Week 5 – Encryption and Usage

Practical Examples of Cryptography

Encryption and Hashing

Channel Encryption
- Network channel encryption
- WiFi encryption
- SSL/TLS encryption
- Secure Email

Machine Encryption and Hashing
- Disk encryption
- Password protection

E-Payment

Octopus card

Bitcoin

NFC Payment

E-Cheque

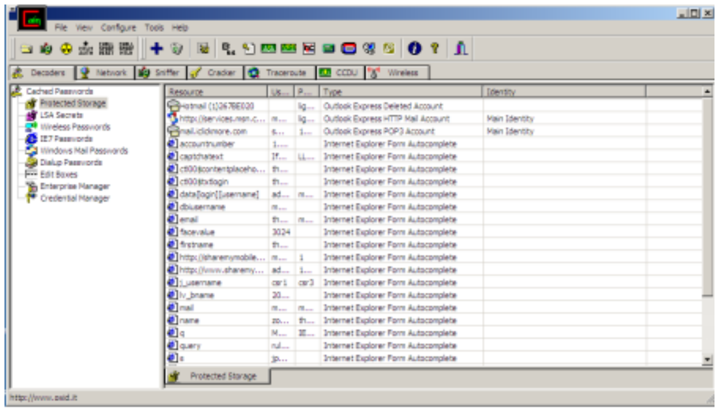
Peer-to-peer payment

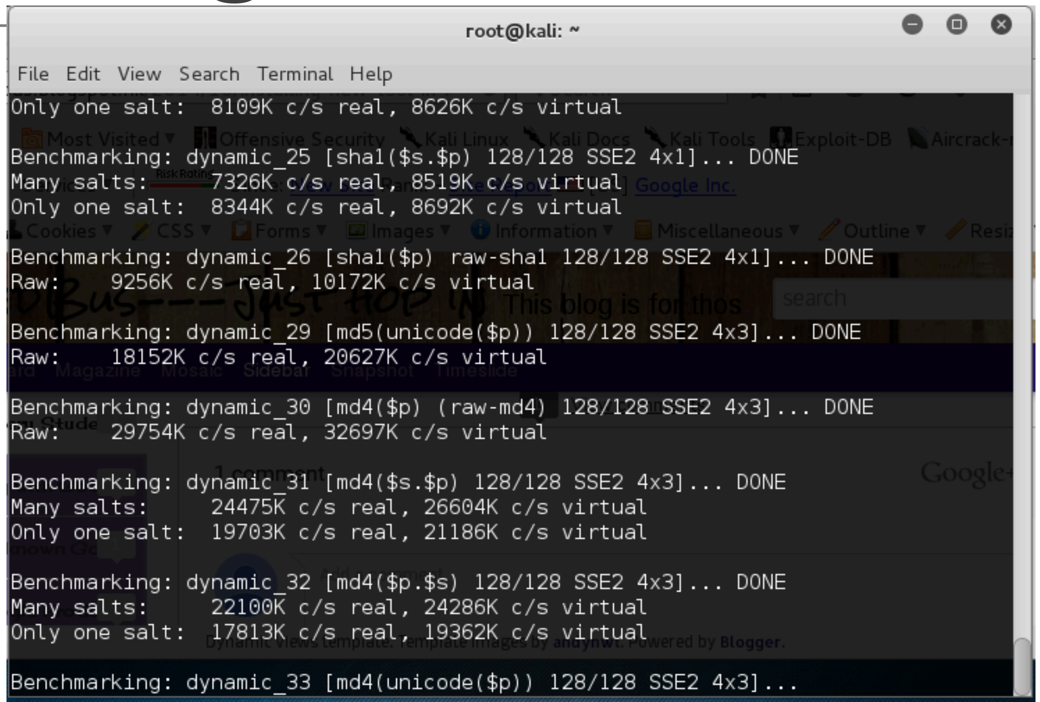
Password cracking

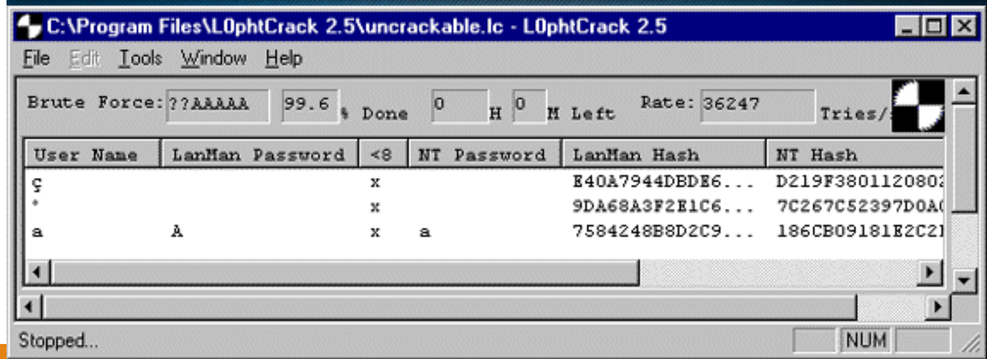
John the ripper

L0PHTCrack

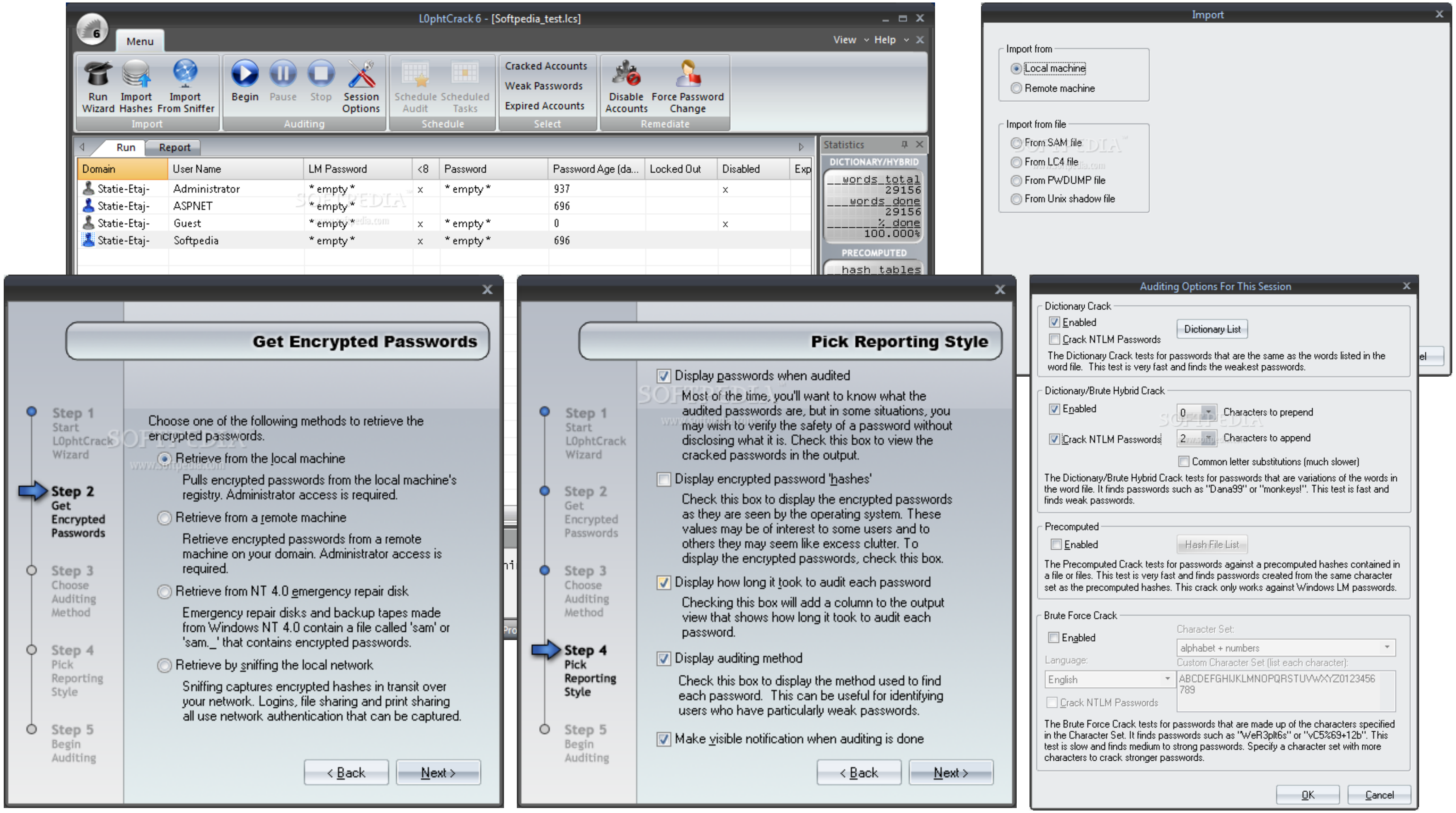
Cain & Abel







Password cracking



Cryptography and Data Security

Fundamental

Terms

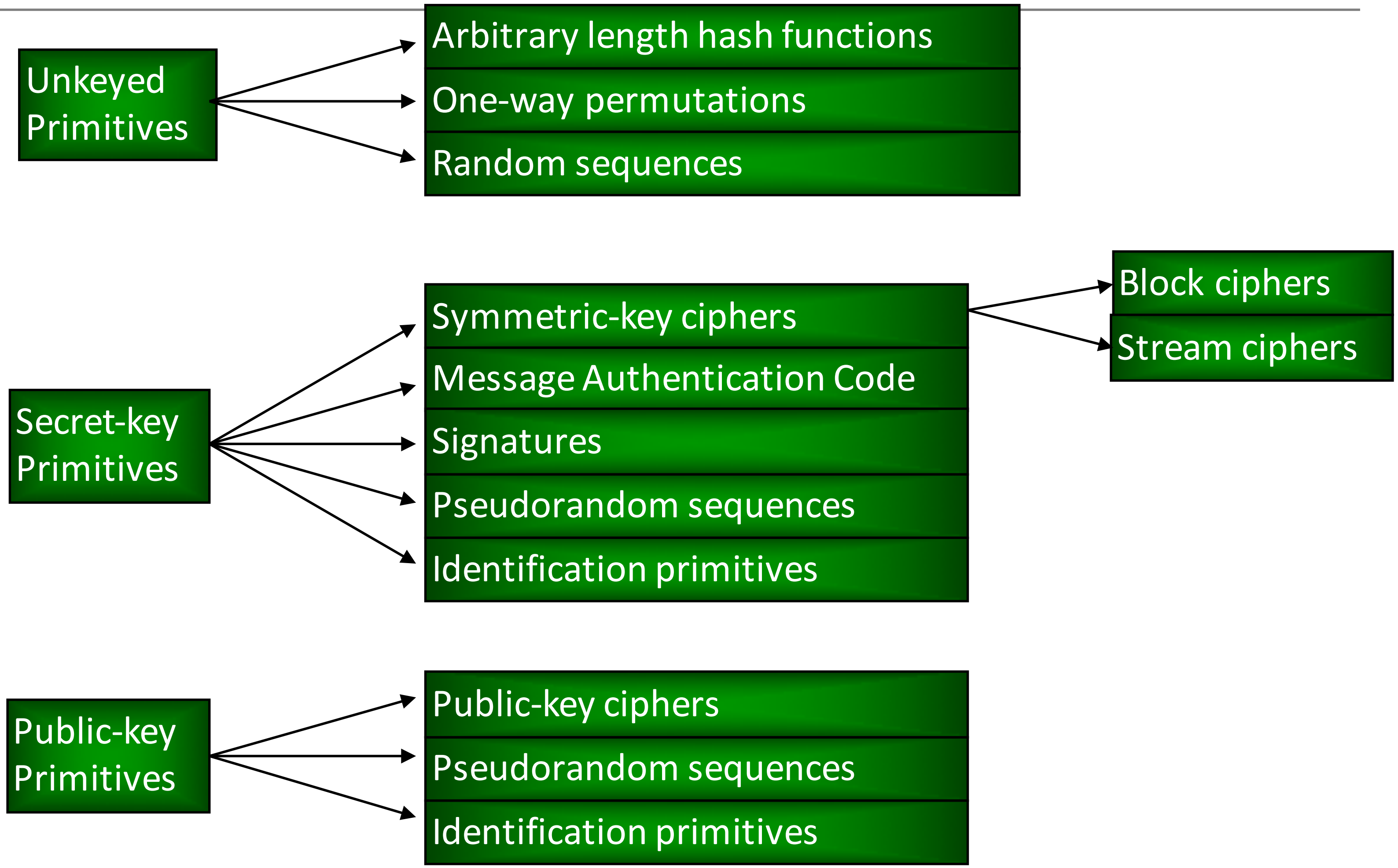
- Cryptography means “hidden writing”
- Encryption is coding a message in such a way that its meaning is concealed
- Decryption is the process of transforming an encrypted message into its original form
- Plaintext is a message in its original form
- Ciphertext is a message in its encrypted form

Why Encryptions ?

Keep secret

Maintain integrity

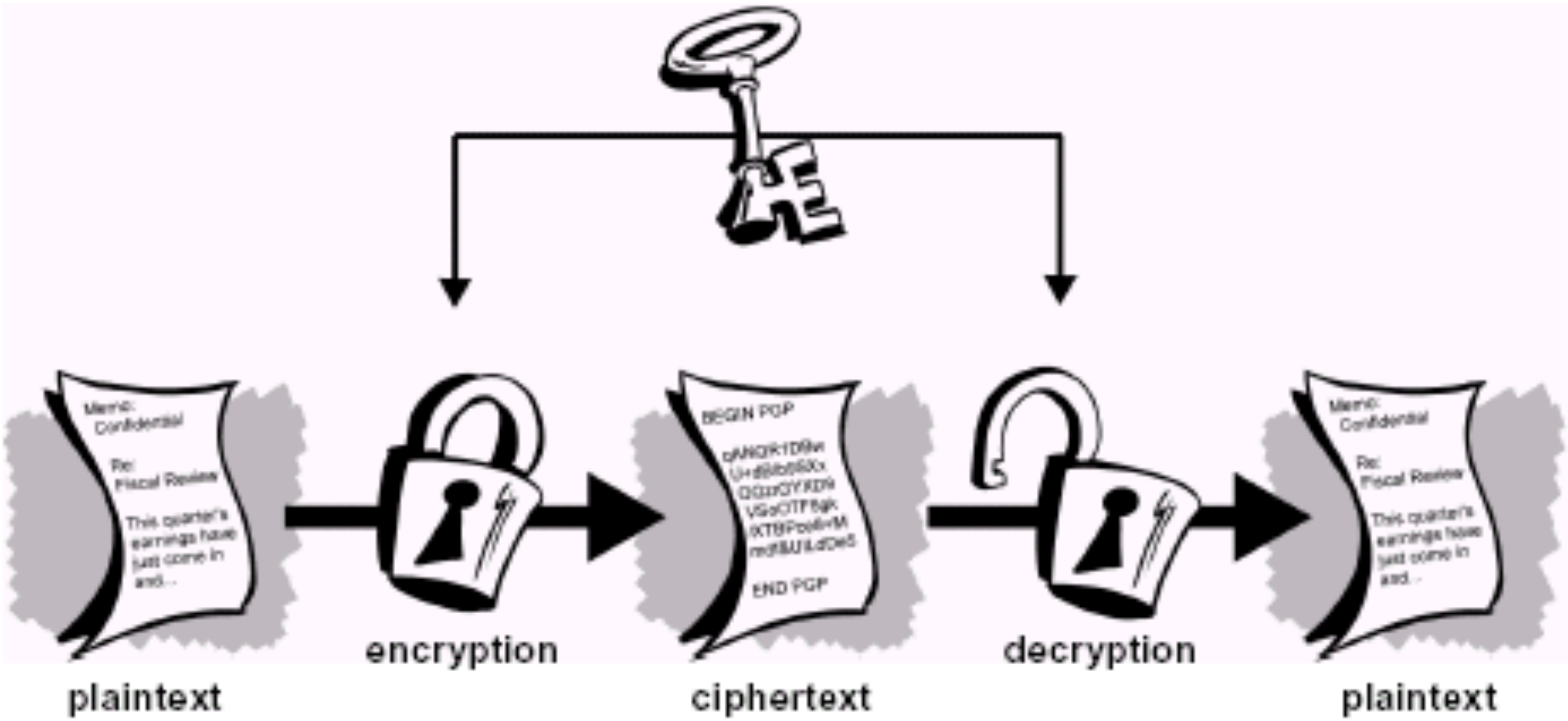
Taxonomy of Cryptographic Techniques



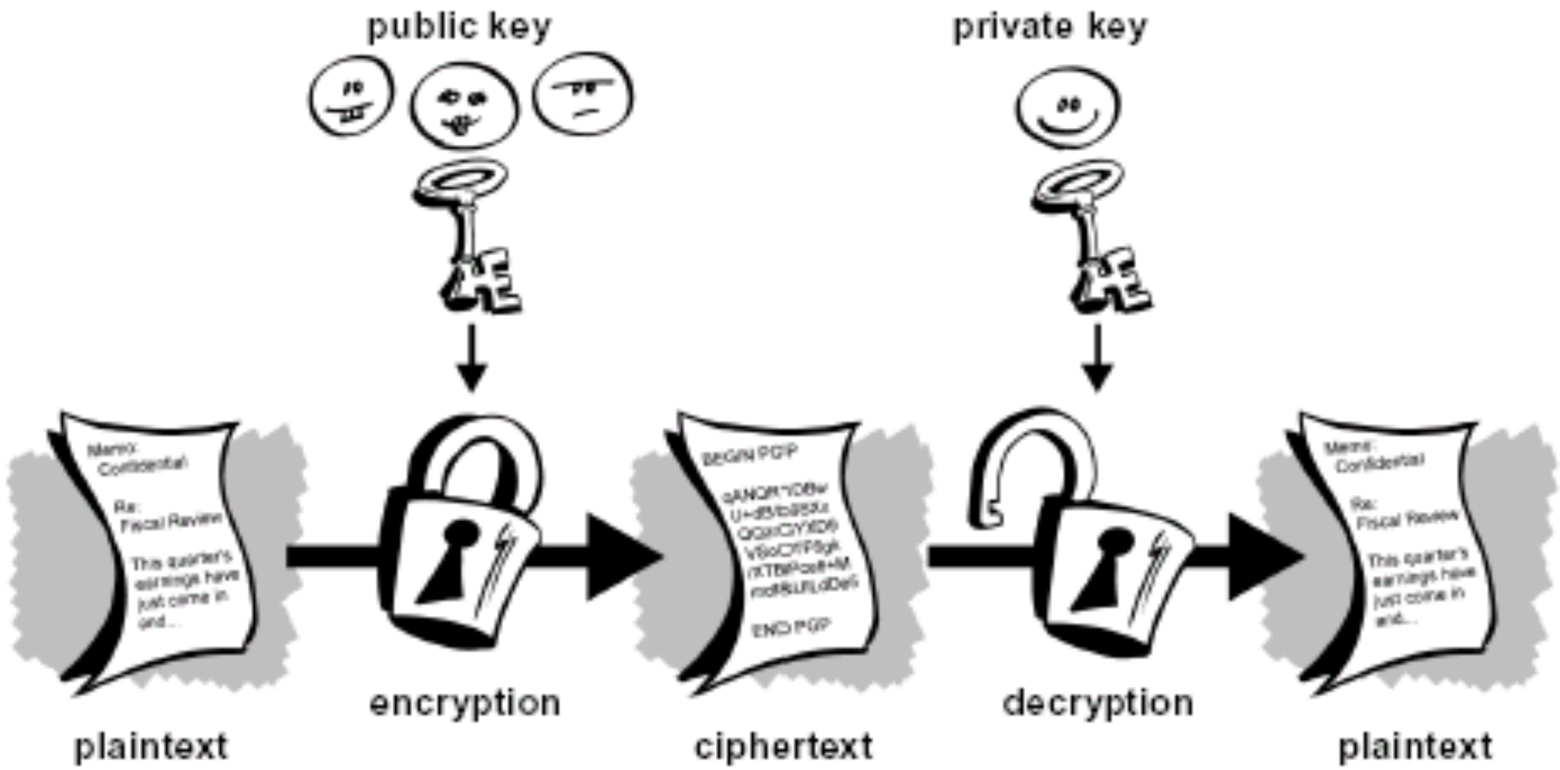
Simplified Explanation of Cryptography

ABC ?.9 ABC

Symmetric (DES, 3DES, RC4, IDEA, AES)



Asymmetric (RSA, Diffie-Hellman, Elliptic Curve, ElGamal)



Types of Cryptography

Symmetric / Secret Key

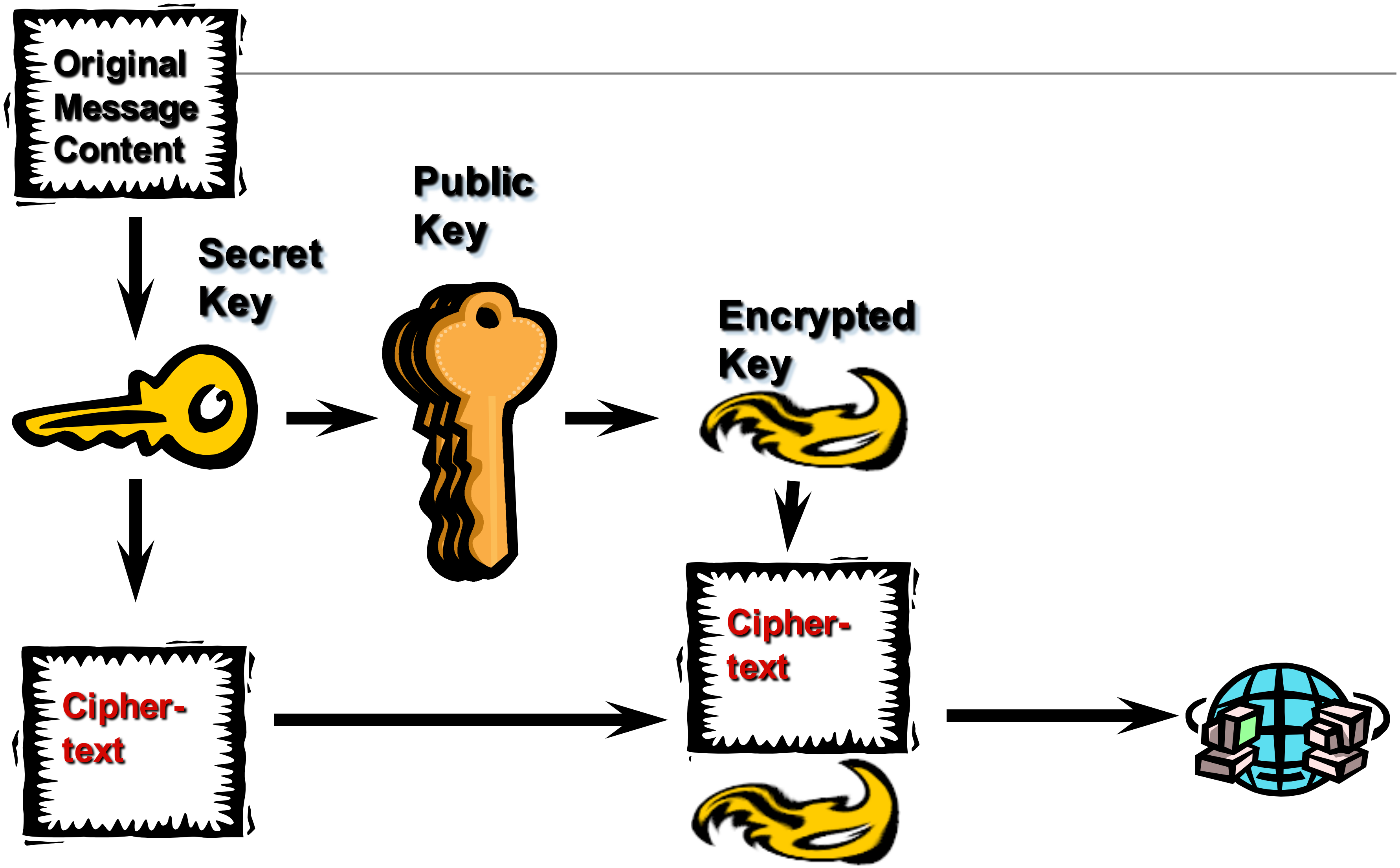
- Fast
- Serious problem in key management

