

C. 11. Security Attacks, Countermeasures and protocols

- Intrusion and Intruder
- Password Attacks
- Denial of Service Attacks
- What layer for security
- Firewalls
- Web Security

Intruders

- significant issue for networked systems is hostile or unwanted access
- either via network or local
- can identify classes of intruders:
 - masquerader
 - misfeasor
 - clandestine user
- varying levels of competence

Intruders

- clearly a growing publicized problem
 - from “Wily Hacker” in 1986/87
 - to clearly escalating CERT stats
- may seem benign, but still cost resources
- may use compromised system to launch other attacks
- awareness of intruders has led to the development of CERTs

Intruders: What Can They Do?

- Eavesdrop--(compromise routers, links, routing algorithms, or DNS)
- Send arbitrary messages (including IP hdr)
- Replay recorded messages
- Modify messages in transit
- Write malicious code and trick people into running it

Intrusion Techniques

- aim to gain access and/or increase privileges on a system
- basic attack methodology
 - target acquisition and information gathering
 - initial access
 - privilege escalation
 - covering tracks
- key goal often is to acquire passwords
- so then exercise access rights of owner

Intrusion Detection

- inevitably will have security failures
- so need also to detect intrusions so can
 - block if detected quickly
 - act as deterrent
 - collect info to improve security
- assume intruder will behave differently to a legitimate user
 - but will have imperfect distinction between

Approaches to Intrusion Detection

- statistical anomaly detection
 - threshold
 - profile based
- rule-based detection
 - anomaly
 - penetration identification

Audit Records

- fundamental tool for intrusion detection
- native audit records
 - part of all common multi-user O/S
 - already present for use
 - may not have info wanted in desired form
- detection-specific audit records
 - created specifically to collect wanted info
 - at cost of additional overhead on system

Statistical Anomaly Detection

- threshold detection
 - count occurrences of specific event over time
 - if exceed reasonable value assume intrusion
 - alone is a crude & ineffective detector
- profile based
 - characterize past behavior of users
 - detect significant deviations from this
 - profile usually multi-parameter

Audit Record Analysis

- foundation of statistical approaches
- analyze records to get metrics over time
 - counter, gauge, interval timer, resource use
- use various tests on these to determine if current behavior is acceptable
 - mean & standard deviation, multivariate, markov process, time series, operational
- key advantage is no prior knowledge used

Rule-Based Intrusion Detection

- observe events on system & apply rules to decide if activity is suspicious or not
- rule-based anomaly detection
 - analyze historical audit records to identify usage patterns & auto-generate rules for them
 - then observe current behavior & match against rules to see if conforms
 - like statistical anomaly detection does not require prior knowledge of security flaws

Rule-Based Intrusion Detection

- rule-based penetration identification
 - uses expert systems technology
 - with rules identifying known penetration, weakness patterns, or suspicious behavior
 - compare audit records or states against rules
 - rules usually machine & O/S specific
 - rules are generated by experts who interview & codify knowledge of security admins
 - quality depends on how well this is done

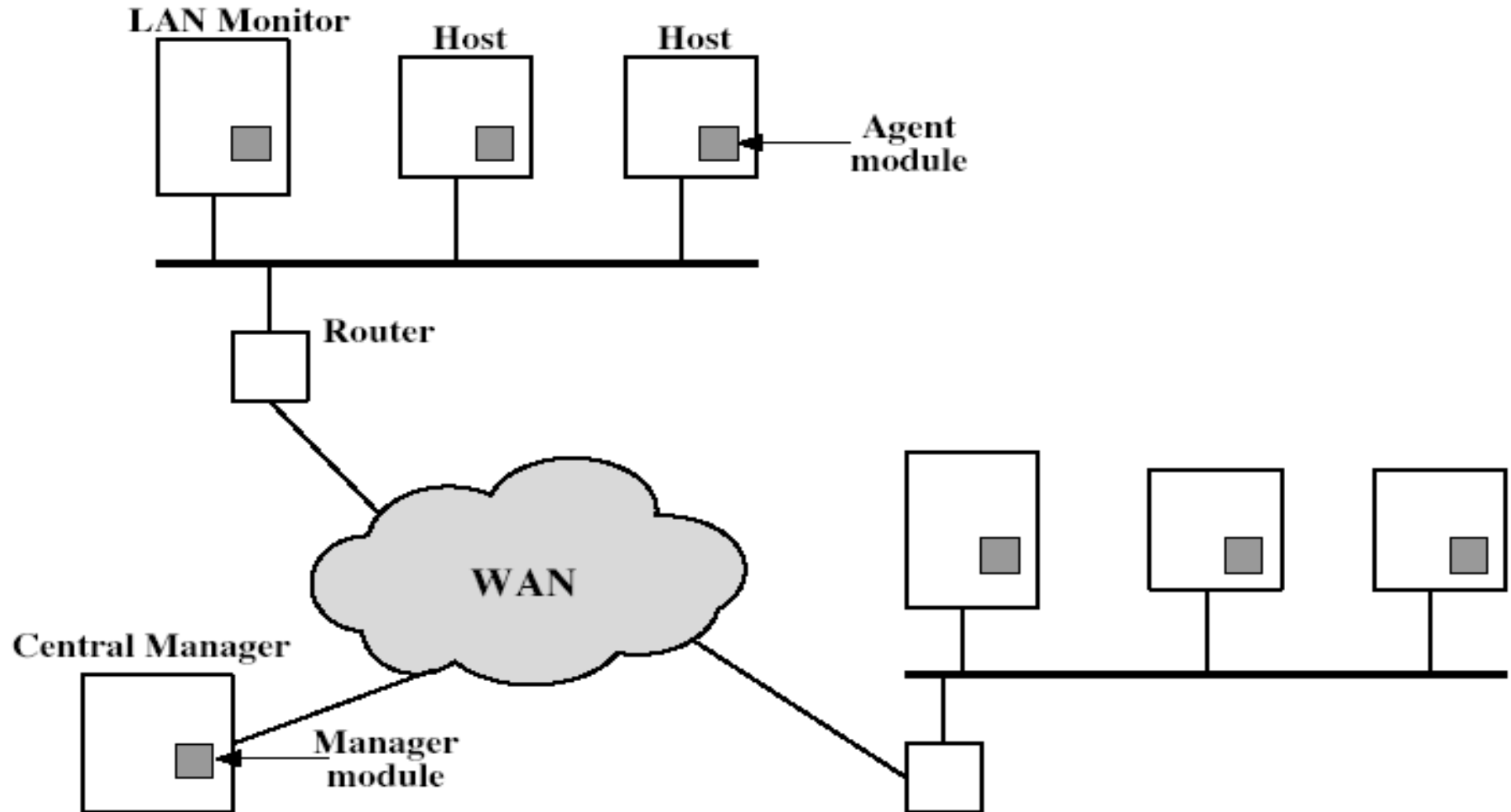
Base-Rate Fallacy

- practically an intrusion detection system needs to detect a substantial percentage of intrusions with few false alarms
 - if too few intrusions detected -> false security
 - if too many false alarms -> ignore / waste time
- this is very hard to do
- existing systems seem not to have a good record

