Certificates, Authentication Network Security

Lecture 9

November 18, 2003

Cryptographic Key Infrastructure

- Goal: bind identity to key
- Classical Crypto:
  - Not possible as all keys are shared
- Public key Crypto:
  - Bind identity to public key
  - Crucial as people will use key to communicate with principal whose identity is bound to key
  - Erroneous binding means no secrecy between principals
  - Assume principal identified by an acceptable name

Certificates

- Create token (message) containing
  - Identity of principal (here, Alice)
  - Corresponding public key
  - Timestamp (when issued)
  - Other information (perhaps identity of signer)

signed by trusted authority (here, Cathy)

$$C_A = \{ e_A \,||\, \text{Alice} \,||\, T \}\, d_C$$

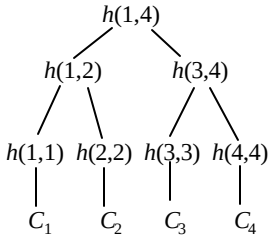
C_A is A's certificate

Use

- Bob gets Alice's certificate
  - If he knows Cathy's public key, he can decipher the certificate
    - When was certificate issued?
    - Is the principal Alice?
  - Now Bob has Alice's public key
- Problem: Bob needs Cathy's public key to validate certificate
  - Merkle's tree,
  - Signature chains

Merkle's Tree Scheme

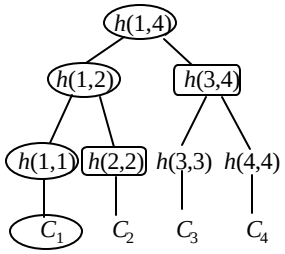
- Keep certificates in a file
  - Changing any certificate changes the file
  - Use crypto hash functions to detect this (data integrity)
- Define hashes recursively
  - h is hash function
  - C_i is certificate for i
- Hash of file ($h(1,4)$ in example) known to all



Details

- f: $D \times D \rightarrow D$ maps bit strings to bit strings
- h: $N \times N \rightarrow D$ maps integers to bit strings
  - if $i = j$, $h(i, j) = f(C_i, C_j)$
  - if $i < j$,
    $h(i, j) = f(h(i, \lfloor (i+j)/2 \rfloor), h(\lfloor (i+j)/2 \rfloor + 1, j))$

Validation



- To validate C_1:
  - Compute $h(1, 1)$
  - Obtain $h(2, 2)$
  - Compute $h(1, 2)$
  - Obtain $h(3, 4)$
  - Compute $h(1,4)$
  - Compare to known $h(1, 4)$
- Need to know hashes of children of nodes on path that are not computed

Problem

- File must be available for validation
  - Otherwise, can't recompute hash at root of tree
  - Intermediate hashes would do
- Not practical in most circumstances
  - Too many certificates and users
  - Users and certificates distributed over widely separated systems

Certificate Signature Chains

- Create certificate
  - Generate hash of certificate
  - Encipher hash with issuer's private key
- Validate
  - Obtain issuer's public key
  - Decipher enciphered hash
  - Recompute hash from certificate and compare
- Problem:
  - Validating the certificate of the issuer and getting issuer's public key

X.509 Chains

- Key certificate fields in X.509v3:
  - Version
  - Serial number (unique)
  - Signature algorithm identifier: hash algorithm
  - Issuer's name; uniquely identifies issuer
  - Interval of validity
  - Subject's name; uniquely identifies subject
  - Subject's public key
  - Signature:
    - Identifies algorithm used to sign the certificate
    - Signature (enciphered hash)

X.509 Certificate Validation

- Obtain issuer's public key
  - The one for the particular signature algorithm
- Decipher signature
  - Gives hash of certificate
- Recompute hash from certificate and compare
  - If they differ, there's a problem
- Check interval of validity
  - This confirms that certificate is current

Issuers

- *Certification Authority (CA)*: entity that issues certificates
  - Multiple issuers pose validation problem
  - Alice's CA is Cathy; Bob's CA is Don; how can Alice validate Bob's certificate?
  - Have Cathy and Don cross-certify
    - Each issues certificate for the other

Validation and Cross-Certifying

- Certificates:
  - Cathy<<Alice>>
    - represents the certificate that C has generated for A
  - Dan<<Bob>>
  - Cathy<<Dan>>
  - Dan<<Cathy>>
- Alice validates Bob's certificate
  - Alice obtains Cathy<<Dan>>
  - Alice uses (known) public key of Cathy to validate Cathy<<Dan>>
  - Alice uses Cathy<<Dan>> to validate Dan<<Bob>>
    - Cathy<<Dan>> Dan<<Bob>> is a signature chain
  - How about Bob validating Alice?