Asymmetric / Public Key

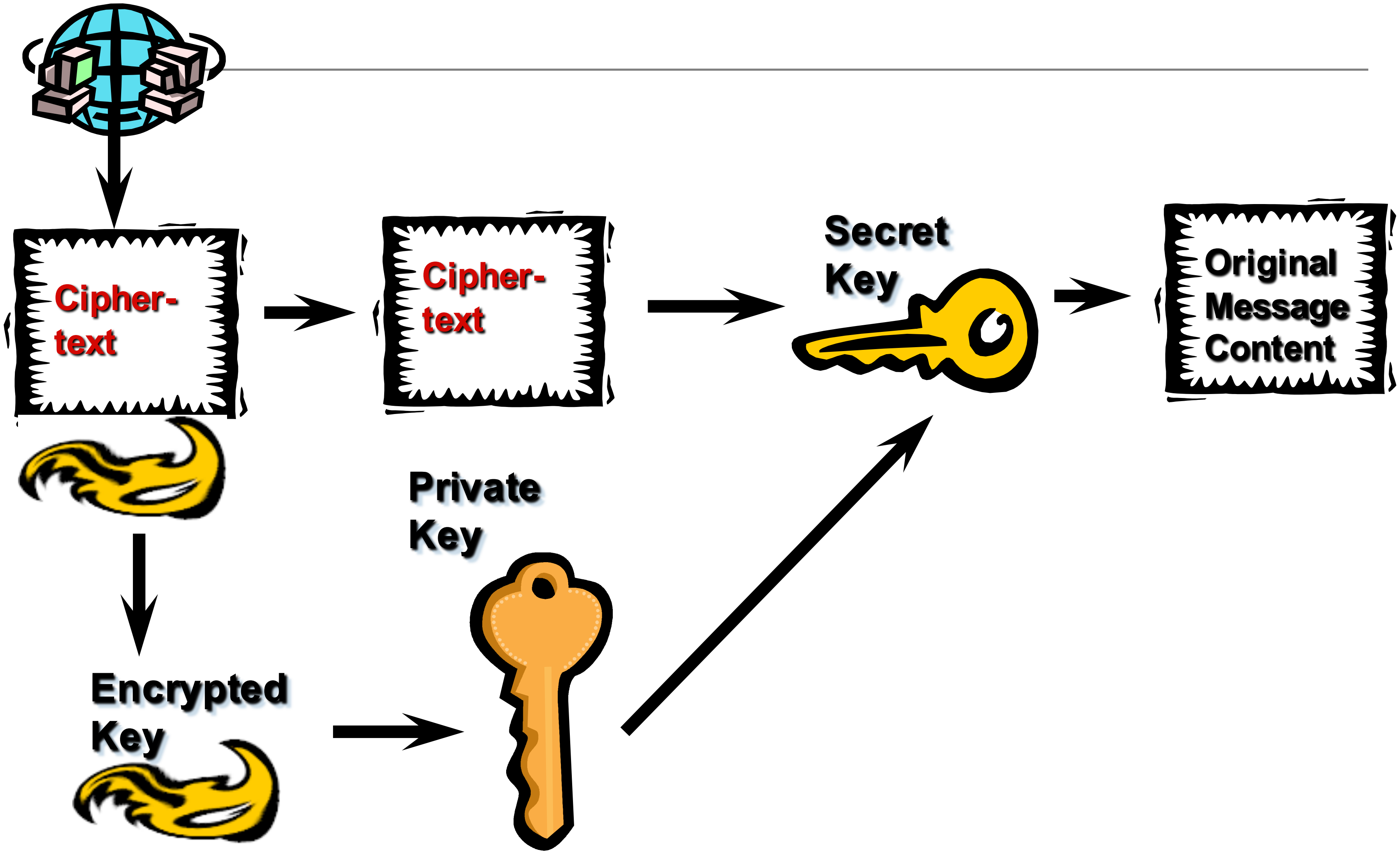
- Slow
- Minor problem in key management

Hash / One-way transformation

Secure envelope approach -- encryption



Secure Envelope Approach – Decryption



Encryption Cryptographic Systems

Ciphers type

- Block ciphers
- Stream ciphers

Encryption Scheme

- Symmetric key system (secret key)
  - DES, IDEA
  - AES
- Asymmetric key system (public key)
  - RSA, DSA
- hybrid key system

Block Ciphers vs. Stream Ciphers

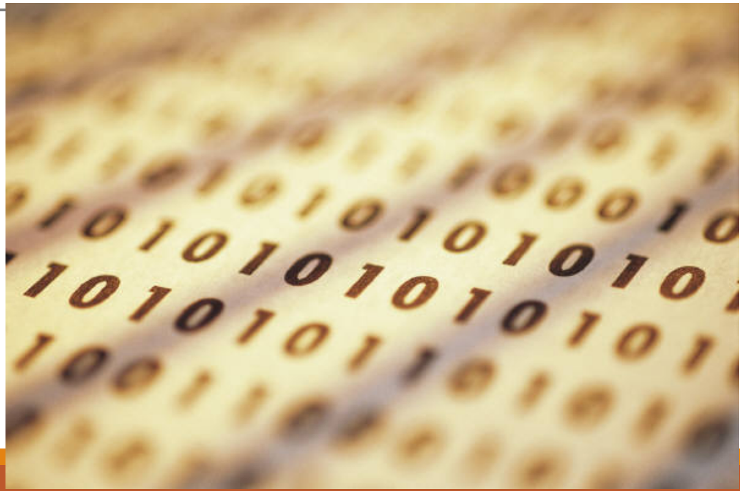
Stream

- Not suitable for software implementation
  - time consuming manipulation of bits
- Easier to analyze mathematically
- Single error can damage only a single bit of data
- Application: T-1 link between 2 computers

Block

- Easy to implement in software
- General in use and algorithms are more strong
- Single error can damage a block's worth of data
- Application: data on computer desk

Hash and Digital Signature



Hashing

Hash functions, is to produce message digest

- Computationally infeasible to find a message which hashes to the same digests as a given message
- Computationally infeasible to find any two strings which hash to the same value

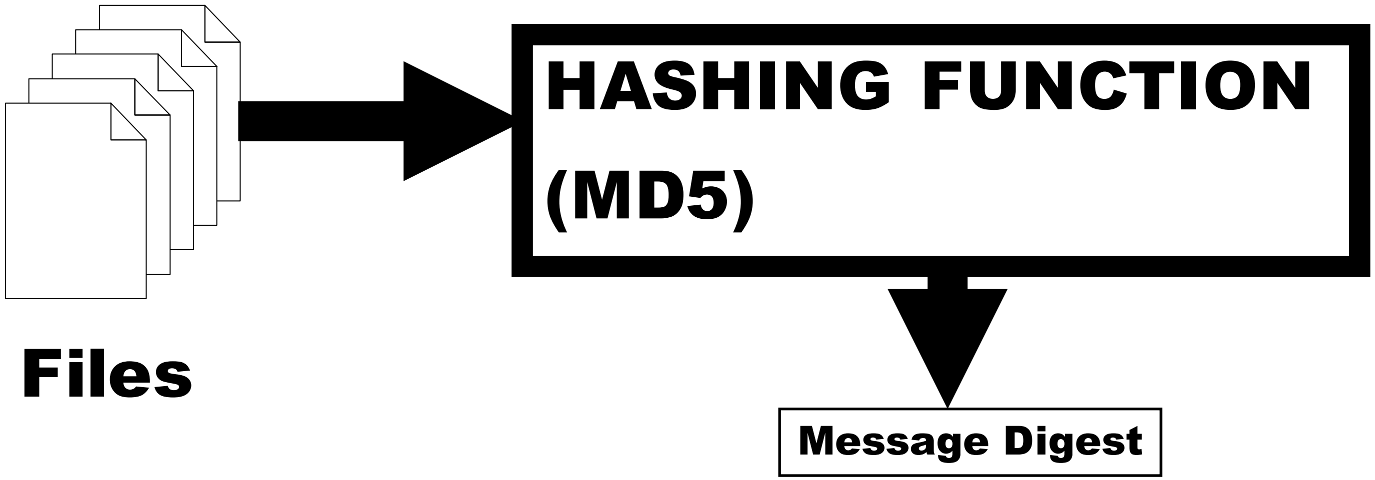
Message digest

- Fixed size result of hashing a message and is smaller than the full message

Digital signature

- Electronic signature of a digital message

Hash (MD4, MD5, SHA-1, SHA-256)



Digital Signature

To achieve non-repudiation

- Prevent senders from denying they have sent messages

Digital signature shall provide:

- Receiver must be able to validate sender's signature
- Signature must not be forgeable
- Sender of a signed message must not be able to repudiate it later

Digital signature cannot be constant

- A function of the entire document to sign

Digital Signature (cont'd)

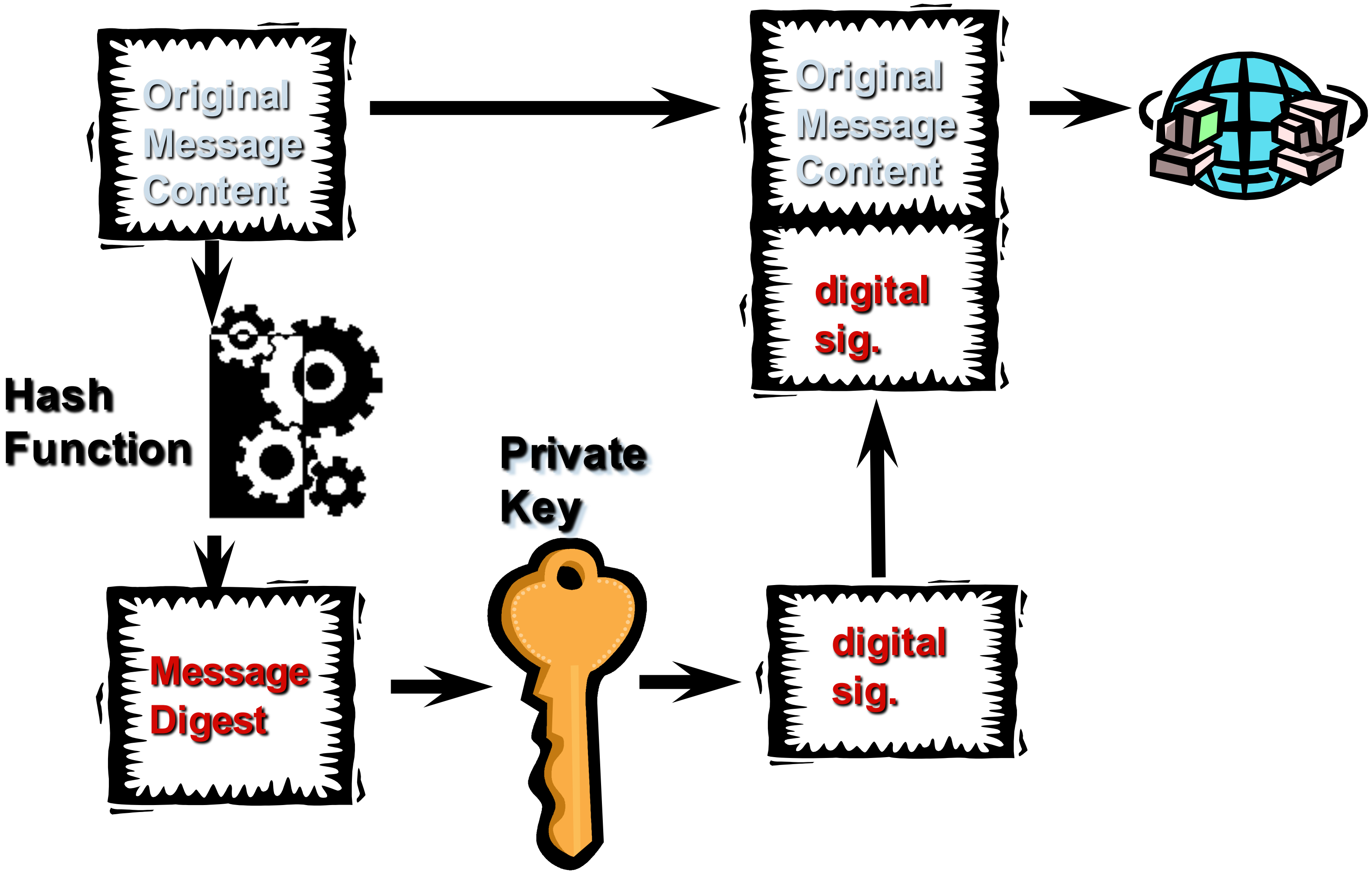
True signature

- Signed messages are forwarded directly from signer to recipient

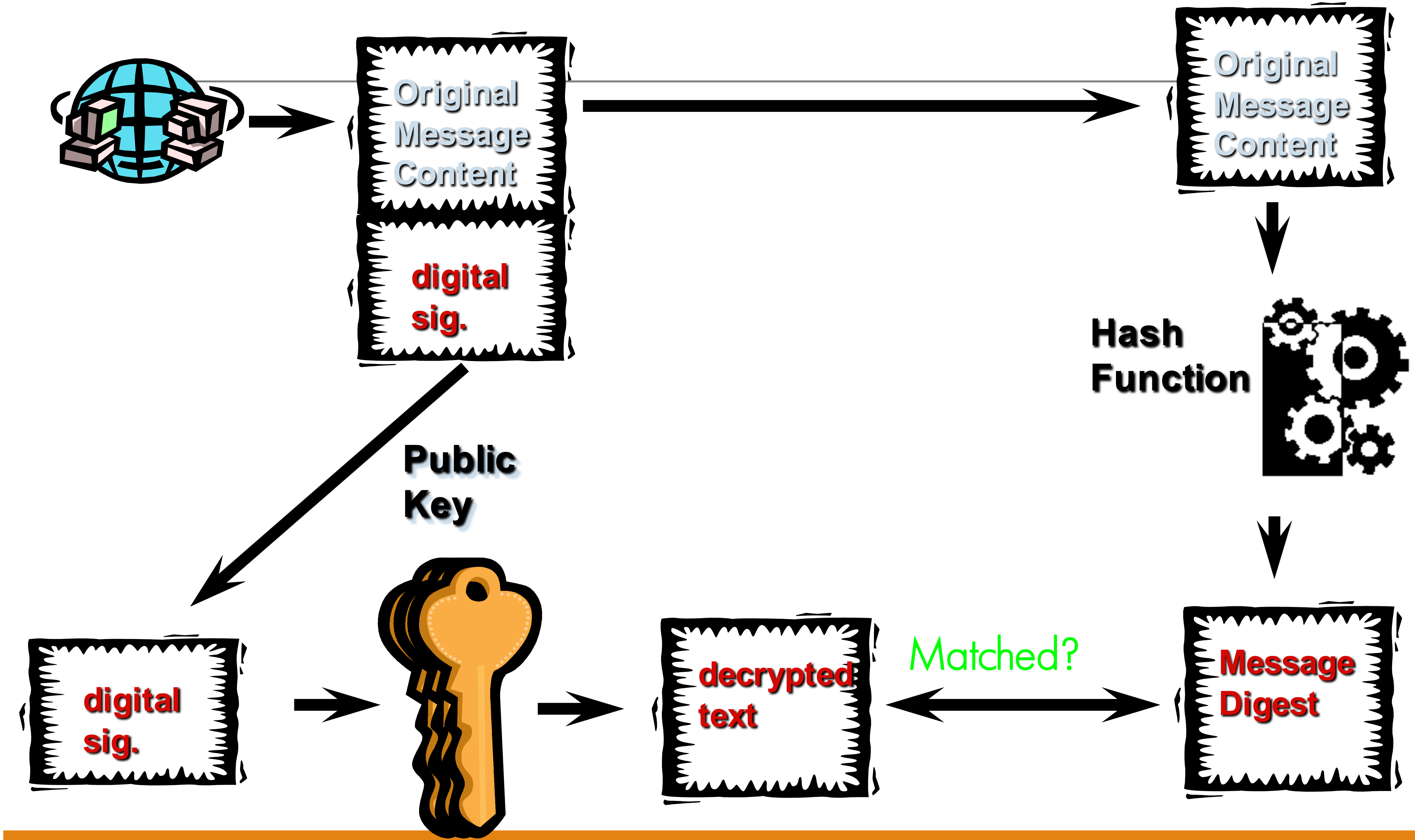
Arbitrated signature

- A witness validates a signature and transmits the message on behalf of the senders

Digital Signature Generation Description



Digital Signature Verification Description



AshleyMadison case (Aug 2015)

The company used a salt-hash-and-stretch password storage system called bcrypt.

- Bcrypt generates a random string of characters (the salt) and mixes it with your password;
- scrambles the password cryptographically (the hash); and does so over and over again (the stretch).

A blogger who went after the the bcrypt hashes recovered only 4000 passwords in a week.

CynoSure Prime recovered the passwords for over 11 million of the MD5 hashes in about 10 days

https://nakedsecurity.sophos.com/2015/09/10/11-million-ashley-madison-passwords-cracked-in-10-days/

24 AshleyMadison: $500K Bounty for Hackers

AUG 15

AshleyMadison.com, an online cheating service whose motto is "Life is Short, Have an Affair," is offering a $500,000 reward for information leading to the arrest and prosecution of the individual or group of people responsible for leaking highly personal information on the company's more than 30 million users.

The bounty offer came at a press conference today by the police in Toronto — where AshleyMadison is based. At the televised and Webcast news conference, Toronto Police Staff Superintendant **Bryce Evans** recounted the key events in "Project Unicorn," the code name law enforcement officials have assigned to the investigation into the attack. In relaying news of the reward offer, Evans appealed to the public and "white hat" hackers for help in bringing the attackers to justice.

A snippet of the message left behind by the Impact Team.

"The ripple effect of the impact team's actions has and will continue to have a long term social and economic impacts, and they have already sparked spin-offs of crimes and further victimization," Evans

More advanced type of password storing mechanism

PBKDF2 (Password-Based Key Derivation Function 2) is a key derivation function that is part of RSA Laboratories' Public-Key Cryptography Standards (PKCS) series, specifically PKCS #5 v2.0, also published as Internet Engineering Task Force's RFC 2898

https://www.youtube.com/watch?v=425_1-eFel4

Bcrypt

bcrypt is a key derivation function for passwords designed by Niels Provos and David Mazières, based on the Blowfish cipher

Incorporating a salt to protect against rainbow table attacks

BSD and SUSE Linux used brcrypt for default encryption

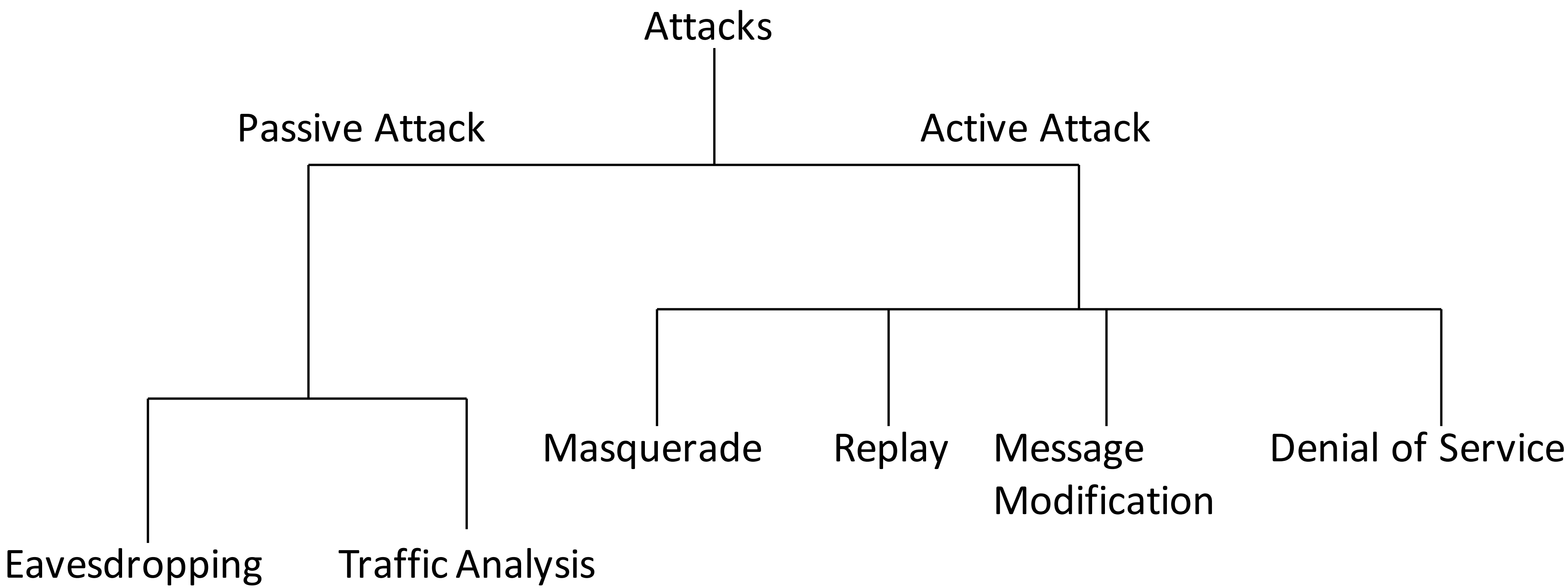
- The prefix "$2a$" or "$2b$" (or "$2y$") in a hash string in a shadow password file indicates that hash string is a bcrypt hash in modular crypt format

Why "To safely store a password using bcrypt"

- A modern server can calculate the MD5 hash of about 330MB every second. If your users have passwords which are lowercase, alphanumeric, and 6 characters long, you can try every single possible password of that size in around 40 seconds.