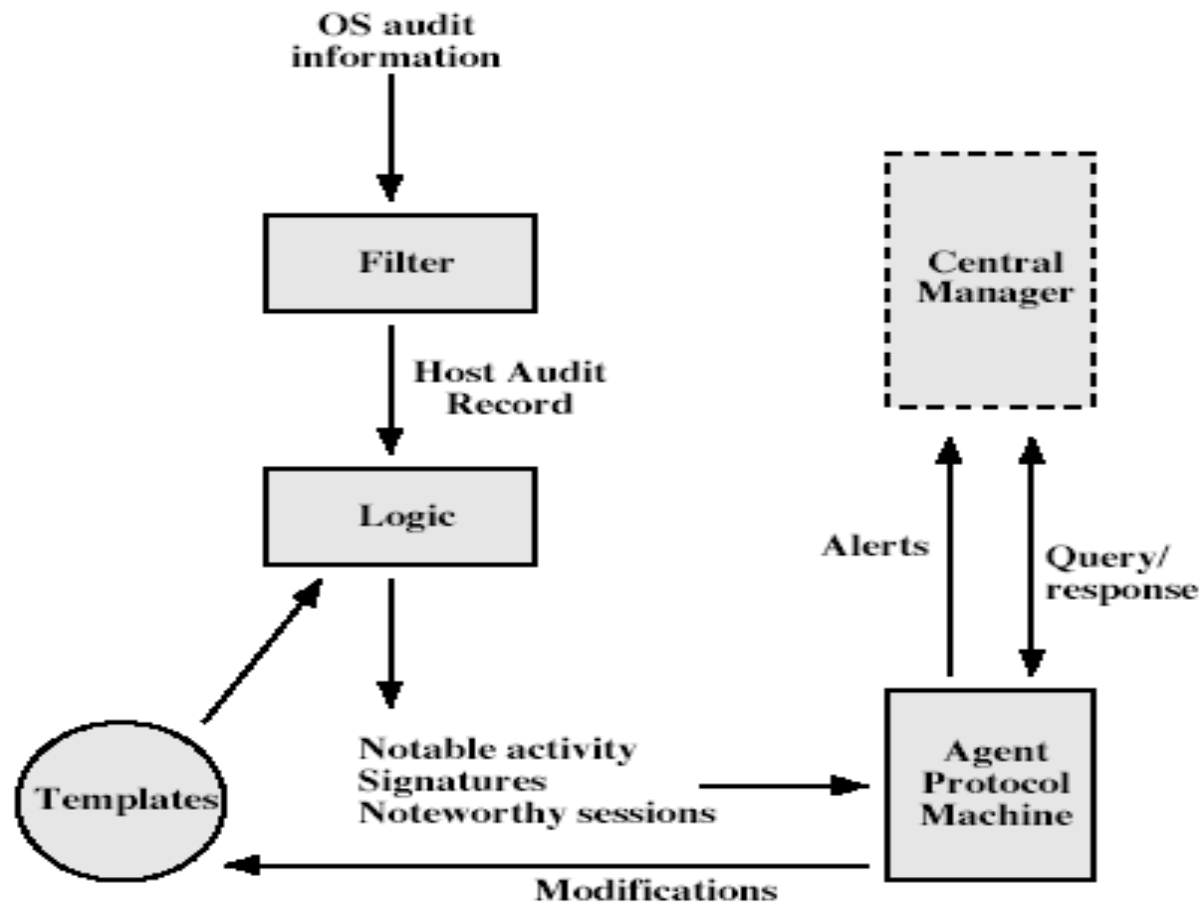
Distributed Intrusion Detection

- traditional focus is on single systems
- but typically have networked systems
- more effective defense has these working together to detect intrusions
- issues
 - dealing with varying audit record formats
 - integrity & confidentiality of networked data
 - centralized or decentralized architecture

Distributed Intrusion Detection - Architecture



Distributed Intrusion Detection – Agent Implementation



Honeypots

- decoy systems to lure attackers
 - away from accessing critical systems
 - to collect information of their activities
 - to encourage attacker to stay on system so administrator can respond
- are filled with fabricated information
- instrumented to collect detailed information on attackers activities
- single or multiple networked systems
- cf IETF Intrusion Detection WG standards

Password Guessing

- one of the most common attacks
- attacker knows a login (from email/web page etc)
- then attempts to guess password for it
 - defaults, short passwords, common word searches
 - user info (variations on names, birthday, phone, common words/interests)
 - exhaustively searching all possible passwords
- check by login or against stolen password file
- success depends on password chosen by user
- surveys show many users choose poorly

Password Capture

- another attack involves **password capture**
 - watching over shoulder as password is entered
 - using a trojan horse program to collect
 - monitoring an insecure network login
 - eg. telnet, FTP, web, email
 - extracting recorded info after successful login (web history/cache, last number dialed etc)
- using valid login/password can impersonate user
- users need to be educated to use suitable precautions/countermeasures

Password Management

- front-line defense against intruders
- users supply both:
 - login – determines privileges of that user
 - password – to identify them
- passwords often stored encrypted
 - Unix uses multiple DES (variant with salt)
 - more recent systems use crypto hash function
- should protect password file on system

Password Studies

- Purdue 1992 - many short passwords
- Klein 1990 - many guessable passwords
- conclusion is that users choose poor passwords too often
- need some approach to counter this

Managing Passwords - Education

- can use policies and good user education
- educate on importance of good passwords
- give guidelines for good passwords
 - minimum length (>6)
 - require a mix of upper & lower case letters, numbers, punctuation
 - not dictionary words
- but likely to be ignored by many users

Managing Passwords - Computer Generated

- let computer create passwords
- if random likely not memorisable, so will be written down (sticky label syndrome)
- even pronounceable not remembered
- have history of poor user acceptance
- FIPS PUB 181 one of best generators
 - has both description & sample code
 - generates words from concatenating random pronounceable syllables