PGP Chains

- Pretty Good Privacy:
  - Widely used to provide privacy for electronic mail
  - Sign files digitally
- OpenPGP certificates structured into packets
  - One public key packet
  - Zero or more signature packets
- Public key packet:
  - Version (3 or 4; 3 compatible with all versions of PGP, 4 not compatible with older versions of PGP)
  - Creation time
  - Validity period (not present in version 3)
  - Public key algorithm, associated parameters
  - Public key

OpenPGP Signature Packet

- Version 3 signature packet
  - Version (3)
  - Signature type (level of trust)
  - Creation time (when next fields hashed)
  - Signer's key identifier (identifies key to encipher hash)
  - Public key algorithm (used to encipher hash)
  - Hash algorithm
  - Part of signed hash (used for quick check)
  - Signature (enciphered hash using signer's private key)

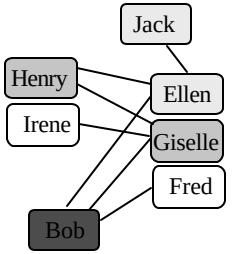
Signing

- Single certificate may have multiple signatures
- Notion of "trust" embedded in each signature
  - Range from "untrusted" to "ultimate trust"
  - Signer defines meaning of trust level (no standards!)
- All version 4 keys signed by subject
  - Called "self-signing"

Validating Certificates

- Alice needs to validate Bob's OpenPGP cert
  - Does not know Fred, Giselle, or Ellen
- Alice gets Giselle's cert
  - Knows Henry slightly, but his signature is at "casual" level of trust
- Alice gets Ellen's cert
  - Knows Jack, so uses his cert to validate Ellen's, then hers to validate Bob's

Arrows show signatures
Self signatures not shown



Authentication and Identity

Courtesy of Professors Chris Clifton & Matt Bishop

What is Authentication?

- Authentication:
  - Binding identity and external entity to subject
- How do we do it?
  - Entity *knows* something (secret)
    - Passwords, id numbers
  - Entity *has* something
    - Badge, smart card
  - Entity *is* something
    - Biometrics: fingerprints or retinal characteristics
  - Entity is in *someplace*
    - Source IP, restricted area terminal

Authentication System: Formal Definition

- *A*: Set of *authentication information*
  - used by entities to prove their identities (e.g., password)
- *C*: Set of *complementary information*
  - used by system to validate authentication information (e.g., hash of a password or the password itself)
- *F*: Set of *complementation functions* (to generate *C)*
  - $f : A \rightarrow C$
  - Generate appropriate $c \in C$ given $a \in A$
- *L*: set of *authentication functions*
  - $l : A \times C \rightarrow \{$ **true, false** $\}$
  - verify identity
- *S*: set of *selection functions*
  - Generate/alter *A* and *C*
  - e.g., commands to change password

Authentication System: Passwords

- Example: plaintext passwords
  - $A = C$ = alphabet*
  - f returns argument: $f(a)$ returns a
  - I is string equivalence: $I(a, b)$ is true if $a = b$
- Complementation Function
  - Null (return the argument as above)
    - requires that c be protected; i.e. password file needs to be protected
  - One-way hash – function such that
    - *Complementary information* $c = f(a)$ easy to compute
    - $f^{-1}(c)$ difficult to compute

Passwords

- Example: Original Unix
  - A password is up to eight characters each character could be one of 127 possible characters;
  - A contains approx. 6.9×10^{16} passwords
  - Password is hashed using one of 4096 functions into a 11 character string
  - 2 characters pre-pended to indicate the hash function used
  - C contains passwords of size 13 characters, each character from an alphabet of 64 characters
    - Approximately 3.0×10^{23} strings
  - Stored in file */etc/passwd* (all can read)

Authentication System

- Goal of (A, C, F, L, S)
  - For all $a \in A$, $c \neq f(a) \in C$
    - $\exists (f, I)$, $f \in F$, $\forall I \in L$ in the system such that
      - $I(a, f(a))$? **true**
      - $I(a, c)$? **false (**with high probability)
- Approaches
  - Hide enough information so that one of a, c or f cannot be found
    - Make C readable only to root (*use shadow password files*)
    - Make F unknown
  - Prevent access to the authentication functions L
    - *root* cannot log in over the network (L exist but fails)