Wireless Network Attack

Potential attack methods



Security Threats in Wireless Environment

Compromise of encryption key

Hardware theft is *equivalent* to *key* theft

Packet spoofing, disassociation attack

Rogue AP

Known plain-text attack

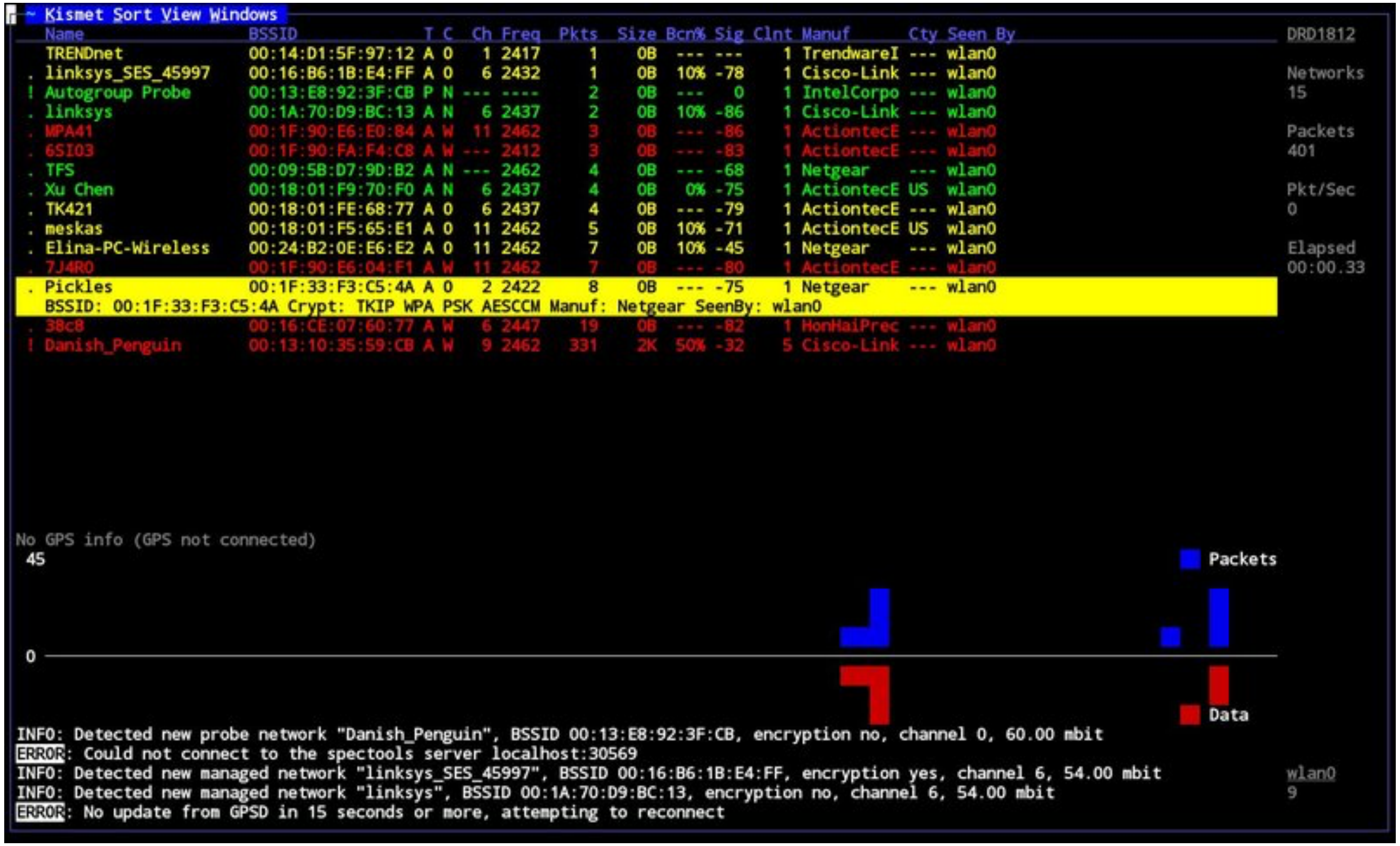
Brute force attack, Dictionary attack

Passive monitoring

Replay attack, insertion attacks, jamming

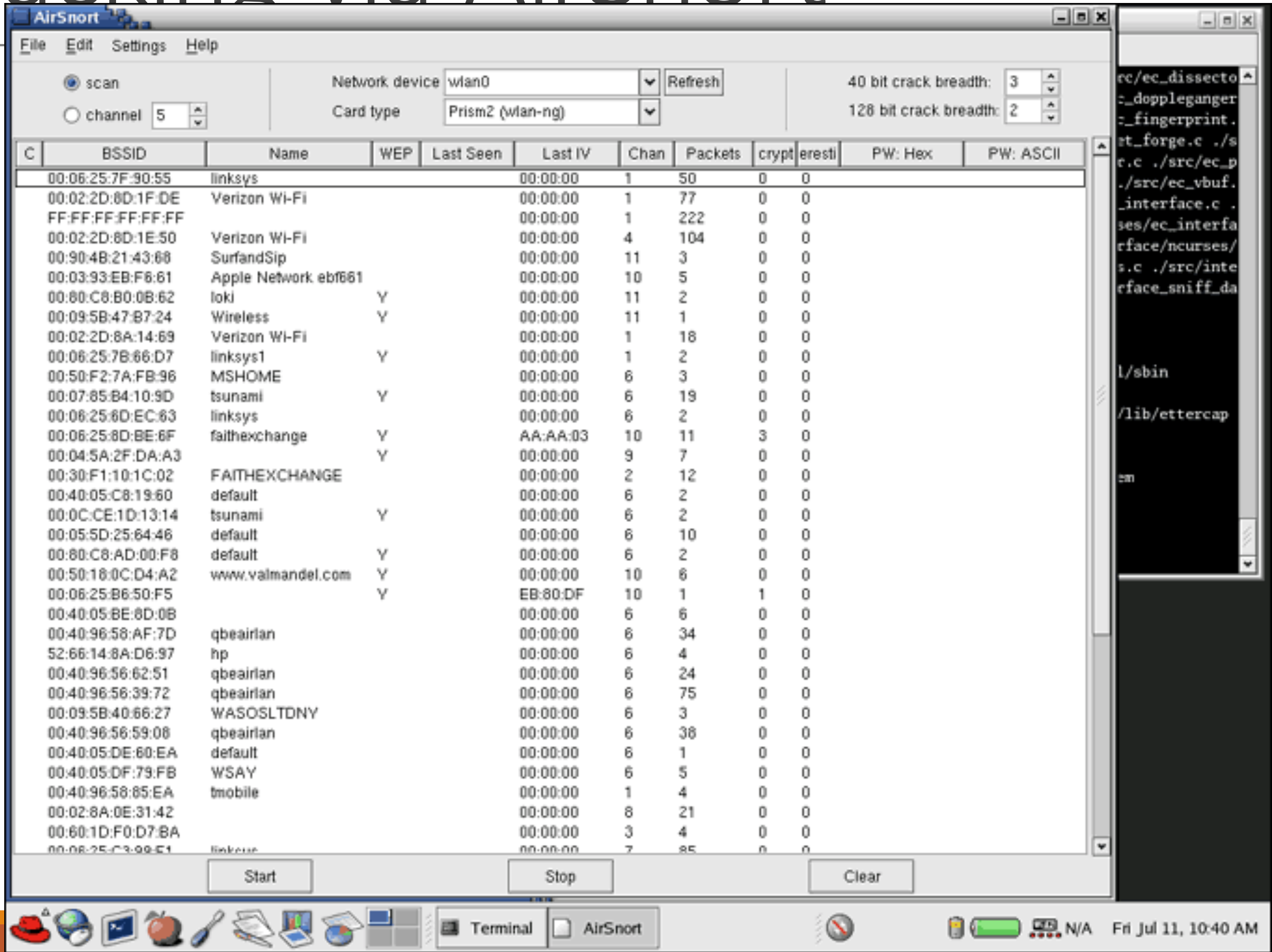
Packet Integrity

Using Kismet



https://www.kismetwireless.net/screenshot.shtml

Cracking via AirSnort



An example of Rogue/Fake AP

No Wi-fi but Hong Kong's Ocean Park is among world's riskiest attractions for phone hacking

Danny Lee
danny.lee@scmp.com

PUBLISHED : Monday, 24 August, 2015, 1:42pm
UPDATED : Monday, 24 August, 2015, 5:54pm

Ocean Park's Wi-fi network will not be up and running until later this year. Photo: Felix Wong

SHARE

3 Like | 3 Share | 24 Tweet | 1 reddit | 12 Share | 0 +1 | 0 Comments

Email | Print

Top Hong Kong destination Ocean Park has been branded one of the riskiest tourist attractions for exposing mobile devices to cyberattacks, alongside New York's Times Square and Disneyland Paris, according to a US security survey.

From Skycure Study: 2015 Best & Worst Tourist Attractions for Mobile Security (18 Aug 2015)

https://www.skycure.com/blog/skycure-study-2015-best-worst-tourist-attractions-for-mobile-security/

http://www.scmp.com/news/hong-kong/law-crime/article/1852083/no-wi-fi-ocean-park-listed-among-worlds-riskiest

Intercepting WiFi traffic

Intercepting WiFi traffic through

- Additional of access points
- Capture of the traffic through peer-to-peer attacks (attack capturing of sensitive data file, password files)

Denial of Services Attacks

Mainly 3 types of Wireless DoS attacks

- RF jamming
  - Jamming of DSSS WLAN using broadcast signal
- Data flooding
  - Flooding of the WLAN by pulling/pushing very large file from/to the Internet
  - Flooding of the WLAN using a packet generator software package
- Hijacking
  - Hijacking occurs at OSI layer 3 where intruder attempt to initiate an attack (pretend to be authorized access point)

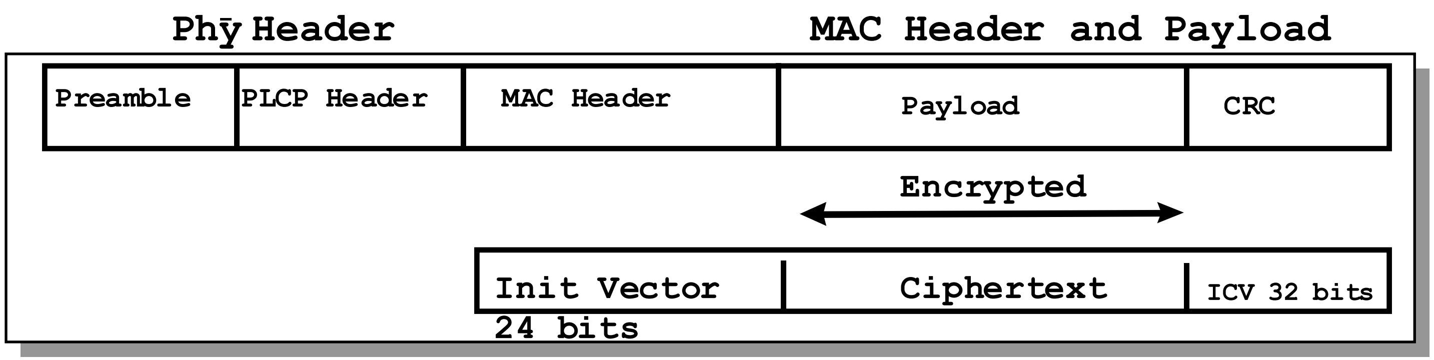
WEP Attack

Wired Equivalency Privacy

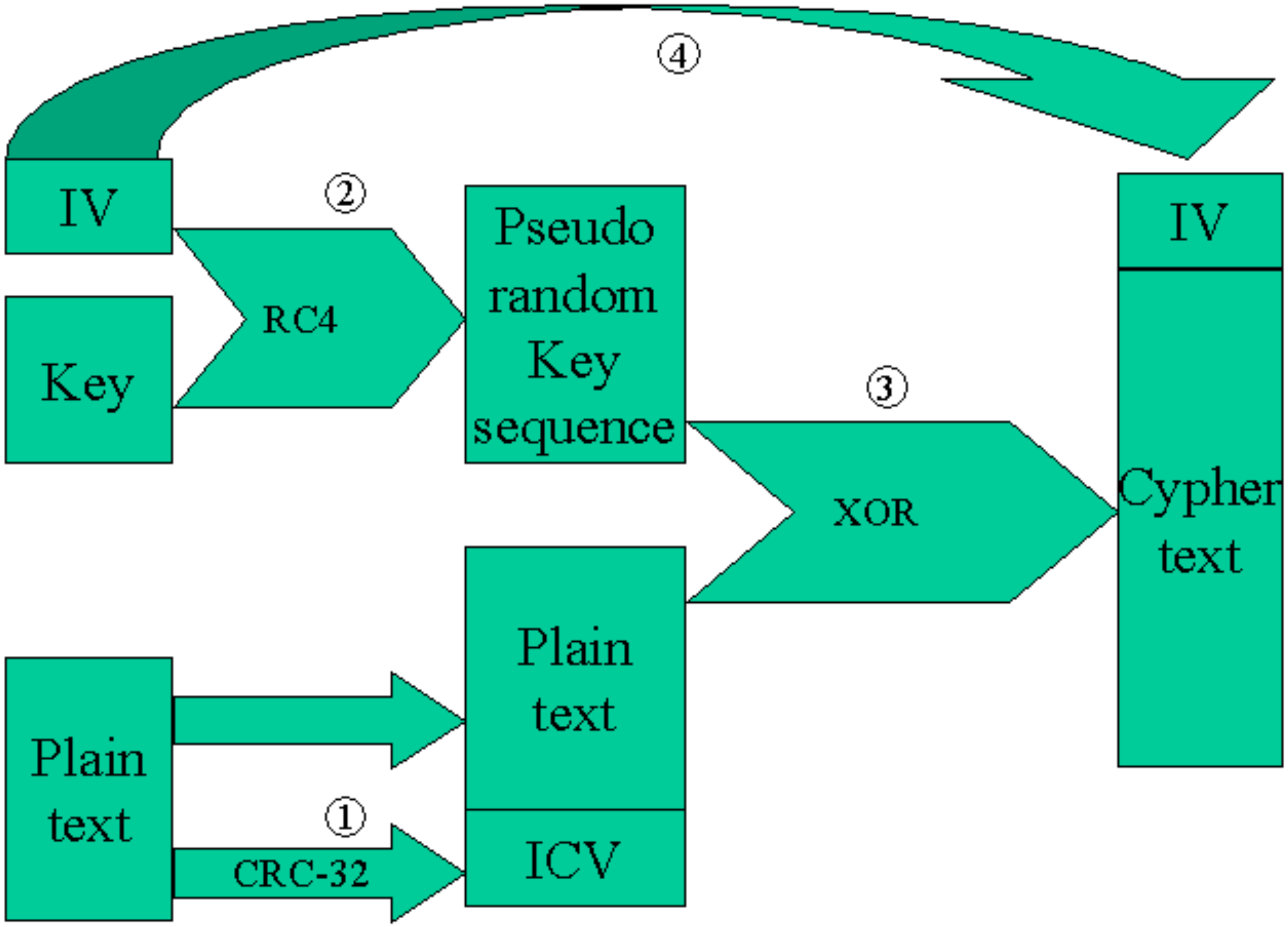
WEP: symmetric encryption (shared key), defines method but not how to share and distribute/manage keys

RC4 algorithm (40+24 bits keys) WIFI compliant

104 + 24 bits proprietary (non IEEE standards) but interoperable implementations (ie Lucent/Compaq - Cisco)



WEP Algorithm



WEP Broken

Due to 24-bit IV

- 50% probability the same IV will repeat after 5000 packets for 40-bit WEP

So, in high volume network traffic, it's easy to crack the WEP key within minutes

In 2007, Erik Tews, Andrei Pychkine, and Ralf-Philipp Weinmann optimizes the attack

WEP Attack

http://www.twistedethics.com/2007/09/12/notes-cracking-wep-with-aircrack-ng-and-airpcap-tx/

http://www.twistedethics.com/2007/06/11/aircrack-ptw-for-windows/

http://www.twistedethics.com/2007/06/11/cracking-wep-with-aircrack-ptw-in-windows-with-airpcap-and-cain/

http://www.twistedethics.com/2007/05/26/cracking-wep-with-airpcap-and-cain-and-abel/

WEP and WPA

Difference between WEP and WPA

- WEP doesn't obscure password in an effective way.
- That is a huge security risk because hackers can directly extract it from packets sent during authentication.
- This makes it easy for those same folks to sit in parking lot or lounge around in a mall and break into networks.

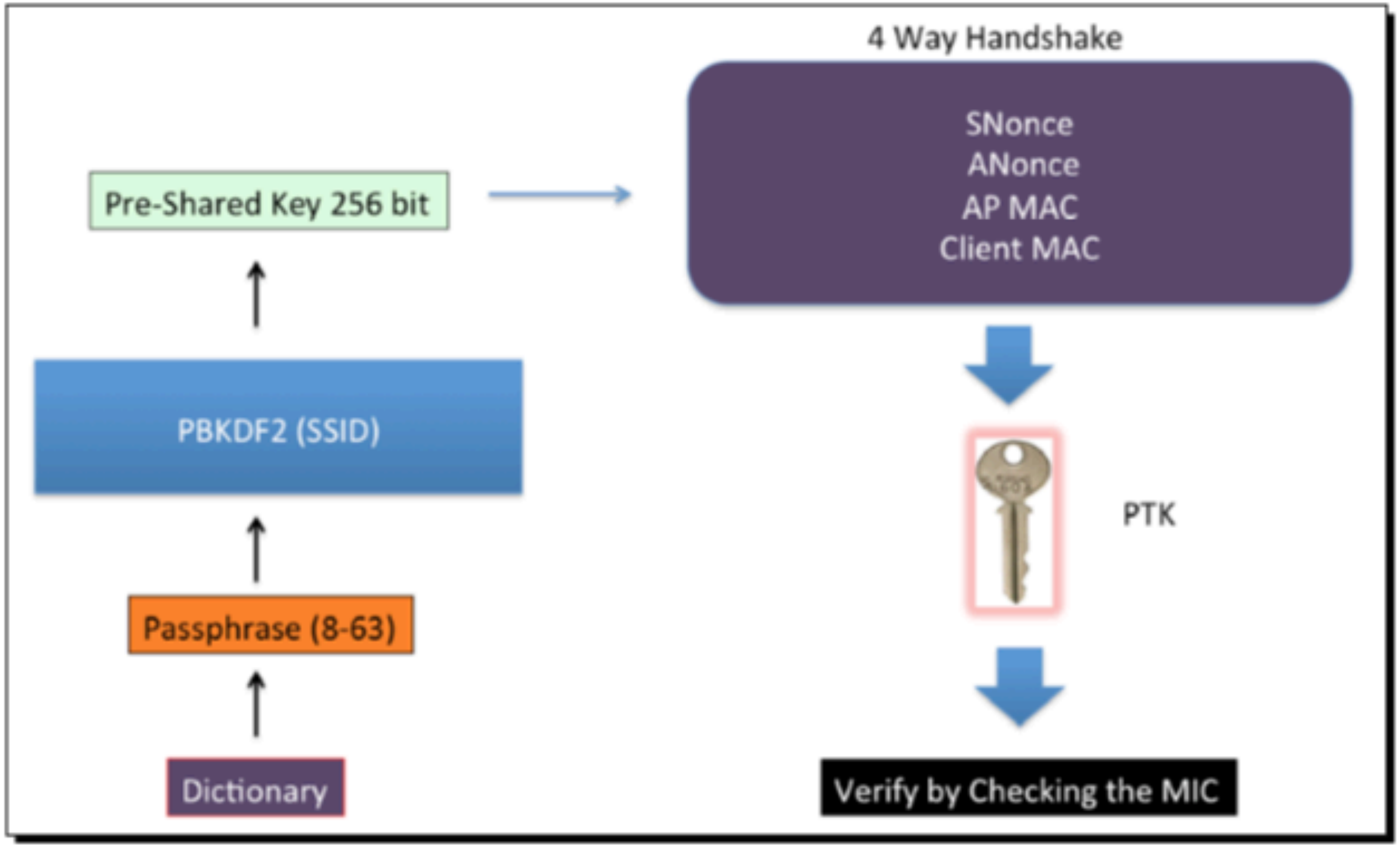
WPA attacks

http://www.hak5.org/episodes/episode-3x06-release

Tools used:

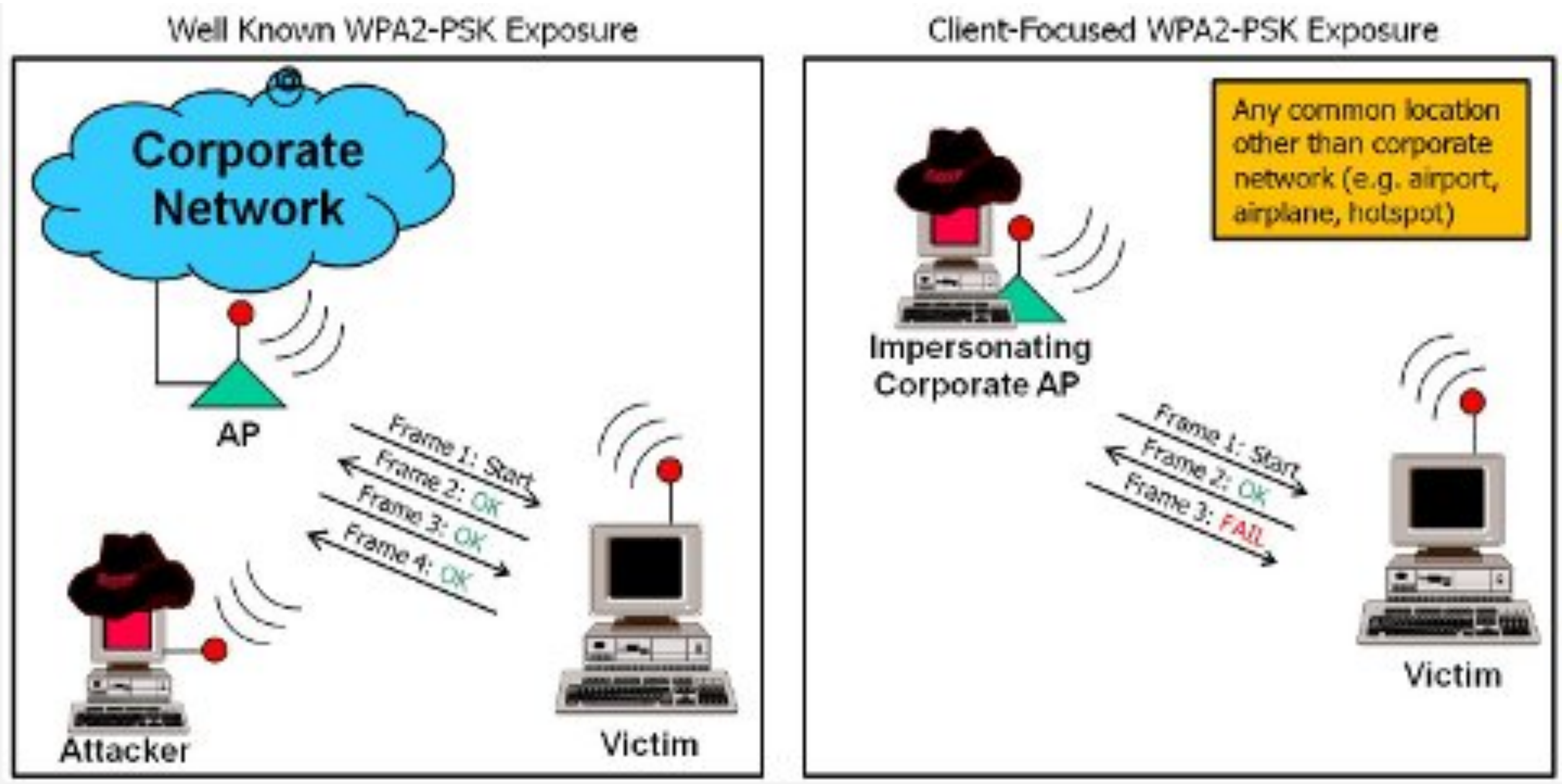
- Backtrack 3
  - Kismet (search for BSSID and channel)
  - Switch the WIFI to monitor mode
    - Airmon-ng stop ath0
    - Airmon-ng start wifi0 11
  - Dump the network packets
    - airodump-ng –c 11 –bssid <BSSID> -w psk ath0
  - Replay to create more packets
    - Aireplay-ng -0 15 –a <MAC Address of Access Point> -c <CLIENT> aht0
  - Crack the packets
    - Aircrack-ng –w word.lst –b <BSSID> psk *.cap

WPA 4 ways handshake for key creation



From: Kali Wireless Penetration Testing

WPA2 – PSK Attack



WPA/WPA2 TKIP Attack

WPA is a security framework whose:

- (a) encryption component is called Temporal Key Integrity Protocol (TKIP), and
- (b) authentication component can be either
  - Pre-Shared Key (PSK) which was designed for home users or
  - RADIUS (based on 802.1x) which was designed for enterprise usage.