Managing Passwords - Reactive Checking

- reactively run password guessing tools
 - note that good dictionaries exist for almost any language/interest group
- cracked passwords are disabled
- but is resource intensive
- bad passwords are vulnerable till found

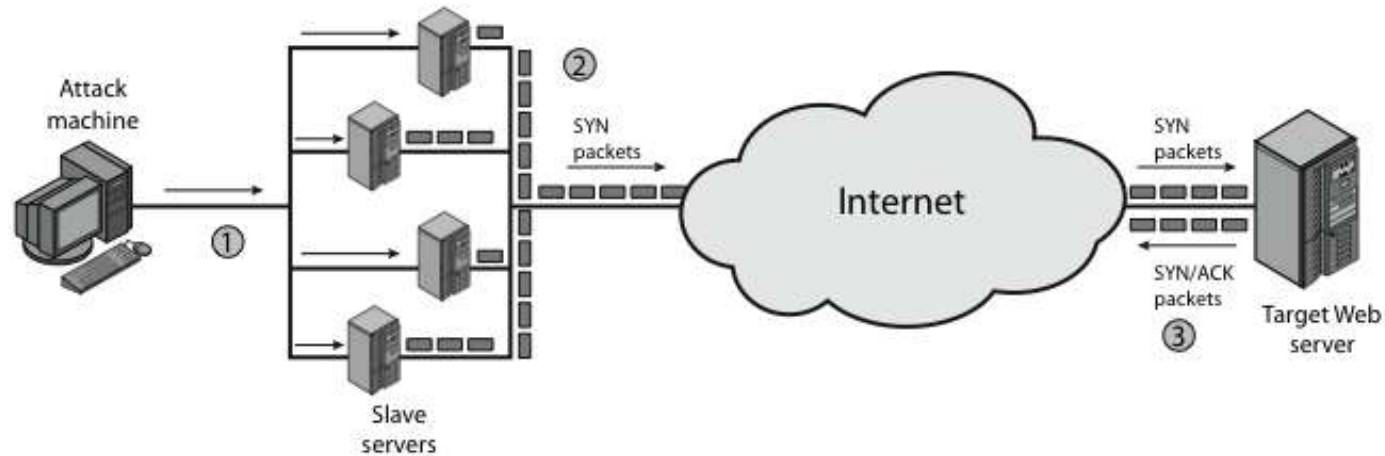
Managing Passwords - Proactive Checking

- most promising approach to improving password security
- allow users to select own password
- but have system verify it is acceptable
 - simple rule enforcement (see earlier slide)
 - compare against dictionary of bad passwords
 - use algorithmic (markov model or bloom filter) to detect poor choices

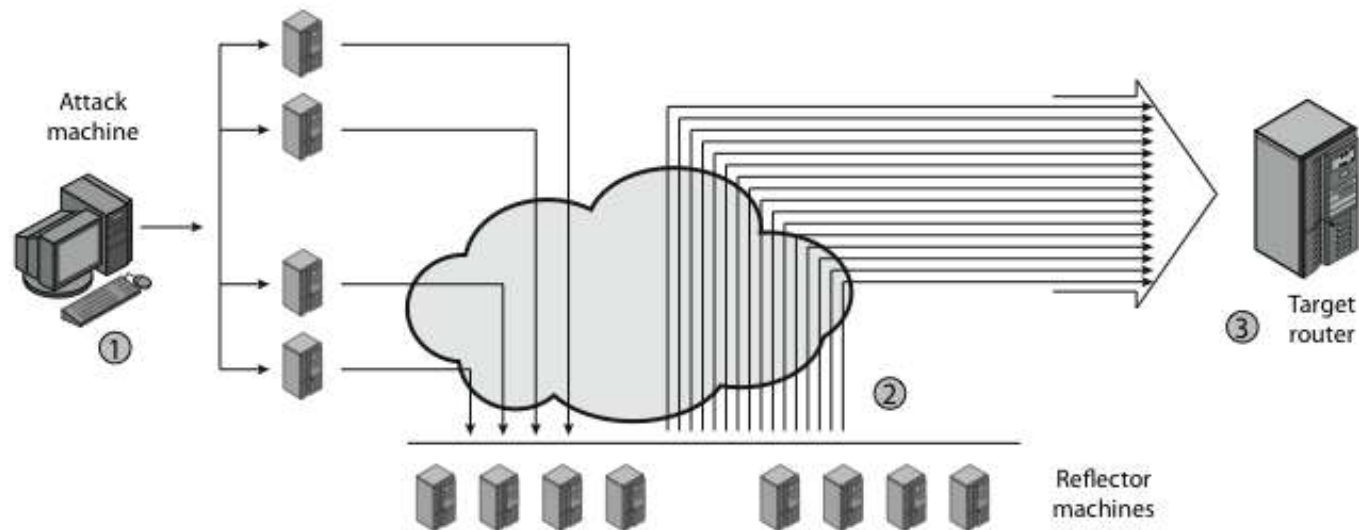
Distributed Denial of Service Attacks (DDoS)

- Distributed Denial of Service (DDoS) attacks form a significant security threat
- making networked systems unavailable
- by flooding with useless traffic
- using large numbers of “zombies”
- growing sophistication of attacks
- defense technologies struggling to cope

Distributed Denial of Service Attacks (DDoS)



(a) Distributed SYN flood attack



(a) Distributed ICMP attack

Constructing the DDoS Attack Network

- must infect large number of zombies
- needs:
 1. software to implement the DDoS attack
 2. an unpatched vulnerability on many systems
 3. scanning strategy to find vulnerable systems
 - random, hit-list, topological, local subnet

DDoS Countermeasures

- three broad lines of defense:
 1. attack prevention & preemption (before)
 2. attack detection & filtering (during)
 3. attack source traceback & ident (after)
- huge range of attack possibilities
- hence evolving countermeasures

Communication Security, What layer?

