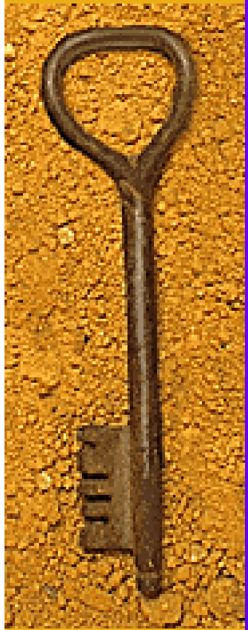
Securing Communications



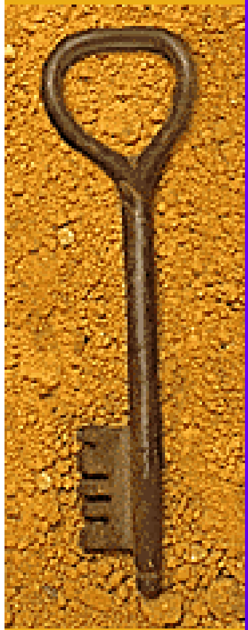
Overview

- Client/Server Authentication (Kerberos)
- Remote User Authentication Service (RADIUS)
- Public-Key Infrastructure (PKI)
- IP Layer Security (IPSec)
- Web Access Security (SSL)
- E-mail Confidentiality (PGP, S/MIME)
- Wireless LANs Security (802.11b)
- Cellular Phone Security (WPKI)

Client/Server Authentication

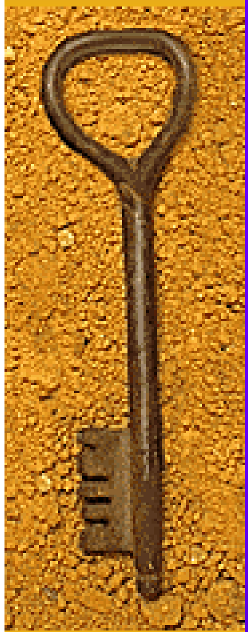
Kerberos

Main sources: Stallings, Schneier, Kaufman et al

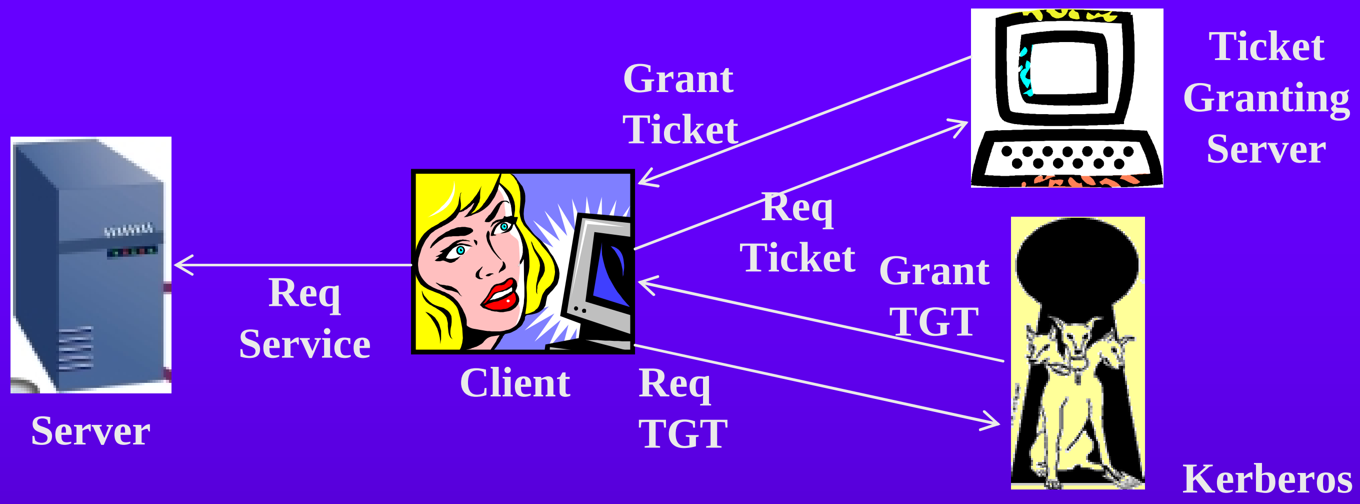


Kerberos

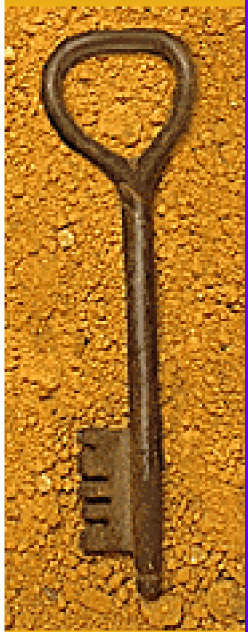
- Client / Server Authentication service
  - Deployed as a network service that allows users and servers to mutually authenticate
  - Uses conventional symmetric key as proof of identity (DES)
  - Developed in MIT by Project Athena.
- Types of concerns addressed
  - User impersonation
  - Alteration of a device identity
  - Replay attacks
- Requirements
  - Security:
    - eavesdropper cannot get enough information
    - Kerberos itself should be secure
  - Reliability and high availability
  - Transparency to the User



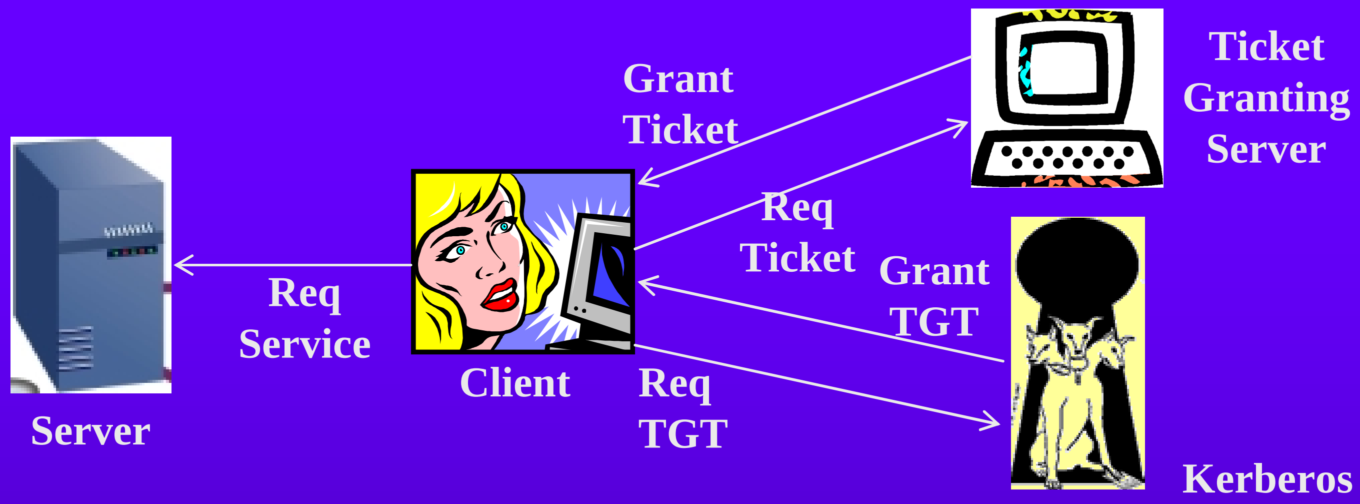
Kerberos Protocol



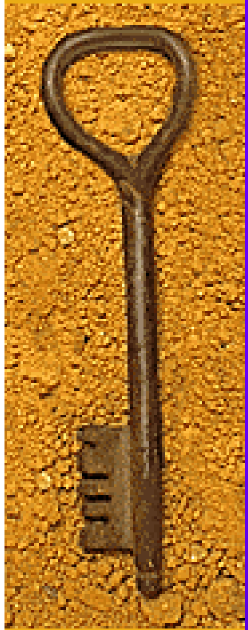
- Ticket: $T(c,s) = s,E_{Ks}(c,a,v,K_{c,s})$
  - c-client, s-server, a-client address, v-validity time
  - Used as a “pass” until expiration
- Authenticator: $A(c,s) = E_{Kc,s}(c,t,k)$
  - t-time stamp, k-additional session key
  - Used once, but the client can generate as many as she wishes



Kerberos Protocol

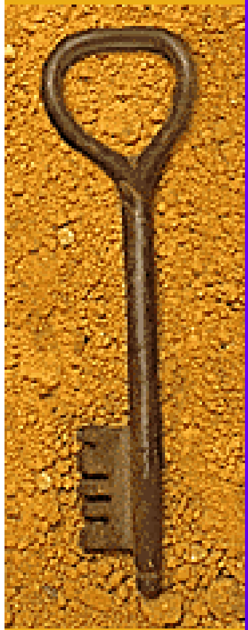


- Req TGT: Send c,tgs
- Grant TGT: Gen $K_{c,tgs}$; Send $E_{Kc}(K_{c,tgs})$, $E_{Ktgs}(T(c,tgs))$
- Req Ticket: Send $E_{Kc,tgs}(A(c,tgs))$, $E_{Ktgs}(T(c,tgs))$, s
- Grant Ticket: Gen $K_{c,s}$; Send $E_{Kc,tgs}(K_{c,s})$, $E_{Ks}(T(c,s))$
- Req Service: $E_{Kc,s}(A(c,s))$, $E_{Ks}(T(c,s))$



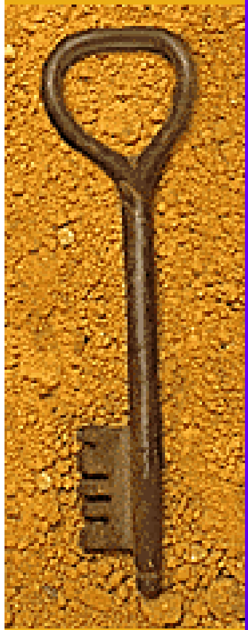
Other Kerberos Features

- Kerberos Replication
  - In large organizations, it is possible to replicate the TGT/Ss, with one copy serving as a master and the others being read-only
- Realms
  - It is common to divide the network services into groups, covered by different Kerberos servers
  - It is possible to create trust between two realms, by defining the one Kerberos TGS as a server in the other realm



Kerberos Security Features

- Kerberos verifies client identity of client through key, and comparing identity and address to a database
- Tickets T(c,tgs/s) is given to the client but is locked
- Server verifies client through session key in authenticator
- Timestamps used to ensure synchronicity and against original ticket validity (typically 8 hours)
- With a simple addition, client can verify server
- It is common to quickly replace use of client long-term key with a session key



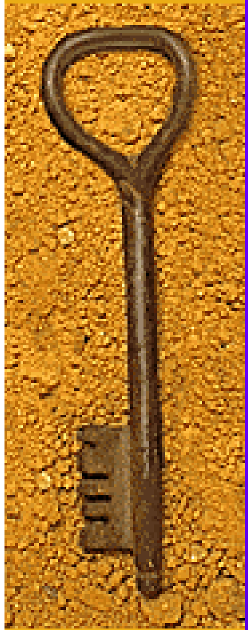
Attacks on Kerberos Security

- Kerberos itself stores many keys and should be protected
- Tickets may be replayed within allowed lifetime. Server should store recent requests and check for replays
- Adversary may cache many TGTs and work offline to decrypt them. Clients shall use safe passwords
- By changing server clocks, adversary may replay tickets. Hosts shall synchronize clocks often
- Kerberos will be enhanced with public-key cryptography and smart card-based key management