Attacks on Passwords

- Dictionary attack: Trial and error guessing
  - Type 1: attacker knows A, f, c
    - Guess g and compute $f(g)$ for each f in F
  - Type 2: attacker knows A, I
    - I returns **True** for guess g
  - Difficulty based on $|A|$, Time
    - Probability P of breaking in time T
    - G be the number of guesses that can be tested in one time unit
    - $P = TG/|A|$
    - Assumptions: time constant; all passwords are equally likely

Password Selection

- Random
  - Depends on the quality of random number generator; size of legal passwords
  - 8 characters: humans can remember only one
  - Will need to write somewhere
- Pronounceable nonsense
  - Based on unit of sound (phoneme)
    - "Helgoret" vs "pxnftr"
  - Easier to remember
- User selection (proactive selection)
  - Controls on allowable
  - Reasonably good:
    - At least 1 digit, 1 letter, 1 punctuation, 1 control character
    - Obscure poem verse

Password Selection

- Reusable Passwords susceptible to dictionary attack (type 1)
  - *Salting* can be used to increase effort needed
    - makes the choice of complementation function a function of randomly selected data
    - Random data is different for different user
    - Authentication function is chosen on the basis of the salt
    - Many Unix systems:
      - A salt is randomly chosen from 0..4095
      - Complementation function depends on the salt

Password Selection

- Password aging
  - Change password after some time: based on expected time to guess a password
  - Disallow change to previous *n* passwords
- Fundamental problem is *reusability*
  - Replay attack is easy
  - Solution:
    - Authenticate in such a way that the transmitted password changes each time

Authentication Systems: Challenge-Response

- Pass algorithm
  - authenticator sends message *m*
  - subject responds with *f*(*m*)
    - *f* is a secret encryption function
    - In practice: key known only to subject
  - Example: ask for second input based on some algorithm

Authentication Systems: Challenge-Response

- One-time password: *invalidated after use*
  - f changes after use
    - Challenge is the number of authentication attempt
    - Response is the one-time password
- S/Key uses a hash function (MD4/MD5)
  - User chooses an initial seed k
  - Key generator calculates
    - $k_1 = h(k), k_2 = h(k_1)$ …, $k_n = h(k_{n-1})$
  - Passwords used in the order
    - $p_1 = k_n$, $p_2 = k_{n-1}$, …, $p_n = k_1$
  - Suppose $p_1 = k_n$ is intercepted;
    - the next password is $p_2 = k_{n-1}$
    - Since $h(k_{n-1}) = k_n$, the attacker needs to know h to determine the next password

Authentication Systems: Biometrics

- Used for human subject identification based on physical characteristics that are tough to copy
  - Fingerprint (optical scanning)
    - Camera's needed (bulky)
  - Voice
    - Speaker-verification (identity) or speaker-recognition (info content)
  - Iris/retina patterns (unique for each person)
    - Laser beaming is intrusive
  - Face recognition
    - Facial features can make this difficult
  - Keystroke interval/timing/pressure

Attacks on Biometrics

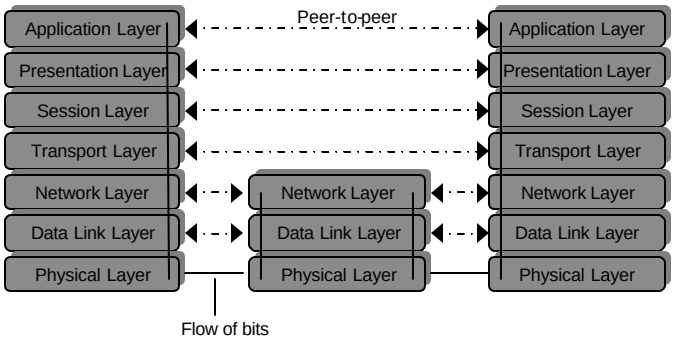
- Fake biometrics
  - fingerprint "mask"
  - copy keystroke pattern
- Fake the interaction between device and system
  - Replay attack
  - Requires careful design of entire authentication system

Authentication Systems: Location

- Based on knowing physical location of subject
- Example: Secured area
  - Assumes separate authentication for subject to enter area
  - In practice: early implementation of challenge/response and biometrics
- What about generalizing this?
  - Assume subject allowed access from limited geographic area
    - I can work from (near) home
  - Issue GPS Smart-Card
  - Authentication tests if smart-card generated signature within spatio/temporal constraints
  - Key: authorized locations known/approved in advance

