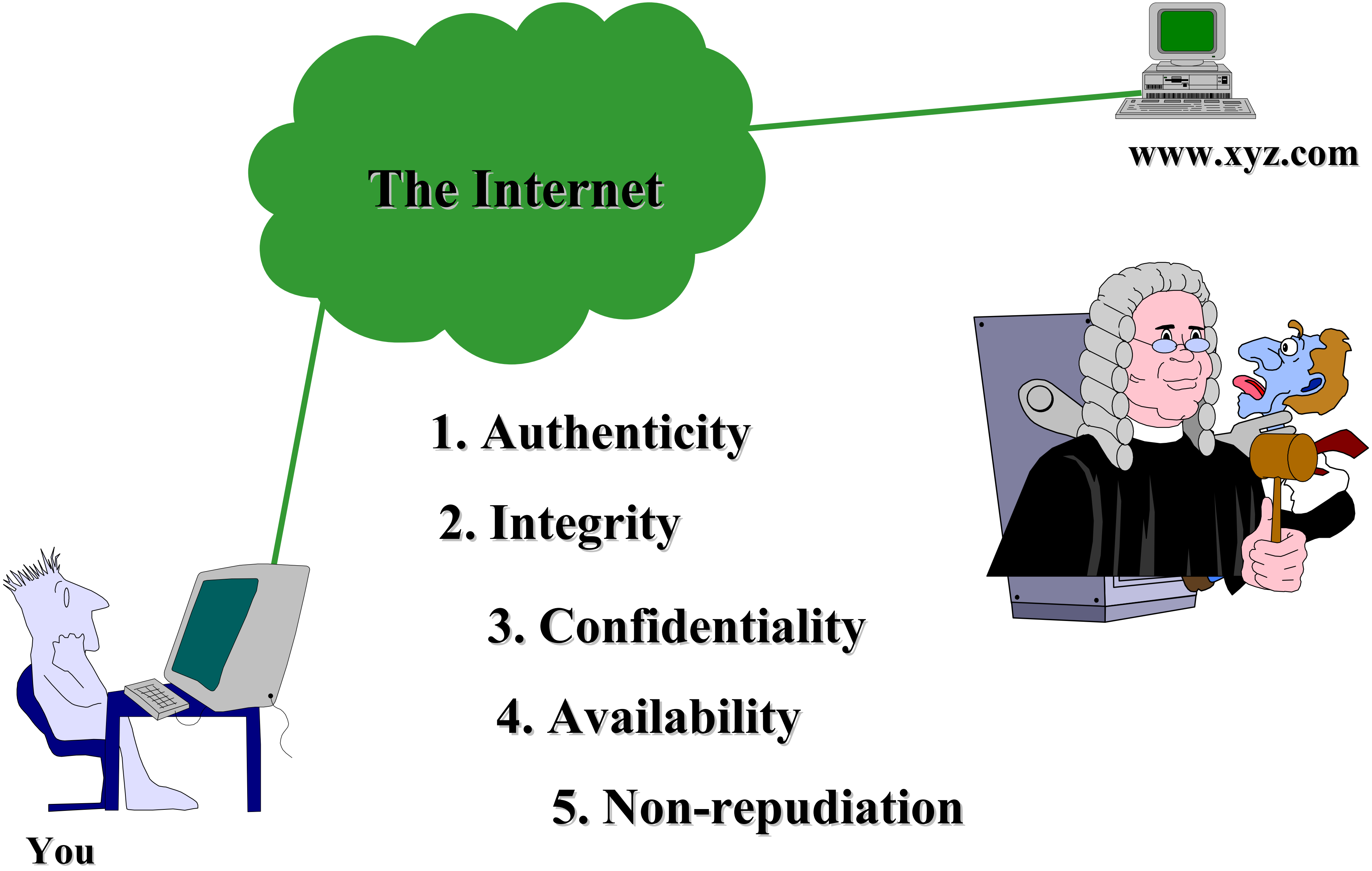
1. Authenticity
2. Integrity
3. Confidentiality

Security Requirements and User Needs



1. Authenticity
2. Integrity
3. Confidentiality
4. Availability

Security Requirements and User Needs



1. Authenticity
2. Integrity
3. Confidentiality
4. Availability
5. Non-repudiation

Solutions

Protection at Two Levels :

1. Lower Level (Channel protection)
   (Communication security services)

2. Application/User Level
   (Application level security services)

Secure Socket Layer History

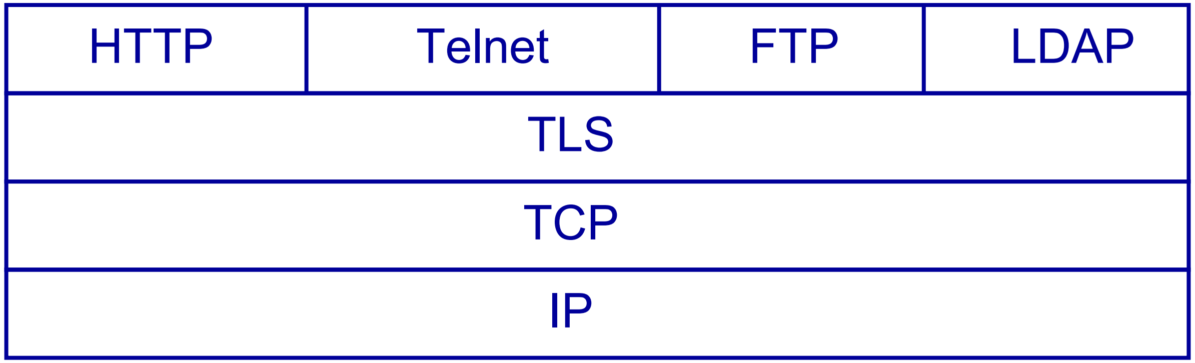
- SSL 1.0 Netscape 1994
- S-HTTP (web only)
- SSL 2.0 Netscape (buggy)
- PCT Microsoft (loser) 1996
- SSL 3.0 Netscape
- TLS 1.0 IETF 1999
- TLS 1.2 now dominant

TLS: Transport Layer Security

- *formerly known as*
  SSL: Secure Sockets Layer
- Addresses issues of privacy, integrity and authentication
  - What is it?
  - How does it address the issues?
  - How is it used

What is TLS?

- Protocol layer
- Requires reliable transport layer (e.g. TCP)
- Supports any application protocols