Remote User Authentication Service

RADIUS

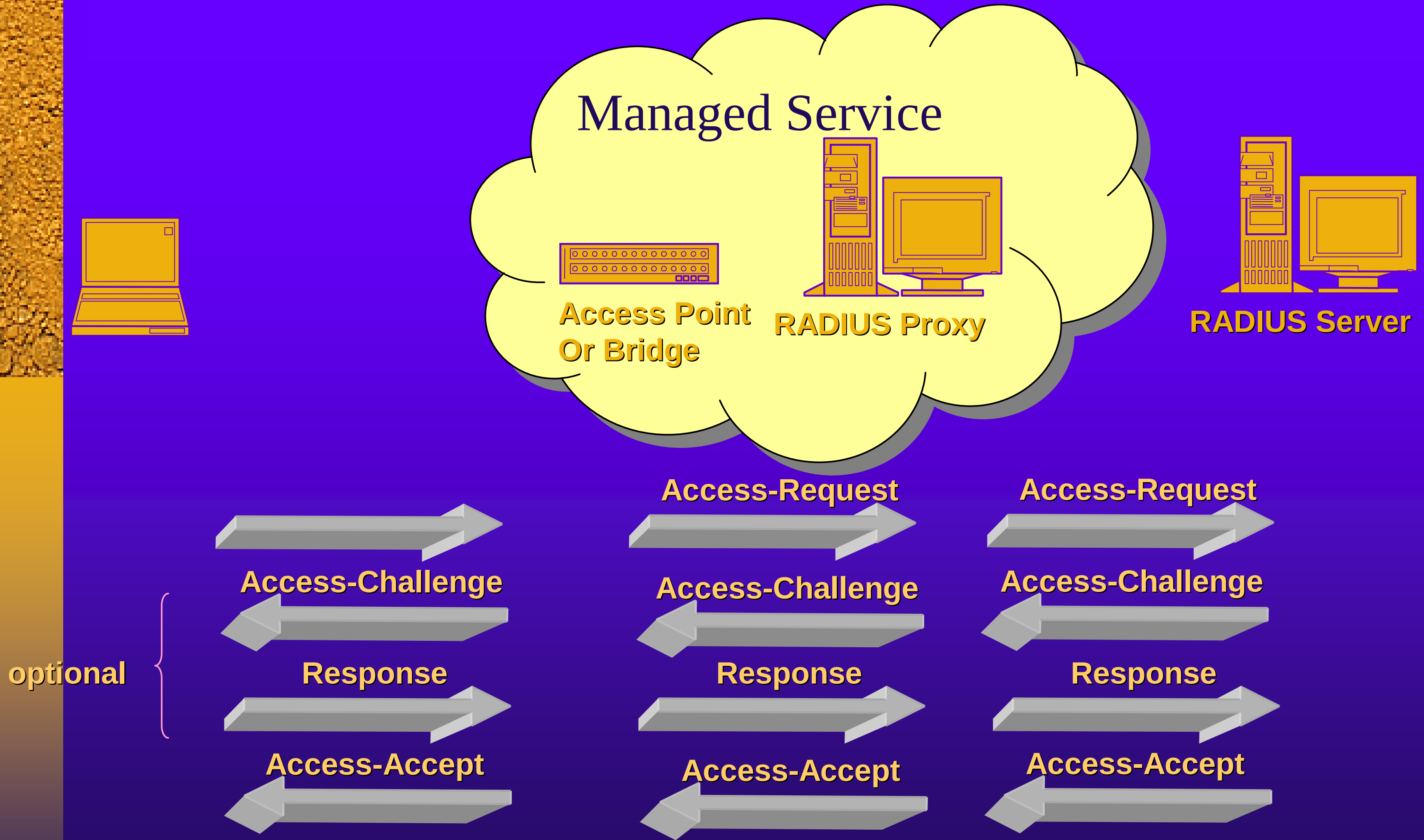
Main resources: IETF, Josh Hill

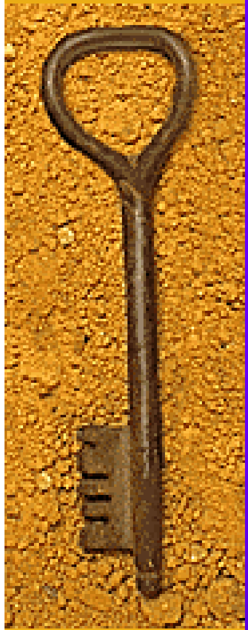


RADIUS

- Remote Authentication Dial In User Service
  - Originally developed for dial-up access
- Widely implemented client/server network protocol
  - Implemented in transport layer (using UDP)
  - Clients are all types of Network Access Servers (NAS)
  - Provides 3A (authentication, authorization, accounting)
  - Example: NT4.0 IAS
- Supports mobile and remote users
  - physical ports (modems, DSL, wireless)
  - virtual ports (extranets, VPNs)
- Allows centralized/remote control and accounting
- Proxy RADIUS protocol allows distributed authentication

How it works


Managed Service
Access Point
Or Bridge
RADIUS Proxy
RADIUS Server
Access-Request
Access-Request
Access-Challenge
Access-Challenge
Access-Challenge
optional
Response
Response
Response
Access-Accept
Access-Accept
Access-Accept

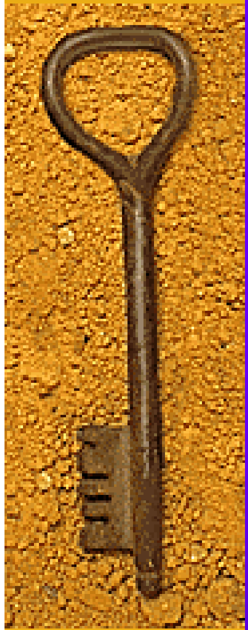


RADIUS Security Mechanisms

- RADIUS client and server share a secret (usually entered as a string password)
- Each request receives an authenticator (nonce)
- Messages are encrypted using a stream cipher, generated using MD5 applied to the secret and authenticator
  - Plaintext (user and password fields) are XORed with stream
  - Chained CBC-style if password is too large
- A few weaknesses were discovered
  - MD5 was not meant to be a stream cipher
  - By XORing two captured ciphertexts, the eavesdropper gets the XOR of the two plaintexts; if one password is shorter, the suffix of the other appears in plaintext
  - Similarly, enables an offline attack on the shared secret
- A few improvements were suggested, including use of symmetric encryption
- Better yet, RADIUS exchange can be encrypted via VPN (IPSec)

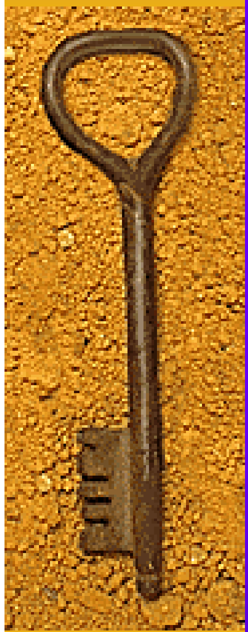
Public Key Infrastructure (PKI)

Main sources: Stallings, IETF



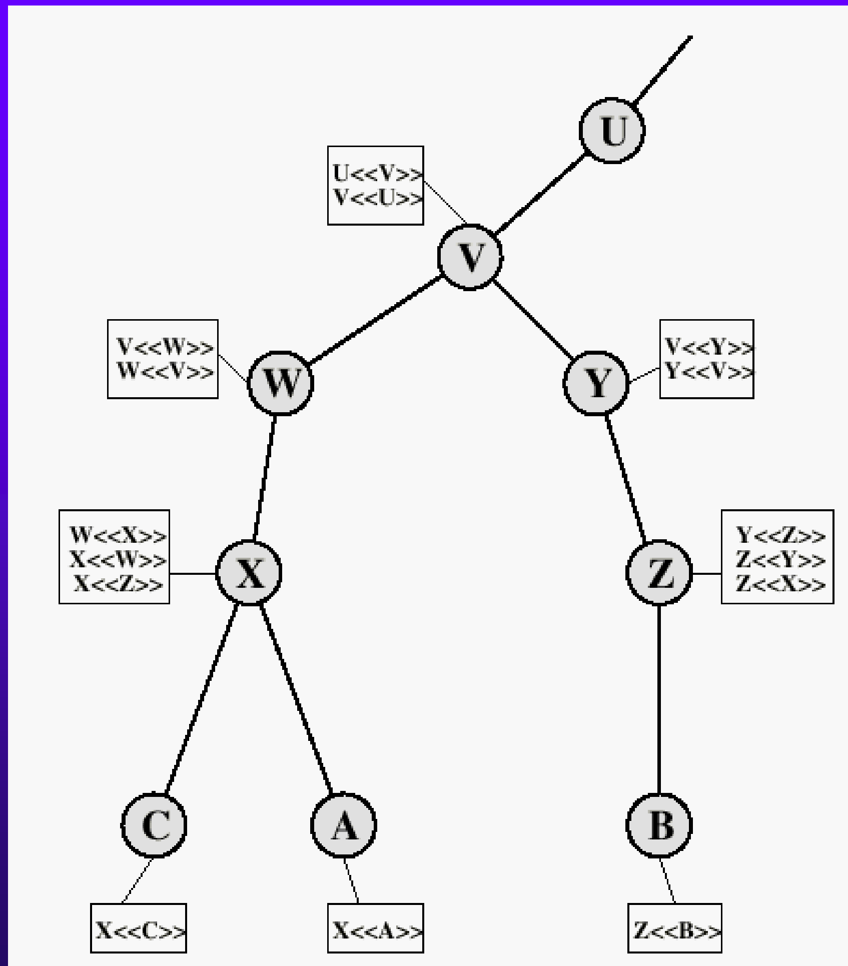
Public Key Infrastructure (PKI)

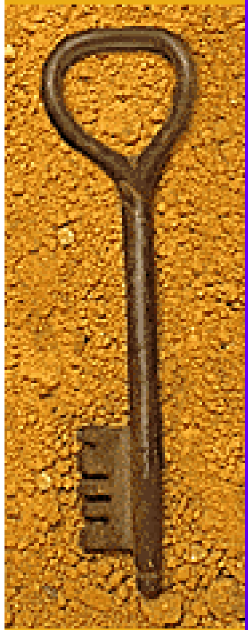
- IETF X.500 Directory Services Protocol is a distributed directory of resources, users, and access policies
  - A.k.a. Directory Access Protocol (DAP)
  - For each user, the directory stores a set of attributes, e.g., UserID, Organization, etc.
  - Most common implementations are of the Lightweight Directory Access Protocol (LDAP) variant, e.g., MS Active Directory
  - Directories are distributed, with protocol running above TCP
- The X.509 sub-protocol provides authentication service
  - Implemented in organizations using PKI servers, which provide access to authentication information, and local CA functionality
  - For each user, the directory may store a certificate that contains some user information and her public key, signed by a CA
  - Works with most common crypto-hash and signature algorithms



X.509 CA Hierarchy

- Stores forward- and reverse certificates for each CA
  - CA<<X>> is X's certificate signed by the CA
- Each certificate contains user attributes, as well as expiration
- Any user with the public key of the CA can get the full path to a specific user
  - e.g., for Z you can get U<<V>>, V<<Y>>, Y<<Z>>
- In case of distributed CAs, one can go back on the chain to obtain (securely) the public key of his counterpart CA
- Certificates can be revoked by CA through published CRLs



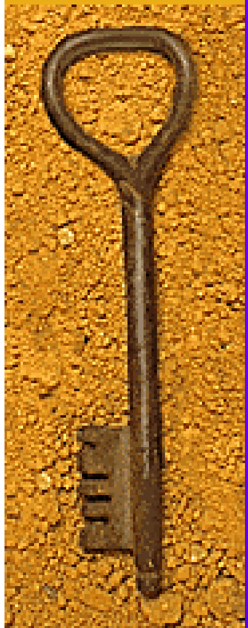


Example: VeriSign Certificates

- Information on certificate
  - Owner name, address, e-mail
  - Public key
  - Certificate expiration date
  - Name of issuing CA
  - CA digital signature

- Digital ID (certificate) classes
  - Class 1: only e-mail is verified
  - Class 2: verification of postal address and other information from consumer databases
  - Class 3: requires appearing in person and/or notarized documentation

X.509 Authentication



- One-way authentication
  - Alice sends authenticating message, and signed hash of same message
  - Message includes:
    - Timestamp
    - Random identifier (against replay)
    - Bob’s identifier
    - Her own certificate
    - Session key encrypted with Bob’s public key