- Layer 2
 - protects link hop-by-hop
 - IP headers can be hidden from eavesdropper (protects against “traffic analysis”)
- Layer 3/4 (more on next slide)
 - protects end-to-end real-time conversation
- Upper layer (e.g., PGP, S/MIME)
 - protects msgs. Store/forward, not real-time

“Key Exchange”

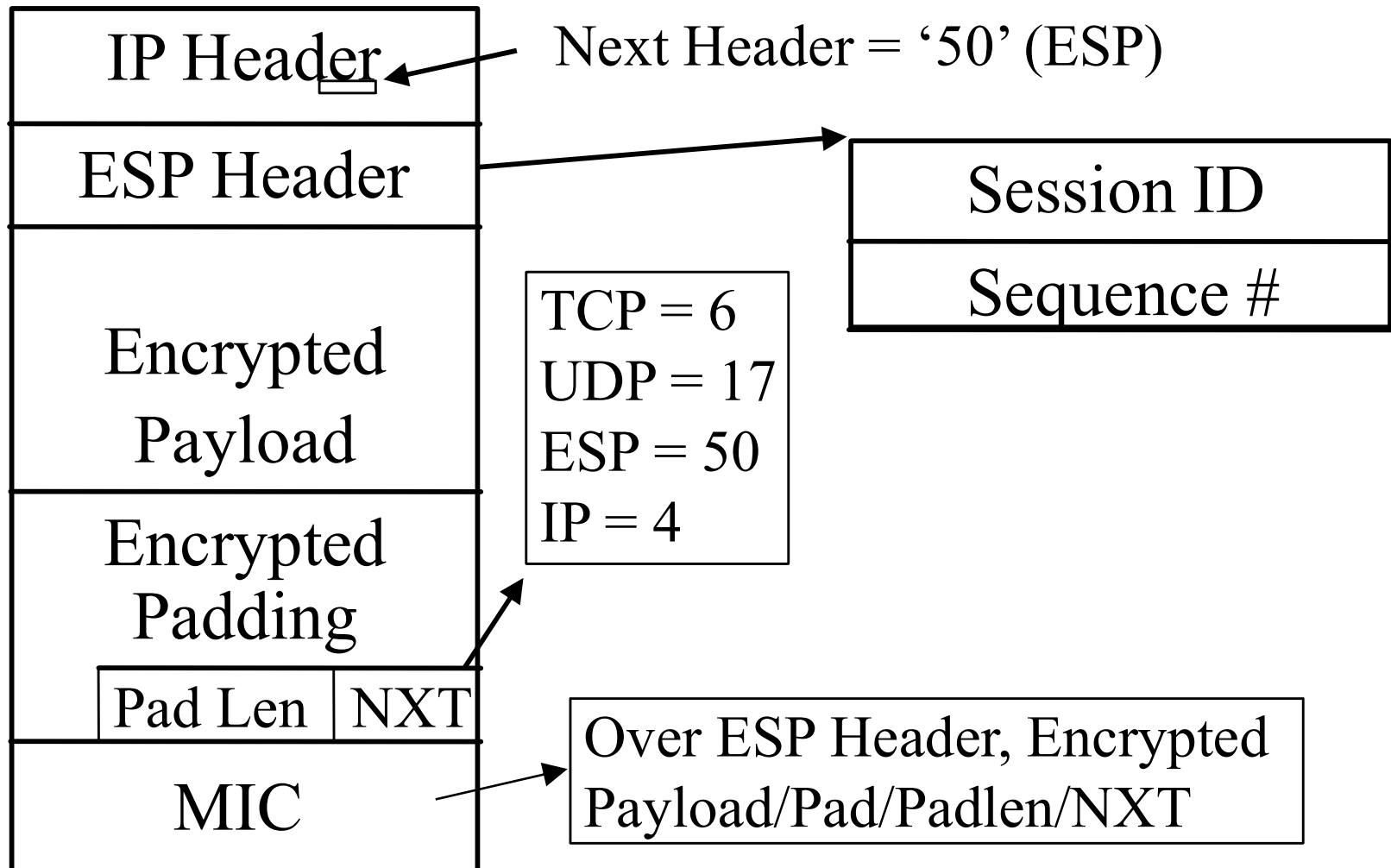
- Mutual authentication/session key creation (create “security association”)
- Good to cryptographically protect entire session (not just initial authentication)
- Good to have new key for each session
- Examples
 - SSL/TLS or Secure Shell (“layer 4”)
 - IPsec (“layer 3”)

Layer 3 vs layer 4

- Layer 3 idea: don't change applications or API to applications, just OS
- layer 4 idea: don't change OS, only change application. They run on top of layer 4 (TCP/UDP)

ESP

Encapsulating Security Payload



Layer 3 vs layer 4

- layer 3 technically superior
 - Rogue packet problem
 - TCP doesn't participate in crypto, so attacker can inject bogus packet, no way for TCP to recover
 - easier to do outboard hardware processing (since each packet independently encrypted)
- layer 4 easier to deploy
- And unless API changes, layer 3 can't pass up authenticated identity

What's going on in IETF Security Area

- Kerberos
- PKIX (certificate format) (see next slide)
- S/MIME, PGP
- IPsec, SSL/TLS, Secure Shell
- SASL (syntax for negotiating auth protocol)
- DNSSEC (public keys, signed data in DNS)
- sacred (downloading credentials)

Examples

- SNMP
- Should be straightforward end-to-end security
- But it has to work when the network is flaky
 - DNS not available
 - LDAP database for retrieving certificates might be down, as might revocation infrastructure

Firewall

- seen evolution of information systems
- now everyone want to be on the Internet
- and to interconnect networks
- has persistent security concerns
 - can't easily secure every system in org
- typically use a **Firewall**
- to provide **perimeter defence**
- as part of comprehensive security strategy

What is a Firewall?

- a **choke point** of control and monitoring
- interconnects networks with differing trust
- imposes restrictions on network services
 - only authorized traffic is allowed
- auditing and controlling access
 - can implement alarms for abnormal behavior
- provide NAT & usage monitoring
- implement VPNs using IPSec
- must be immune to penetration

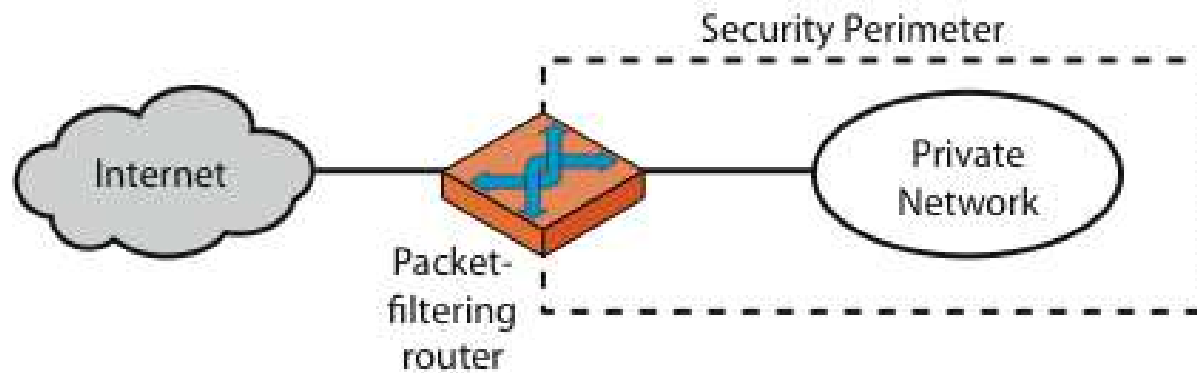
Firewall Limitations

- cannot protect from attacks bypassing it
 - eg sneaker net, utility modems, trusted organisations, trusted services (eg SSL/SSH)
- cannot protect against internal threats
 - eg disgruntled or colluding employees
- cannot protect against transfer of all virus infected programs or files
 - because of huge range of O/S & file types