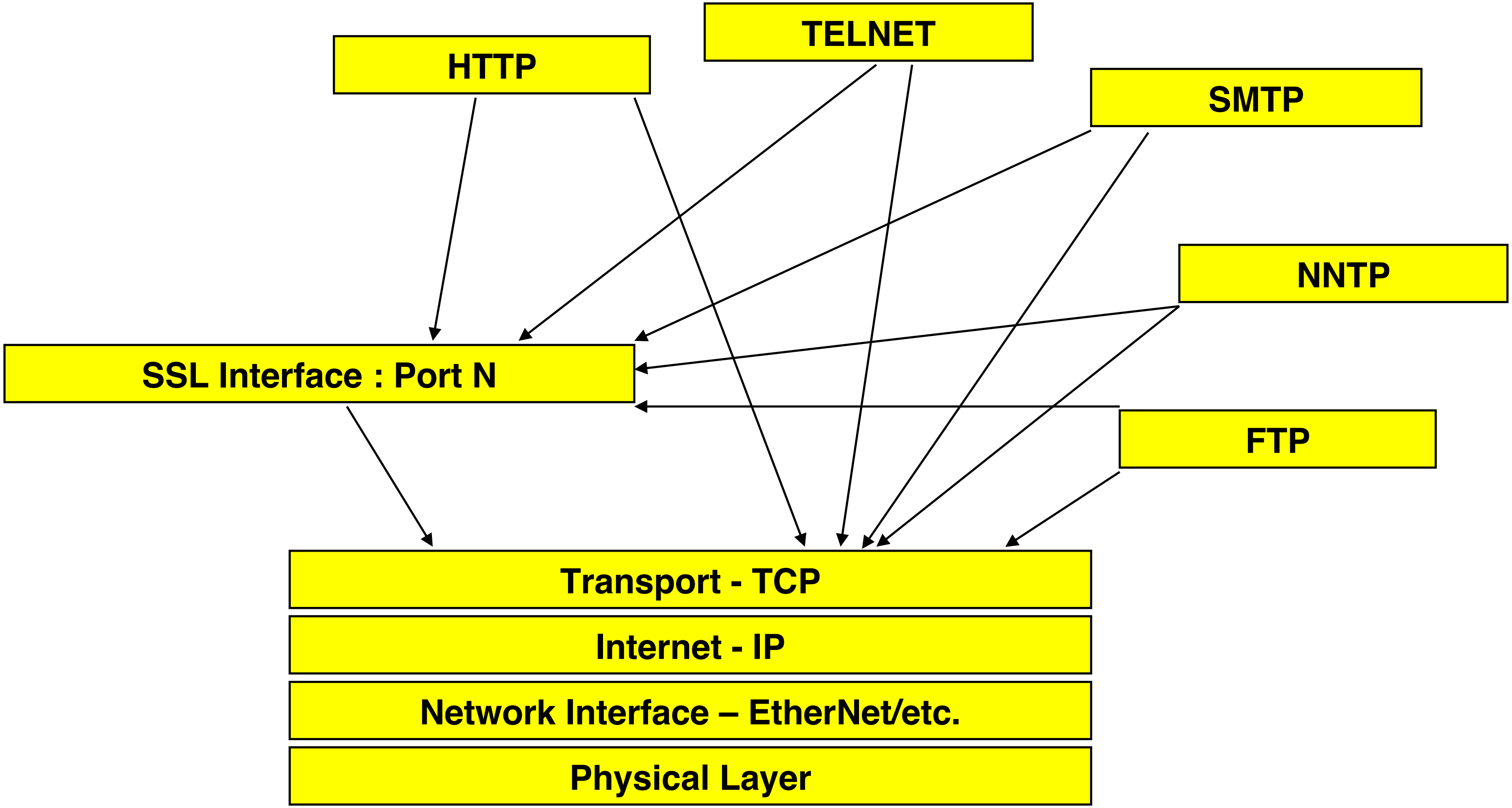
HTTP Telnet FTP LDAP

TLS

TCP

IP

Protocol Stack

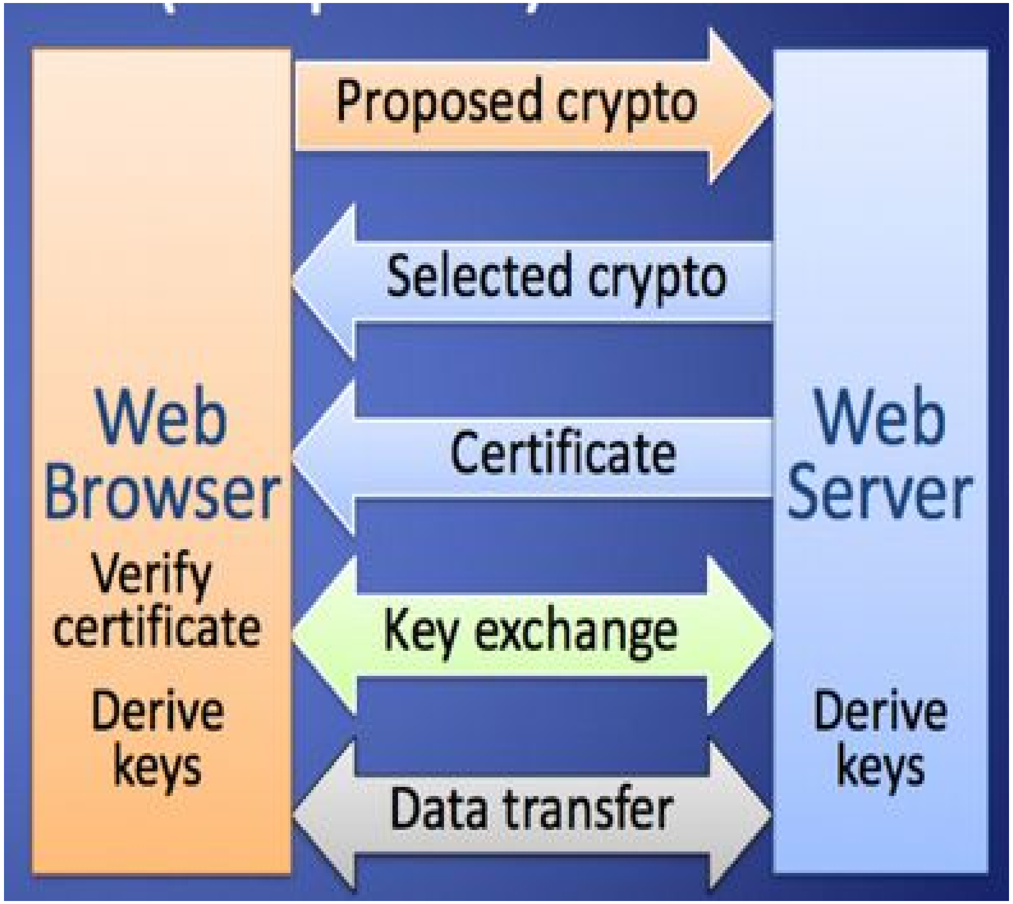


TLS: Overview

- Establish a session
  - Agree on algorithms
  - Share secrets
  - Perform authentication
- Transfer application data
  - Ensure privacy and integrity

TLS Overview

- Browser sends supported crypto algorithms
- Server picks strongest algorithms it supports
- Server sends certificate (chain)
- Client verifies certificate (chain)
- Client and server agree on secret value R by exchanging messages
- Secret value R is used to derive keys for symmetric encryption and hash-based authentication of subsequent data transfer

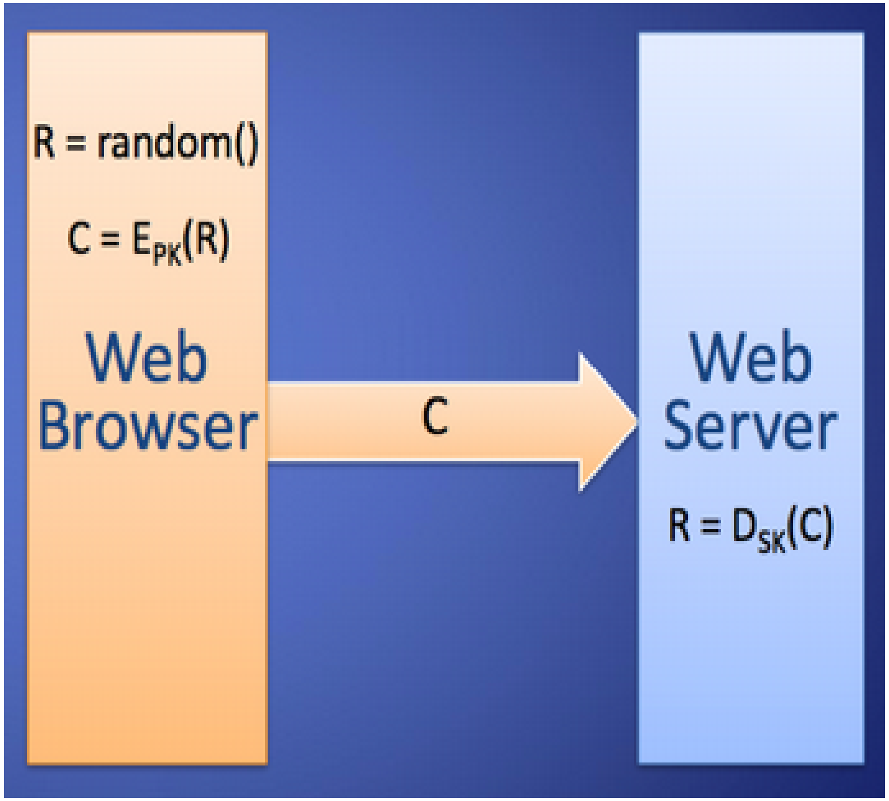


TLS:Key Exchange

- Need secure method to exchange secret key
- Use public key encryption for this
  - “key pair” is used - either one can encrypt and then the other can decrypt
  - slower than conventional cryptography
  - share one key, keep the other private
- Choices are RSA or Diffie-Hellman

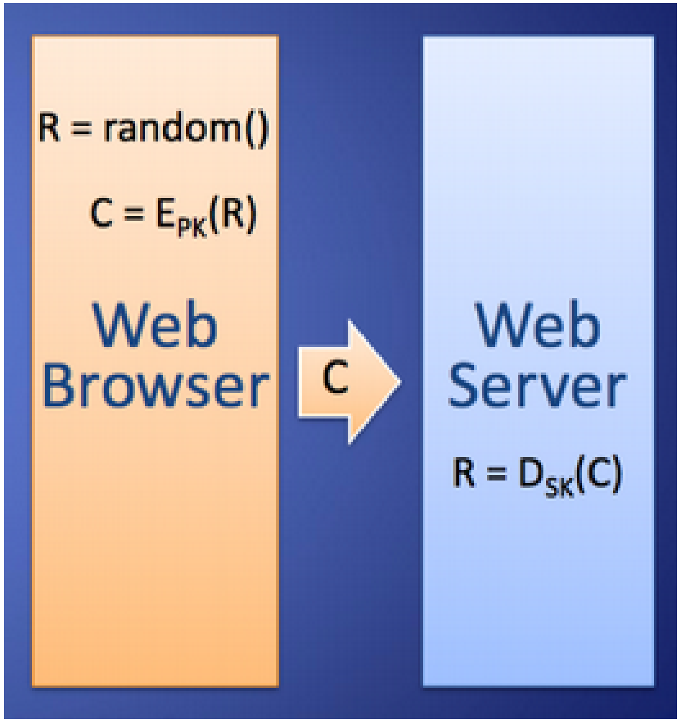
Basic Key Exchange

- Called RSA key exchange for historical reasons
- Client generates random secret value R
- Client encrypts R with public key, PK, of server C = EPK(R)
- Client sends C to server
- Server decrypts C with private key, SK, of server R = DSK(C)