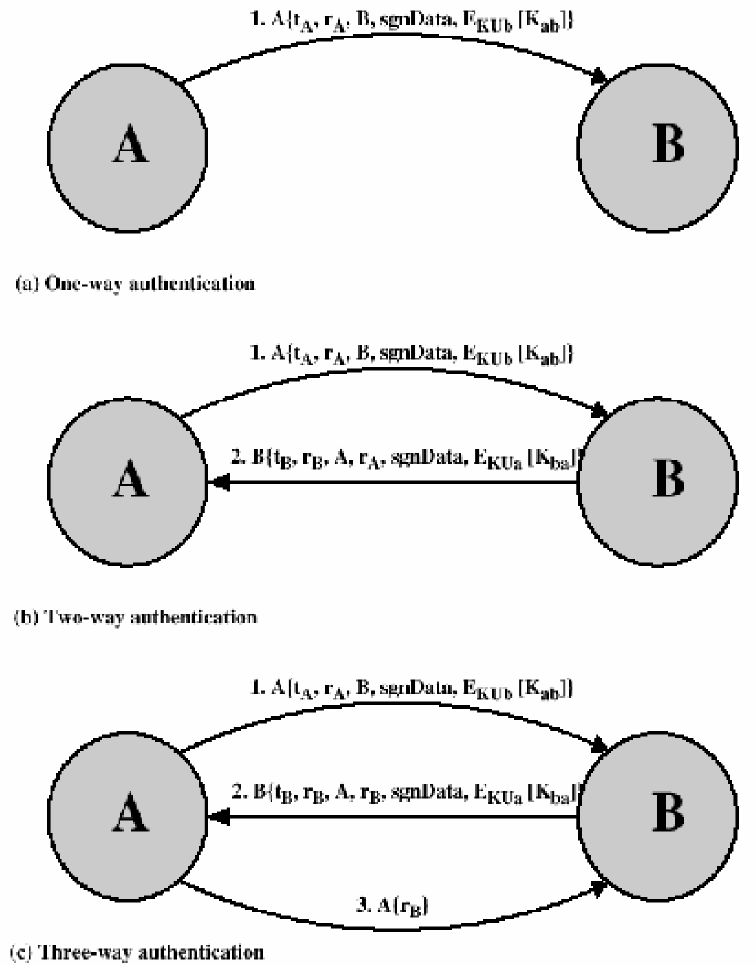
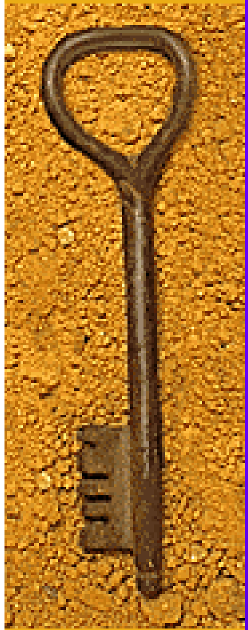


Figure 4.5 X.509 Strong Authentication Procedures



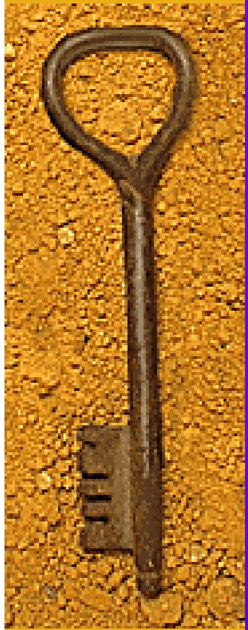
PKI Servers Functionality

- Main functions
  - Issuing (CA) and registering (RA) certificates
  - Storing and retrieving certificates
  - Revoking certificates
  - Key Lifecycle management
- Applications
  - E-mail (S/MIME)
  - Web browsing (SSL and IPSec)
  - Digitally signed mobile code and documents
  - Other applications, through API

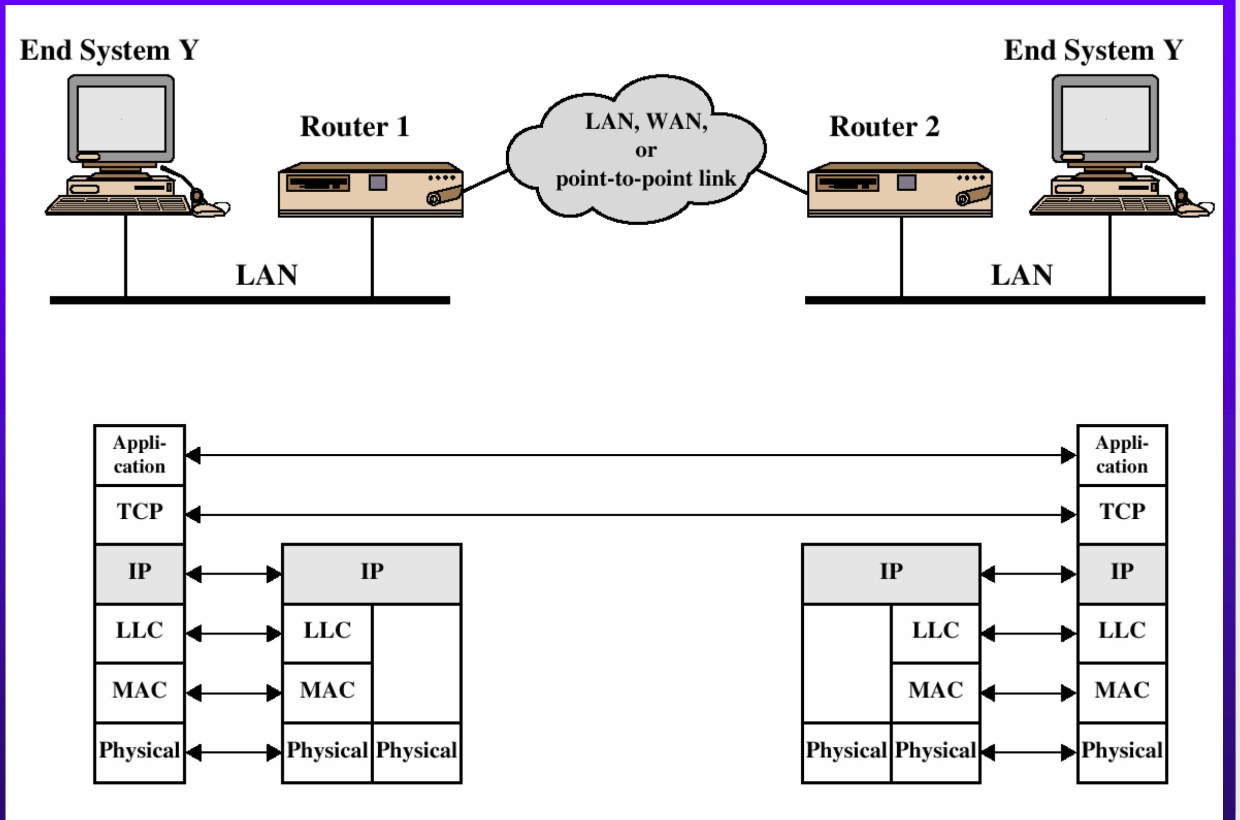
IP Layer Security

IPSec

Main Source: Stallings



Network (IP) Layer



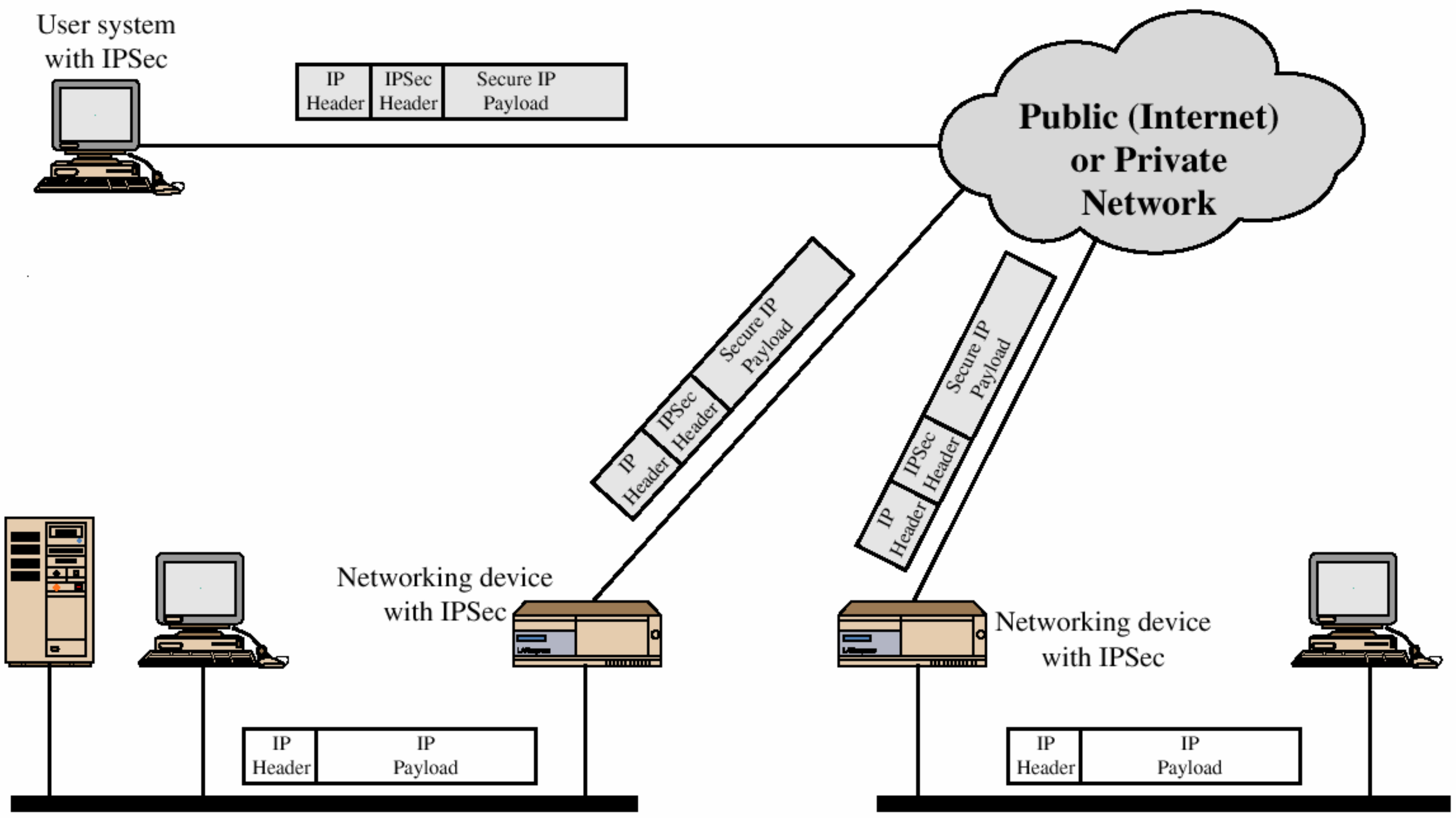
OSI 7 layers

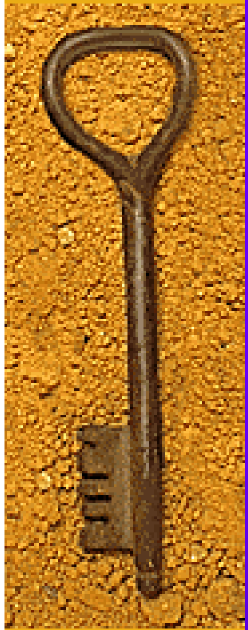
- Application
- Presentation
- Session
- Transport
- Network (IP)
- Data Link
- Physical

IP Security

- IPSec is not a single protocol, but rather a framework, and set of algorithms that address security concerns at IP layer
  - Authentication
  - Confidentiality
  - Key Management
- Designed for IPv6 but implemented in most IPv4
- IPSec is carried out at the packet level
  - Implemented in transport level in routers or in PC-based software
  - All packets going out are encrypted
  - All packets coming in are authenticated and decrypted
- IPSec is implemented in the transport layer
  - Transparent to applications
  - Gives certain peace of mind to security ignorant applications
  - Routers can authenticate neighboring routers and routing requests