Firewalls – Packet Filters

- simplest, fastest firewall component
- foundation of any firewall system
- examine each IP packet (no context) and permit or deny according to rules
- hence restrict access to services (ports)
- possible default policies
 - that not expressly permitted is prohibited
 - that not expressly prohibited is permitted

Firewalls – Packet Filters



(a) Packet-filtering router

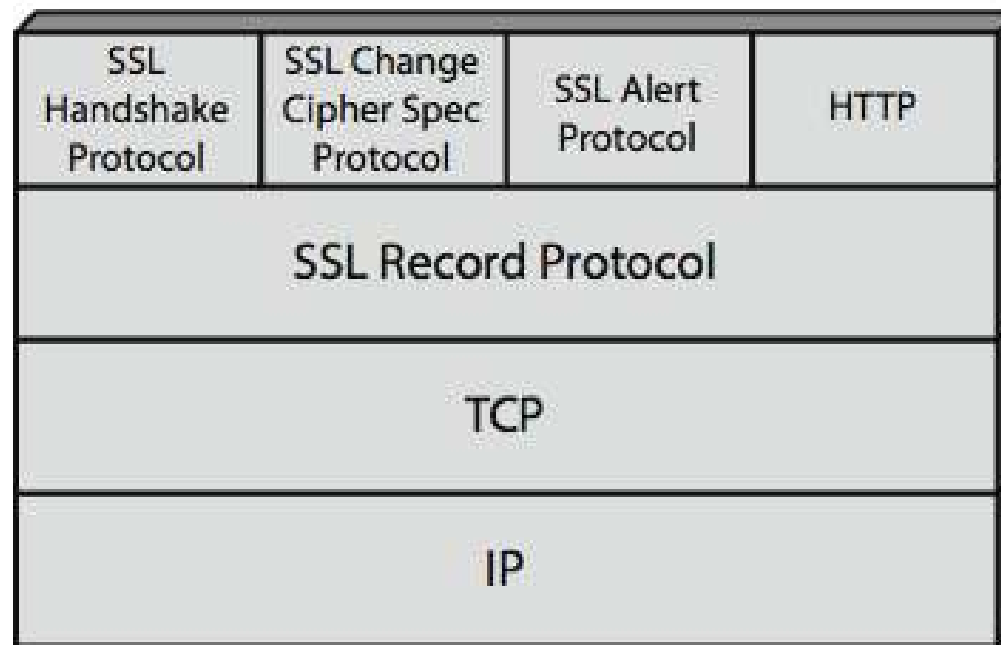
Web Security

- Web now widely used by business, government, individuals
- but Internet & Web are vulnerable
- have a variety of threats
 - integrity
 - confidentiality
 - denial of service
 - authentication
- need added security mechanisms

SSL (Secure Socket Layer)

- transport layer security service
- originally developed by Netscape
- version 3 designed with public input
- subsequently became Internet standard known as TLS (Transport Layer Security)
- uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL Architecture