In November 2008, TKIP encryption component of WPA was found to be vulnerable to a packet injection exploit.

WPA and WPA2 networks that use the more robust AES-CCMP encryption algorithm are immune to the attack.

TKIP Attack

TKIP was introduced in 2003, and amongst other enhancements, included a new per packet hashing algorithm, the Message Integrity Check (MIC).

MIC is based on a weak algorithm, designed to be accommodated on legacy WEP hardware

If more than two MIC failures are observed in a 60 second window, both the Access Point (AP) and client station shut down for 60 seconds

New TKIP attack uses a mechanism similar to the "chopchop" WEP attack to decode one byte at a time by using multiple replays and observing the response over the air. (http://www.aircrack-ng.org/doku.php?id=korek_chopchop)

Breaking Wi-Fi Protected Access Temporal Key Integrity Protocol within An Hour, http://thehackernews.com/2015/07/crack-rc4-encryption.html (Video)

Small packets like ARP frames can typically be decoded in about 15 minutes by leveraging this exploit.

WiFi Network Attack

Beck-Tews attack

- TKIP is vulnerable to a keystream recovery attack
  - permits an attacker to transmit 7-15 packets of the attacker's choice on the network
  - Targets small ARP packets
  - (http://en.wikipedia.org/wiki/Temporal_Key_Integrity_Protocol)

Ohigashi-Morii attack

- Simpler and faster implementation of Beck-Tews attack using man-in-the-middle attack scheme

SSL, TLS, VPN, and IPSEC

SSL protocol

Application Layer - HTTP

```
GET / HTTP/1.1
Accept: */*
Accept-Language: en-us,zh-hk;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (…)
Host: www.ust.hk
Connection: Keep-Alive
Cookie: WTO_CLIENT=1
```

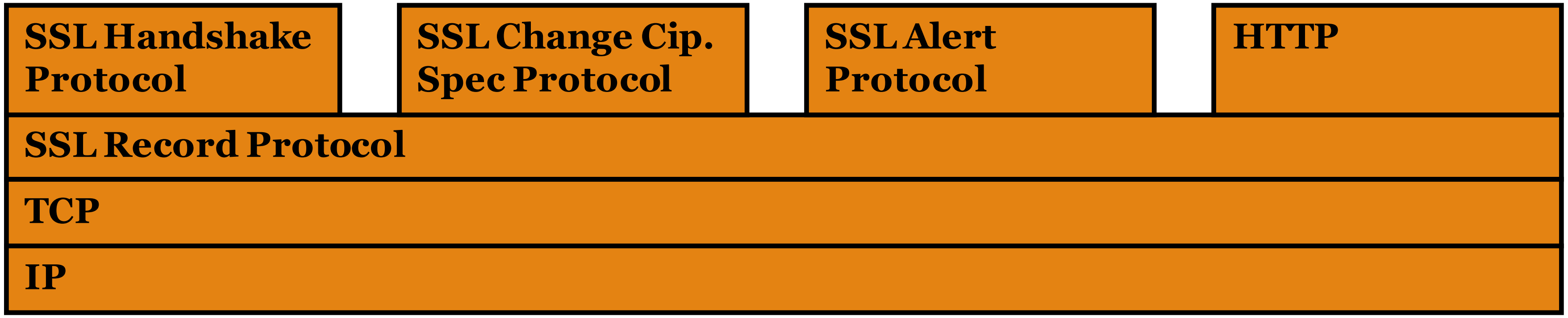
```
HTTP/1.1 200 OK
Date: Sat, 03 Jul 2004 12:01:30 GMT
Server: Apache/1.3.12 (Unix) mod_ssl/2.6.3 OpenSSL/0.9.5a
Last-Modified: Tue, 25 Jun 2002 09:59:10 GMT
ETag: "1a0a64-e4-3d183eee"
Accept-Ranges: bytes
Content-Length: 228
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<data……>
```

```
GET /en/index.html HTTP/1.1
Accept: */*
Accept-Language: en-us,zh-hk;q=0.5
Accept-Encoding: gzip, deflate
If-Modified-Since: Thu, 15 Jan 2004 06:21:45 GMT; length=208
User-Agent: Mozilla/4.0 (…)
Host: www.ust.hk
Connection: Keep-Alive
Cookie: WTO_CLIENT=1
```

```
HTTP/1.0 304 Not Modified
Date: Sat, 03 Jul 2004 12:01:27 GMT
Server: Apache/1.3.27 (Unix) mod_ssl/2.8.12 OpenSSL/0.9.6b
ETag: "439a3-d0-40063179"
```

SSL - Secure Socket Layer Protocol



SSL connection

- Transport (RM OSI) that provides suitable type of services. Every connection is associated with one session.

SSL session

- An association between a client and a server. Sessions are created by the Handshake protocol (defines set of cryptographic security parameters, that can be shared among multi

SSL - Secure Socket Layer Protocol (Cont.)

SSL Record Protocol - provides two services for SSL connections

- Confidentiality - handshake protocol defines shared secret key for encryption of SSL payloads
- Integrity - handshake protocol defines shared secret key to form message authentication code MAC

Operations of SSL Record Protocol

- Fragment
- Compress
- Add MAC
- Encryption

SSL - Secure Socket Layer Protocol (Cont.)

Change Cipher Spec Protocol

- Uses the SSL Record protocol
- The simplest of SSL protocols
- Only single byte message with value 1 and causes the pending state to be copied into the current state (updates cipher suite to be used in this connection)

SSL - Secure Socket Layer Protocol (Cont.)

Alert Protocol

- Used to convey SSL-related alerts to the peer entity
- Two types of alerts: warning and fatal
- In the case of fatal alert SSL immediately terminates the connection (the other connections of the session can continue but no new one is established)

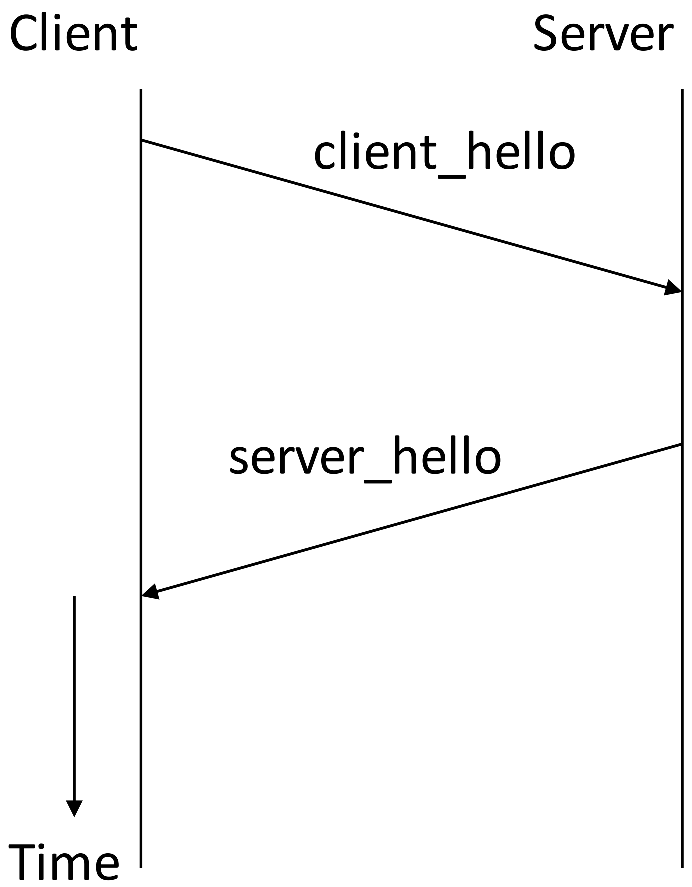
SSL - Secure Socket Layer Protocol (Cont.)

Handshake Protocol

- The most complex part of SSL
- Allows server and client to authenticate each other
- Negotiates an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL
- Is used before any application data are transmitted
- Consists of four phases

SSL - Secure Socket Layer Protocol (Cont.)

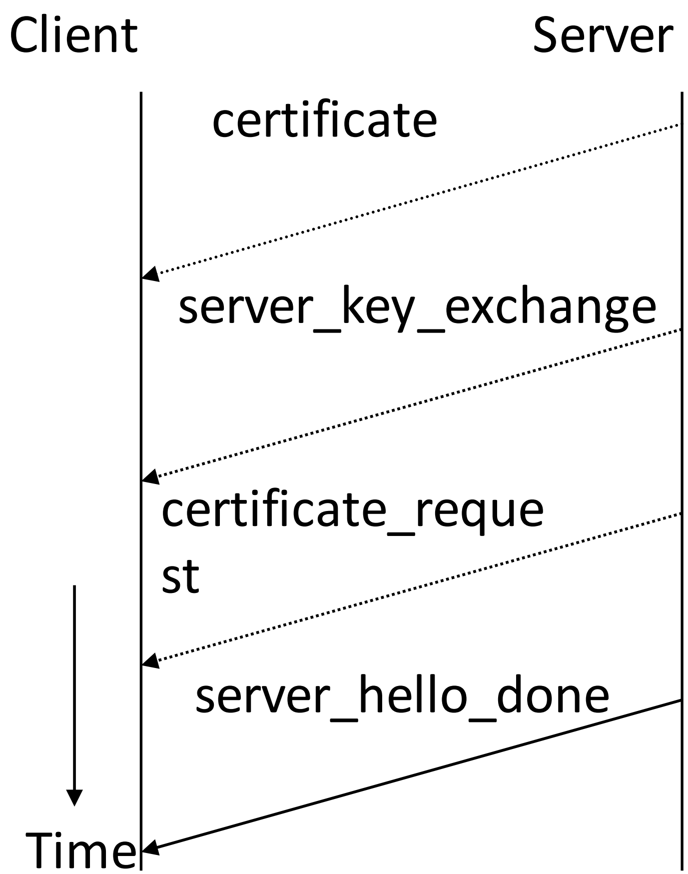
Handshake Protocol, Phase 1- Establish Security Capabilities



Establishes security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers

SSL - Secure Socket Layer Protocol (Cont.)

Handshake Protocol, Phase 2-Server Authentication and Key Exchange

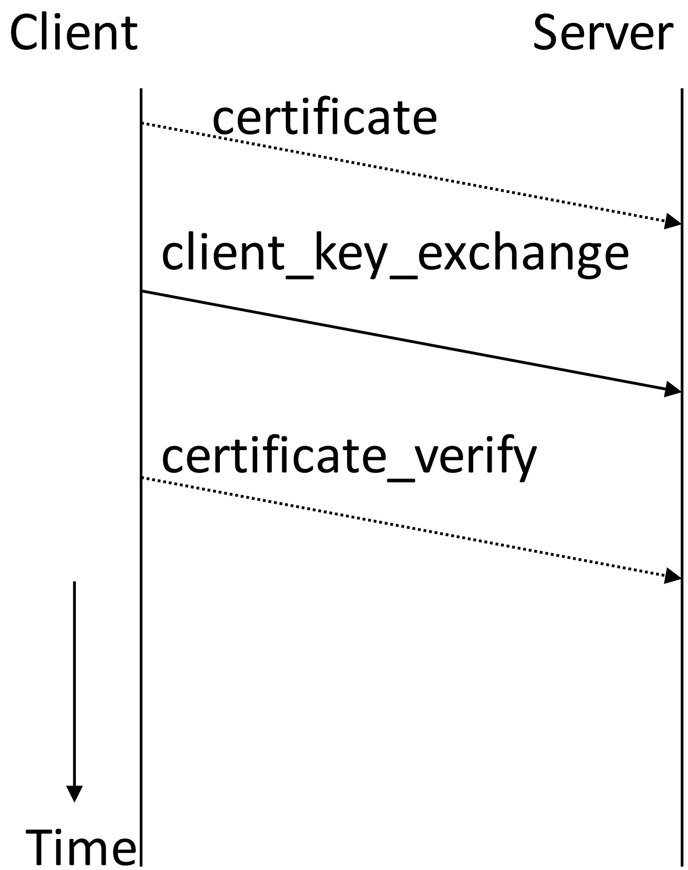


Server may send certificate, key exchange and request certificate. Server signals end of hello message phase.

Dotted transfers are optional or situation-dependent messages that are not always sent

SSL - Secure Socket Layer Protocol (Cont.)

Handshake Protocol, Phase 3-Client Authentication and Key Exchange

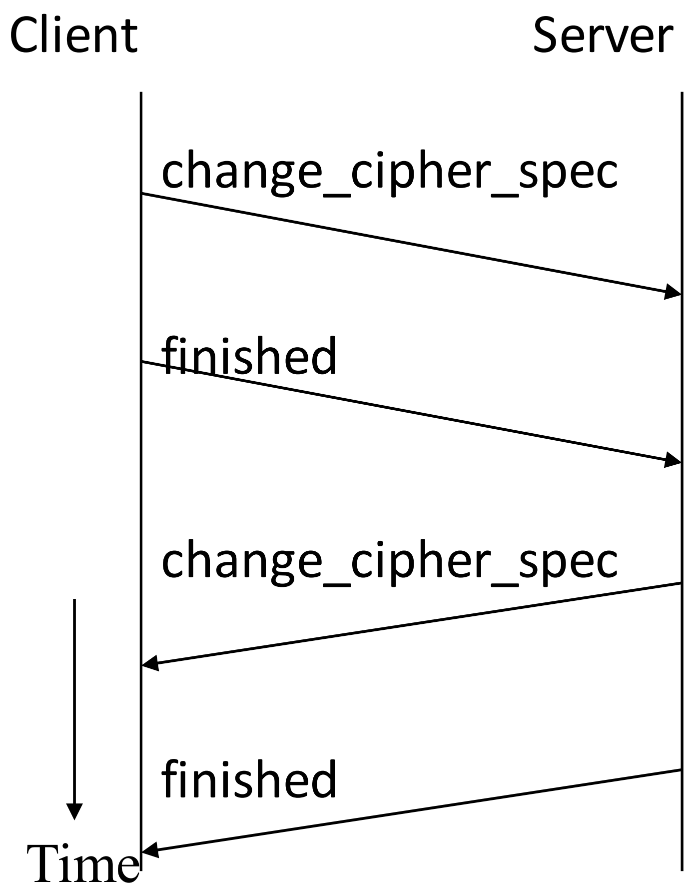


Client sends certificate, if requested. Client sends key exchange. Client may send certificate verification.

Dotted transfers are optional or situation-dependent messages that are not always sent

SSL - Secure Socket Layer Protocol (Cont.)

Handshake Protocol, Phase 4-Finish



Changes cipher suite and finishes handshake protocol.

SSL - Secure Socket Layer Protocol (Cont.)

Supported key exchange methods

- RSA - secret key is encrypted with the receiver’s RSA public key (receiver’s certificate available)
- Fixed Diffie-Hellman - server’s certificate contains the D-H public parameters
- Ephemeral Diffie-Hellman - the D-H public keys are exchanged, signed using the sender’s private RSA or DSS key
- Anonymous Diffie-Hellman - base D-H algorithm is used with no authentication
- Fortezza

SSL - Secure Socket Layer Protocol (Cont.)

Supported cipher algorithms:

- RC4, RC2, DES, 3DES, DES40, IDEA, Fortezza
- Supported MAC algorithms:
- MD5, SHA-1

Transport Layer Security (TLS)

- IETF standardisation initiative for producing an Internet standard version of SSL. (Current version of TLS is very similar to SSLv3.)

TLS

The Transport Layer Security (TLS) protocol was released in January 1999 to create a standard for private communications.

implementation of the TLS protocol on two levels: the TLS record protocol and TLS handshake protocol

There are seven main differences between SSL and TLS.

- Protocol version number
- Alert protocol message types
- message authentication
- Key material generation
- Certificate verify
- Finished
- Baseline cipher suites

Cryptography in Network

Cryptographic modules can be applied at different OSI layers:

- Application
- Presentation
- Network
- Transport

Granularity of protection is better at application and presentation layer

Individual user has complete control over the encryption algorithms and keys at application layer

Encryption Mode

Link encryption

- ✓ Encrypt all data along a communication path
- ✓ Communication node need to decrypt all data
- ✓ Easily incorporate into network protocols
- ✗ Encrypted and decryption many times across nodes
- ✗ Compromised single node disclosure
- ✗ Loses control over algorithm used along the path

End-to-end encryption

- ✓ Encrypted and decrypted only at endpoints
- ✗ Routing information remain visible

Network Security

Another way to classify web security threats is in terms of the location of the threat

- Web server
- Web browser
- Network traffic between browser and server.

Issues of server and browser security fall into the category of computer system security

Issues of network traffic between server and browser fall into the category network security

TCP/IP Model of Network Architecture

Application layer - offers services to users, provides network management (TELNET, HTTP, SMTP, .. protocols)

Transport layer - provides data flow between two end nodes of network. TCP protocol (reliable, with connection) a UDP (connectionless, unreliable)

Network layer - provides packet transmission over network. IP protocol (connectionless, unreliable), ICMP protocol.

Network interface layer - provides the same services as link and physical layer in RM OSI.

IPSec

Internet Threats in IP network

Security problems in IP v4
- Packet Sniffing
- Loss of Data integrity
- Identity spoofing
- Replay old packets

IPSec functional objectives
- Data Confidentiality
- Data integrity
  - Connectionless integrity
- Origin Identification
  - Data origin authentication [no more spoof attacks!]
  - Access Control
- Replay Attack Prevention
  - Rejection of replayed packets [no more session hijacking!]