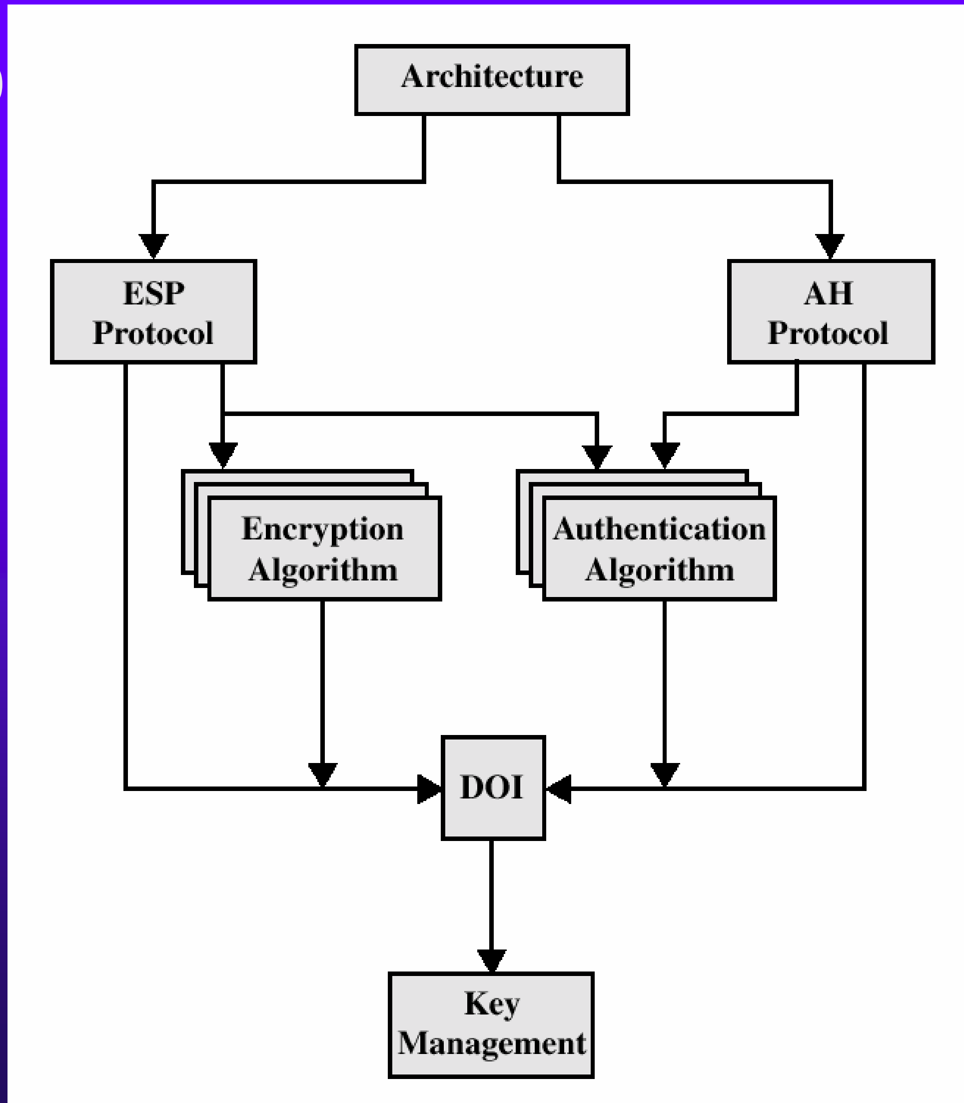
IPSec Usage: A Typical Scenario

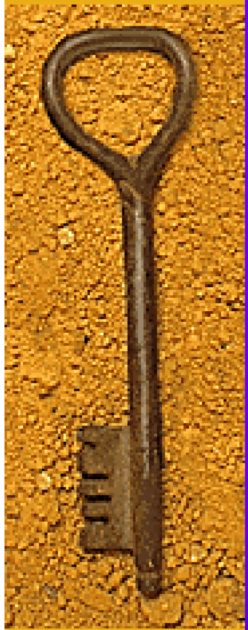




IPSec Architecture

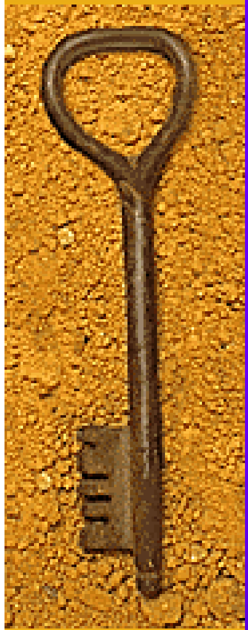
- Authentication Header (AH)
  - only authentication service
- Encapsulating Security Payload (ESP)
  - Packet Encryption
  - Packet Authentication (optional)
- Domain Of Interpretation (DOI)
  - Specific parameters for encryption and authentication algorithms
- Key Management





IPSec Services

	AH	ESP	ESP with authentication
Access Control			
Connectionless Integrity			
Data origin authentication			
Reject replayed packets			
Confidentiality			
Limited traffic flow confidentiality			

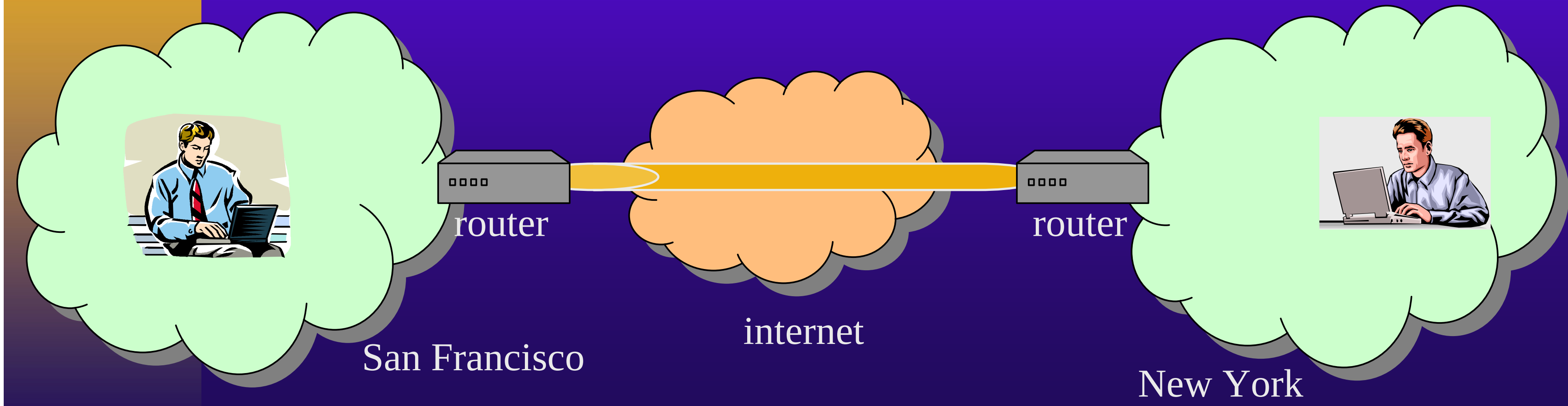


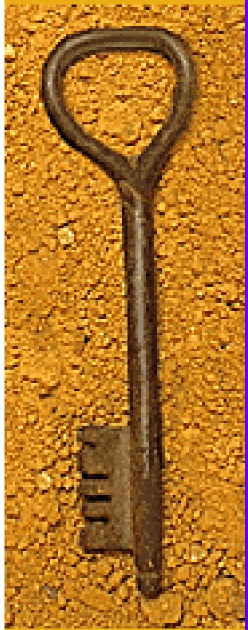
Security Association (SA)

- A one-way relationship between sender and receiver
  - Security Parameters Index (SPI)
    - Identifies the SA in the SA database
  - IP Destination
    - Address of destination endpoint
  - Security Protocol Identifiers
    - Specifies whether ESP or AH should be used
- SA Database stores all SA entries
  - AH info: authentication algorithm, keys, key lifetime,…
  - ESP info: encryption and authentication algorithms, keys, IVs,…
  - Running sequence number, used to prevent packet replays
  - SA lifetime
  - IPSec protocol mode: Tunnel, Transport
- Security Policy Database (SPD) specifies SA selectors that determine mapping of outbound packets to specific SAs

Transport and Tunnel Modes

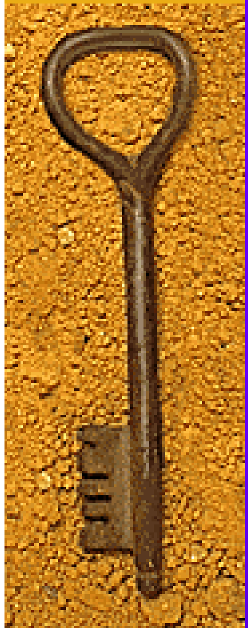
- Transport Mode
  - Protects upper layers
  - IP Payload is encrypted
- Tunnel Mode
  - Protects all layers
  - New outer packet is created at the network boundary, with original packet as its payload, and the entire inner packet is encrypted





Authentication Header (AH)

- Authentication of data and source
  - Prevent modifications of payload while in transit
  - Prevent IP spoofing
  - AH contains Integrity Check Value (ICV)
    - Calculated HMAC over payload and all transit-immutable values, concatenated with shared key (truncated to 96 bits)
    - IPSec requires supoprt of at least SHA-1 and MD5
- Counter replay attacks
  - Prevent capture and replay of packets
  - For every SA, source generates up to 2^{32} sequence numbers, then starts a new SA with a new key
  - Receiver authenticates using a sliding window (w=64)

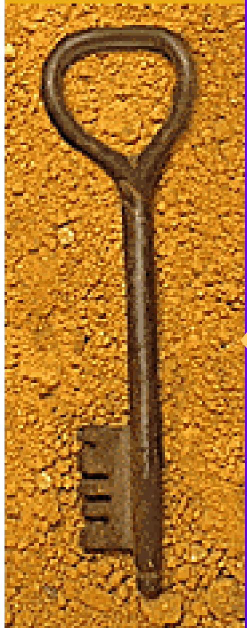


Encapsulating Security Payload (ESP)

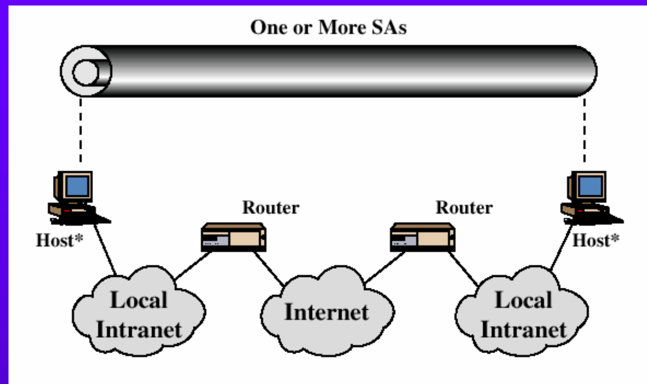
- Adds encryption of the payload
  - Encryption Algorithms: 3DES, RC5, IDEA, 3IDEA, CAST, Blowfish
  - Plaintext payload is replaced with ciphertext by source and is routed as new payload
- Optionally provides authentication
  - HMAC with SHA-1 or MD5
- Anti-replay sequence number

- Note: This is not a repetition because each SA can only use either ESP or AH but not both

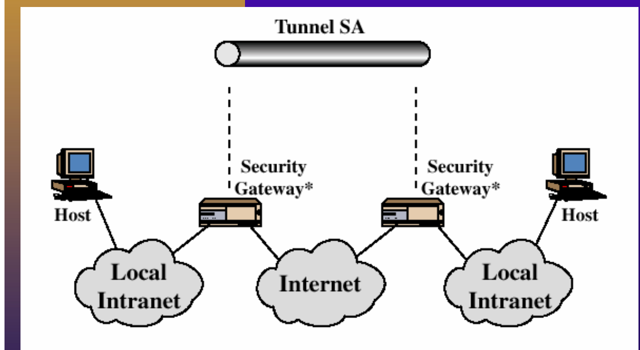
SA Bundles and Tunneling



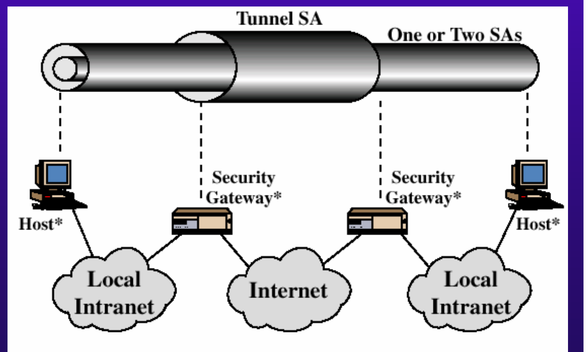
- SA bundles allow a sequence of SAs to be applied to same packet, or within a tunnel
  - Transport adjacency
    - Transport ESP SA (without authen) followed by Transport AH SA (covering also ESP fields)
  - Iterated Tunneling