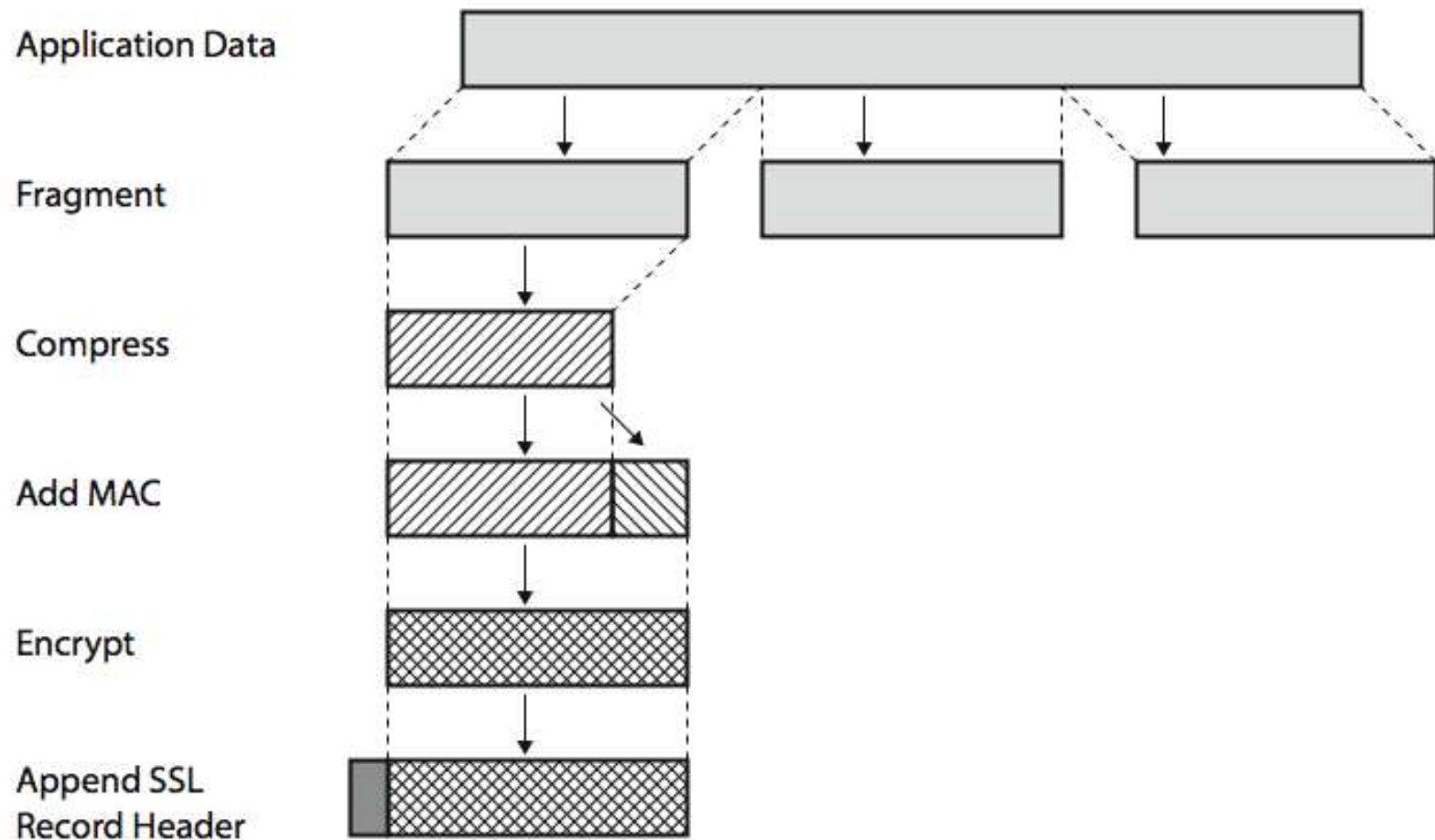
SSL Architecture

- **SSL connection**
 - a transient, peer-to-peer, communications link
 - associated with 1 SSL session
- **SSL session**
 - an association between client & server
 - created by the Handshake Protocol
 - define a set of cryptographic parameters
 - may be shared by multiple SSL connections

SSL Record Protocol Services

- **message integrity**
 - using a MAC with shared secret key
 - similar to HMAC but with different padding
- **confidentiality**
 - using symmetric encryption with a shared secret key defined by Handshake Protocol
 - AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
 - message is compressed before encryption

SSL Record Protocol Operation



SSL Change Cipher Spec Protocol

- one of 3 SSL specific protocols which use the SSL Record protocol
- a single message
- causes pending state to become current
- hence updating the cipher suite in use

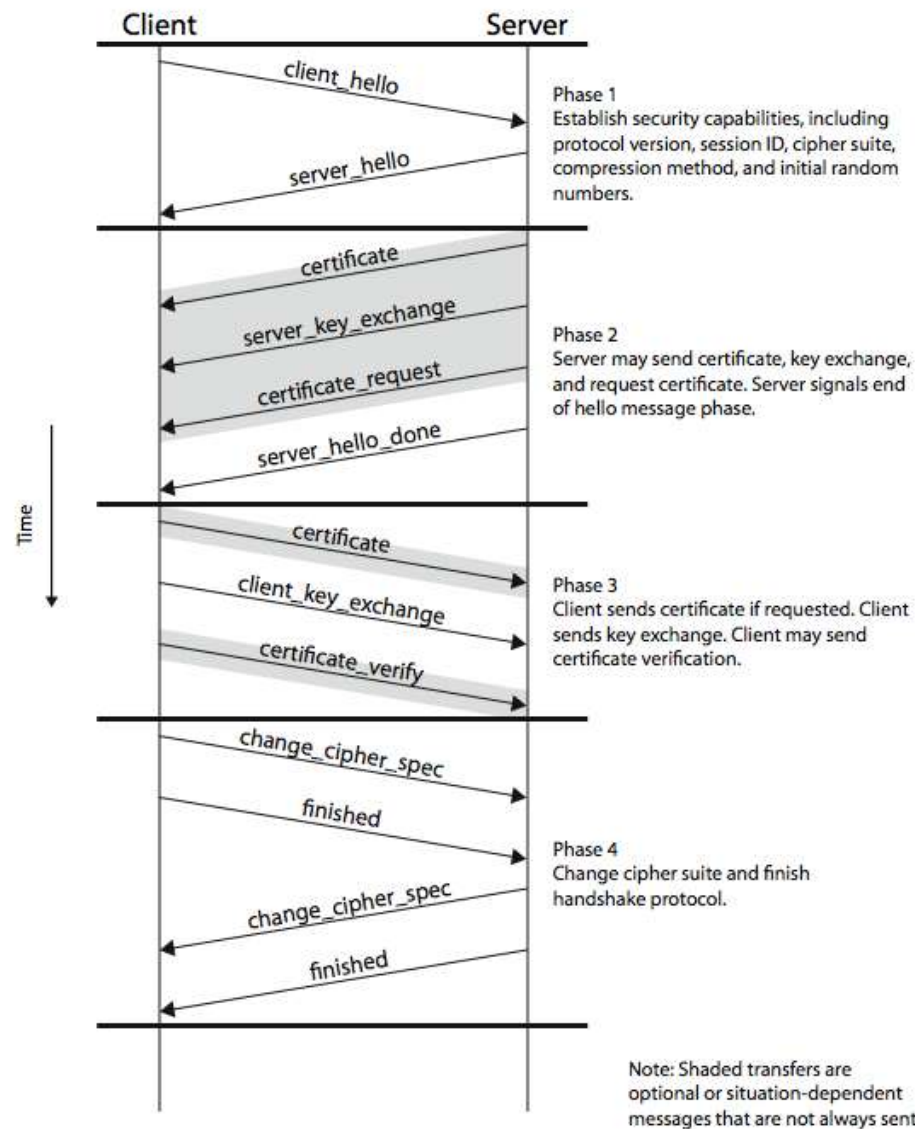
SSL Alert Protocol

- conveys SSL-related alerts to peer entity
- severity
 - warning or fatal
- specific alert
 - fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
 - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- compressed & encrypted like all SSL data

SSL Handshake Protocol

- allows server & client to:
 - authenticate each other
 - to negotiate encryption & MAC algorithms
 - to negotiate cryptographic keys to be used
- comprises a series of messages in phases
 1. Establish Security Capabilities
 2. Server Authentication and Key Exchange
 3. Client Authentication and Key Exchange
 4. Finish