IP Security (IPSec)

Developed by IETF, IPSec Working Group

Transparent to applications & users

Transport mode & Tunnel mode

Security Association

- Represent an agreement between 2 peers on a set of security services to be applied to the IP traffic stream between these nodes

IPSec Protocol

IP Authentication Header (AH)

- Connectionless integrity
  - Mutable field
- Data origin authentication
- Protection against replay

IP Encapsulating Security Payload (ESP)

- Confidentiality
  - Secret-key Cryptography

Transport and Tunnel Mode

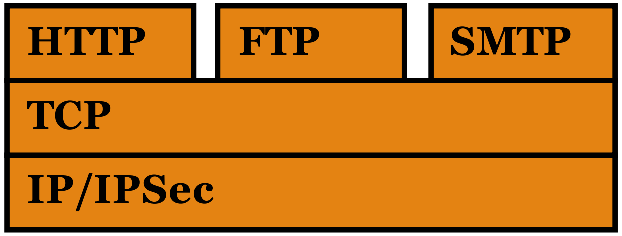
Key management

- Oakley Key Determination Protocol
- ISAKMP

IPSec - IP Secure Protocol

Ensures authentication, confidentiality and integrity of IP packets between two nodes (extensions of IP protocol)

- AH and EPS headers
- Key and algorithm management according IKE (Internet Key Exchange)
  - combination of ISAKMP and Oakley
- Creates a tunnel at the network layer



IPsec Security Architecture

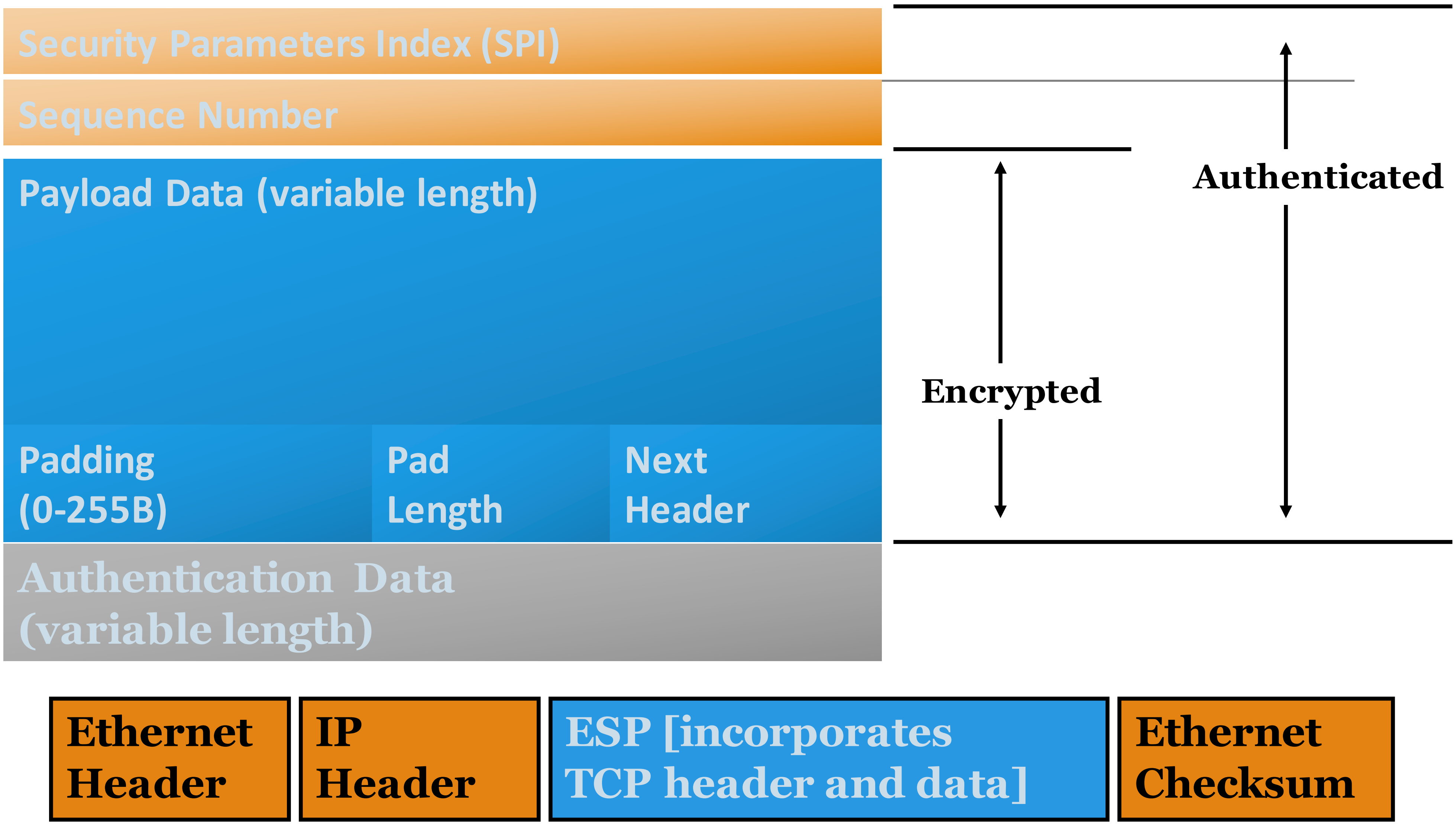
Cryptographic protocols:

- Securing Packet flows
  - Authentication Header (AH)
    - Digitally signing the packet
  - Encapsulating Security Payload (ESP)
    - Closed envelope for encryption and integrity
- Internet Key Exchanges
  - Based on IKE

IPSec - IP Secure Protocol
IPSec Services

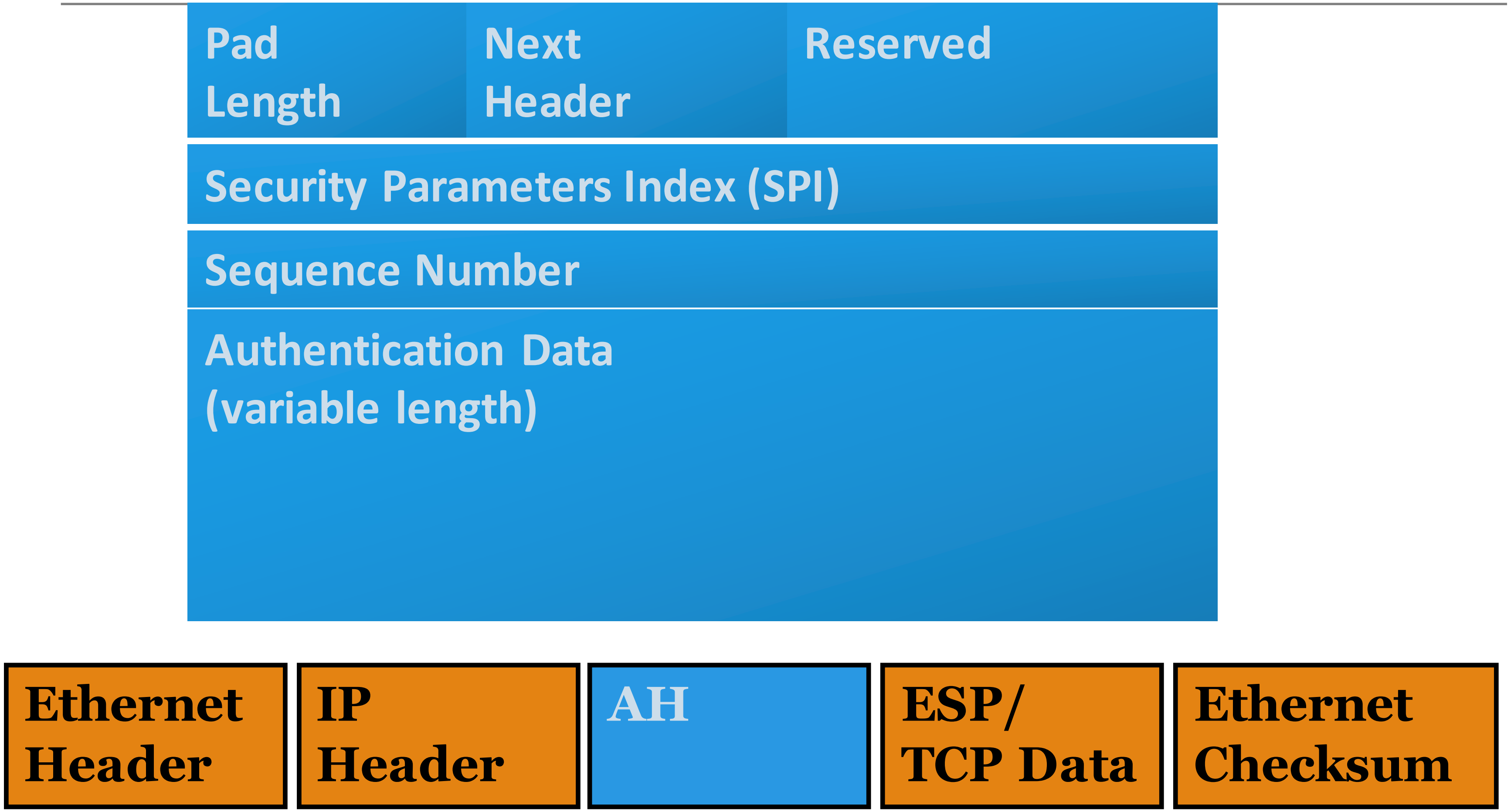
	AH	ESP (Encryption only)	ESP (Encryption and Authentication)
Access control	YES	YES	YES
Connectionless integrity	YES		YES
Data origin authentication	YES		YES
Rejection of replayed packets	YES	YES	YES
Confidentiality		YES	YES
Limited traffic flow confidentiality		YES	YES

IP Sec - ESP



Security Parameters Index (SPI)
Sequence Number
Payload Data (variable length)
Padding (0-255B)
Pad Length
Next Header
Authentication Data (variable length)
Authenticated
Encrypted
Ethernet Header
IP Header
ESP [incorporates TCP header and data]
Ethernet Checksum

IP Sec - AH



Pad Length
Next Header
Reserved
Security Parameters Index (SPI)
Sequence Number
Authentication Data (variable length)
Ethernet Header
IP Header
AH
ESP/ TCP Data
Ethernet Checksum

Internet Security Association and Key Management Protocol (ISAKMP)

Defines procedures and packet formats to establish, negotiate, modify and delete security associations in IPSec

Different payloads

- Security associate   Proposal   Transform
- Key exchange   Identification   Certificate
- Certificate request   Hash   Signature
- Nonce   Notification   Delete

Exchanges

- Base   Identity protection
- Authentication only   Aggressive
- Information

IPSec: Applications

Secure connection to Extranet

Secure connection to Intranet at remote site

Secure remote access over the Internet

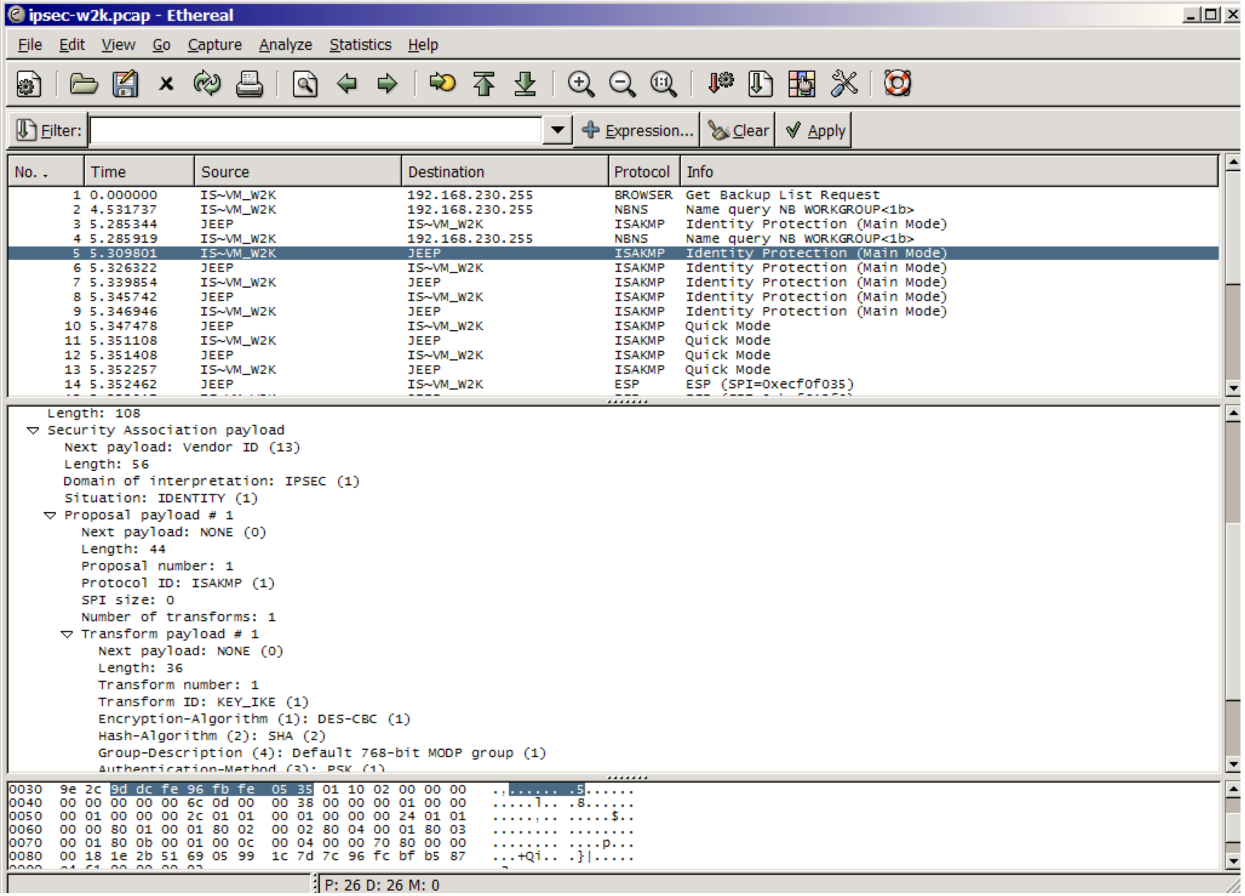
Enhancing network security

- EFT
- Key distribution/exchange

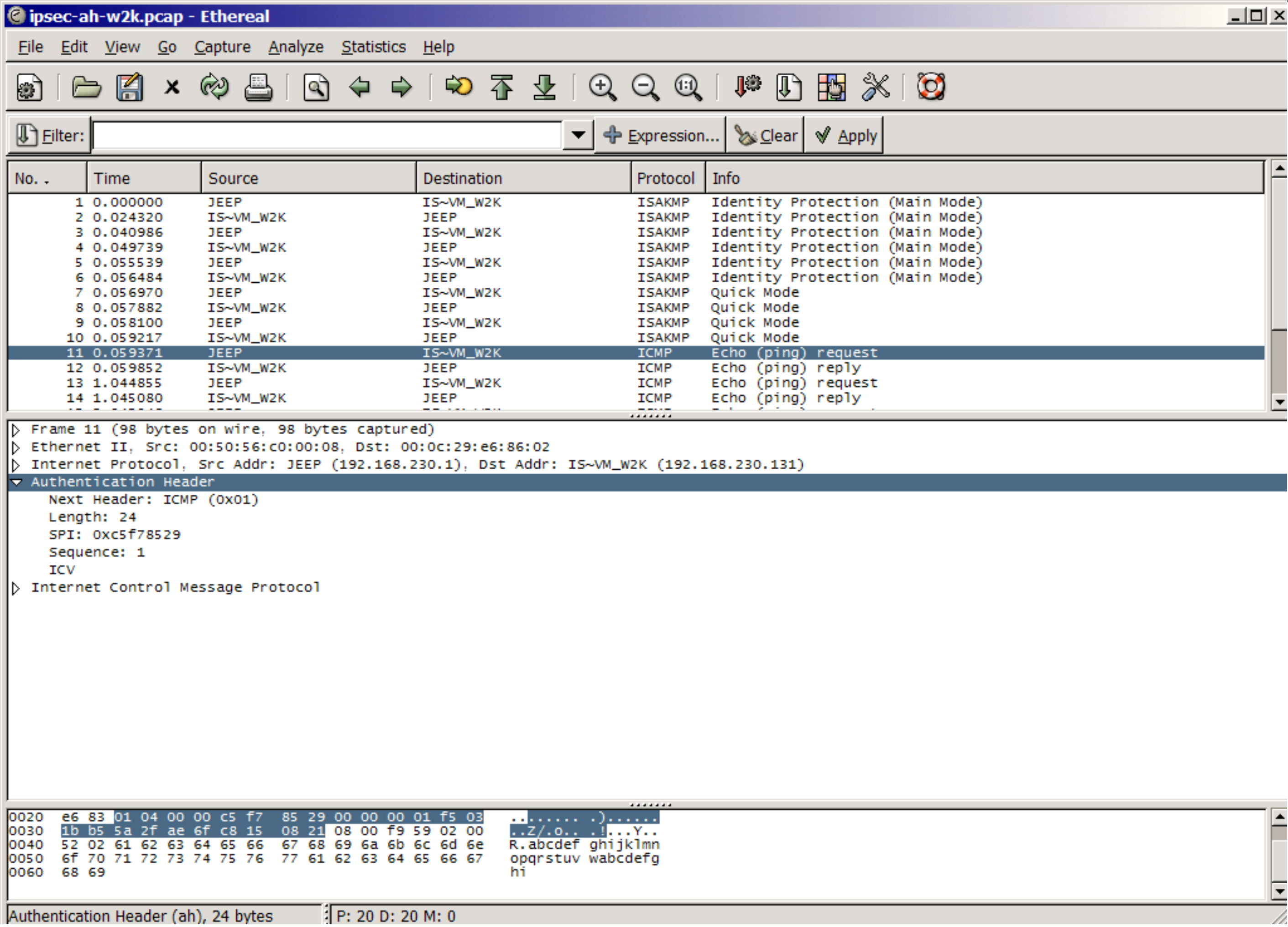
Enhance other network applications

- Provide basic security requirements

IPSec Demonstration



IPSec Demonstration



IPSec Demonstration

```
C:\WINDOWS\System32\cmd.exe - ping -t 192.168.230.131
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.230.131:
    Packets: Sent = 8, Received = 8, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
D:\Documents and Settings\pong>
D:\Documents and Settings\pong>
D:\Documents and Settings\pong>ping -t 192.168.230.131

Pinging 192.168.230.131 with 32 bytes of data:

Negotiating IP Security.
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
Reply from 192.168.230.131: bytes=32 time<1ms TTL=128
```

IPSec Demonstration

```
C:\WINDOWS\System32\cmd.exe
D:\Documents and Settings\pong>windump -i 6 -n
windump: listening on\Device\Packet_{3A7DEC6A-15AF-48B4-9BFD-697D491E6920}
00:18:05.169810 192.168.230.1 > 192.168.230.131: ESP(spi=0xecf0f035,seq=0xf)
00:18:05.170197 192.168.230.131 > 192.168.230.1: ESP(spi=0xbcf612f9,seq=0xf)
00:18:06.170727 192.168.230.1 > 192.168.230.131: ESP(spi=0xecf0f035,seq=0x10)
00:18:06.170965 192.168.230.131 > 192.168.230.1: ESP(spi=0xbcf612f9,seq=0x10)
00:18:07.170916 192.168.230.1 > 192.168.230.131: ESP(spi=0xecf0f035,seq=0x11)
00:18:07.171142 192.168.230.131 > 192.168.230.1: ESP(spi=0xbcf612f9,seq=0x11)
00:18:08.171109 192.168.230.1 > 192.168.230.131: ESP(spi=0xecf0f035,seq=0x12)
00:18:08.171365 192.168.230.131 > 192.168.230.1: ESP(spi=0xbcf612f9,seq=0x12)
00:18:09.172300 192.168.230.1 > 192.168.230.131: ESP(spi=0xecf0f035,seq=0x13)
00:18:09.172555 192.168.230.131 > 192.168.230.1: ESP(spi=0xbcf612f9,seq=0x13)
00:18:10.173501 192.168.230.1 > 192.168.230.131: ESP(spi=0xecf0f035,seq=0x14)
00:18:10.173732 192.168.230.131 > 192.168.230.1: ESP(spi=0xbcf612f9,seq=0x14)
00:18:11.147407 192.168.230.1.500 > 192.168.230.131.500: isakmp: phase 2/others
? inf[E]: [|hash]
00:18:11.172535 192.168.230.131.500 > 192.168.230.1.500: isakmp: phase 2/others
? inf[E]: [|hash]
00:18:11.173085 192.168.230.1.500 > 192.168.230.131.500: isakmp: phase 2/others
? inf[E]: [|hash]
00:18:11.189037 192.168.230.1 > 192.168.230.131: icmp: echo request
00:18:16.425406 192.168.230.1 > 192.168.230.131: icmp: echo request
00:18:21.431364 192.168.230.1 > 192.168.230.131: icmp: echo request
00:18:21.431593 192.168.230.131 > 192.168.230.1: icmp: echo reply
```

IPSec Demonstration

C:\WINDOWS\System32\cmd.exe

```
00:24:38.380359 192.168.230.1 > 192.168.230.131: icmp: echo request
00:24:43.388329 192.168.230.1 > 192.168.230.131: icmp: echo request
00:24:48.395240 192.168.230.1.500 > 192.168.230.131.500: isakmp: phase 1 I ident
: [|sa]
00:24:48.419801 192.168.230.131.500 > 192.168.230.1.500: isakmp: phase 1 R ident
: [|sa]
00:24:48.436301 192.168.230.1.500 > 192.168.230.131.500: isakmp: phase 1 I ident
: [|ke]
00:24:48.446589 192.168.230.131.500 > 192.168.230.1.500: isakmp: phase 1 R ident
: [|ke]
00:24:48.452392 192.168.230.1.500 > 192.168.230.131.500: isakmp: phase 1 I ident
[E]: [|id]
00:24:48.454327 192.168.230.131.500 > 192.168.230.1.500: isakmp: phase 1 R ident
[E]: [|id]
00:24:48.454879 192.168.230.1.500 > 192.168.230.131.500: isakmp: phase 2/others
I oakley-quick[E]: [|hash]
00:24:48.455810 192.168.230.131.500 > 192.168.230.1.500: isakmp: phase 2/others
R oakley-quick[EC]: [|hash]
00:24:48.456028 192.168.230.1.500 > 192.168.230.131.500: isakmp: phase 2/others
I oakley-quick[EC]: [|hash]
00:24:48.456912 192.168.230.131.500 > 192.168.230.1.500: isakmp: phase 2/others
R oakley-quick[EC]: [|hash]
00:24:48.457053 192.168.230.1 > 192.168.230.131: AH(spi=0xec10b130,seq=0x1): icm
p: echo request
00:24:48.459577 192.168.230.131 > 192.168.230.1: AH(spi=0xfa84acf1,seq=0x1): icm
p: echo reply
00:24:49.394524 192.168.230.1 > 192.168.230.131: AH(spi=0xec10b130,seq=0x2): icm
p: echo request
```

Security Associations

Security Parameters Index (SPI)
- Local significance

IP Destination Address
- Unicast address

Security Protocol Identifier
- Association with AH and ESP

Other parameters
- Sequence Number Counter
- Sequence Counter Overflow
- Antireplay windows
- AH information
- ESP information
- Lifetime of this security association
- IPSec Protocol Mode
- Path MTU

Virtual Private Network

What is Virtual Private Network

Secure private communications over public internet

Private IP packets encapsulated within public packets (tunnel)

Protecting the Network channel over untrusted network

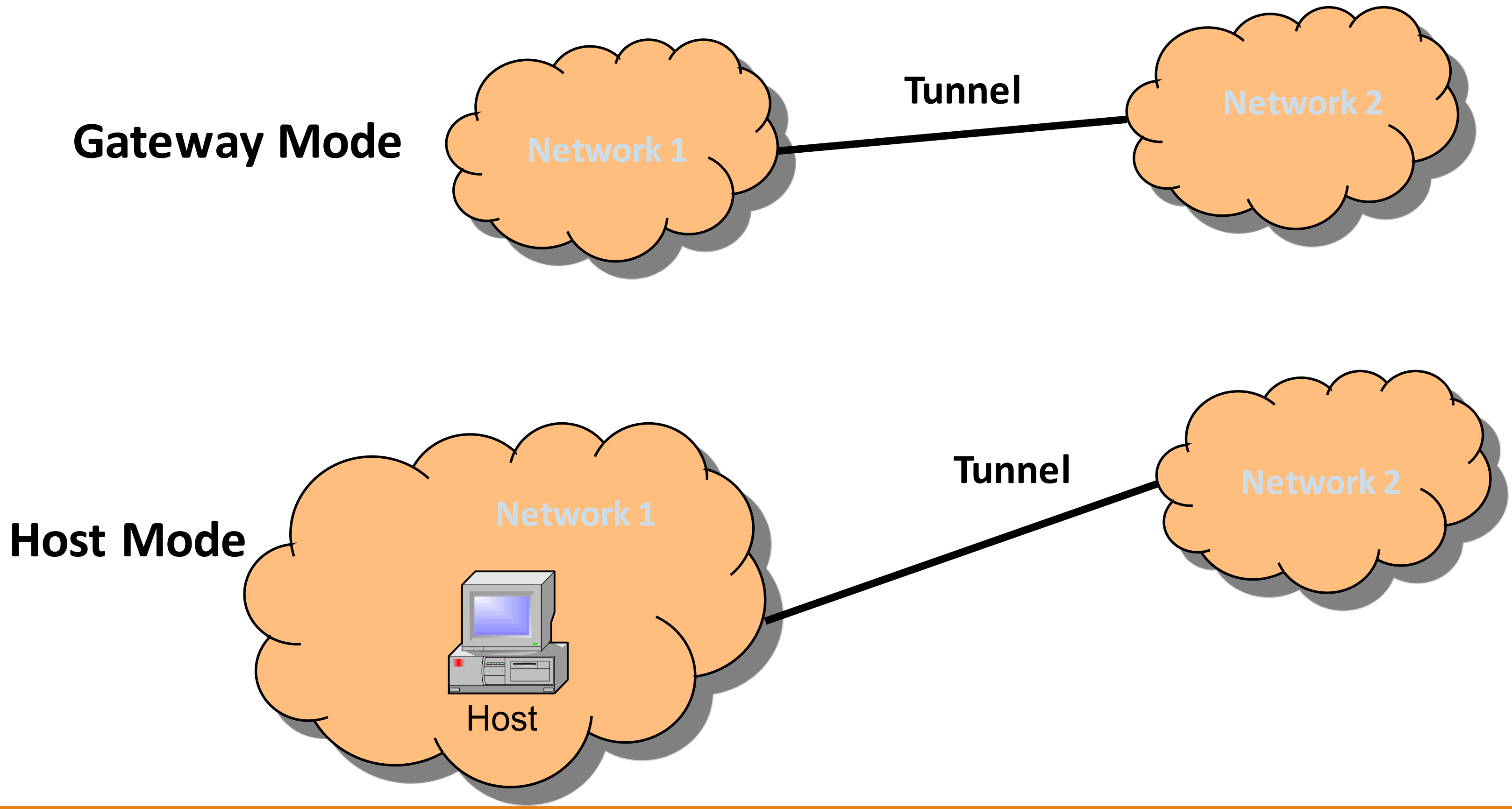
Protect through the use of encryption
- Either through IPSEC, or other encryption scheme

Compliance issue

Categorized in
- Remote Access
- Site-to-Site
- Extranet

Virtual Private Network (VPN)

Create a virtual network with a "tunnel" between two different networks: IPSec, PPTP, IPv6, …



VPN

Normally secured

- Encryption and Integrity check
  - Key Exchange
- Network control

Provide remote connectivity

- Employees
- Business Partners

Compared to SSL?