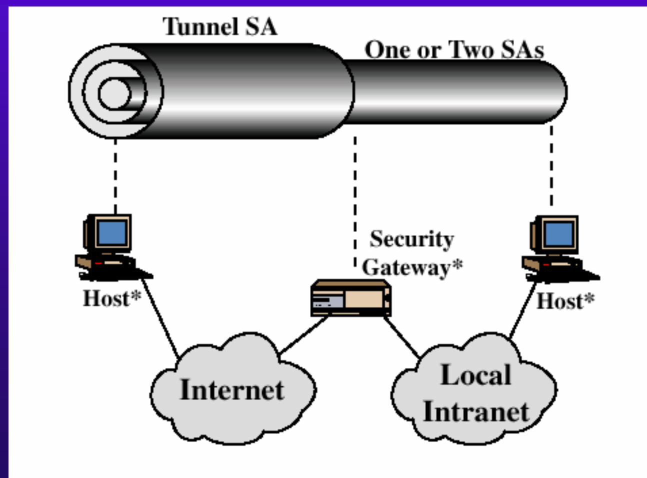
(a) Case 1



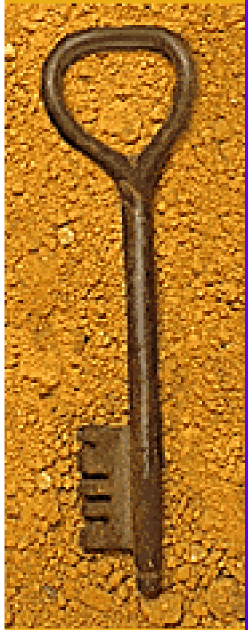
(b) Case 2



(c) Case 3



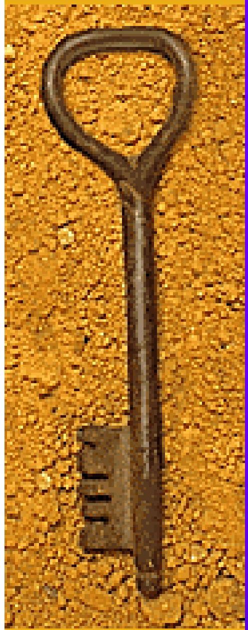
(d) Case 4



Key Management

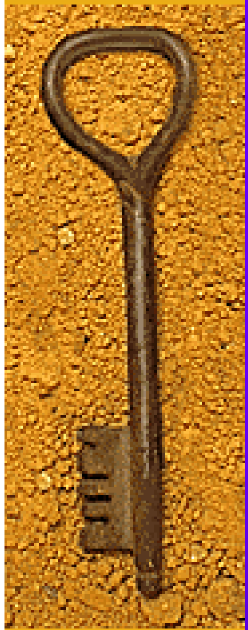
- Option 1: Manual configuration

- Option 2: Automated on-demand creation of keys (ISAKMP/Oakley)
  - ISAKMP – default SA and key management protocol
    - Does not mandate a specific key determination and exchange protocol, but implements at least Oakley
  - Oakley – default key determination protocol



Oakley

- A refinement of Diffie-Hellman
  - Reminder: session key = g^{xy} mod p, where x and y are private keys of parties
- DH weaknesses
  - Clogging attack: attacker forces Alice to exponentiate endlessly
  - Man-in-the-middle attack: attacker impersonates Alice to Bob and impersonates Bob to Alice
- Oakley hardening
  - Uses cookies, exchange of authenticating party-dependent random numbers, hence attacker can only clog with acknowledge requests
  - Authenticates DH exchange to prevent impersonation
  - Uses nonces against replay attacks
- Options:
  - Choice of “groups”: setup parameters for DH exchange
  - Choice of authentication method



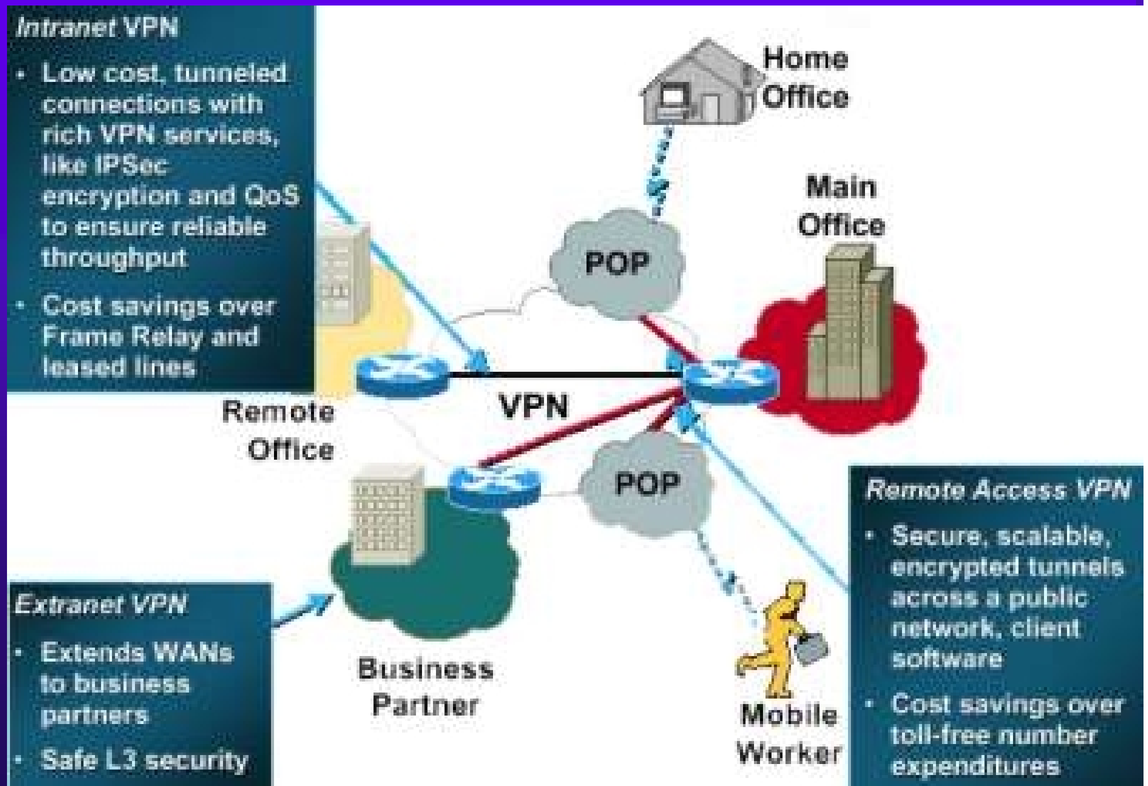
ISAKMP

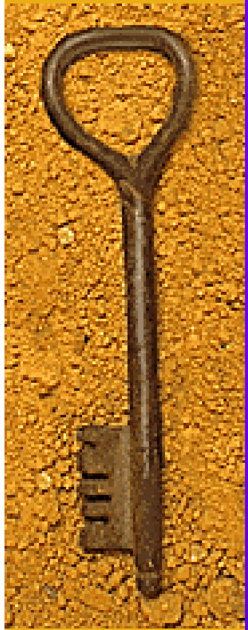
- Protocol to establish, negotiate, modify, and delete SAs
- ISAKMP messages:
  - Security Association – establish new SA (initial parameters)
  - Proposal – indicates the protocol to be used (ESP or AH)
  - Transform – the algorithms to be used, e.g., 3DES, HMAC-SHA-1
  - Key Exchange – which key exchange protocol, e.g., Oakley, RSA
  - Identification – the identity of the peers, e.g., IP address, User ID
  - Certificate – certificates of the peers
  - Certificate Request
  - Hash – data generated by the hash function
  - Signature – data generated by digital signature function
  - Nonce – the current nonce
  - Notification - messages
  - Delete – revoke an SA

Virtual Private Networks (VPN)

- Types of VPNs
  - Remote Access, a.k.a. Virtual Private Dialup Network (VPDN), where a user dials into the network
  - Site-to-Site – intranet, and/or extranet

- VPDN is currently most used form of VPN; Intranet is second
- VPDN is usually set up by a third-party Network Access Server (NAS)





VPN Implementations

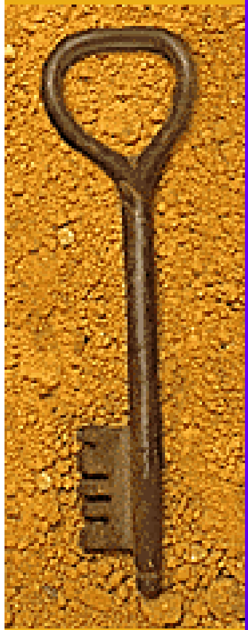
- IPSec tunneling or transport encryption
- Simple encryption for systems that are not IPSec enabled
  - Symmetric encryption using a physically-delivered shared key
  - Public-key encryption, e.g., using RSA or PGP
- Most implementations include
  - Authentication, Authorization, and Accounting (3A) servers
  - Firewalls/ QoS servers
- Actual implementation
  - Desktop client for remote users
  - VPN concentrator (hardware)
  - Part of firewalls/routers

Web Access Security

Secure Socket Layer (SSL)

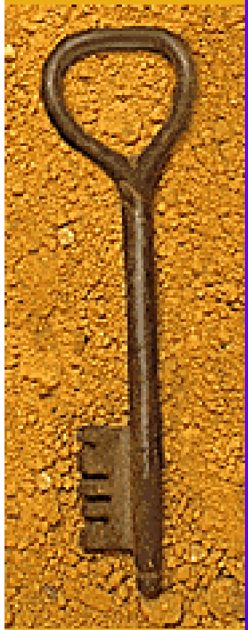
Transport Layer Security (TLS)

Main Source: Stallings



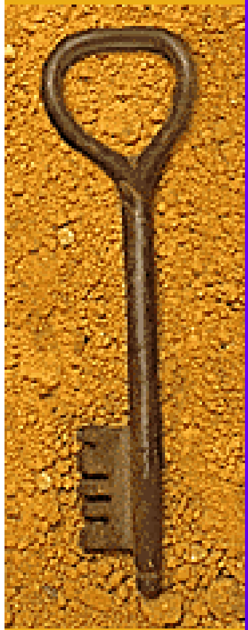
Web Security Considerations

- In principle, Web access is simply client-server
  - Protocols such as Kerberos apply
- Special characteristics of Web access
  - Web servers are “out there” accessible to anyone
  - Web servers often must be connected to corporate databases, and can be dangerous if subverted
  - Applicative software is quickly developed for web servers, and is often security-ignorant
  - Web users are often not subject to corporate rules
  - Web users are often not knowledgeable
  - Web users cannot be counted on to fulfill their part in a security protocol



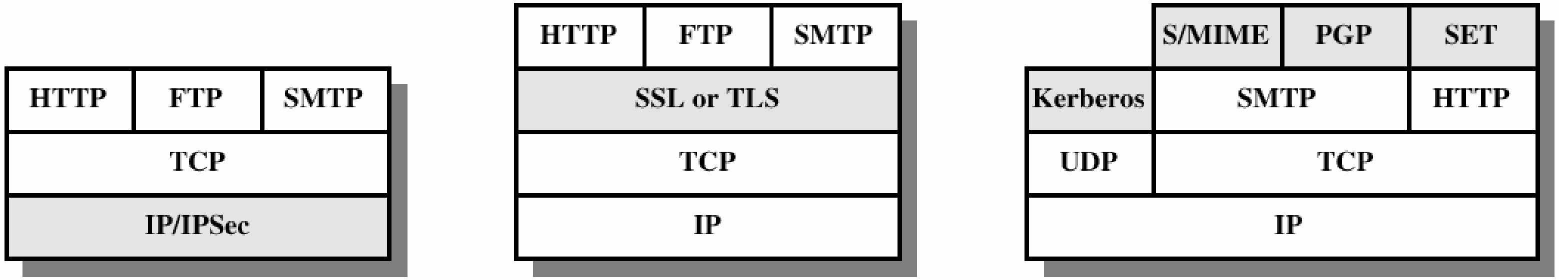
Security Threats on the Web

- Integrity
  - Modification of data on servers ← System Security
  - Modification of messages ← Communication Security
- Confidentiality
  - Theft of data from server, or from client ← System Security
  - Eavesdropping on communication ← Communication Security
  - Info on network configuration ← System Security
  - Info on network traffic ← Communication Security
- Interruption
  - Denial of Service and DDOS ← System Security
- Authentication
  - Impersonation of legitimate users ← Communication Security
  - Data forgery on server (or client) ← System Security

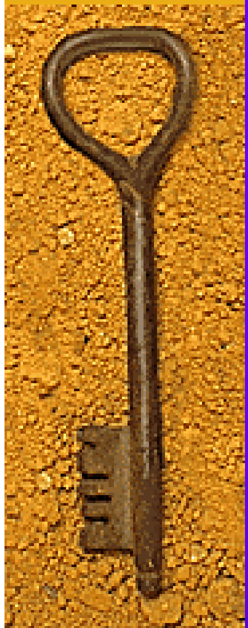


Alternative Security Facilities for Web Communications

- Network layer: IPSec
- Application layer
- SSL/TLS protocols
  - As a protocol above TCP in transport and session layers
  - As part of application software: browser on client side and web server (SSL was developed by Netscape)