Network Security

ISO/OSI Model

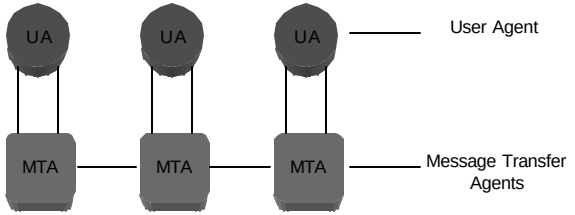


Protocols

- End-to-end protocol
  - Communication protocol that involves end systems with one or more intermediate systems
  - Intermediate host play no part other than forwarding messages
    - Example: telnet
- Link protocol
  - Protocol between every directly connected systems
    - Example: IP – guides messages from a host to one of its immediate host
- Link encryption
  - Encipher messages between intermediate host
  - Each host share a cryptographic key with its neighbor
    - Attackers at the intermediate host will be able to read the message
- End-to-end encryption
  - Example: telnet with messages encrypted/decrypted at the client and server
  - Attackers on the intermediate hosts cannot read the message

Electronic Mail

- UA interacts with the sender
- UA hands it to a MTA
- Attacker can read email on any of the computer with MTA
- Forgery possible



Security at the Application Layer: Privacy-enhanced Electronic Mail (PEM)

- Study by Internet Research Task Force on Privacy or Privacy Research Group to develop protocols with following services
  - Confidentiality, by making the message unreadable except to the sender and recipients
  - Origin authentication, by identifying the sender precisely
  - Data integrity, by ensuring that any changes In the message are easy to detect
  - Non-repudiation of the origin (if possible)

Design Considerations/goals for PEM

- Not to redesign existing mail system protocols
- To be compatible with a range of MTAs, UAs and other computers
- To make privacy enhancements available separately so they are not required
- To enable parties to use the protocol to communicate without prearrangement

PEM Basic Design

- Defines two keys
  - Data Encipherment Key (DEK) to encipher the message sent
    - Generated randomly
    - Used only once
    - Sent to the recipient
  - Interchange key: to encipher DEK
    - Must be obtained some other way than the through the message

Protocols

- Confidential message (DEK: k_s)

Alice $\xrightarrow{\{m\}k_s \,\|\, \{k_s\}k_{Bob}}$ Bob

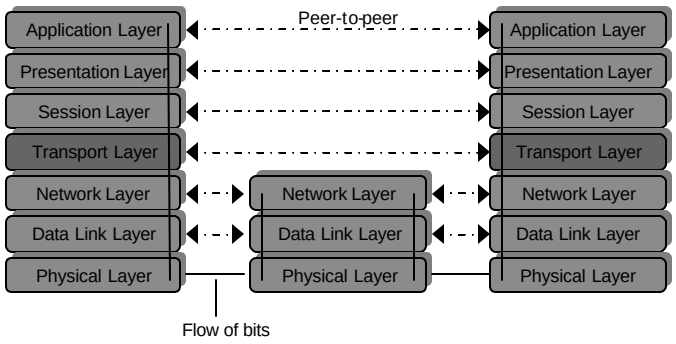
- Authenticated, integrity-checked message

Alice $\xrightarrow{m \,\|\, \{h(m)\}k_{Alice}}$ Bob

- Enciphered, authenticated, integrity checked message

Alice $\xrightarrow{\{m\}k_s \,\|\, \{h(m)\}k_{Alice} \,\|\, \{k_s\}k_{Bob}}$ Bob

ISO/OSI Model
SSL: Security at Transport Layer



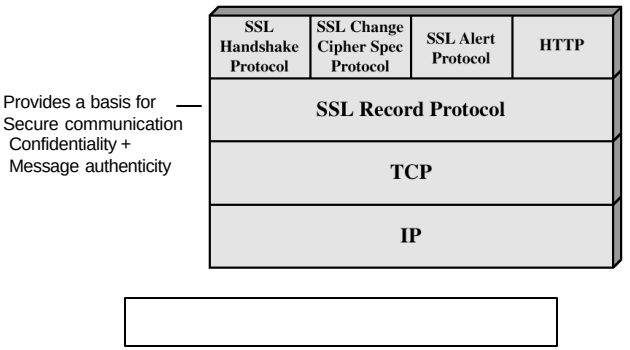
Security at the Transport Layer
Secure Socket Layer (SSL)