- Provide network-level connectivity: support all kinds of network application

SSL-VPN is a new trend

Variations

VPN connection types

- Client to Server, Server to Server

Types of VPN

- Hardware, software, firewall

Protocols

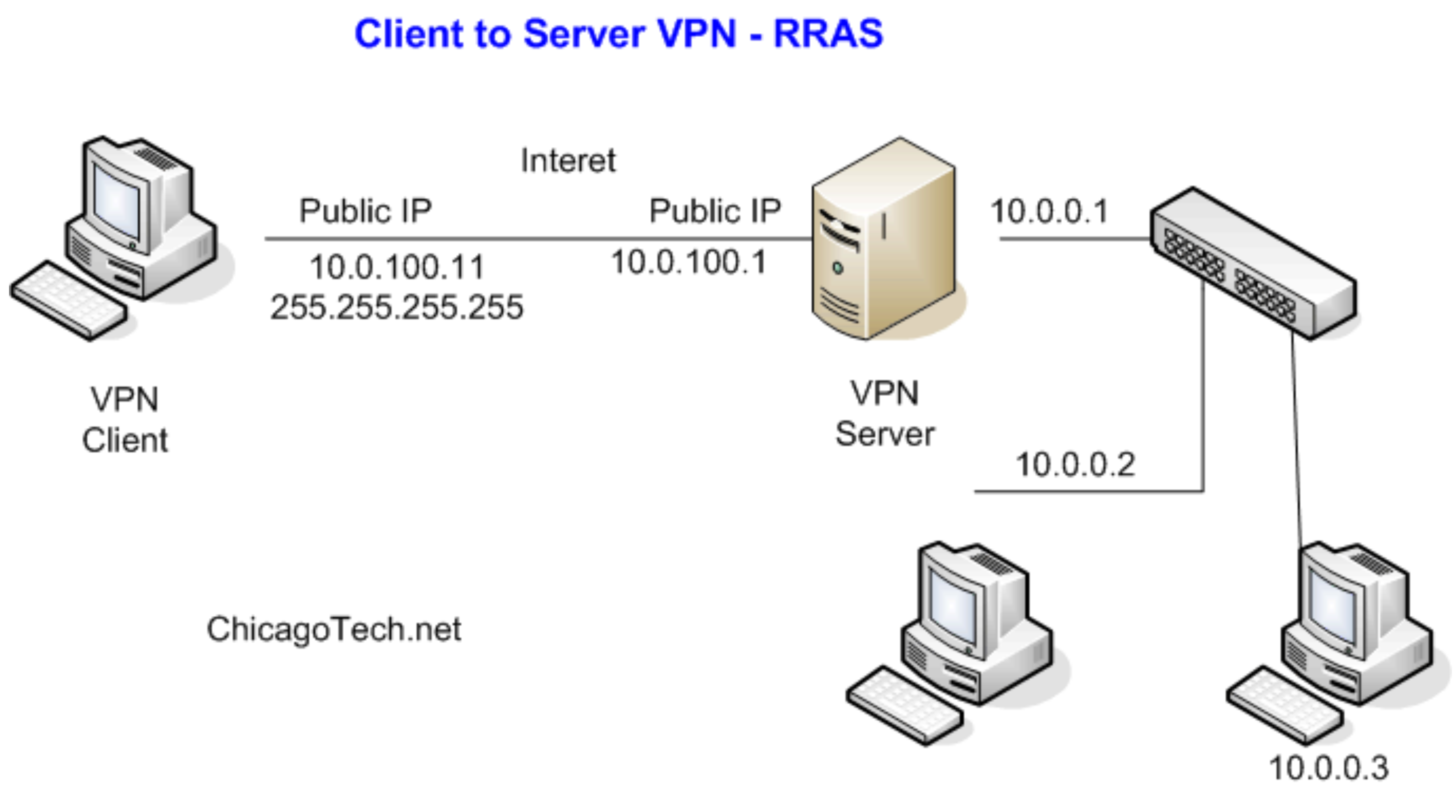
- PPTP, L2F, L2TP, IPSec

VPN connection types

Peer-to-peer

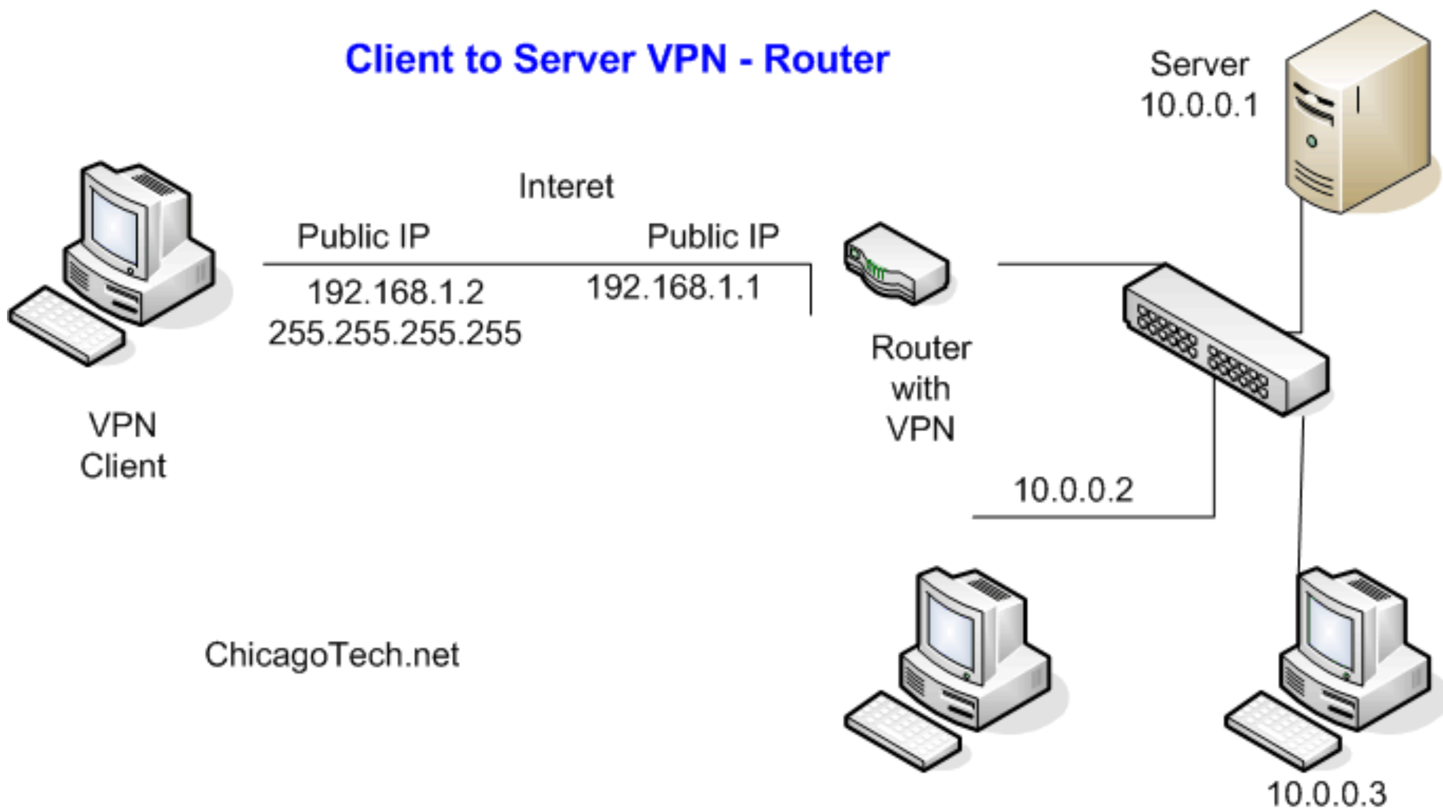
VPN connection types

Client to server VPN - RRAS



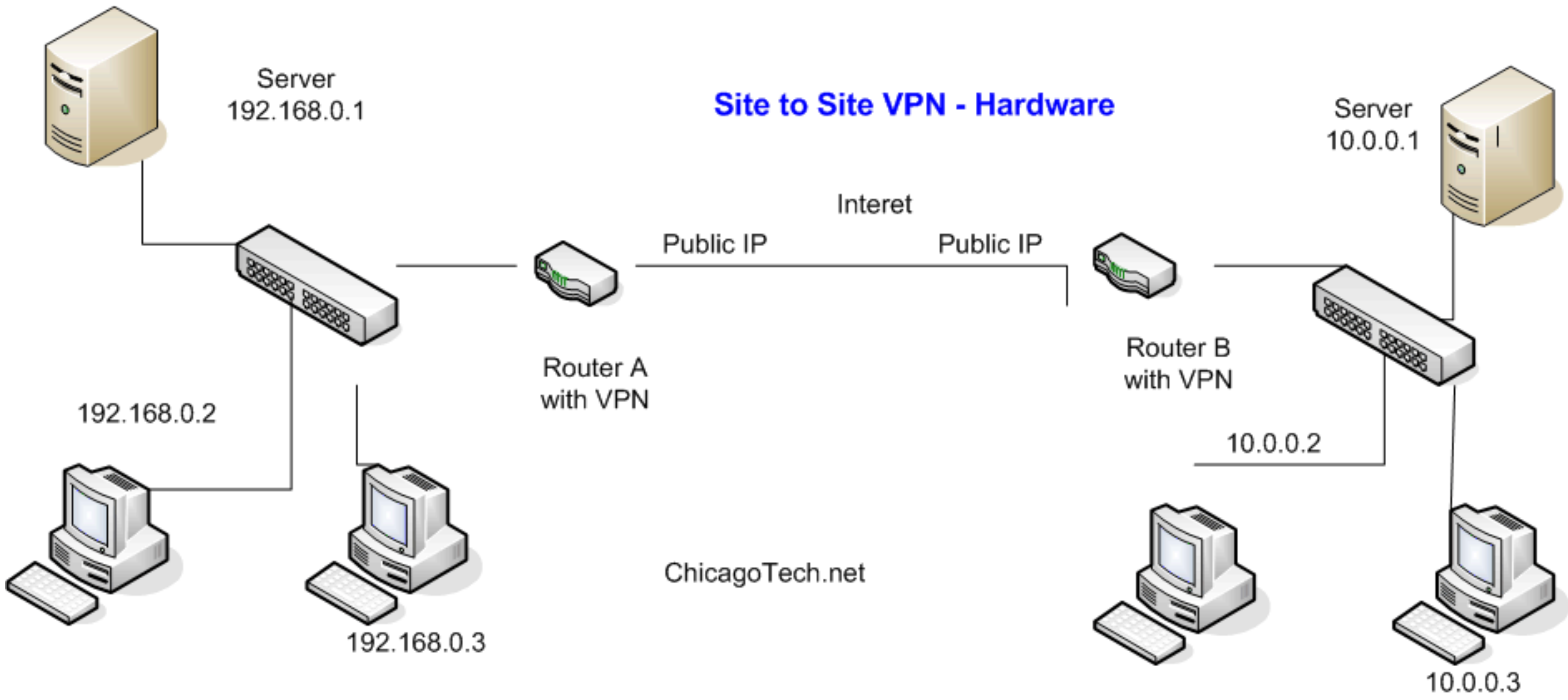
VPN connection types

Client to server VPN router



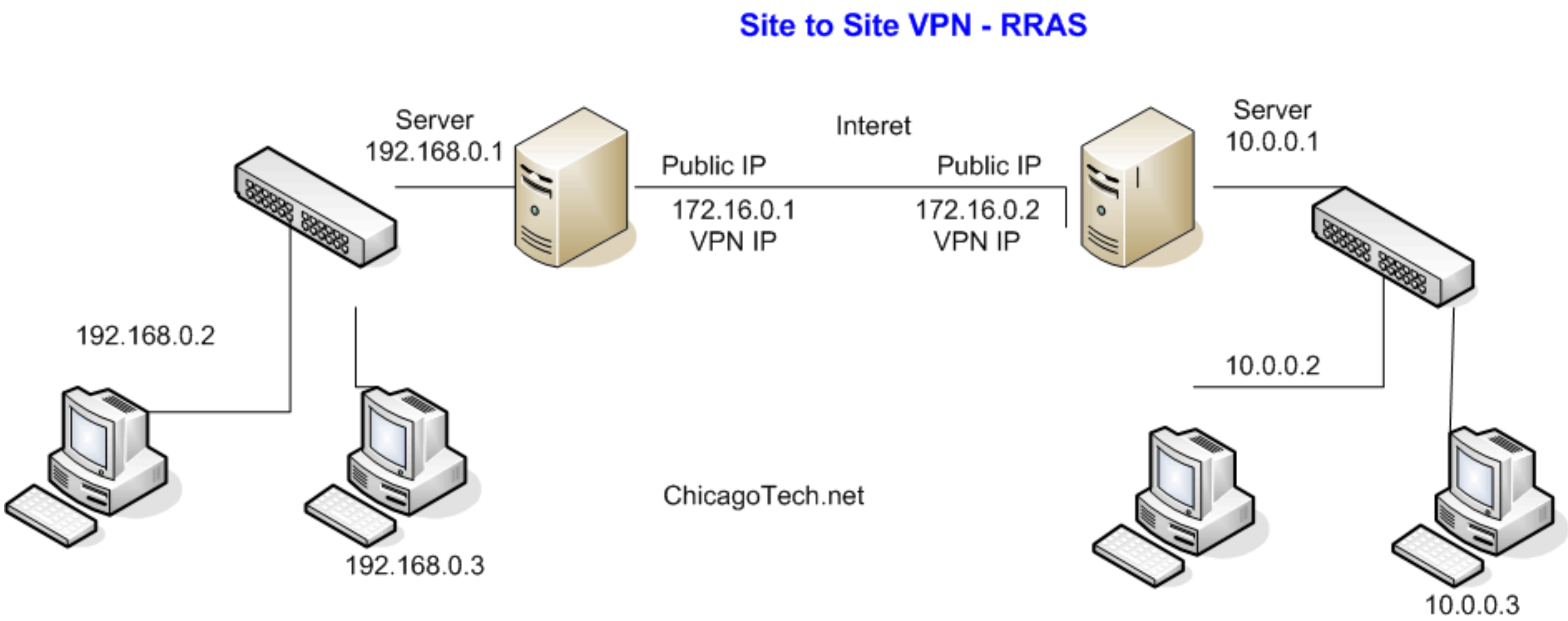
VPN connection types

Site to site VPN



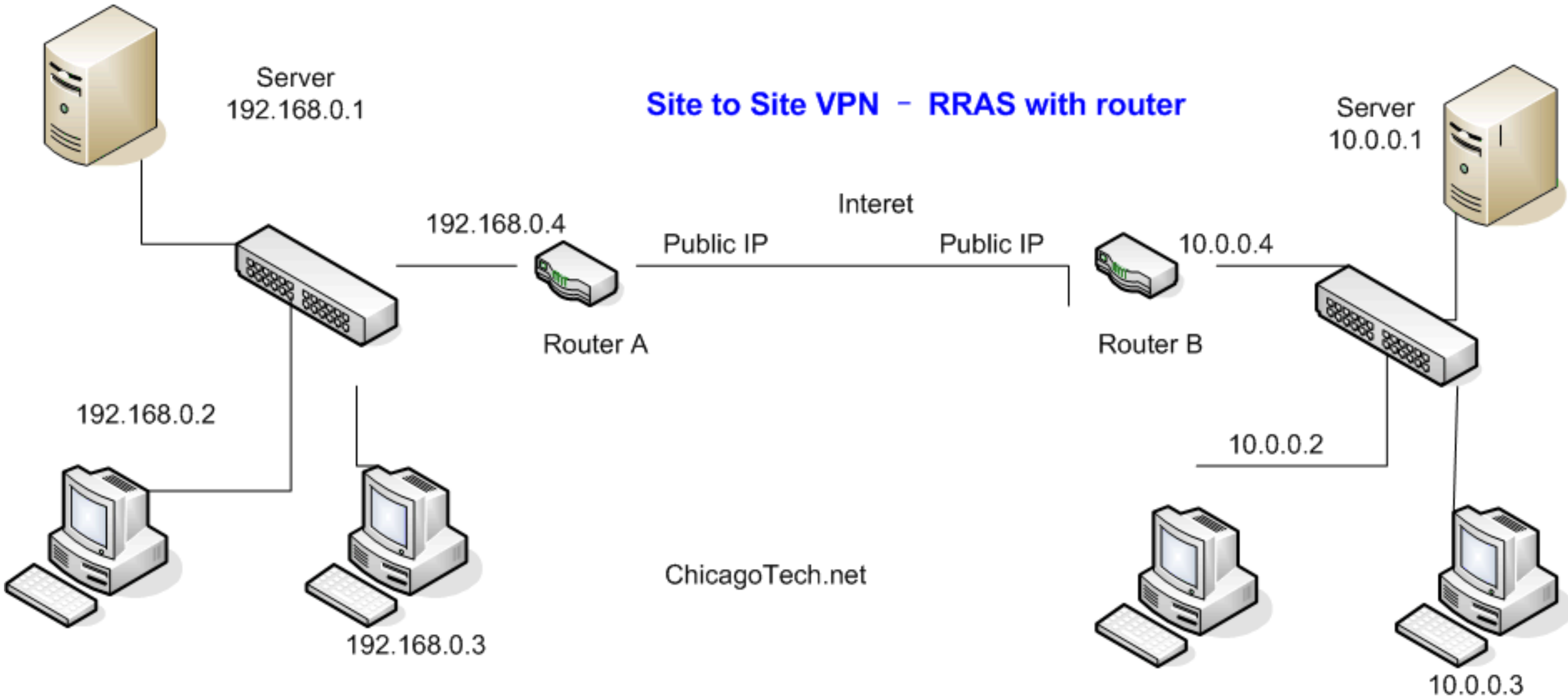
VPN connection types

Site to site VPN – RRAS



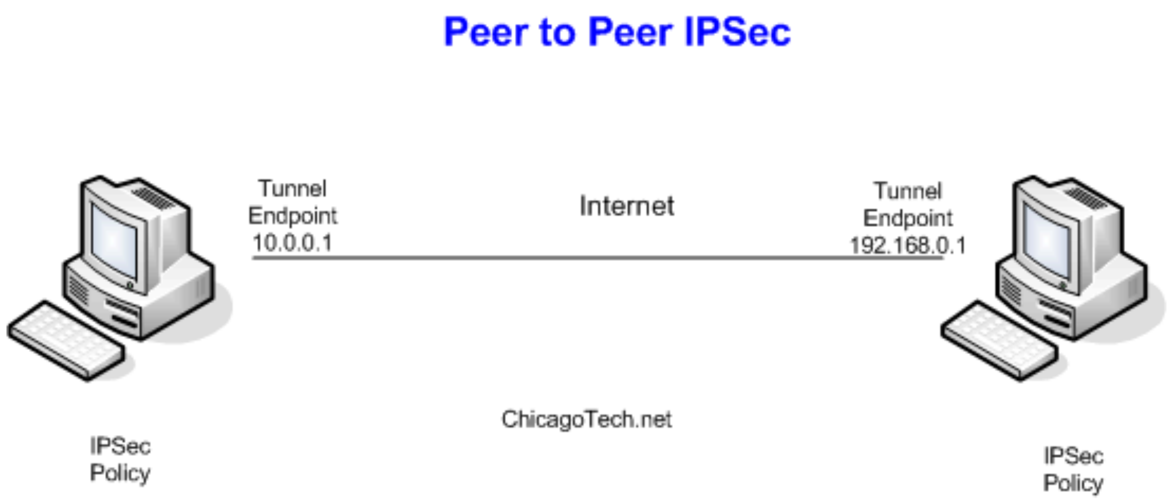
VPN connection types

Site to site VPN – RRAS with router



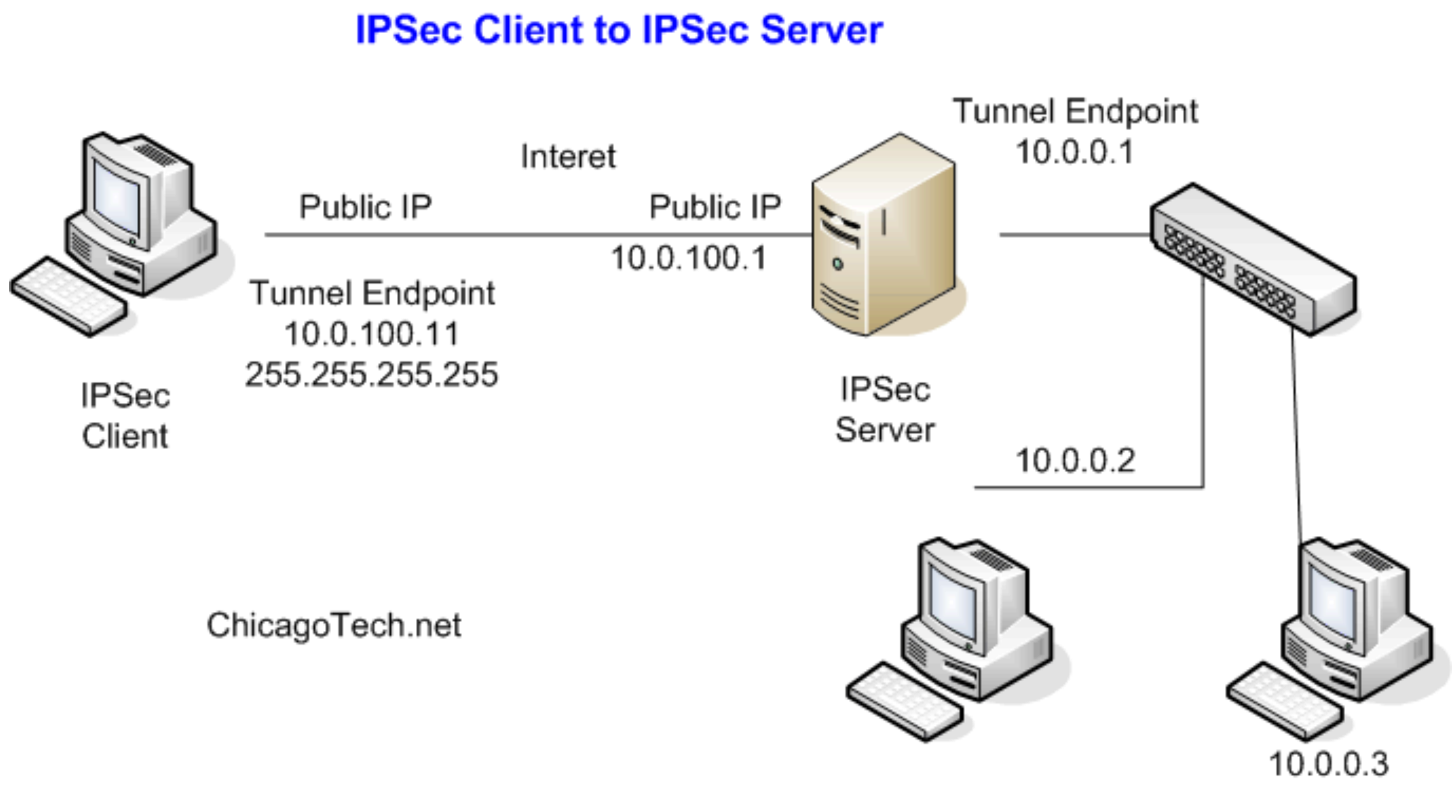
IPSEC connection types

Peer to Peer IPSEC



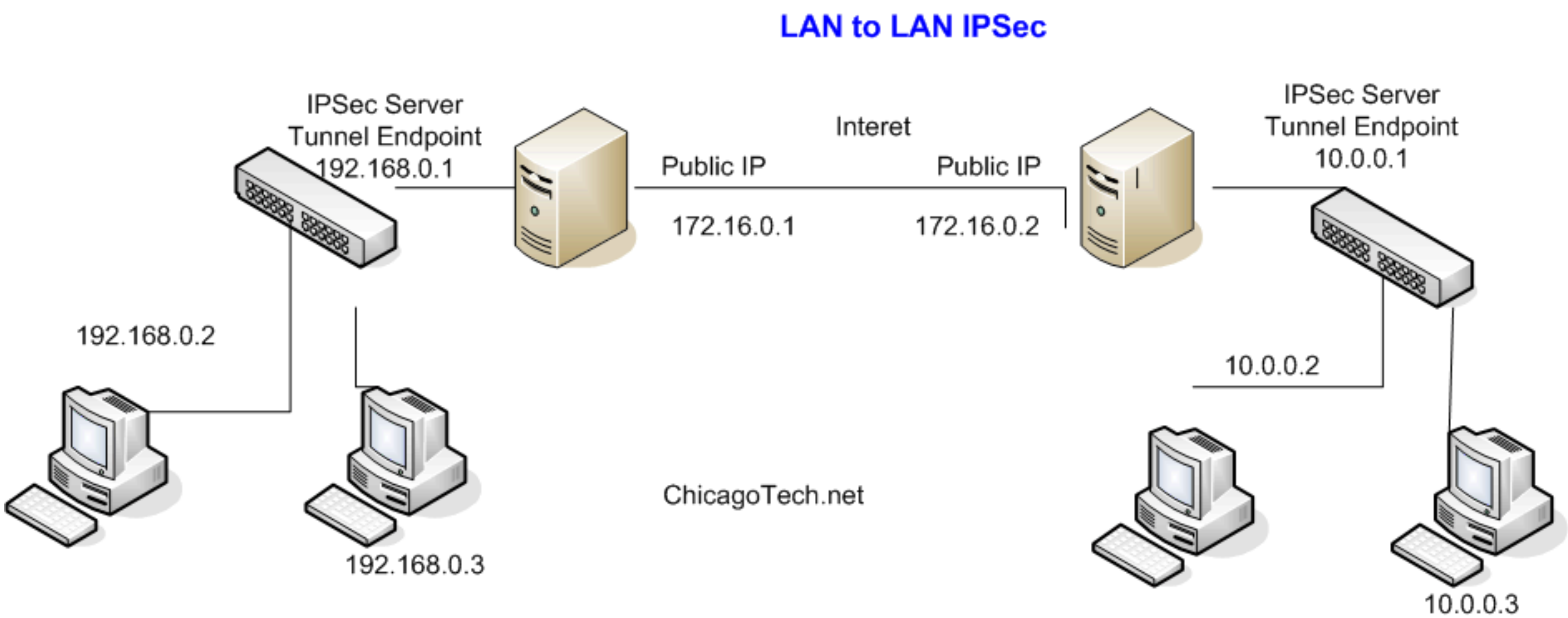
IPSEC connection types

IPSEC Client to IPSEC Server



IPSEC connection types

LAN to LAN IPSEC



VPN Protocols

Layer 5 (TCP/IP)

- SSL

Layer 3 (IP)

- IPSEC

Layer 2

- PPTP, L2TP

Summary of VPN Tunneling Protocols

	PPTP	L2F	L2TP	IPSec	SSL/TLS
Layer	2	2	2	3	Higher layers (apps/ transport)
Encryption	PPP based, MPPE	PPP based, MPPE	PPP, encryption, MPPE	DES, 3DES, DES-CBC, CAST 128, IDEA	DES, 3DES, RC2, RC4
Authentication	PPP based (PAP, CHAP, MS-CHAP)	PPP based (PAP, CHAP, MS-CHAP, EAP)	PPP based (PAP, CHAP, MS-CHAP, EAP)	Digital certs, public keys	Digital certs
Data integrity	None	None	None	HMAC-MD5, SHA-1	MD5, SHA-1, HMAC

Summary of VPN Tunneling Protocols

	PPTP	L2F	L2TP	IPSec	SSL/TLS
Multi-protocol support	No	Yes	Yes	No (IP only)	Yes
Main VPN type supported	User-site	User-site	User-site	User-site, site-site	User-site
RFC reference	RFC 2637	RFC 2341 informatio nal)	RFC 2661	RFC 2401-2409	RFC 2246

PPTP (Microsoft VPN)

Microsoft based Point-to-Point Tunnel Protocol.

Layer 2 protocol

Uses enhanced version of CHAP (MS-CHAP v2)

Support

- 40-bits encryption (for Win95, Win98 clients)
- 128-bits encryption (for recent Windows version)

PPTP Limitations

PPTP is mainly used for Windows and MAC clients only

Performance depends on client system.