(a) Network Level (b) Transport Level (c) Application Level



Secure Socket Layer (SSL)

- Developed by Netscape as part of their browser
  - SSLv3 was subjected to public review
  - Transport Layer Security (TLS) designed as successor to SSLv3
- SSL works is a session-based protocol, and each session may consist of multiple connections
- SSL consists of two layers
  - SSL Record Protocol provides basic security services, e.g. https
  - Handshake protocol is used to initiate sessions
  - Alert protocol for peer messaging

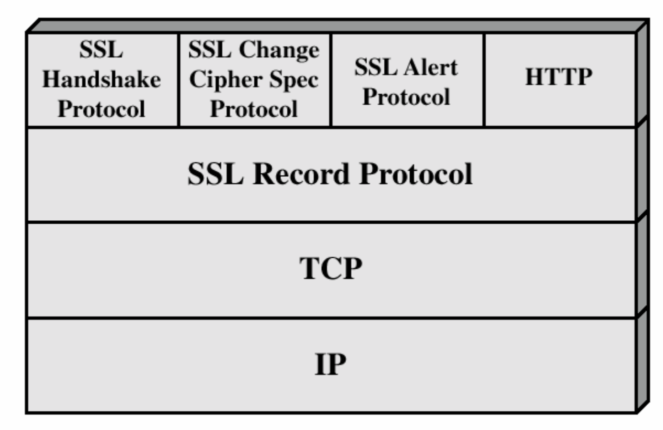
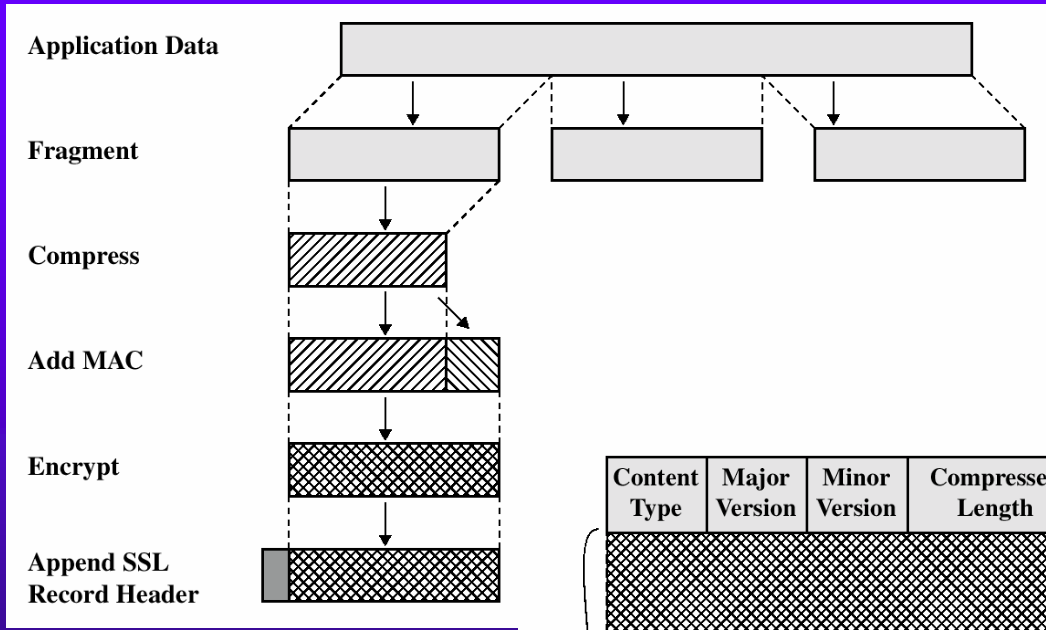


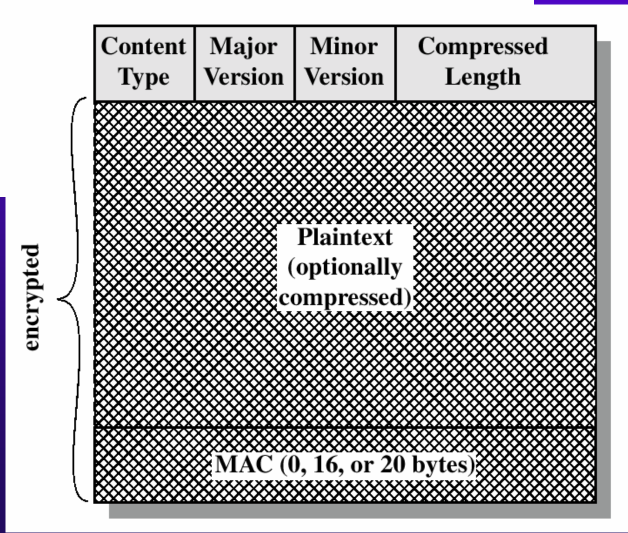
Figure 7.2 SSL Protocol Stack

- SSL session states:
  - Security algorithms
  - Compression methods
  - Master keys
  - Certificates

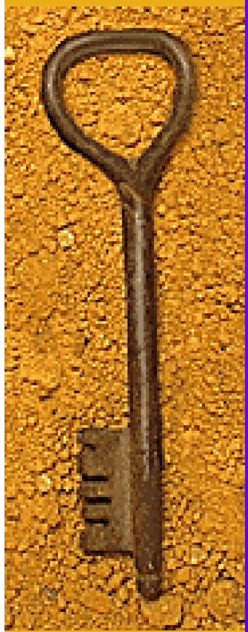
SSL Record Protocol

- Services: Confidentiality, Message Integrity
- Several encryption algorithms are permitted
- HMAC standard



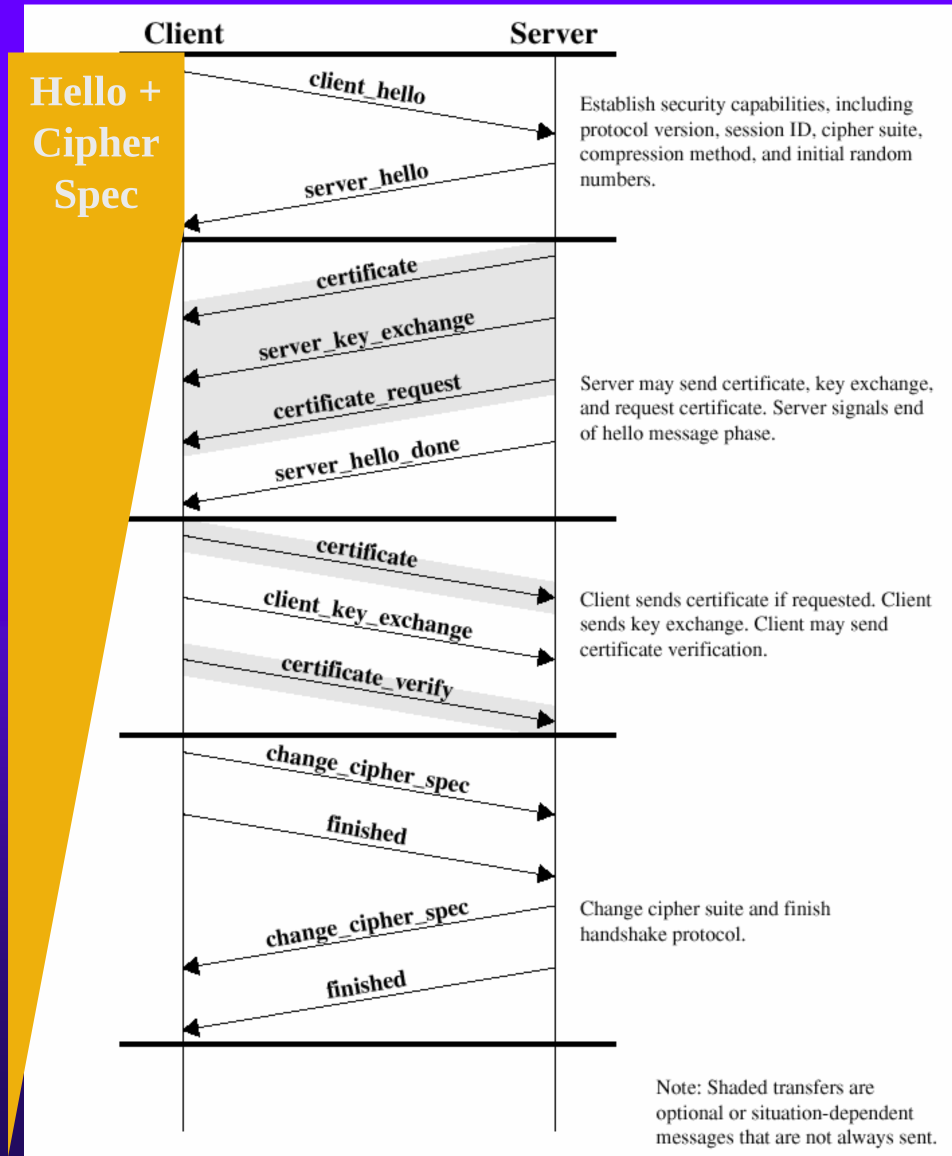


- Header:
  - Content type: SSL-specific, or application (e.g. HTTP)
  - SSL version

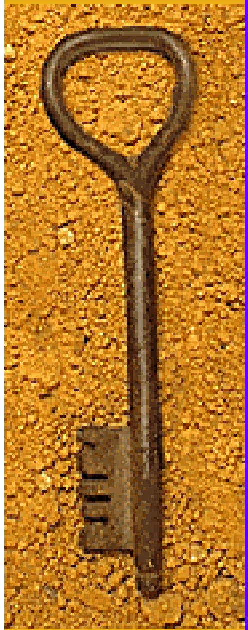


Handshake Protocol

- Client suggests; Server chooses
- SSL version: lower version will be used
- Nonce: timestamp+random
- Session ID: existing or new
- Alternative CipherSpec suites, in decreasing preference
  - Key exchange
  - Encryption algorithms
  - MAC algorithm
  - Parameters
- Compression methods supported

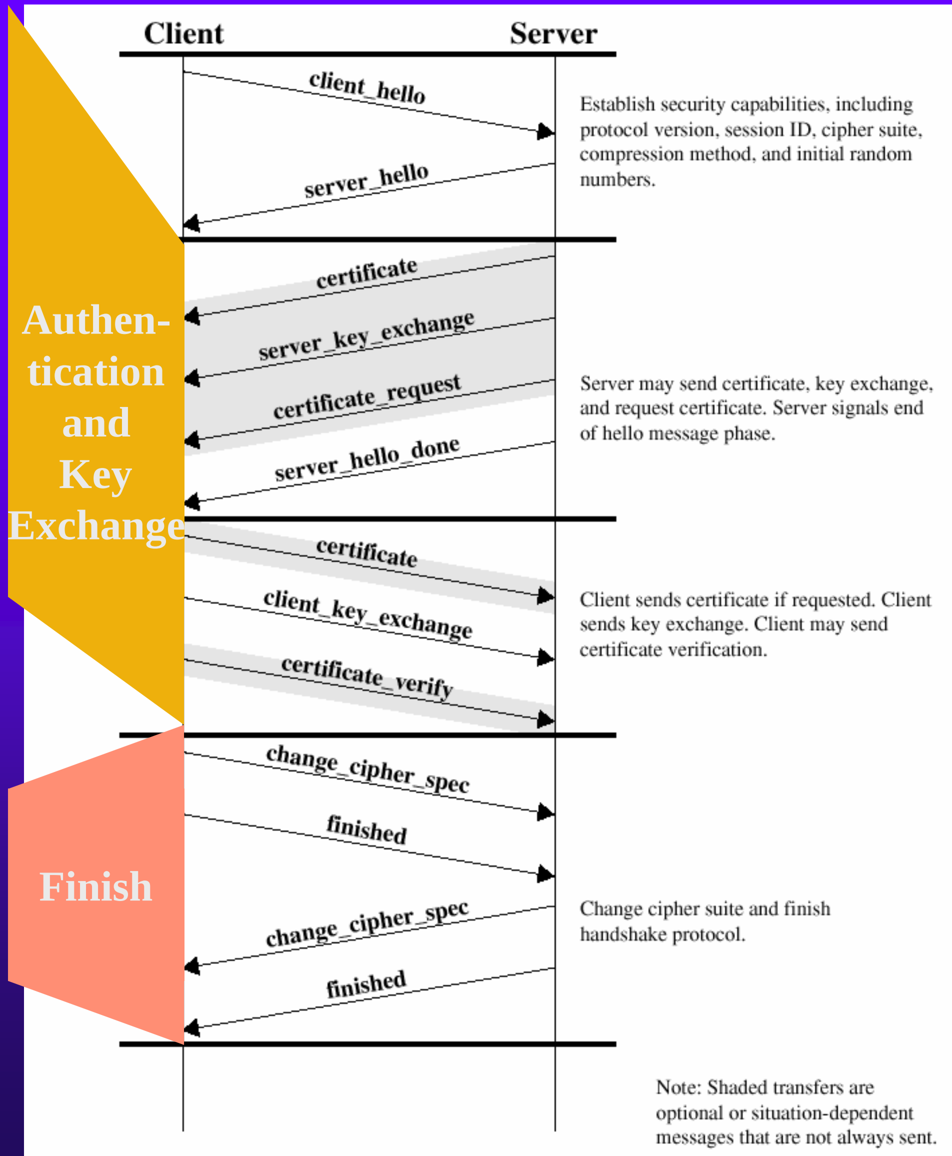


Handshake Protocol



- Server starts; client follows
- Server sends certificate
- Server sends key exchange message
- Server may ask for client certificate
- Client responds