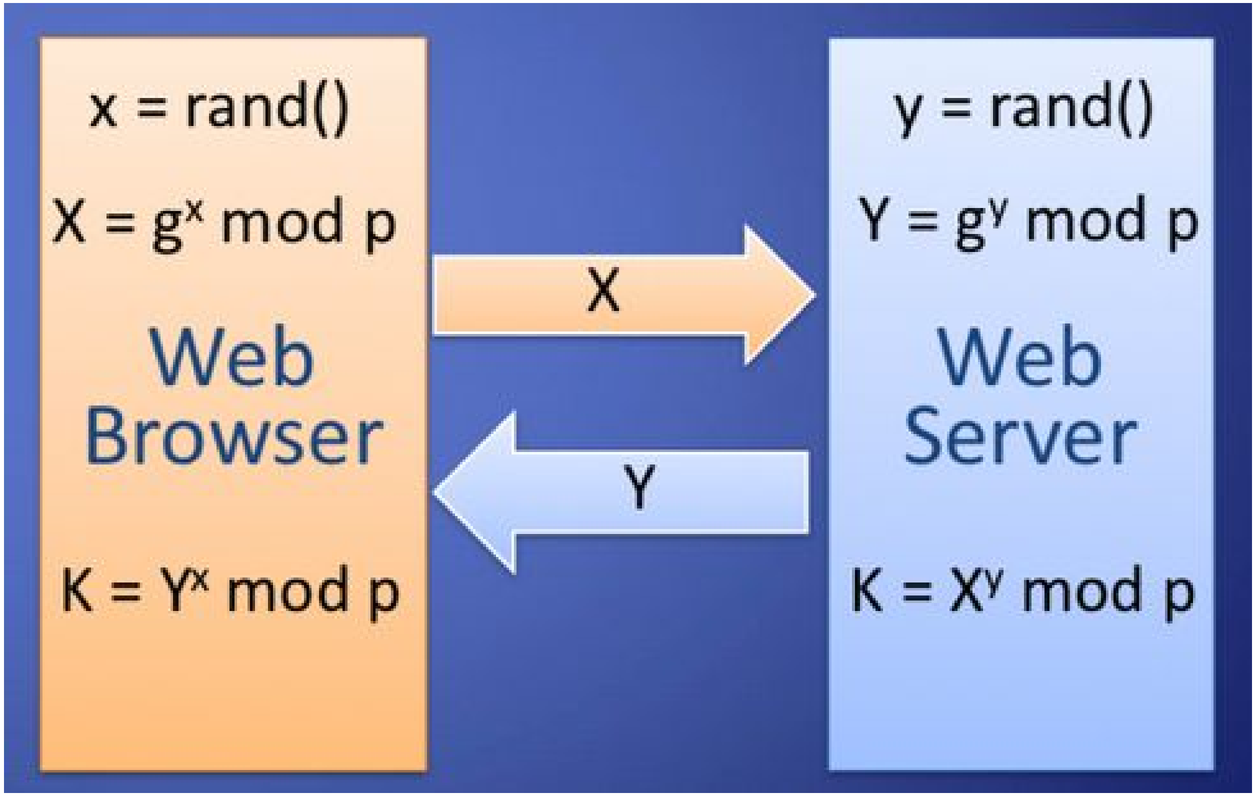


Forward Secrecy

- Compromise of public-key encryption private keys does not break confidentiality of past messages
- TLS with basic key exchange does not provide forward secrecy
- Attacker eavesdrop and stores communication
- If server's private key is compromised, attacker finds secret value R in key exchange and derives encryption keys



Diffie Hellman Key Exchange



$x = \text{rand}()$

$X = g^x \bmod p$

$K = Y^x \bmod p$

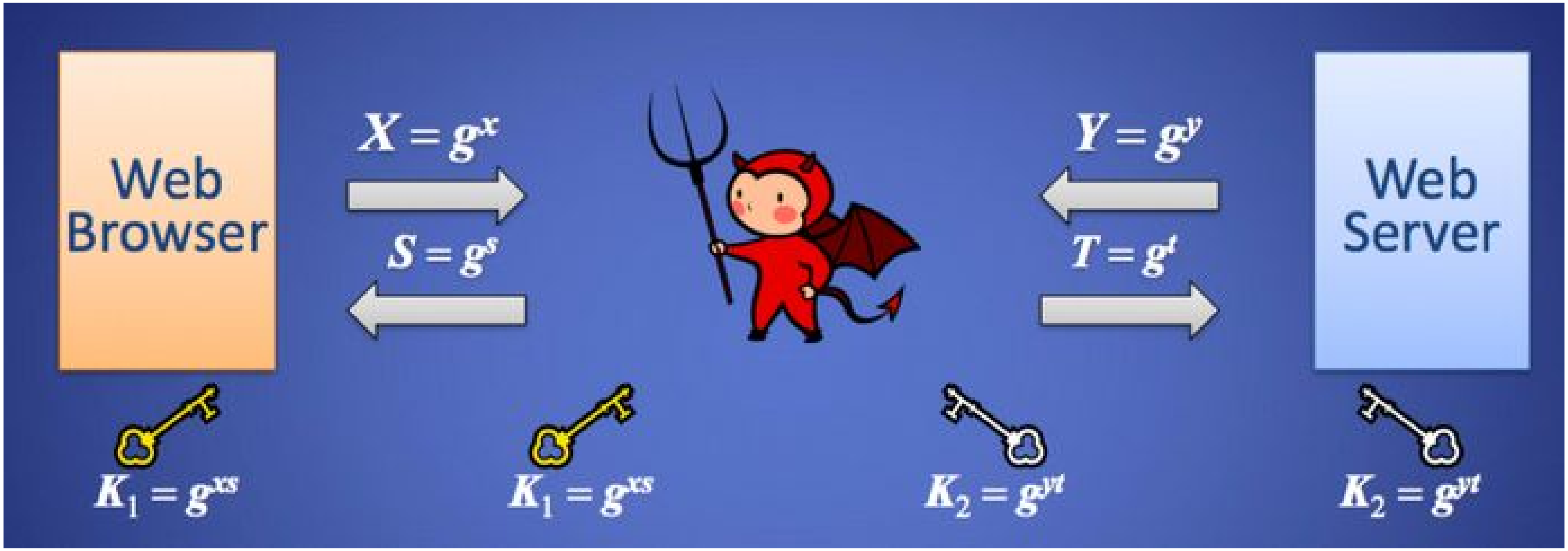
$y = \text{rand}()$

$Y = g^y \bmod p$

$K = X^y \bmod p$

Achieves forward secrecy

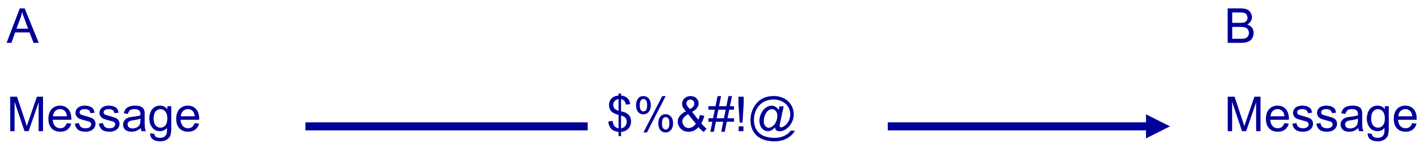
Attacker in the Middle



Solution:
Browser and server send signed X and Y respectively
Requires each to know the public key of the other

TLS: Privacy

- Encrypt message so it cannot be read
- Use conventional cryptography with shared key
  - DES, 3DES, AES
  - RC2, RC4
  - IDEA