- With Win98, 80% - 85% of the underlying connection speed only.

L2TP

Layer 2 Tunnel Protocol

- Combined from L2F (Cisco) and PPTP (Microsoft)

L2TP offers the following benefits:

- Vendor interoperability.
- Can be used as part of the wholesale access solution
- Can be operated as a client initiated VPN solution
- Supports Multihop, which enables Multichassis Multilink PPP in multiple home gateways.

L2TP is, in fact, a layer 5 protocol session layer, and uses the registered UDP port 1701.

The entire L2TP packet, including payload and L2TP header, is sent within a UDP datagram

L2TP/IPsec

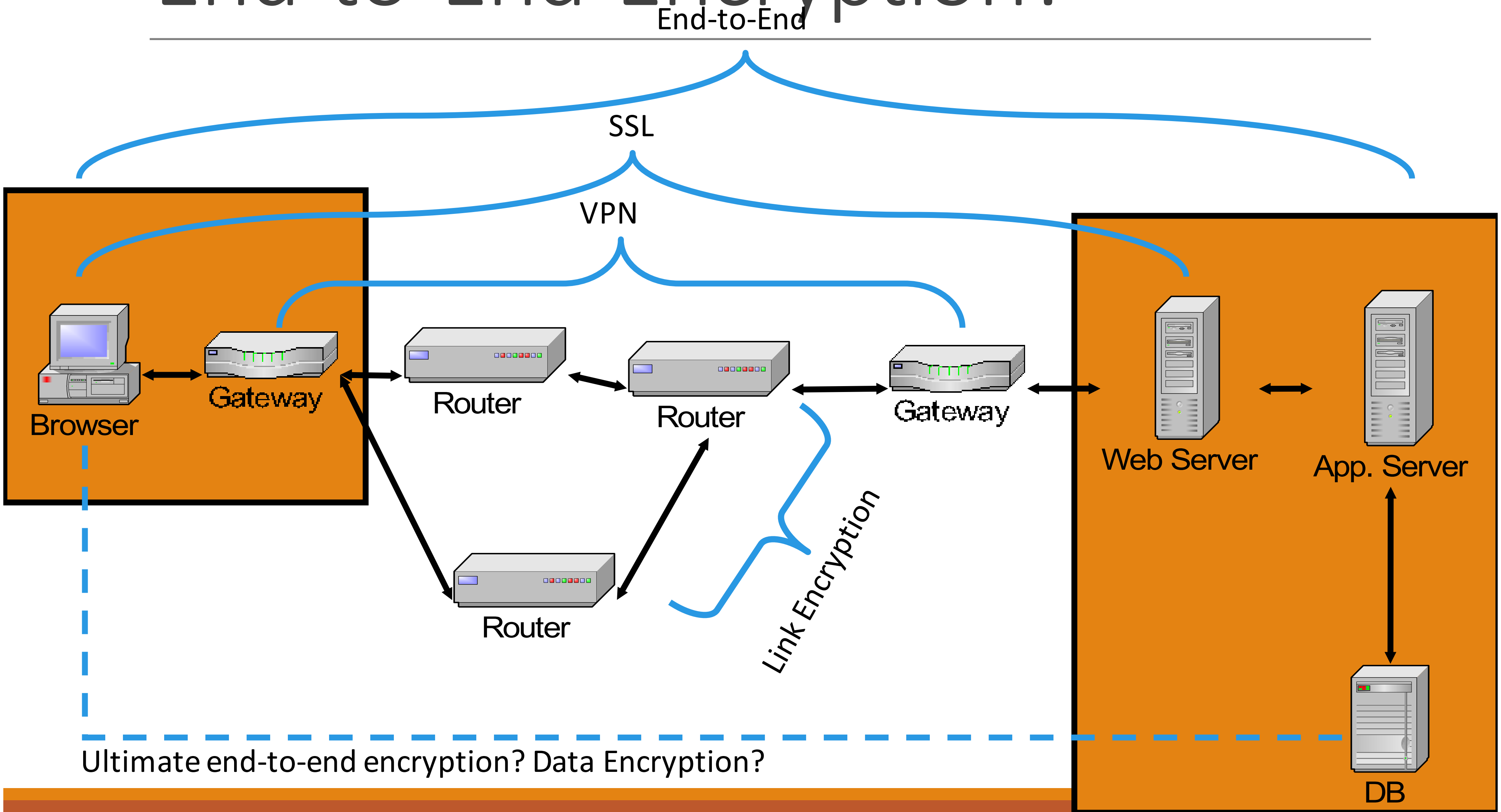
Negotiation through IKE

When the process is complete, L2TP packets between the endpoints are encapsulated by IPsec.

Since the L2TP packet itself is wrapped and hidden within the IPsec packet, no information about the internal private network can be garnered from the encrypted packet.

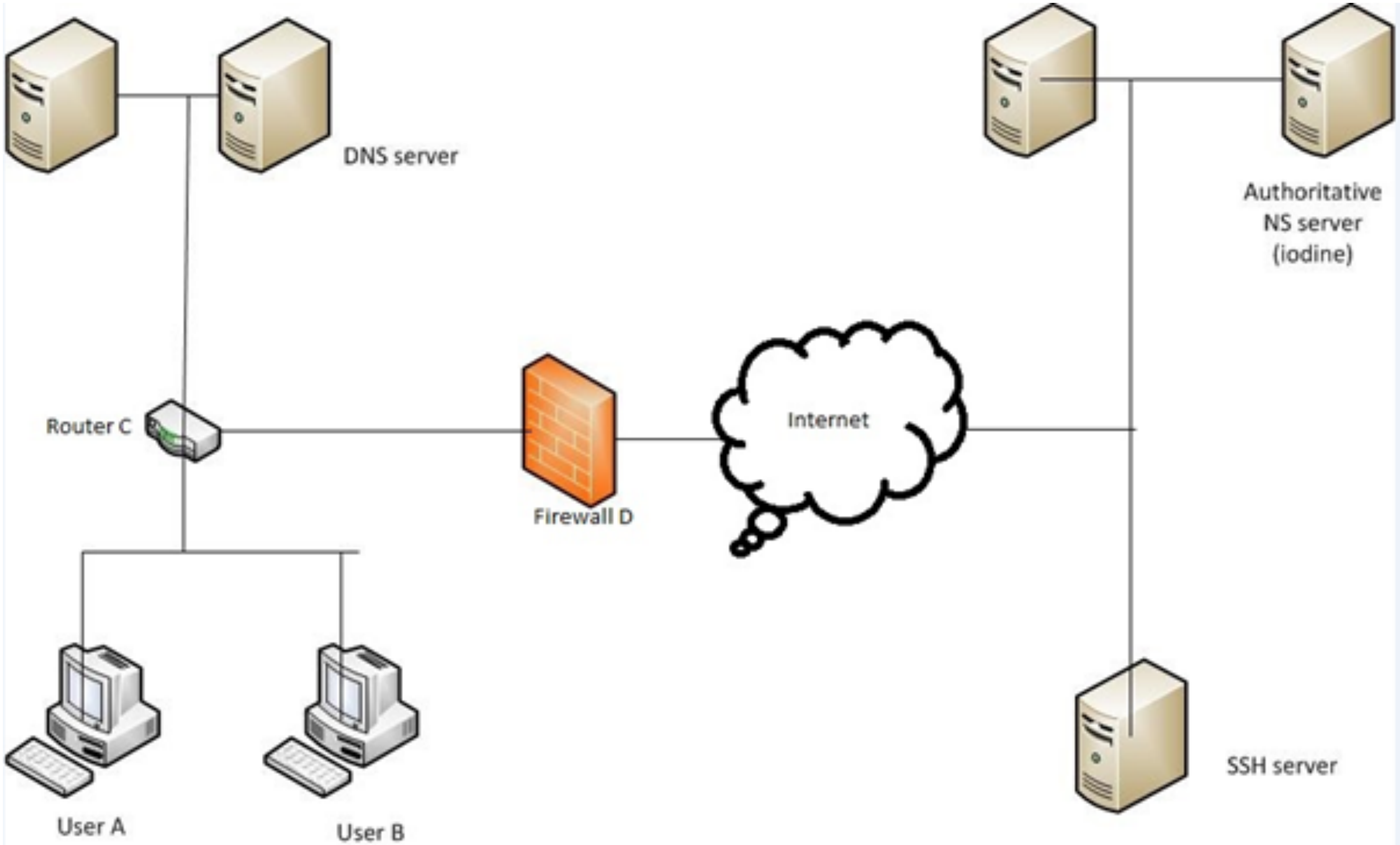
Older Windows version do not support (Limitation)

End-to-End Encryption?



Ultimate end-to-end encryption? Data Encryption?

Covert Channel



Example: DNS Tunneling: http://resources.infosecinstitute.com/dns-tunnelling/

PKI, Key Management

Pretty Good Privacy (PGP)

Offered as a freeware by Philip R. Zimmermann at 1991

De facto standard program for secure e-mail and file encryption on the Internet

Now available in both freeware and commercial versions

Working Principles

- PGP enables you to make your own public and secret key pairs
- PGP public keys are distributed and certified via an informal network called "the web of trust," which is kind of like the letters of introduction popular in the pre-electronic era

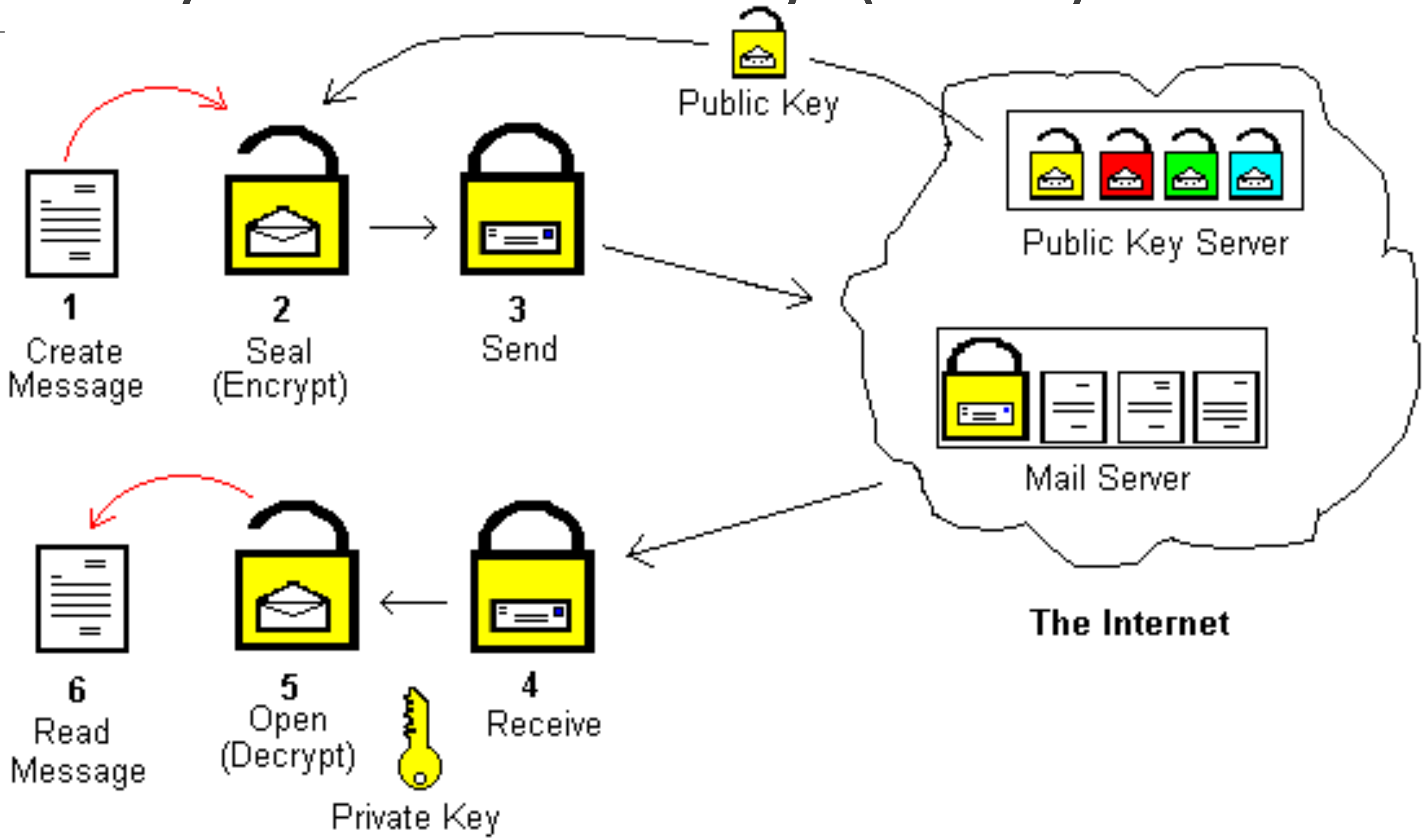
Pretty Good Privacy (PGP)

PGP uses a concept known as a ``web of trust'', in which any party can certify the identity of any other party (in PGP parlance, ``sign their key'')

A signature on a key or document can be trusted if, and only if, there is a path of signatures between the verifier's key and the key used to make the signature in question

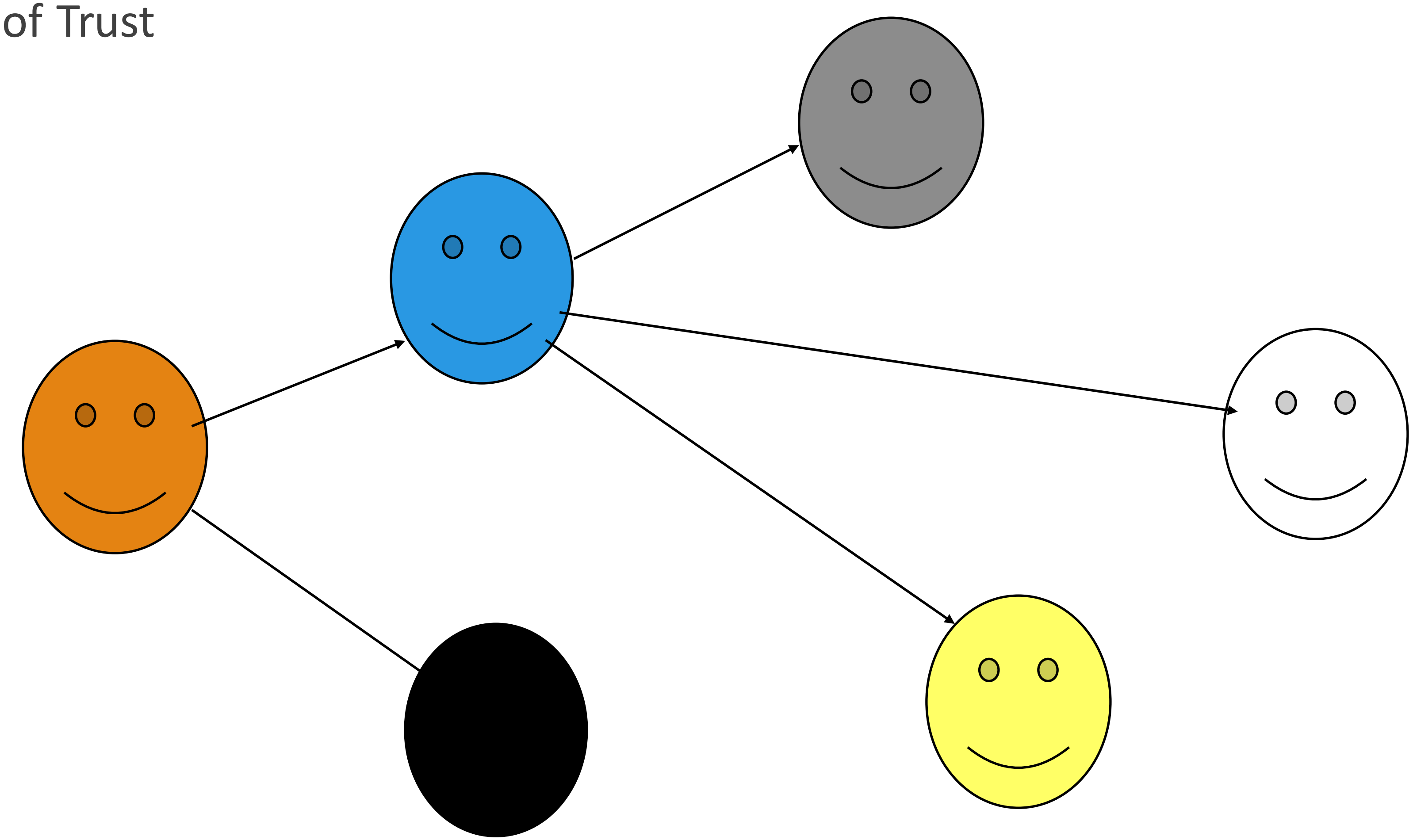
The number, shortness, and quality of such paths determine how well the key, and therefore the signature, can be trusted