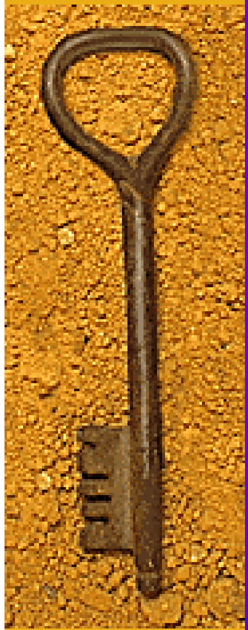
- State changed to pending cipher_spec
- Handshake done



Secure E-mail

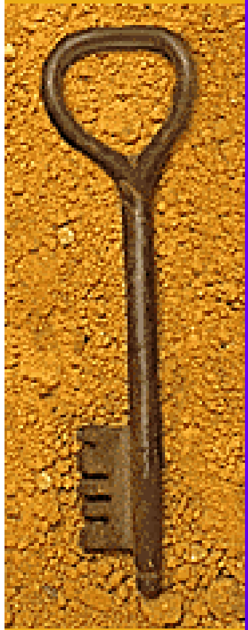
Pretty Good Privacy (PGP)

Secure MIME



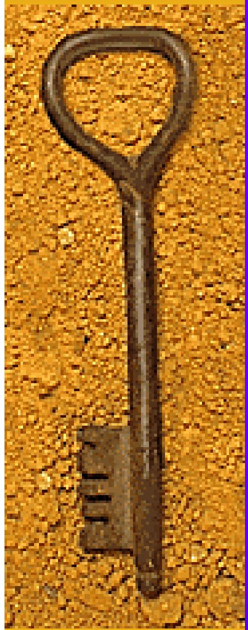
E-mail Security Requirements

- E-mail is most widely used network application
  - Compatibly available on virtually any platform and OS
- Security services
  - Confidentiality
  - Source Authentication
  - Message Integrity Authentication
- Other Requirements
  - Cross-platform compatibility
  - Asynchronous availability: no need for both parties to be simultaneously logged-in



Pretty Good Privacy (PGP)

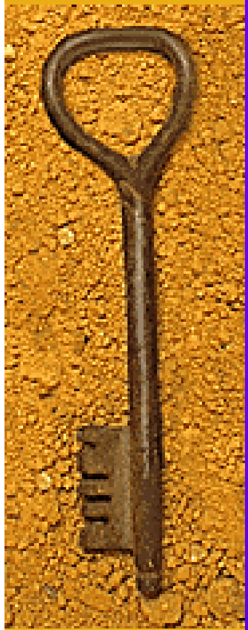
- Created by Philip Zimmerman
  - Freely available on most platforms   http://www.pgpi.org
  - Was adopted at OpenPGP RFC that can be freely implemented
  - Commercial version available from Network Associates (pgp.com)
  - Based on a selection of best available algorithms
- Provides the following services:
  - Confidentiality
    - Key exchange: Diffie-Hellman, or RSA
    - Encryption: CAST-128, or IDEA, or 3DES
  - Authentication
    - Digital signature using SHA-1/MD5, and encrypted using DSS/RSA
  - Compression: ZIP
  - Attachments also encrypted using PGP's file encryption protocol
  - Partitioning and reassembly of large messages
- PGP available also for icq and wireless communication



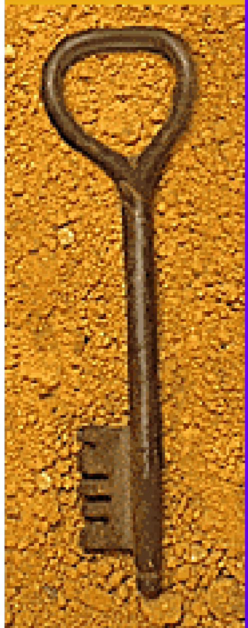
PGP Confidentiality

- Sender processing
  - Generates a distinct session key per message
  - Compresses message
  - Encrypts session key using receiver's public key
    - RSA, or ElGamal/DH for key exchange
  - Encrypts message using session key
    - Using conventional cryptography faster than RSA
  - Appends encrypted key and message, and sends
- Receiver processing
  - Decrypts session key using own private key
  - Decrypts message
  - Unzips

PGP Message Integrity and Source Authentication

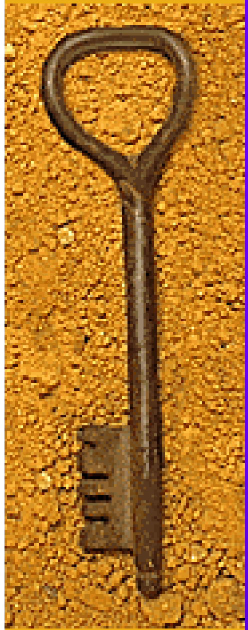


- Sender processing
  - Use SHA-1 to generate 160-bit hash code for the message
  - Hash code is encrypted using sender's private key
  - Encrypted hash code is appended to message and sent
- Receiver processing
  - Decrypts the hash code using the sender's public key
  - Generates a new hash code from the message
  - Compares received and computed hash codes
- Note: Signatures can be kept detached from the message, e.g., allowing multiple non-nested signatures on same document, and for record purposes



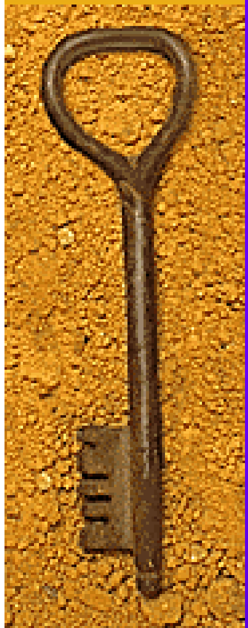
PGP Authentication+Confidentiality

- PGP supports both services
- Sender processing
  - Signature is generated first, and appended to original message
  - Appended message is compressed and encrypted using session key
  - Session key is encrypted using receiver's public key
- Compression is applied *after* the signature
  - Signature can be kept with original message for later verification
  - Compression algorithm is independent and can be changed
- Encryption applied to compressed message
  - has less redundancy than original plaintext – harder cryptanalysis



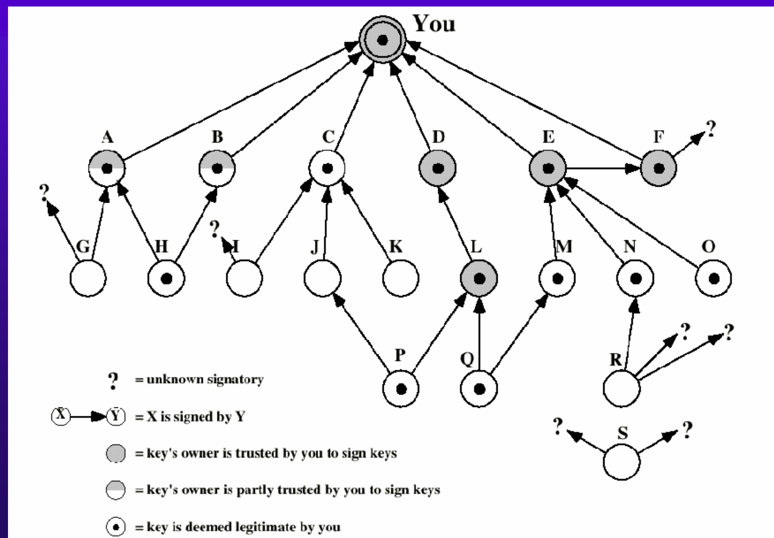
PGP Key Management

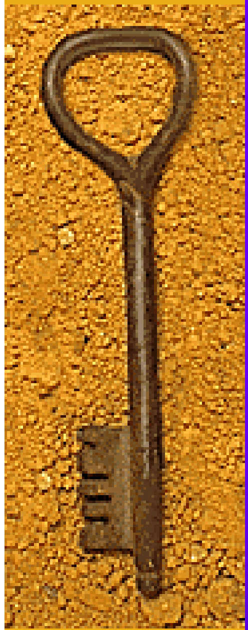
- Session Keys. PGP employs a keystroke-based technique for generating cryptographically strong session keys
  - Next Session Key = $E_{PrevKey}$(keystroke)
- Rings. PGP allows users to maintain “rings” with multiple pairs of private-public keys
  - To be able to decrypt messages encrypted with older keys
  - To communicate with different users using different keys
  - Each key is identified (almost uniquely) by its rightmost 64 bits
  - Each key is also indexed by the User ID
- Passphrases. Private keys are kept encrypted, using the hash code of a user-chosen passphrase as key



PGP Public-Key Management

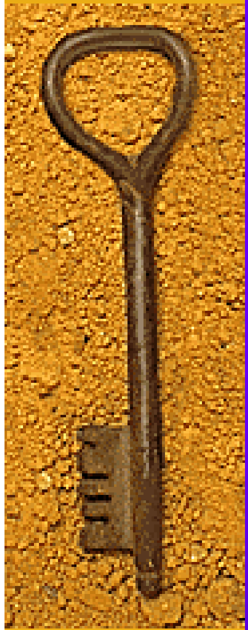
- Key distribution main concern: impersonation
- Options:
  - Alice can physically deliver the key on a floppy
  - Alice can e-mail or dictate key to Bob over the phone; Bob can verify the key with Alice using its hash code "fingerprint"
  - A trusted "introducer" can sign a certificate with Alice's key
  - Obtain Alice's key from a trusted certificate authority
- PGP associates with each key
  - a set of introducers, and Bob's trust in each
  - a level of legitimacy, computed by PGP from the combined legitimacies of the introducers
  - a level of trust in each user to legitimize another user
- Key owner can revoke it by signing revocation certificate





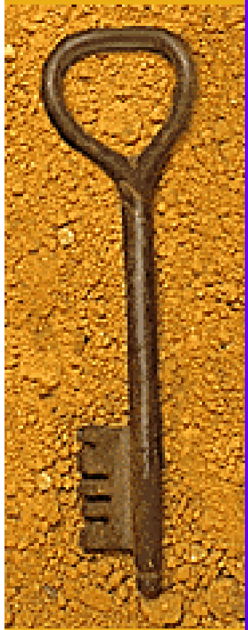
S/MIME

- Developed by RSA Data Security
- Secure / Multipurpose Internet Mail Extension
  - Built on top of MIME, based on technology developed by RSA
  - Likely to become Internet standard
- MIME fixes some of the limitations of SMTP (Simple Mail Transfer Protocol)
  - Large files
  - Non-ASCII characters (binaries, special)
- MIME header allows specification
  - multiple types, e.g., application/postscript, video/mpeg
  - multiple transfer encodings, e.g., 7bit, base64
- MIME Messages can be multi-part and contain multiple different contents