A

B

Message

$%&#!@

Message

TLS Encrypts

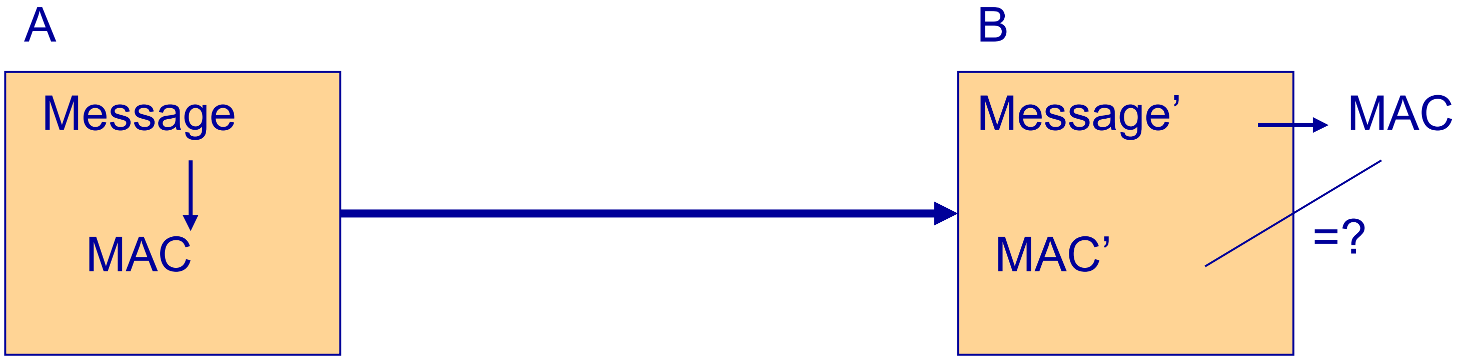
- ALL Browser-Server and Server-Browser except which-browser is talking to which-server
- URL of requested document
- Contents of requested document
- Contents of any submitted form fill-outs
- Cookies sent from browser to server
- Cookies sent from server to browser
- Contents of HTTP header
- Javascript communications
- Etc.

TLS: Integrity

- Compute fixed-length Message Authentication Code (MAC)
  - Includes hash of message
  - Includes a shared secret
  - Include sequence number
- Transmit MAC with message

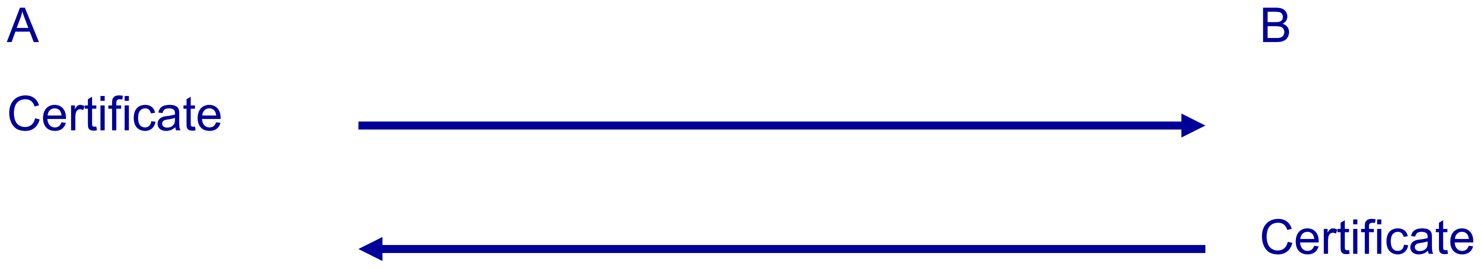
TLS: Integrity

- Receiver creates new MAC
  - should match transmitted MAC
- TLS allows MD5, SHA-1



TLS: Authentication

- Verify identities of participants
- Client authentication is optional
- Certificate is used to associate identity with public key and other attributes



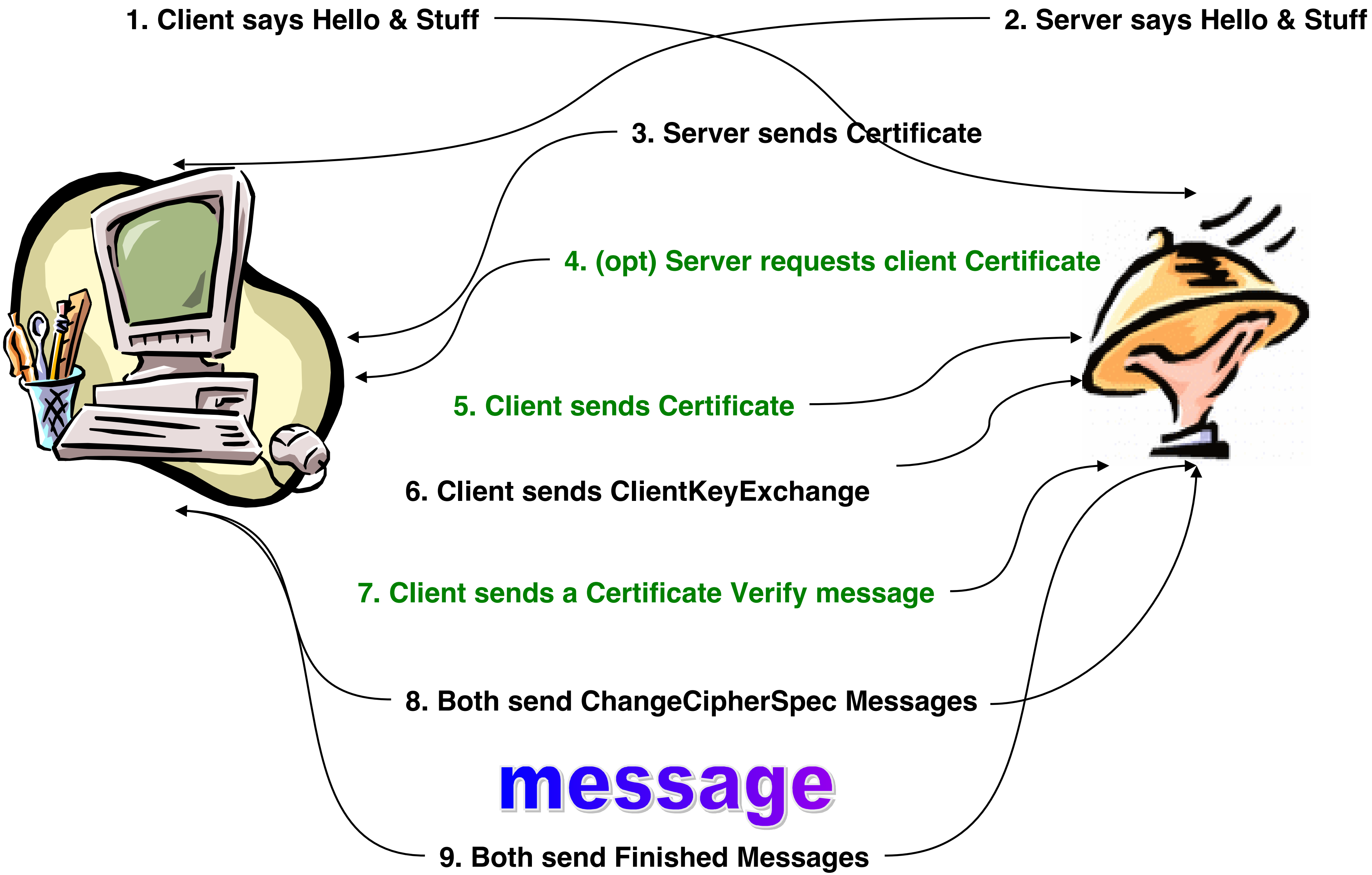
A

B

Certificate

Certificate

TLS Transaction



1. Client says Hello & Stuff

2. Server says Hello & Stuff

3. Server sends Certificate

4. (opt) Server requests client Certificate

5. Client sends Certificate

6. Client sends ClientKeyExchange

7. Client sends a Certificate Verify message

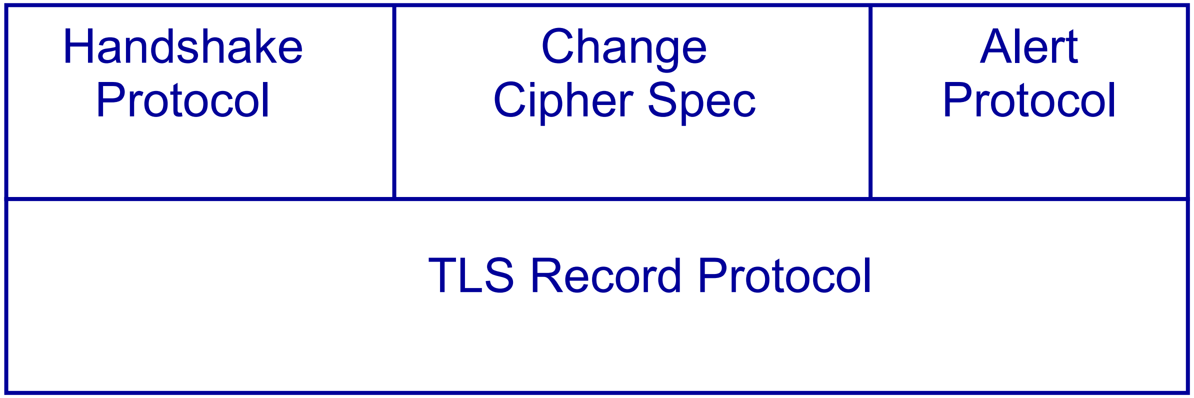
8. Both send ChangeCipherSpec Messages

message

9. Both send Finished Messages

TLS: Architecture

- TLS defines Record Protocol to transfer application and TLS information
- A session is established using a Handshake Protocol