SSL Handshake Protocol



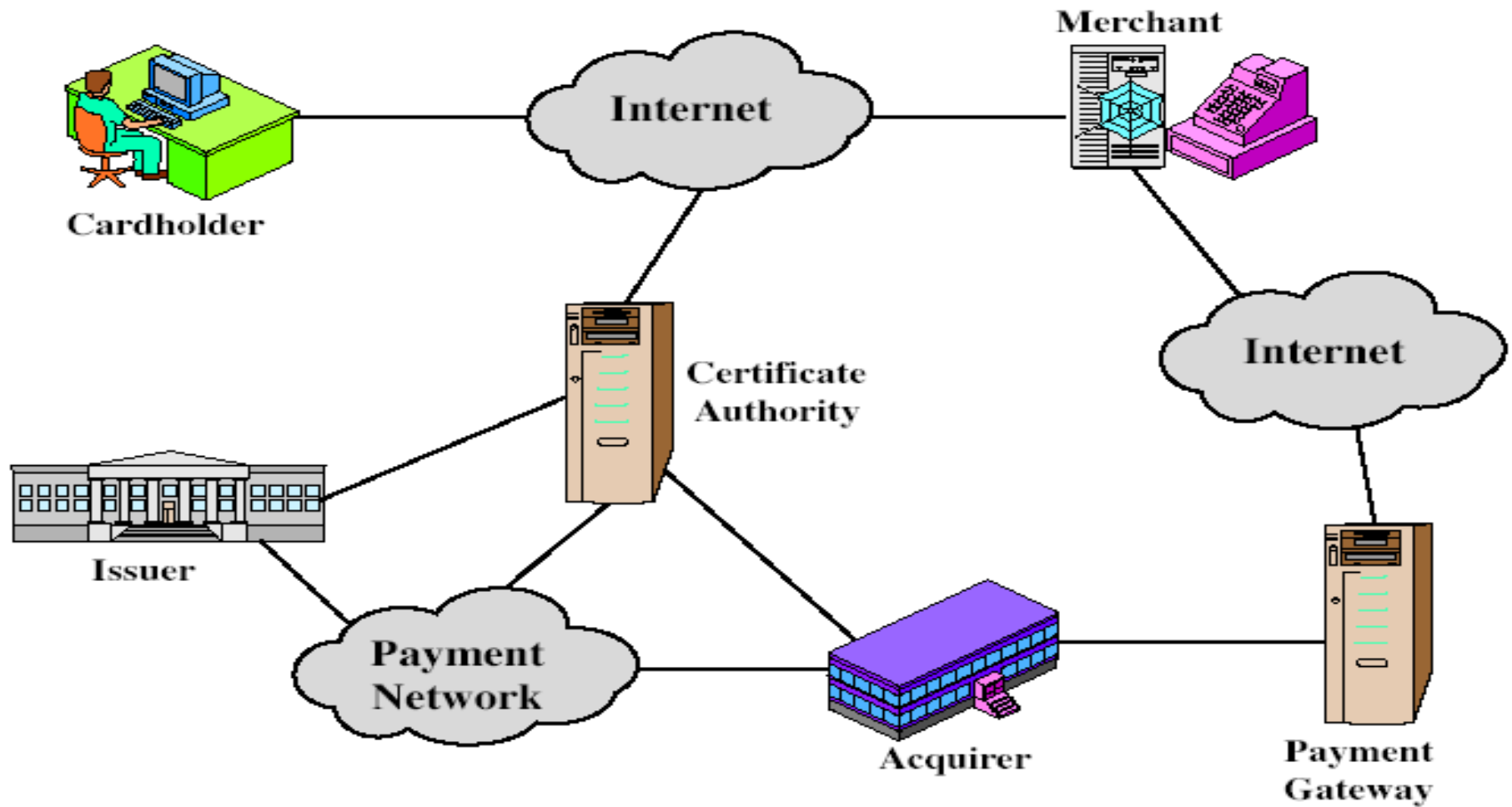
TLS (Transport Layer Security)

- IETF standard RFC 2246 similar to SSLv3
- with minor differences
 - in record format version number
 - uses HMAC for MAC
 - a pseudo-random function expands secrets
 - has additional alert codes
 - some changes in supported ciphers
 - changes in certificate types & negotiations
 - changes in crypto computations & padding

Secure Electronic Transactions (SET)

- open encryption & security specification
- to protect Internet credit card transactions
- developed in 1996 by Mastercard, Visa etc
- not a payment system
- rather a set of security protocols & formats
 - secure communications amongst parties
 - trust from use of X.509v3 certificates
 - privacy by restricted info to those who need it

SET Components



SET Transaction

1. customer opens account
2. customer receives a certificate
3. merchants have their own certificates
4. customer places an order
5. merchant is verified
6. order and payment are sent
7. merchant requests payment authorization
8. merchant confirms order
9. merchant provides goods or service
10. merchant requests payment

Dual Signature

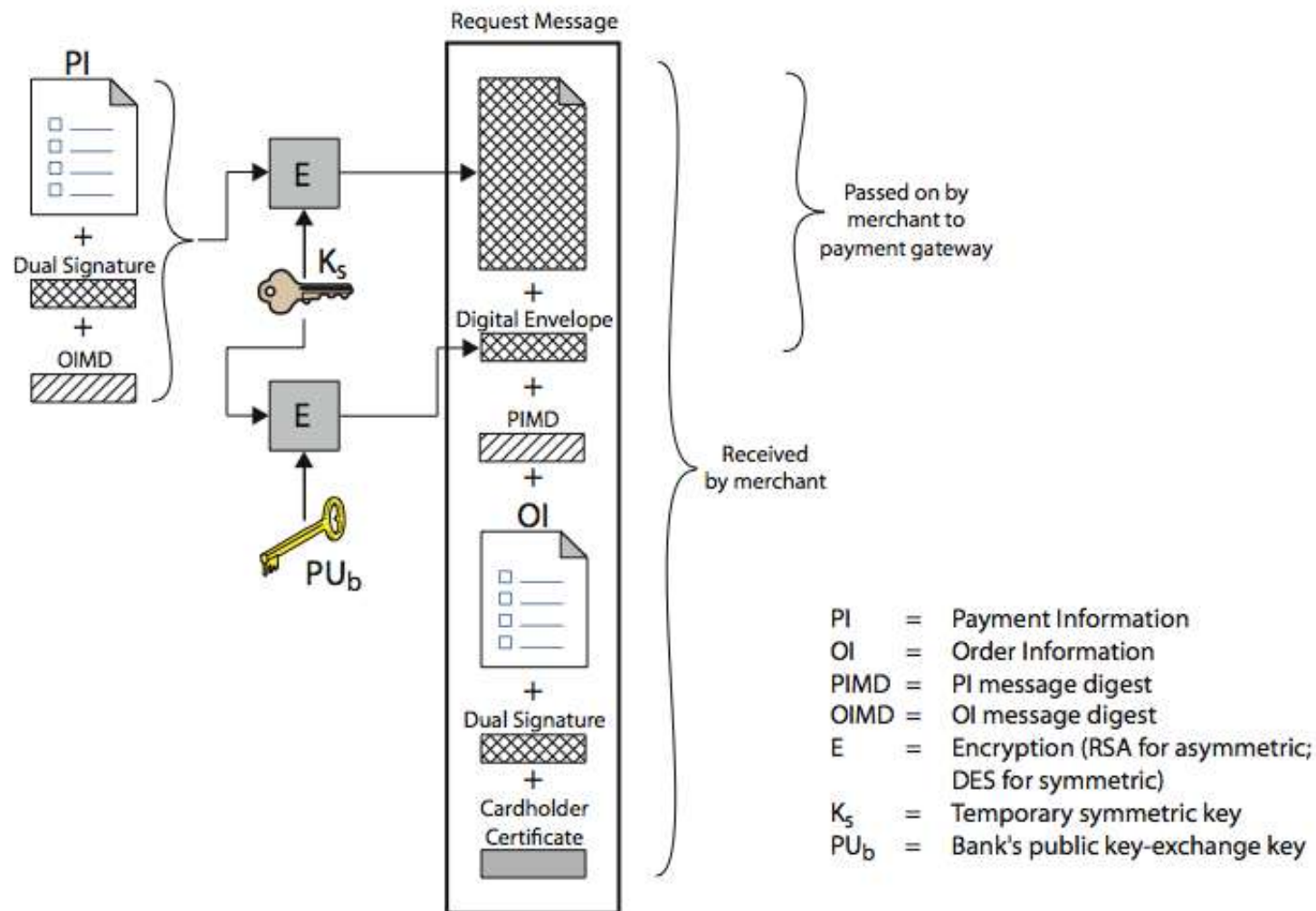
- customer creates dual messages
 - order information (OI) for merchant
 - payment information (PI) for bank
- neither party needs details of other
- but **must** know they are linked
- use a dual signature for this
 - signed concatenated hashes of OI & PI