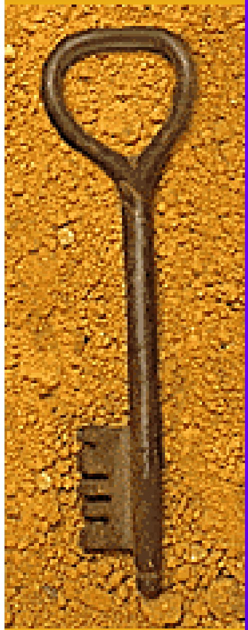
S/MIME Services

- Confidentiality
  – Enveloped data
- Authentication
  – Signed data: digital signature is created and is encoded with content in radix-64 (A-Z,0-9,+,/)
  – Clear-signed data: only signature is encoded in radix-64 and the rest of the message is clear
- Confidentiality and Authentication
  – Nesting of signature and enveloping in either order
- New MIME types (pkcs) added



S/MIME Algorithms

- Message digesting
  - SHA-1, MD5
- Encrypt message digest
  - DSS, RSA-512/1024
- Encrypt session key
  - DH/ElGamal, RSA
- Encrypt message with session key
  - 3DES, RC2-40



S/MIME Key Management

- Based on a hybrid of X.509 (CA) and PGP (local)
- User must establish herself at a recognized CA
- Certificate registration and revocation can be communicated using a special MIME type
- S/MIME uses CA to verify UserID-public key match
- Users manage copies of certificates/keys locally

Wireless LANs Security

802.11b

Main Sources: IEEE standards, SANS, and Berkeley Group

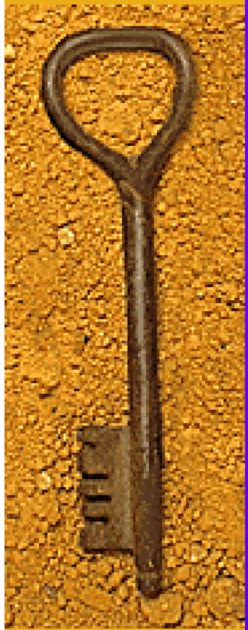
Wireless Networks

- Originally devised for mobile, and/or location-based services, but now gaining popularity due to low cost and easy setup



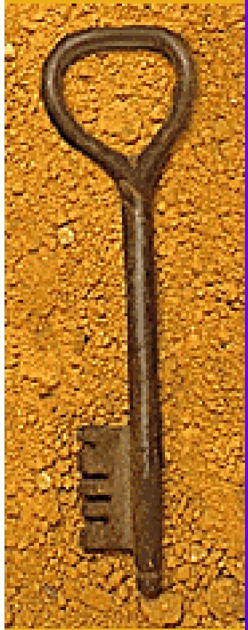


- HomeRF – 1.2Mbps (recently increased to 10Mbps)
- Bluetooth – short range (10m), Personal Area Network (PAN), very low voltage
- 802.11 – IEEE Standard for wireless LANs
  - Frequency hopping, using 2.4GHz unlicensed ISM frequency
- 802.11b (WiFi) – Direct Sequencing Spread Spectrum (DSSS), and increases bit rates to 11Mbps
  - Achieves market sweet spot, in terms of cost, acceptance, interoperability
- 802.11a – forthcoming, will increase rate to 54Mbps, and will address some of the security shortcomings



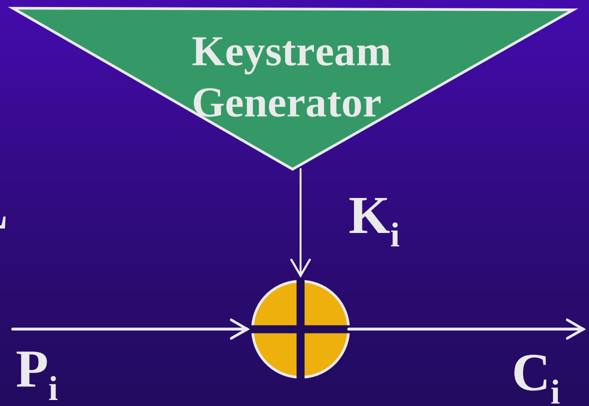
Wired Equivalent Protocol (WEP)

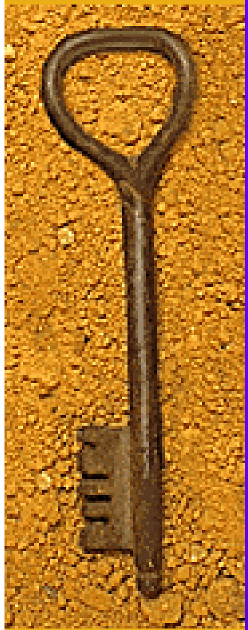
- WEP is the standard security in 802.11b
- WEP security services:
  - Confidentiality
  - Integrity of messages
  - No key management, and no robust authentication
- WEP mechanisms
  - Challenge response mechanism for authentication (very weak)
  - RC4 used to encrypt packets, based on a key shared between mobile unit and access point (link encryption)
  - Integrity Check Vector (ICV) is appended to the packets, to ensure that they were not modified
- Note: in wireless communication, every communication is point-to-multi-point
  - Can simply intercept packets, without need for spoofing



RC4

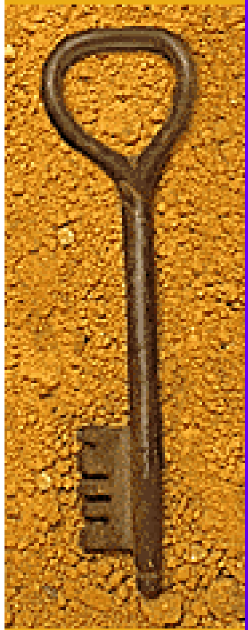
- Developed by RSA, and kept secret till posted on Internet
- Keystream is generated based on initial key, XORed with the plaintext
  - An 8x8 S-box, with all 256 permutations as entries
    - Initial setting based on 256 iterations scrambling the Key
  - In each round, entries are swapped based on the values in other entries
    - i=(i+1) mod 256; j=(j+Si) mod 256; swap Si and Sj
  - One entry, selected based on the values in two other entries, is selected as next one-byte key
    - t=(Si+Sj) mod 256; K=St
- RC4 with 40 bits is exportable
- Also used in Lotus Notes, and SSL





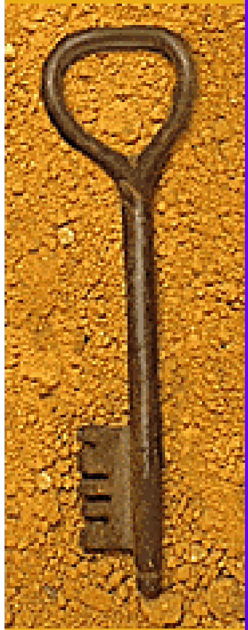
Passive Attacks on WEP's Shortcomings

- A.k.a. "drive-by hacking" or "parking lot attacks"
- Given two ciphertexts, encrypted with same keystream, their XOR will significantly reduce the search space
  - (A XOR K) XOR (B XOR K) = A XOR B
- Keystream depends on key and IV
  - In most implementations, key is 40 bit and IV is 24 bits
  - Key is often shared, so IV is only randomization
- At 11Mbps, IVs are repeated after 5 hours
- Once plaintext is recovered, the key can be obtained from the reverse XOR



Active Attacks on WEP's Shortcomings

- WEP's authentication is based on challenge-response
  - The expected response is the encryption of the challenge
  - But, since the challenge is sent in the clear, with both plaintext and ciphertext, Eve can easily infer keystream and fake her own response
- WEP's ICV is based on Cyclical Redundancy Check (CRC)
  - When modifying content, it is easy to predict the bits that need to be flipped in the CRC
  - Eve can change destination IP address, and have the AP decrypt the packet for her
- Table-based attack: Eve can construct a table of all possible keystreams (2^{24} x 1500 bytes = 24GB)



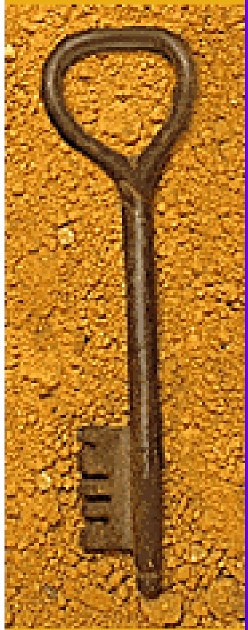
Improving 802.11b Security

- Administrators should use end-to-end encryption
  - Place base stations outside the firewall and use VPN to get inside
  - Use authentication protocols to authenticate remote clients
- WEP and 802.11b are both scrutinized for improvement
  - WEP2 adds to IV space, and uses different and changing keys for different stations, but is suspect due to same vendor interests
  - IEEE's Enhanced Security Network (ESN) will use AES with 128bit keys
  - 802.1X is an authentication protocol that can use multiple auth paradigms (not only for wireless), such as Remote Authentication Dial-In User Service (RADIUS) servers

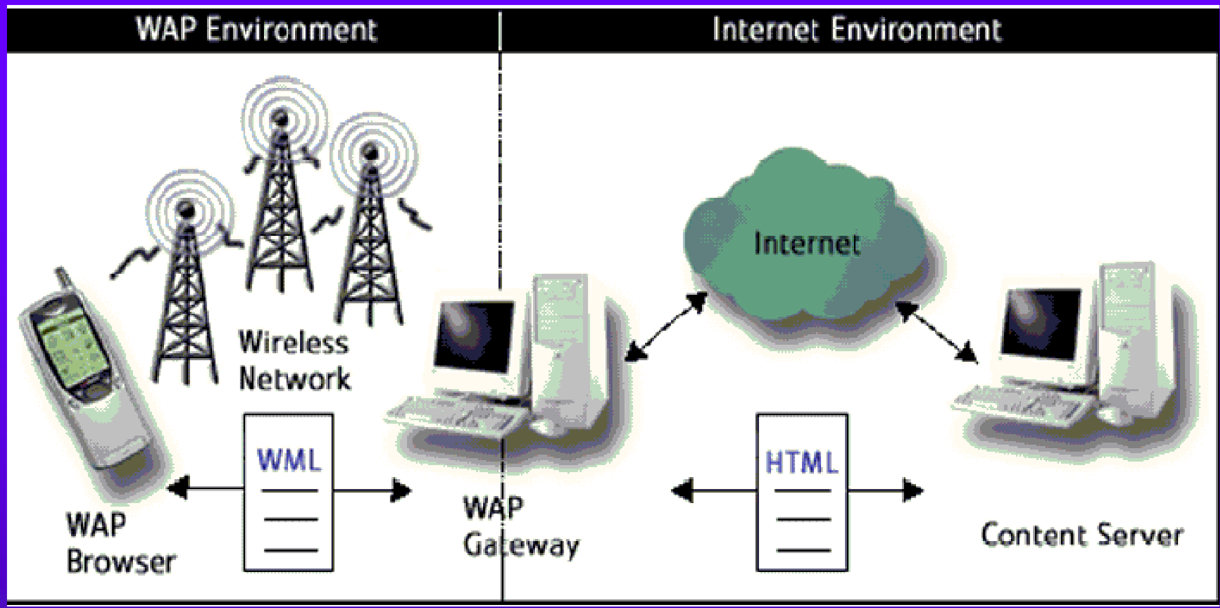
Cellular Security

WAP PKI

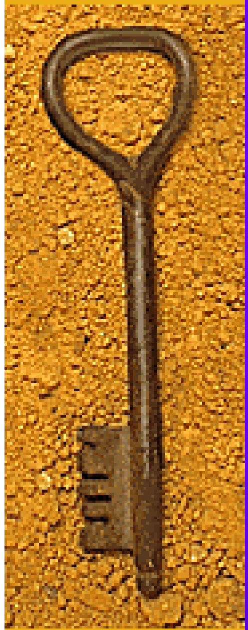
Main Sources: WAP Forum, Certicom



WAP Security Needs



- Main risks (today)
  - Eavesdropping, Impersonation, Malicious Code
  - Interruption
- Needed security services
  - Confidentiality, Authentication, Non-repudiation (m-commerce)
  - Must work in computationally-challenged environment



WAP Security Standards

- WAP Identity Module (WIM)
  - Tamper resistant chip on the handheld that stores key material, typically implemented as a smart card
- WML Script Crypto API (WMLSCrypt)
  - Library of security functions for WAP applications, e.g., key generation and management, encryption, digital signature
  - Elliptic Curve Cryptography (ECC) requires less key material and less computation than traditional public-key encryption algorithms
- Wireless Transport Layer Security (WTLS)
  - Based on TLS, optimized for wireless applications
  - Provides authentication, encapsulation/encryption, integrity check
- WAP Public Key Infrastructure (WPKI)
  - Optimized PKI management of keys and certificates
  - Reduced size certificates