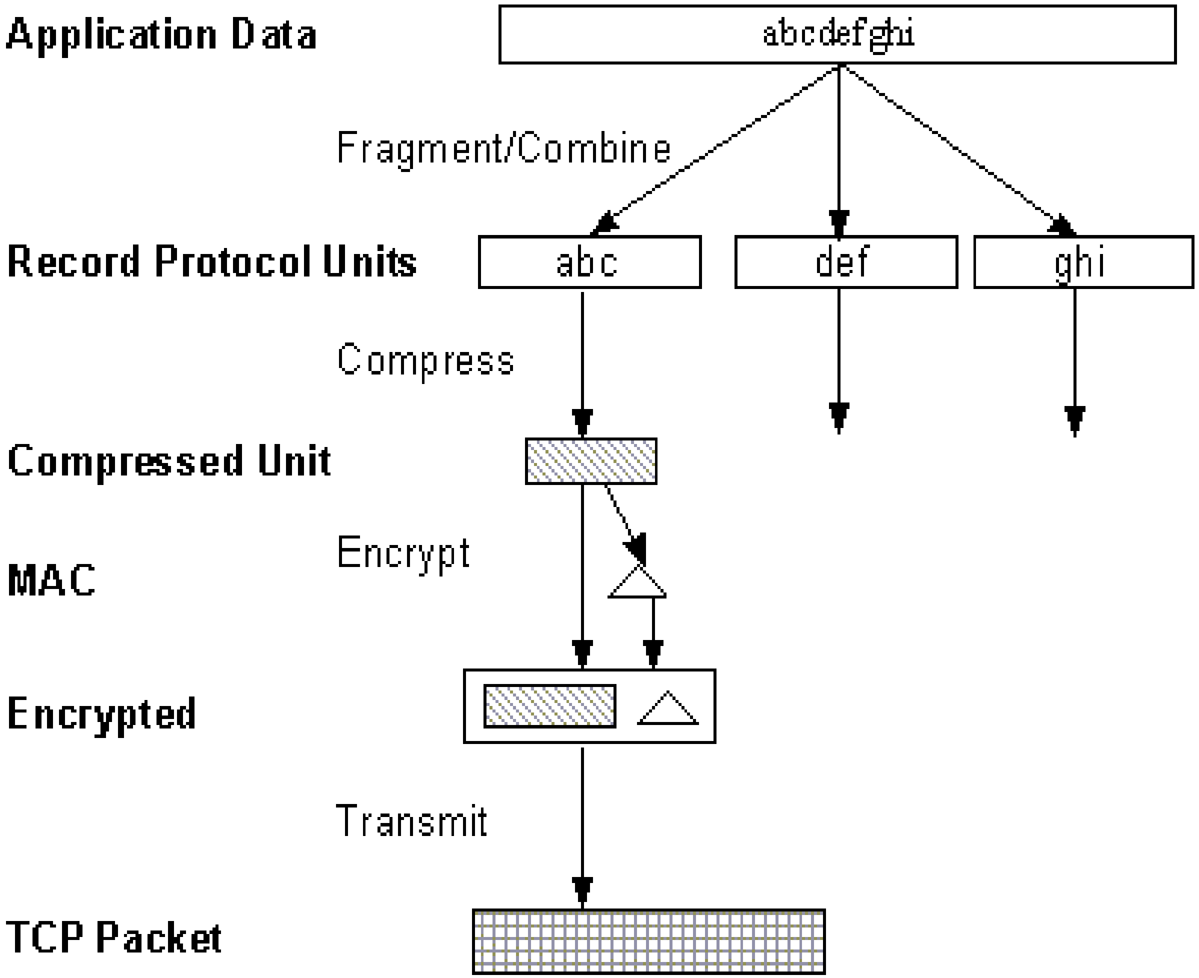
Handshake Protocol

Change Cipher Spec

Alert Protocol

TLS Record Protocol

TLS: Record Protocol



TLS: HTTP Application

- HTTP most common TLS application
  - https://
- Requires TLS-capable web server
- Requires TLS-capable web browser

Public Key Certificates

- X.509 Certificate associates public key with identity
- Certification Authority (CA) creates certificate
  - Adheres to policies and verifies identity
  - Signs certificate
- User of Certificate must ensure it is valid

Subject Names

- X.500 Distinguished Name (DN)
- Associated with node in hierarchical directory (X.500)
- Each node has Relative Distinguished Name (RDN)
  - Path for parent node
  - Unique set of attribute/value pairs for this node

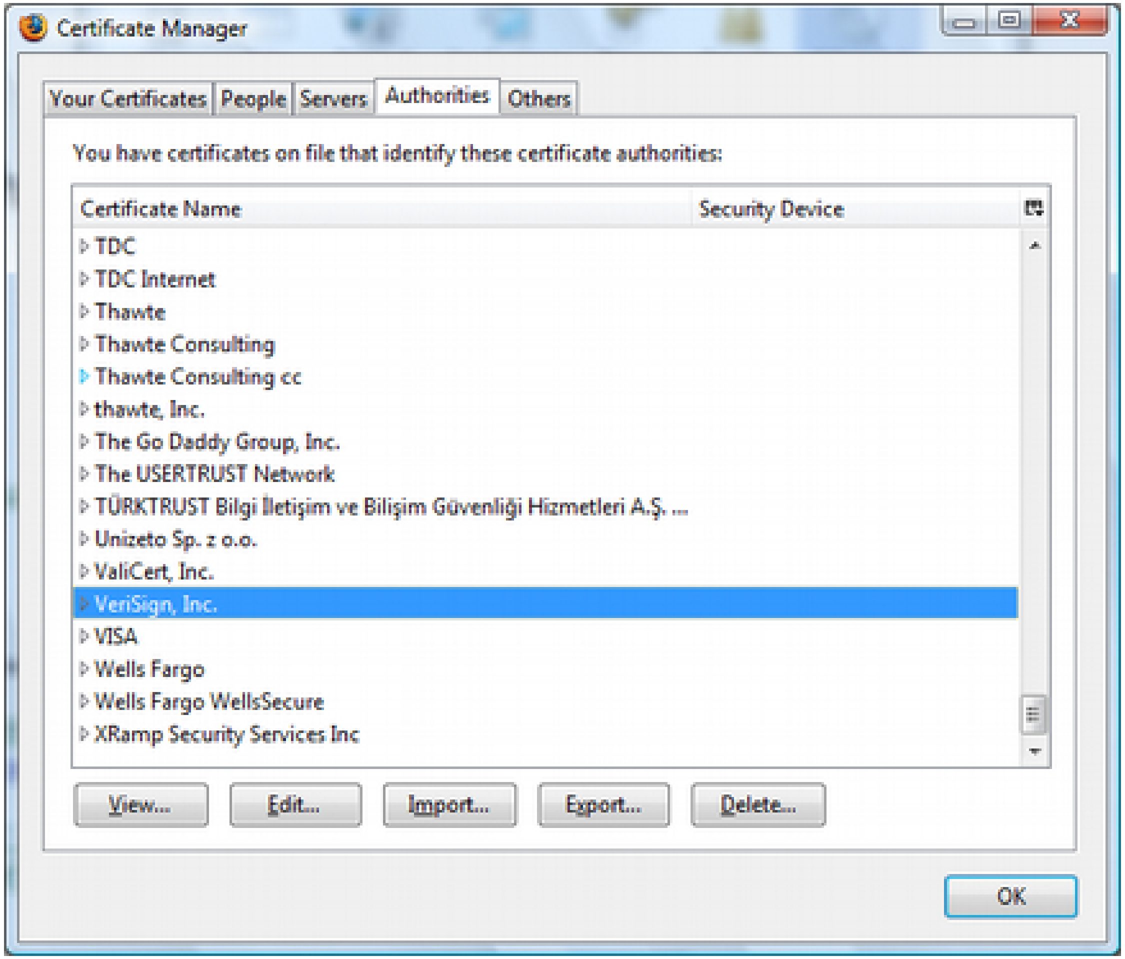
Example Subject Name

- Country at Highest Level (e.g. US)
- Organization typically at next level (e.g. CertCo)
- Individual below (e.g. Common Name "Kasun" with Id = 1)