Pretty Good Privacy (PGP)

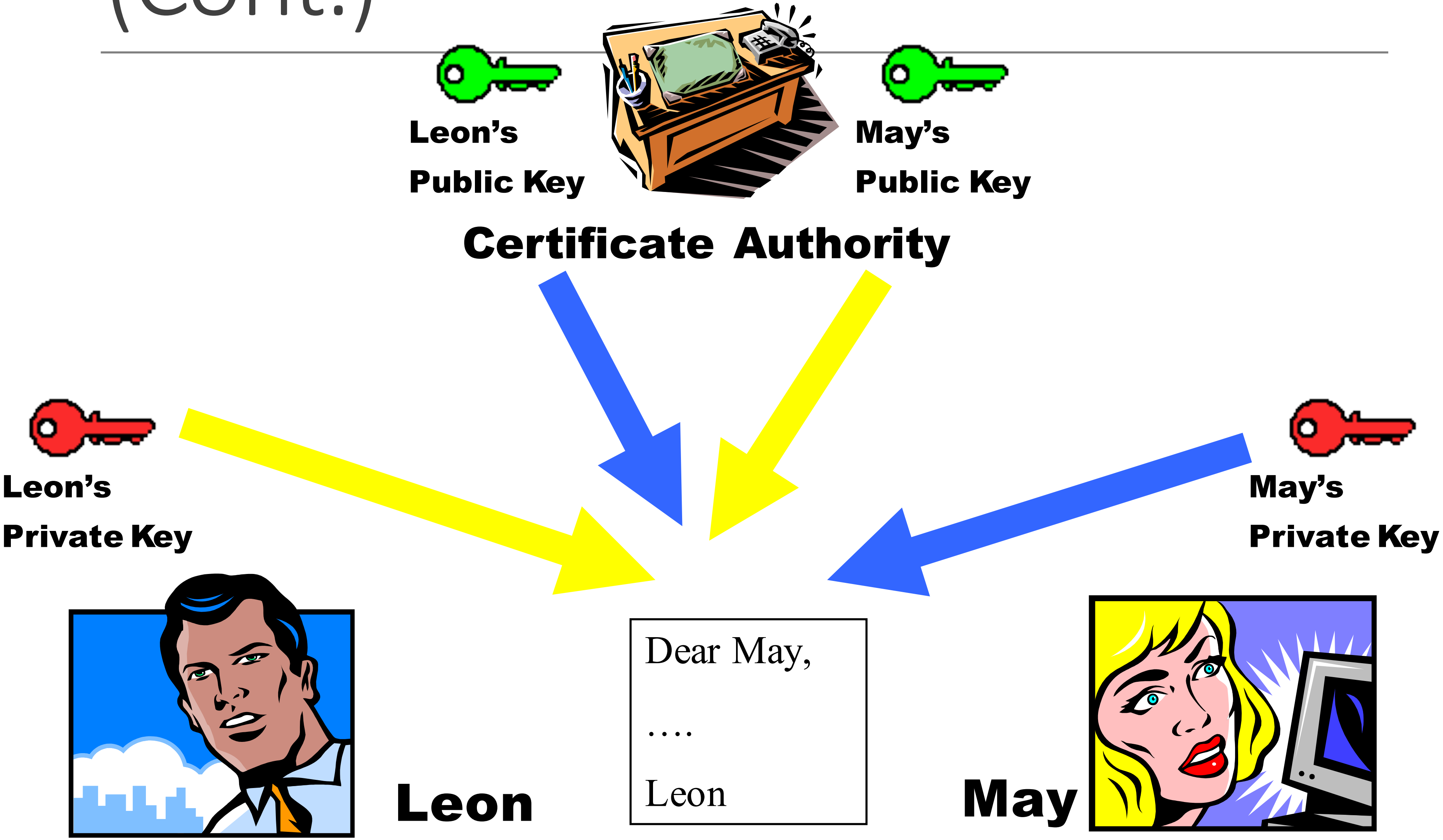


Pretty Good Privacy (PGP)

Web of Trust



Public Key Infrastructure (PKI) (Cont.)



Public Key Infrastructure (PKI) (Cont.)

Why PKI ?

- Non-repudation
  - Private key encrypt and public key decrypt
- Identification & Authentication
  - Access control on private key
- Confidentiality
  - Public key encrypt and private key decrypt
- Integrity
  - Successful decrypt by the decryption key

PKI – Key Management and Distribution

Management of public components in public-key system

- Public components can be managed by on-line or off-line directory service
- Exchange directly by the users

Concerns

- Generation and storage of component pairs
- Hardware support for key management

PKI – Certificate Management

Centralized management – PKI, CA

- ✓ Entire process over insecure channels with excellent security
- ✓ Distribution of certificates are valid at time of receipt
- ✗ Bottleneck
- ✗ Concentration of trust in one entity

Decentralized management – PGP

- Users are responsible for managing their own certificates
- Central authority periodically issue invalid certificates list

Phone book approach

Overview

Concerns the entire life cycle of cryptographic keys employed with cryptographic modules

Required for all cryptographic modules

Cryptographic module will not only have its own key management requirements

Key Life Cycle

Key generation

- Random numbers shall be generated truly randomly or pseudo-randomly
- Seed key shall be entered in the same manner as cryptographic keys
- Intermediate key generate states and values shall NOT be accessible outside the module in unprotected form

Key Life Cycle (cont'd)

Key distribution

- Manual, automated or hybrid methods
- Documentation shall specify key distribution techniques implemented by the module

key entry and output

- Manual or electronic entry methods
- Manually entered keys shall be verified for accuracy and consistency
- Split knowledge procedure shall be considered when keys in manual distribution
- Electronically distributed keys shall be entered and output in encrypted form

Key Life Cycle (cont'd)

Key storage

- Secret and private keys MAY be stored in plaintext format within a cryptographic module, which shall NOT be accessible from outside
- Ensure all keys are associated with correct entities to which keys are assigned

Key destruction

- Cryptographic module shall provide capability to zeroise all plaintext cryptographic keys and other unprotected critical security parameters
- Zeroization is not required if keys and parameters are either encrypted or physically/logically protected

Key Life Cycle (cont'd)

Key archiving

- Optional
- Keys output for archiving shall be encrypted

Key notarization

- Apply additional security to a key utilizing the identity of the originator and ultimate recipient

Key Escrow

Government concerns surveillance and forensics investigation

Enables strong encryption and Government agents to obtain decryption keys held by escrow agents

Only decryption keys are backup, signing keys shall NOT be stored

Split-knowledge procedure

A method of key recovery

PKI Overview

Operations of PKI *MAY* include:

- Registration
- Certification
- Key pair recovery
- Key generation
- Key update
- Cross-certification
- Revocation
- Directory lookup

Digital Certificate

What is Digital Certificate ?

- A digital Certificate is an electronic “credit card” that establishes your credentials when doing business or other transactions on the web
- It is issued by a certification authority
- One of the most popular standards specifying the contents of a digital certificate is X.509, published by the International Telecommunications Union (ITU)
- The most updated version is X.509 version 3
- The most supporting version is X.509 version 2

Digital Certificate (Cont.)

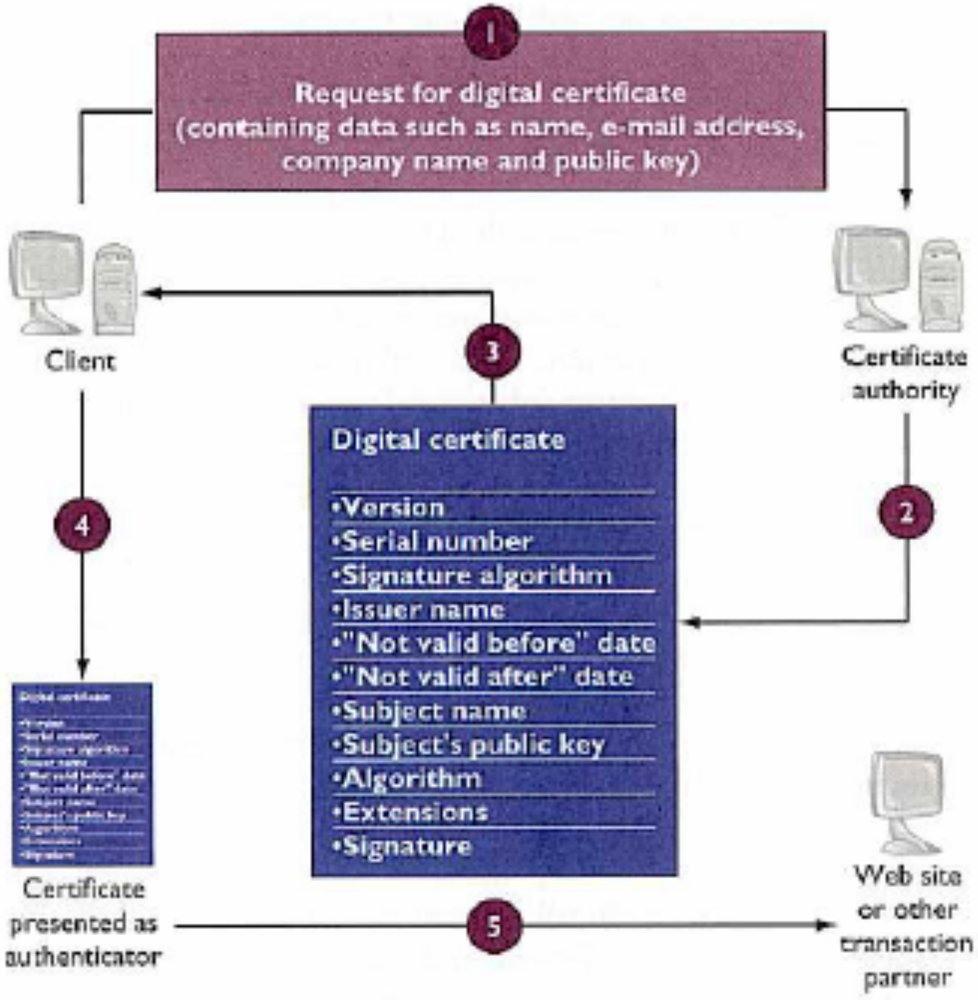
Version	3
Certificate serial no.	123456
Signature algorithm identifier	RSA with MD5
Issuer	c=hk, o=HKPOST
Validity period	Start=01/08/96, expiry=-1/08/97
Subject X.500 name	c=hk, o=issl, cn=Janzon Lo
Subject public key information	ae 5f 03 e6 g7..........
Issuer unique identifier	
Subject unique identifier	
Extensions	
CA's Digital Signature	aUdie8Uy9JkL..........

Digital Certificate (Cont.)

Certificate Revocation List (CRL)

- A collection of electronic data containing information concerning revoked Digital Certificates
- Hong Kong Post Statement:
  - The CRL is a data structure that enumerates public-key certificates (or other kinds of certificates) that have been invalidated by their issuer prior to the time at which they were scheduled to expire.
  - Under normal circumstances, Hongkong Post will publish the latest CRL as soon as possible after the update time. Hongkong Post may need to change the above updating and publishing schedule of the e-Cert CRL without prior notice if such changes are considered to be necessary under unforeseeable circumstances.

Digital Certificate (Cont.)



Certificate Authority

CA performs

- Issue and deliver subordinate and cross certificates
- Accept revocation requests from certificate holders and ORAs for certificate is issued
- Post certificates and CRLs to the repository
- Request CA certificates

Reply upon repository to make X.509 v3 certificates and X.509 v2 CRLs

CA accredit ORAs – an off-line decision to accept ORA generated certification requests

CA identify certificate holders using X.500 distinguished names

Public Key Infrastructure (PKI)

Asymmetric Key Encryptions

- Public Key (Publicly available – stored in Certificate Authority)
- Private Key (Secretly available – stored in your own pocket)

Public Key Infrastructure (PKI) (Cont.)

Mathematical Foundations

- Computational Complexity deals with time and space requirements for the execution of algorithms, such that “impossible” to solve in polynomial time
  - Factorization of the product of two very large prime numbers, such that
  - Key-strength = how large the prime numbers used

Public Key Infrastructure (PKI) (Cont.)

- Rely on impossible computational capability

Public Key Infrastructure (PKI) (Cont.)

FIPS – approved algorithms

- Digital Signature Algorithm (DSA)
- Ron Rivest, Adi Shamir, & Leonard Adleman (RSA)
- Elliptic Curve DSA

Cryptographic attack

Symmetric Key Algorithm – Attacks !

Brute force attack

- Longer key length
- Change secret-key more frequently

Meet-in-the middle attack

Chosen-plaintext attack

Known-plaintext attack

Differential cryptanalysis attack

Related-key cryptanalysis attack

Public Key Algorithm – Attacks !

Man-in-the-middle attack

Chosen-plaintext attack

Factorization attack

Ciphertext-only attack

Common modulus attack

Low exponent attack

SSL related vulnerabilities

Recently, there are a number of SSL related vulnerabilities:

- Heartbleed (2014)
- POODLE (2014)
- Critical SSL flaw that Apple patched in OS X and iOS (2014)
- FREAK (2015)

HeartBleed Vulnerability

The flaw is called 'Heartbleed' because it comes from a programming mistake in OpenSSL's implementation of the TLS/DTLS (transport layer security protocols) 'heartbeat' extension. It affects websites using OpenSSL 1.0.1 through to version 1.0.1f.

Heartbleed bug is only present in the OpenSSL implementation of SSL and TLS

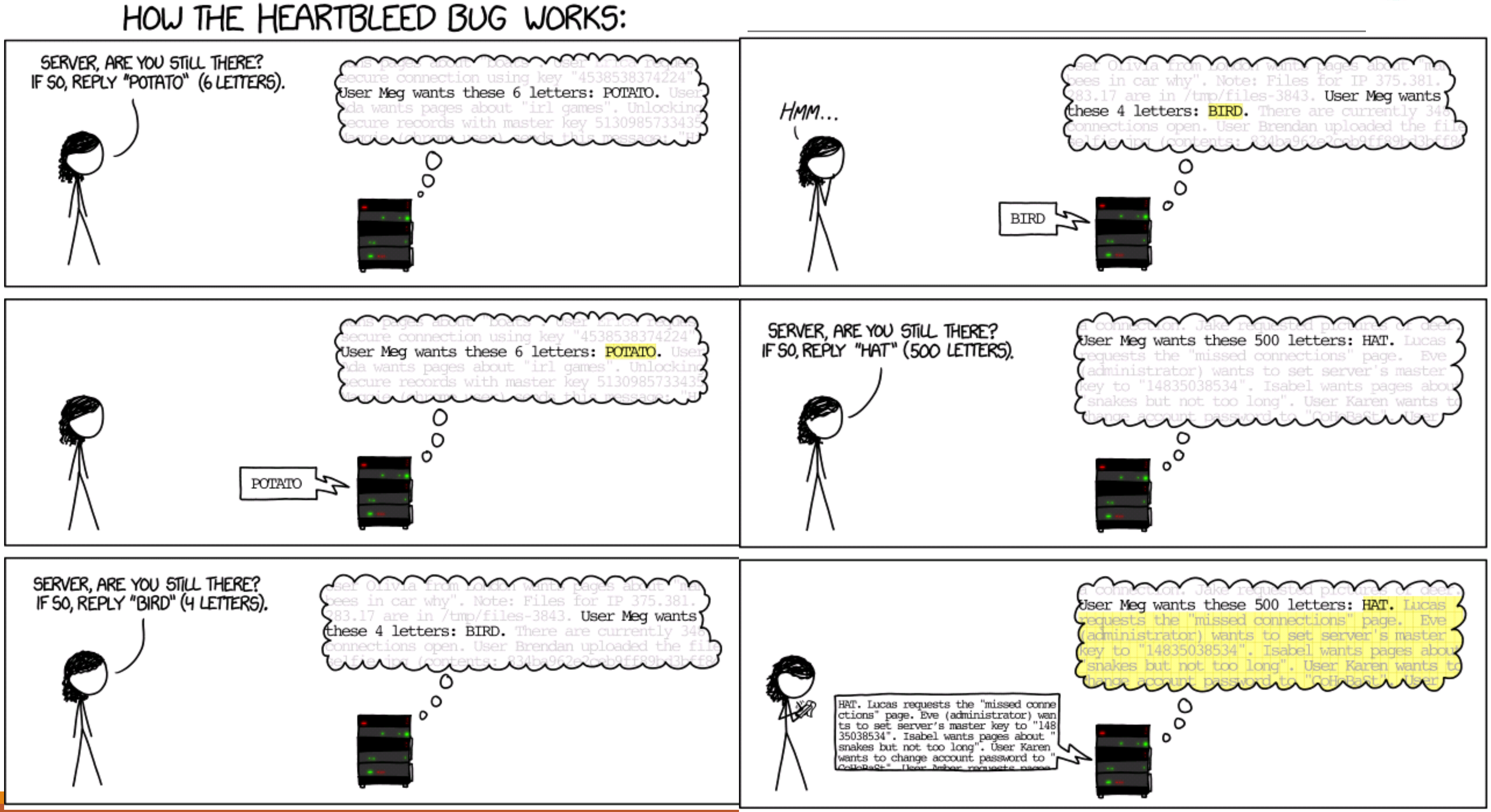
At the time of disclosure, about 17% of the world's "secure" websites were said to be vulnerable to the bug.

The Heartbleed bug itself was introduced in December 2011, in fact it appears to have been committed about an hour before New Year's Eve

Some websites have been set up where CISOs and end users can check whether the websites they run or use are vulnerable. Two such sites are:

- http://filippo.io/Heartbleed/
- https://lastpass.com/heartbleed/

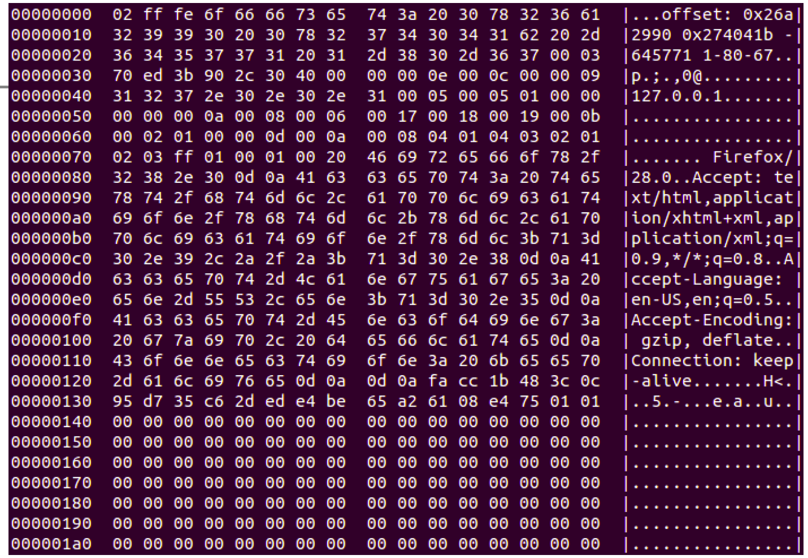
HeartBleed



http://xkcd.com/1354/

HeartBleed

Some of them are session-related information such as session ID, different tokens, keys, and some other sensitive internal information such as queries, internal data, etc. A real example shows what we can receive in the response



Almost all which must be protected by SSL, few examples:

https://www.youtube.com/watch?v=lkst_tSwB9o

https://www.youtube.com/watch?v=D5Igbv-c1dY (Metasploit – OpenSSL Heartbeat)

Transport Layer Security

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to provide communications security over a computer network.

As of 2014 the 3.0 version of SSL is considered insecure as it is vulnerable to the POODLE attack that affects all block ciphers in SSL; and RC4, the only non-block cipher supported by SSL 3.0, is also feasibly broken as used in SSL 3.0

POODLE attack (Sep 2014)

The POODLE attack (which stands for "Padding Oracle On Downgraded Legacy Encryption") is a man-in-the-middle exploit which takes advantage of Internet and security software clients' fallback to SSL 3.0

As of 2014 the 3.0 version of SSL is considered insecure as it is vulnerable to the POODLE attack that affects all block ciphers in SSL; and RC4 (the only non-block cipher supported by SSL 3.0) is also feasibly broken as used in SSL 3.0

In cryptography, a padding oracle attack is an attack which is performed on the padding of a cryptographic message.

The plain text message often has to be padded (expanded) to be compatible with the underlying cryptographic primitive.

Leakage of information about the padding may occur mainly during decryption of the ciphertext. Padding oracle attacks are mostly associated with ECB or CBC mode decryption used within block ciphers.

In symmetric cryptography, the padding oracle attack is most commonly applied to the CBC mode of operation, where the "oracle" (usually a server) leaks data about whether the padding of an encrypted message is correct or not.

Such data can allow attackers to decrypt (and sometimes encrypt) messages through the oracle using the oracle's key, without knowing the encryption key.

POODLE attack (Sep 2014)

If attackers successfully exploit this vulnerability, on average, they only need to make 256 SSL 3.0 requests to reveal one byte of encrypted messages

POODLE to attack it are CVE-2014-3566 and CVE-2014-8730

Migration

- To mitigate the POODLE attack, one approach is to completely disable SSL 3.0 on the client side and the server side
- To prevent this attack, one could append an HMAC (Hash-based message authentication code) to the ciphertext. Without the key used to generate the HMAC, an attacker won't be able to produce valid ciphertexts.