- Developed by Netscape to provide security in WWW browsers and servers
- SSL is the basis for the Internet standard protocol – Transport Layer Security (TLS) protocol (compatible with SSLv3)
- Key idea: *Connections* and *Sessions*
  - A SSL session is an association between two peers
  - An SSL connection is the set of mechanisms used to transport data in an SSL session

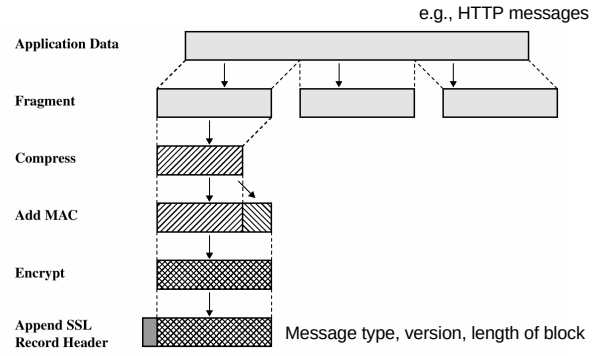
Secure Socket Layer (SSL)

- Each party keeps session information
  - Session identifier (unique)
  - The peer's X.503(v3) certificate
  - Compression method used to reduce volume of data
  - Cipher specification (parameters for cipher and MAC)
  - Master secret of 48 bits
- Connection information
  - Random data for the server & client
  - Server and client keys (used for encryption)
  - Server and client MAC key
  - Initialization vector for the cipher, if needed
  - Server and client sequence numbers
- Provides a set of supported cryptographic mechanisms that are setup during negotiation (handshake protocol)

SSL Architecture



SSL Record Protocol Operation



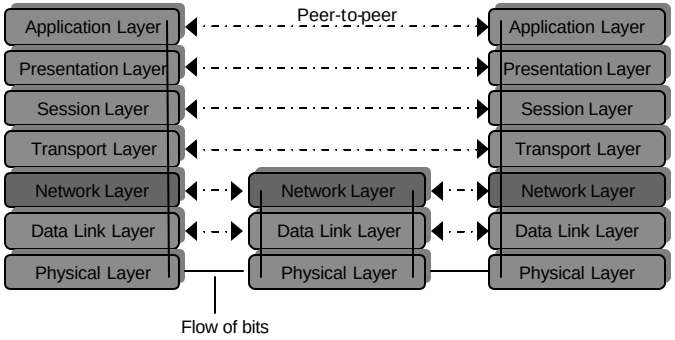
Handshake Protocol

- The most complex part of SSL
- Allows the server and client to authenticate each other
  - Based on interchange cryptosystem (e.g., RSA)
- Negotiate encryption, MAC algorithm and cryptographic keys
  - Four rounds
- Used before any application data are transmitted

Other protocols

- SSL Change Cipher Spec Protocol
  - A single byte is exchanged
  - After new cipher parameters have been negotiated (renegotiated)
- SSL Alert Protocol
  - Signals an unusual condition
  - *Closure alert* : sender will not send anymore
  - *Error alert*: fatal error results in disconnect

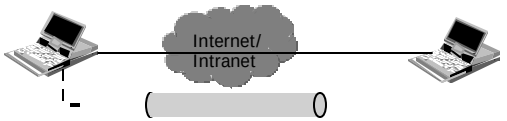
ISO/OSI Model
IPSec: Security at Network Layer



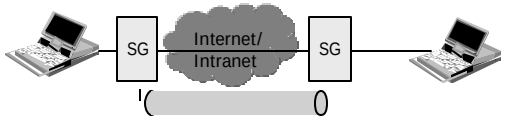
IPSec

- Set of protocols/mechanisms
  - Encrypts and authenticates all traffic at the IP level
    - Protects all messages sent along a path
    - Intermediate host with IPSec mechanism (firewall, gateway) is called a *security gateway*
  - Use on LANs, WANs, public, and private networks
- Application independent (Transparent to user)
  - Web browsing, telnet, ftp…
- Provides at the IP level
  - Access control
  - Connectionless integrity
  - Data origin authentication
  - Rejection of replayed packets
  - Data confidentiality
  - Limited traffic analysis confidentiality