  DN = {
    - C=LK;
    - O=UCSC;
    - CN=Kasun, ID=1}

Certificate Authorities

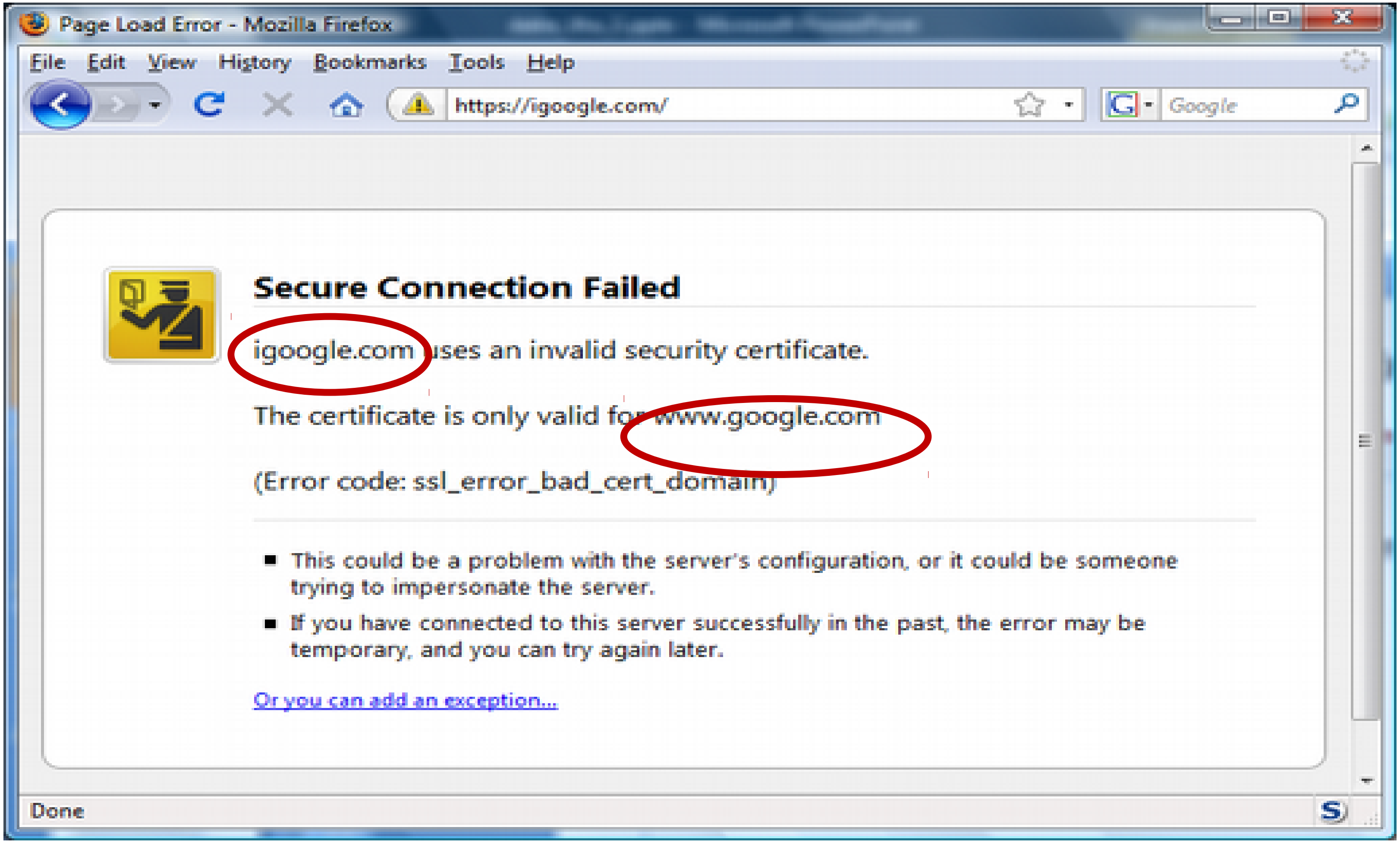
Browsers accept certificates from a large number of CAs



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
TDC	
TDC Internet	
Thawte	
Thawte Consulting	
Thawte Consulting cc	
thawte, Inc.	
The Go Daddy Group, Inc.	
The USERTRUST Network	
TÜRKTRUST Bilgi İletişim ve Bilişim Güvenliği Hizmetleri A.Ş. ...	
Unizeto Sp. z o.o.	
ValiCert, Inc.	
VeriSign, Inc.	
VISA	
Wells Fargo	
Wells Fargo WellsSecure	
XRamp Security Services Inc	

Firefox: Invalid cert dialog


Page Load Error - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://igoogle.com/
Google
Done

Secure Connection Failed

igoogle.com uses an invalid security certificate.

The certificate is only valid for www.google.com

(Error code: ssl_error_bad_cert_domain)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

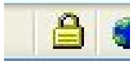
Or you can add an exception…

Firefox 3.0: **Four clicks to get firefox to accept cert**

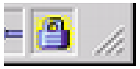
- page is displayed with full HTTPS indicators

SSL Indicators

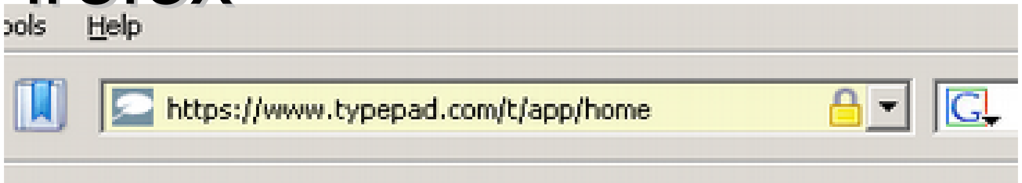
- Microsoft IE



- Mozilla

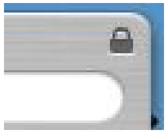


- Firefox

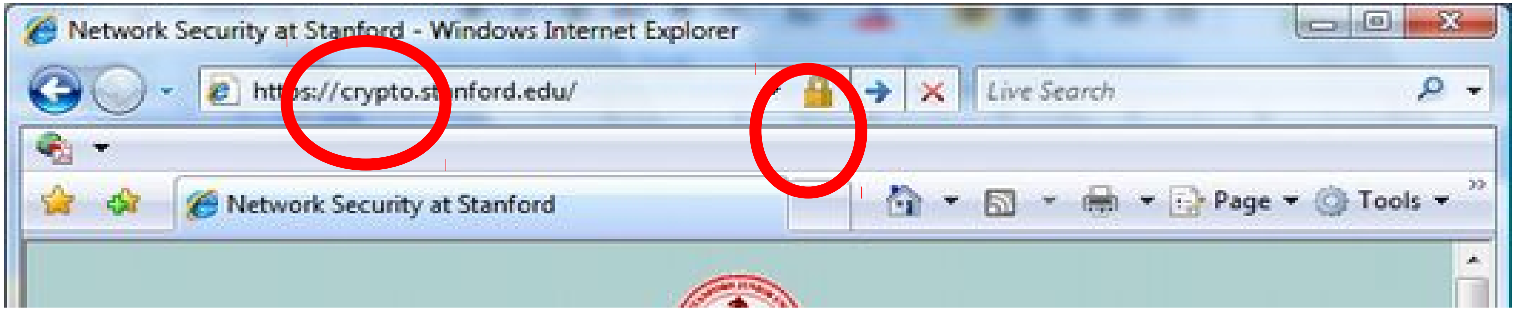


https://www.typepad.com/t/app/home

- Safari



The lock icon: SSL indicator


Network Security at Stanford - Windows Internet Explorer
https://crypto.stanford.edu/
Live Search
Network Security at Stanford
Page
Tools

Intended goal:

- Provide user with identity of page origin
- Indicate to user that page contents were not viewed or modified by a **network attacker**

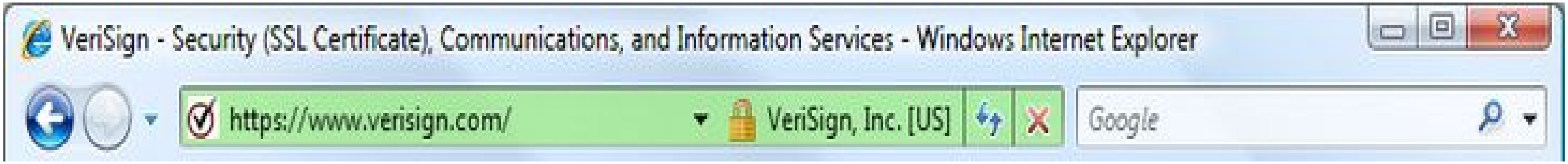
In reality:

- Origin ID is not always helpful
- Many other problems

Version 3 Certificates

- Version 3 X.509 Certificates support alternative name formats as extensions
  - X.500 names
  - Internet domain names
  - e-mail addresses
  - URLs
- Certificate may include more than one name

Extended validation (EV) certs


VeriSign - Security (SSL Certificate), Communications, and Information Services - Windows Internet Explorer
https://www.verisign.com/
VeriSign, Inc. [US]
Google

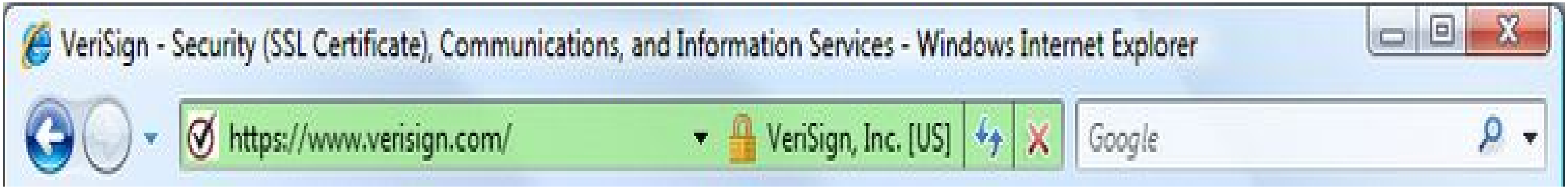
An Extended Validation Certificate (EV) is an X.509 public key certificate issued according to a specific set of identity verification criteria. These criteria require extensive verification of the requesting entity's identity by the certificate authority (CA) before a certificate is issued.

Certificates issued by a CA under the EV guidelines are not structurally different from other certificates (and hence provide no stronger cryptography than other, cheaper certificates),

Extended Validation (EV) Certs

- **Harder to obtain than regular certs**
  - requires human lawyer at CA to approve cert request

- **Designed for banks and large e-commerce sites**



VeriSign - Security (SSL Certificate), Communications, and Information Services - Windows Internet Explorer

https://www.verisign.com/

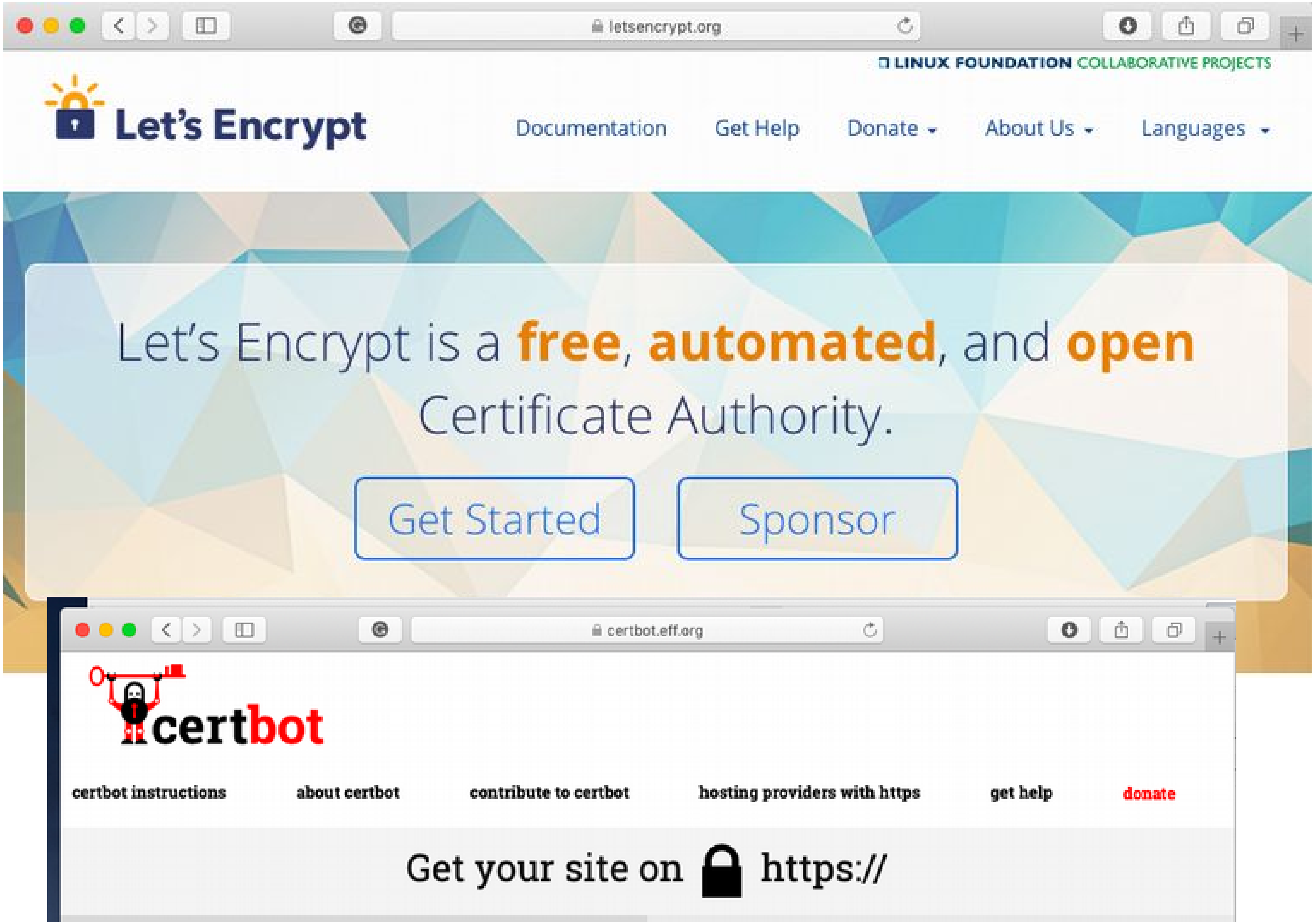
- **Helps block "semantic attacks":** www.bankofthe**vv**est.com

Automatic Certificate Management Environment (ACME)

Certificates in PKI using X.509 (PKIX) are used for a number of purposes, the most significant of which is the authentication of domain names.

Thus, certificate authorities in the Web PKI are trusted to verify that an applicant for a certificate legitimately represents the domain name(s) in the certificate.Today, thi verification is done through a collection of ad hoc mechanisms.

ACME protocol automates process of verification and certificate issuance.



Let's Encrypt is a **free**, **automated**, and **open** Certificate Authority.