$$DS = E(PR_c, [H(H(PI) || H(OI))])$$

SET Purchase Request

- SET purchase request exchange consists of four messages
 1. Initiate Request - get certificates
 2. Initiate Response - signed response
 3. Purchase Request - of OI & PI
 4. Purchase Response - ack order

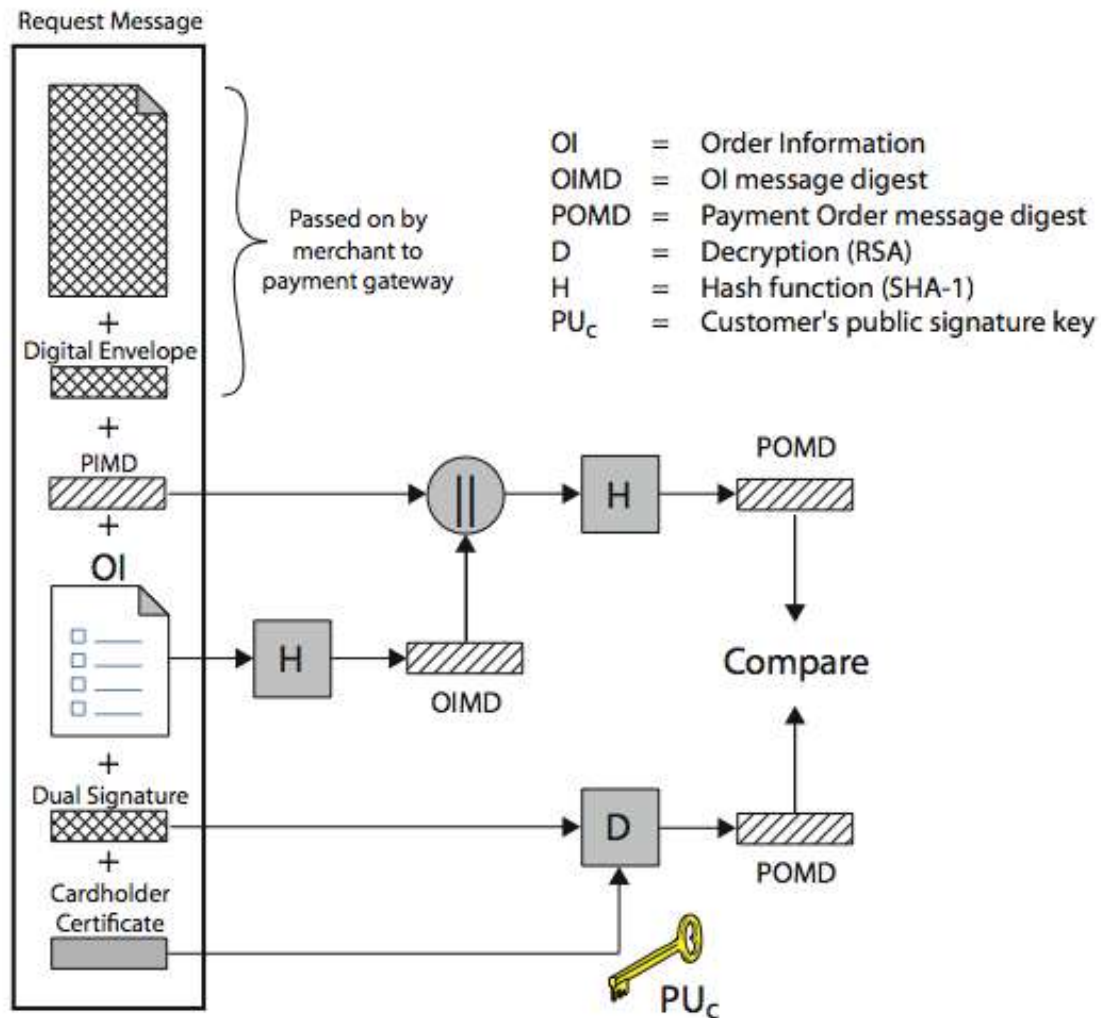
Purchase Request – Customer



Purchase Request – Merchant

1. verifies cardholder certificates using CA sigs
2. verifies dual signature using customer's public signature key to ensure order has not been tampered with in transit & that it was signed using cardholder's private signature key
3. processes order and forwards the payment information to the payment gateway for authorization (described later)
4. sends a purchase response to cardholder

Purchase Request – Merchant



Payment Gateway Authorization

1. verifies all certificates
2. decrypts digital envelope of authorization block to obtain symmetric key & then decrypts authorization block
3. verifies merchant's signature on authorization block
4. decrypts digital envelope of payment block to obtain symmetric key & then decrypts payment block
5. verifies dual signature on payment block
6. verifies that transaction ID received from merchant matches that in PI received (indirectly) from customer
7. requests & receives an authorization from issuer
8. sends authorization response back to merchant

Payment Capture

- merchant sends payment gateway a payment capture request
- gateway checks request
- then causes funds to be transferred to merchants account
- notifies merchant using capture response

Summary

- have considered:
 - need for web security
 - SSL/TLS transport layer security protocols
 - SET secure credit card payment protocols