Cases where IPSec can be used

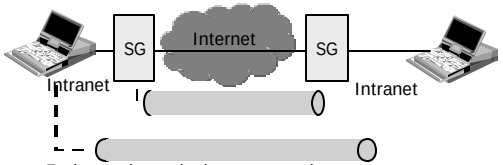


End-to-end security between two hosts

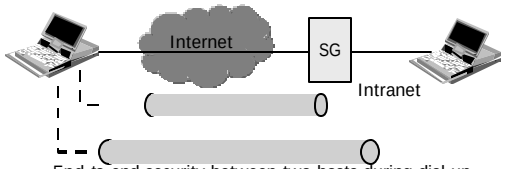


End-to-end security between two security gateways

Cases where IPSec can be used (2)



End-to-end security between two hosts + two gateways



End-to-end security between two hosts during dial-up

IPSec Protocols

- Authentication header (AH) protocol
  - Message integrity
  - Origin authentication
  - Anti -replay services
- Encapsulating security payload (ESP) protocol
  - Confidentiality
  - Message integrity
  - Origin authentication
  - Anti -replay services
- Internet Key Exchange (IKE)
  - Exchanging keys between entities that need to communicate over the Internet
  - What authentication methods to use, how long to use the keys, etc.

Security Association (SA)

- Unidirectional relationship between peers (a sender and a receiver)
- Specifies the security services provided to the traffic carried on the SA
  - Security enhancements to a channel along a path
- Identified by three parameters:
  - IP Destination Address
  - Security Protocol Identifier
    - Specifies whether AH or ESP is being used
  - Security Parameters Index (SPI)
    - Specifies the security parameters associated with the SA

Security Association (2)

- Each SA uses AH or ESP (not both)
  - If both required two SAs are created
- Multiple security associations may be used to provide required security services
  - A sequence of security associations is called *SA bundle*
  - Example: We can have an AH protocol followed by ESP or vice versa

Security Association Databases

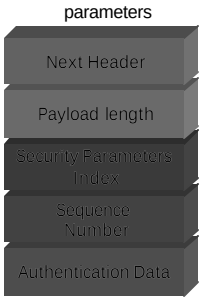
- IP needs to know the SAs that exist in order to provide security services
- Security Policy Database (SPD)
  - IPSec uses SPD to handle messages
  - For each IP packet, it decides whether an IPSec service is provided, bypassed, or if the packet is to be discarded
- Security Association Database (SAD)
  - Keeps track of the sequence number
  - AH information (keys, algorithms, lifetimes)
  - ESP information (keys, algorithms, lifetimes, etc.)
  - Lifetime of the SA
  - Protocol mode
  - MTU

IPSec Modes

- Two modes
  - Transport mode
    - Encapsulates IP packet data area
    - IP Header is not protected
      - Protection is provided for the upper layers
      - Usually used in host-to-host communications
  - Tunnel mode
    - Encapsulates entire IP packet in an IPSec envelope
      - Helps against traffic analysis
      - The original IP packet is untouched in the Internet

Authentication Header (AH)

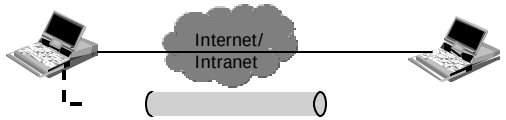
- Next header
  - Identifies what protocol header follows
- Payload length
  - Indicates the number of 32-bit words in the authentication header
- Security Parameters Index
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Authentication Data

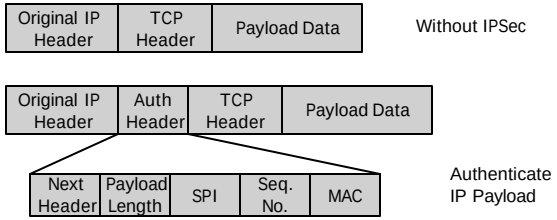


Preventing replay

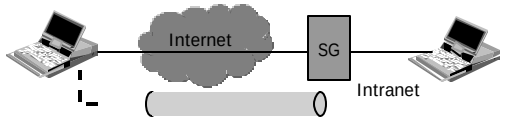
- Using 32 bit sequence numbers helps detect replay of IP packets
- The sender initializes a sequence number for every SA
  - Each succeeding IP packet within a SA increments the sequence number
- Receiver implements a window size of W to keep track of authenticated packets
- Receiver checks the MAC to see if the packet is authentic