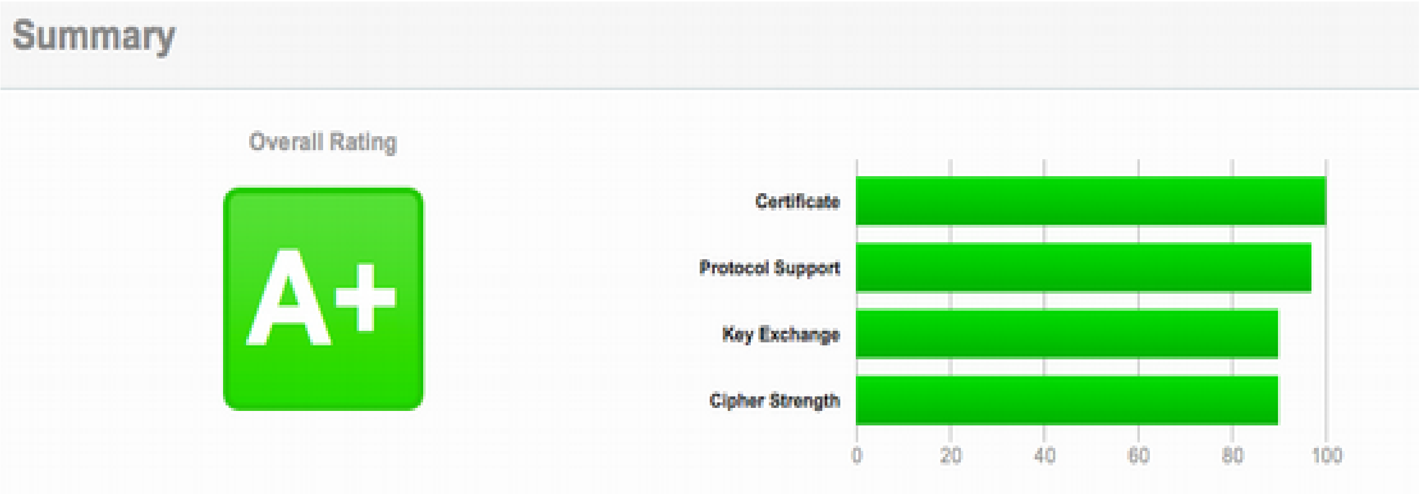
Get your site on 🔒 https://

SSLABS – www.ssllabs.com

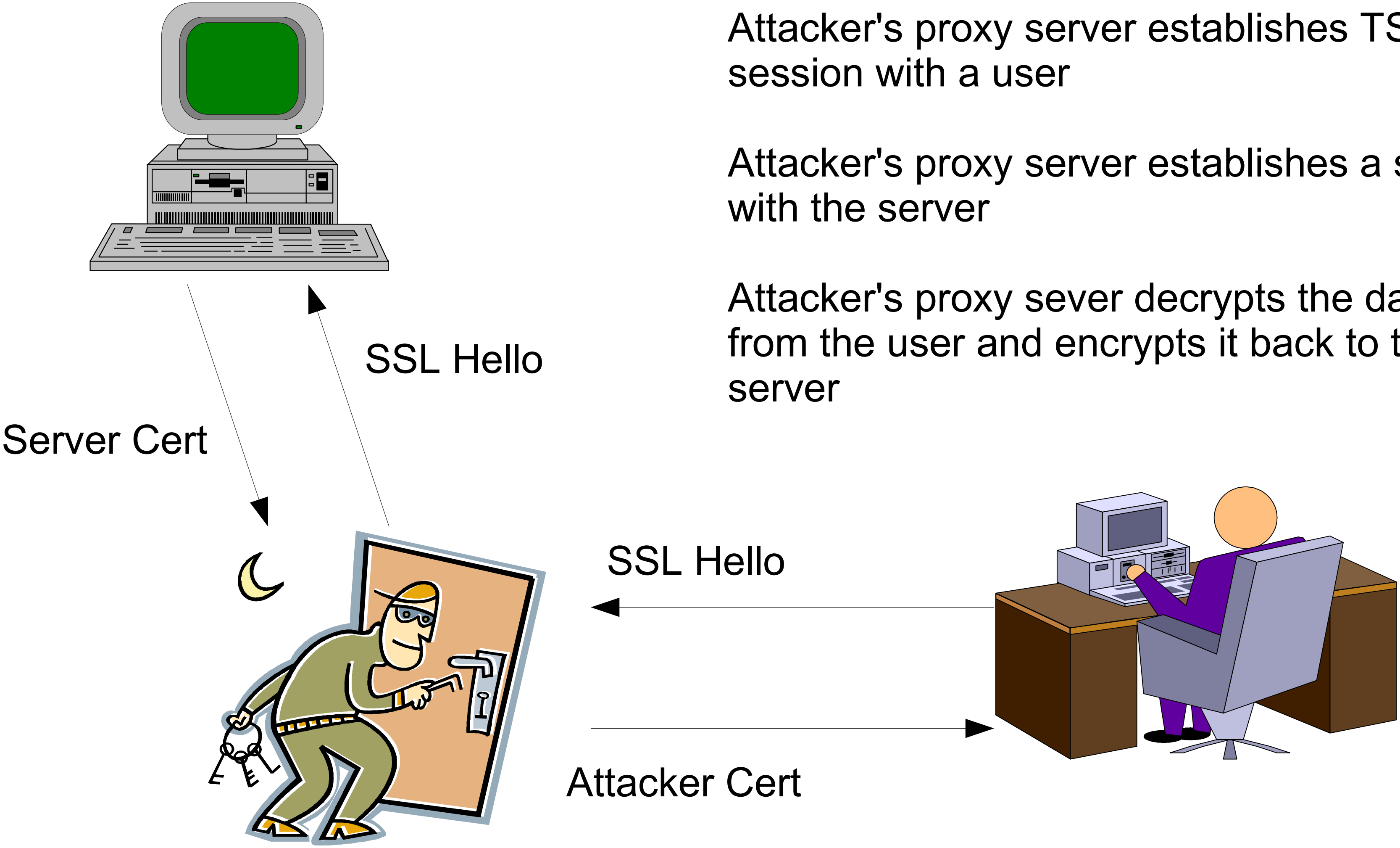
HOW WELL DO YOU KNOW SSL?

If you want to learn more about the technology that protects the Internet, you've come to the right place.

Test your server »
Test your site's certificate and configuration
Test your browser »
Test your browser's SSL implementation
SSL Pulse »
See how other web sites are doing
Documentation »
Learn how to deploy SSL/TLS correctly



Man in the Middle



Server Cert

SSL Hello

SSL Hello

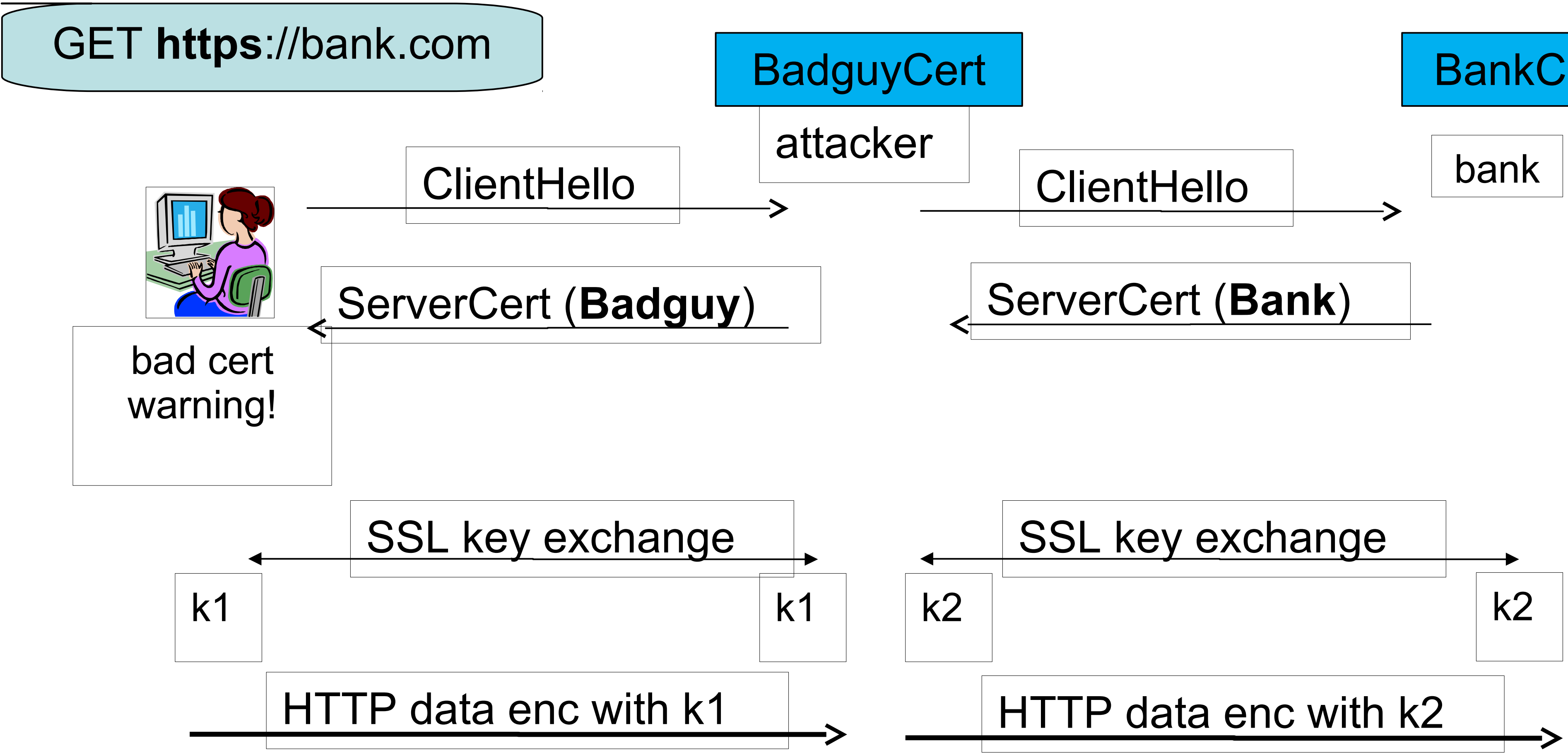
Attacker Cert

Attacker's proxy server establishes TSL session with a user

Attacker's proxy server establishes a session with the server

Attacker's proxy sever decrypts the data from the user and encrypts it back to the server

Man in the middle attack using invalid certs



GET **https**://bank.com

BadguyCert

BankCert

attacker

bank

ClientHello

ClientHello

ServerCert (**Badguy**)

ServerCert (**Bank**)

bad cert warning!

SSL key exchange

SSL key exchange

k1

k1

k2

k2

HTTP data enc with k1

HTTP data enc with k2

Attacker proxies data between user and bank.
Sees all traffic and can modify data at will.

Discussion