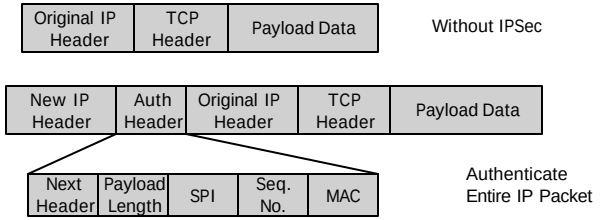
Transport Mode AH





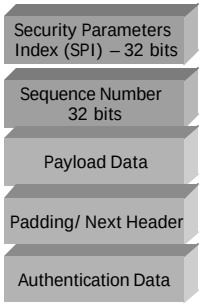
Tunnel Mode AH





ESP – Encapsulating Security Payload

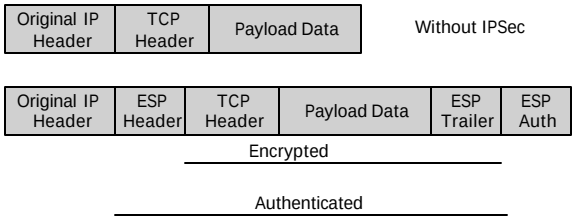
- Creates a new header in addition to the IP header
- Creates a new trailer
- Encrypts the payload data
- Authenticates the security association
- Prevents replay



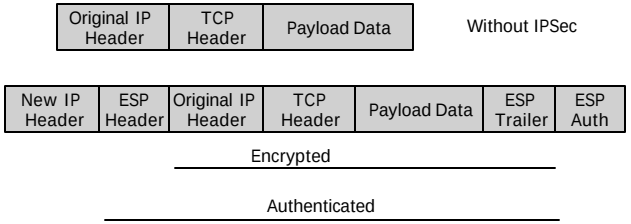
Details of ESP

- Security Parameters Index (SPI)
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Payload
  - Application data carried in the TCP segment
- Padding
  - 0 to 255 bytes of data to enable encryption algorithms to operate properly
  - To mislead sniffers from estimating the amount of data transmitted
- Authentication Data
  - MAC created over the packet

Transport mode ESP



Tunnel mode ESP



Perimeter Defense

- Organization system consists of a network of many host machines –
  - the system is as secure as the weakest link
- Use perimeter defense
  - Define a border and use gatekeeper (firewall)
- If host machines are scattered and need to use public network, use encryption
  - Virtual Private Networks (VPNs)

Perimeter Defense

- Is it adequate?
  - Locating and securing all perimeter points is quite difficult
    - Less effective for large border
  - Inspecting/ensuring that remote connections are adequately protected is difficult
  - Insiders attack is often the most damaging

Firewalls

- Total isolation of networked systems is undesirable
  - Use firewalls to achieve selective border control
- Firewall
  - Is a configuration of machines and software
  - Limits network access
  - Come "for free" inside many devices: routers, modems, wireless base stations etc.
  - Alternate:
    a firewall is a host that mediates access to a network, allowing and disallowing certain type of access based on a configured security policy

What Firewalls can't do

- They are not a panacea
  - Only adds to defense in depth
- If not managed properly
  - Can provide false sense of security
- Cannot prevent insider attack
- Firewalls act a particular layer (or layers)

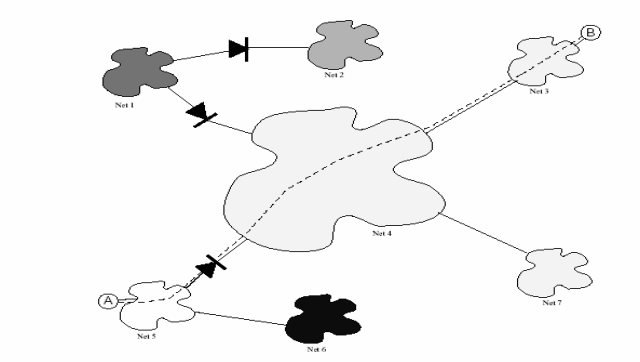
Virtual Private Networks
What is it?

- It is a private network that is configured within a public network
- A VPN "appears" to be a private national or international network to a customer
- The customer is actually "sharing" trunks and other physical infrastructure with other customers
- Security?

What is a VPN? (2)

- A network that supports a *closed* community of authorized users
- There is traffic isolation
  - Contents are secure
  - Services and resources are secure
- Use the public Internet as part of the virtual private network
- Provide security!
  - Confidentiality and integrity of data
  - User authentication
  - Network access control
- IPSec can be used

Tunneling in VPN



"Typical" corporate network