ECE 455: CYBERSECURITY

Lecture #6

Daniel Gitzel

Announcement

- **Read papers for quiz after midterm.**
- **Finish Lab 1 (parts 3 and 4)**
- **Start work on final project.**
  - We'll have a check-in on 11/4.

KEY ESTABLISHMENT

Introduction

- **Crypto transforms security problems into key management problems**
  - **Don’t lose that key!**
- **To use encryption, digital signatures, or MACs, we have to hold the “right” cryptographic keys.**
  - Public key algorithms need authentic public keys.
  - Symmetric key algorithms need shared secret keys.

Session Keys

- **Public key algorithms are more expensive than symmetric key algorithms.**
  - Cost factors: key length, computation time, bandwidth
- **It is desirable to use long-term keys only sparingly to reduce the "attack surface".**
  - Potential problem: attacks that collect a large amount of encrypted material.
  - Solution: long-term keys establish short term session keys.

Key Usage

- **Good practice: restrict the use of keys to a specific purpose.**
- **In key management, we may use key encrypting keys and data encrypting keys.**
- **Examples for key usages:**
  - Encryption or Decryption
  - Signature or Non-repudiation
  - Master key or Transaction key
- **Don’t use a single key pair for both encryption and signatures!**

Key Establishment

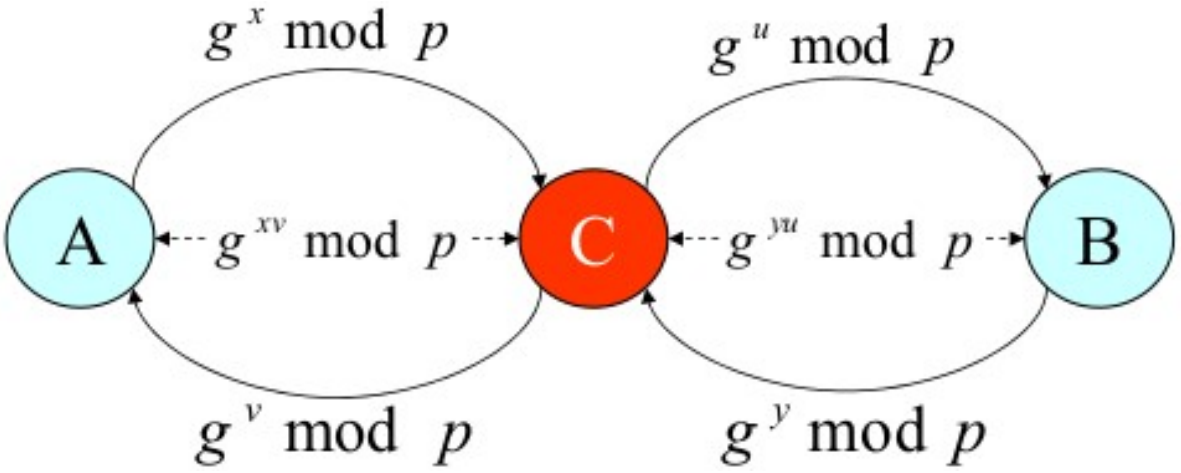
- **Key establishment** shares a secret between two or more parties, for later cryptographic use.
- **Key transport**: One party creates the secret value and securely transfers it to the other(s).
- **Key agreement**: Both parties contribute to the generation of the secret value much that no party can predict the outcome.

Key Authentication

- **Key authentication: assurance that no one except a specific second party may access a particular secret key**
- **Key confirmation: assurance that another party has possession of a particular secret key**
- **Explicit key authentication: both key authentication and key confirmation**

Man-in-the-Middle Attacks

- **Diffie-Hellman doesn't provide authentication: parties do not know whom they are establishing a key with.**
- **An attacker C sitting between A and B can mount a man-in-the-middle attack**



Unauthenticated Diffie-Hellman

- **Unauthenticated Diffie-Hellman is susceptible to man-in-the-middle attacks.**
- **Does this imply that unauthenticated Diffie-Hellman should never be used?**
- **What are its advantages?**

Known Key Attack

- **Denning & Sacco found a problem when more than one protocol run is considered.**
  - e.g. Alice and Bob send multiple messages over time.
- **We have to consider:**
  - Compromise of long-term secret keys.
  - Compromise of past session keys.
- **Known key attack: use a compromised past session key to compromise a future session.**

Perfect Forward Secrecy

- **When a long-term key is compromised, we can no longer protect future sessions.**
  - Design protocols where past sessions remain secure.
- **Forward secrecy: secrecy of old keys is carried forward into the future.**
- **Perfect forward secrecy: compromise of long-term keys does not compromise past session keys.**

Password-based Protocols

- **Use password P to encrypt a randomly generated session key K_s ; use session key to encrypt further data.**
  - A → B: e(P, K_s)
  - B → A: e(K_s, data)
- **Vulnerable to off-line dictionary attack.**
  - Guess password P, decrypt first message and get a candidate session key K'_s ; decrypt the second message with K'_s

Encrypted Key Exchange (EKE)

- **Encrypt data with the password as the key.**
- **Alice generates a random public key/private key pair K_a , K_{a-1}**
  - Alice sends public key K_a to Bob, encrypted under the password P (symmetric encryption)
  - Bob randomly generates session key K_s
  - Bob sends K_s to Alice encrypted first under K_a (public-key encryption) and then under P (symmetric encryption):
- **A → B: e(P, K_a)**
- **B → A: e(P, e(K_a, K_s))**

Public Key Infrastructures

- **Certificates**
  - Binding a name to a key
- **X.509**
  - Standard for public key certs
- **Electronic Signatures**
  - Security service for associating documents (or keys) with persons

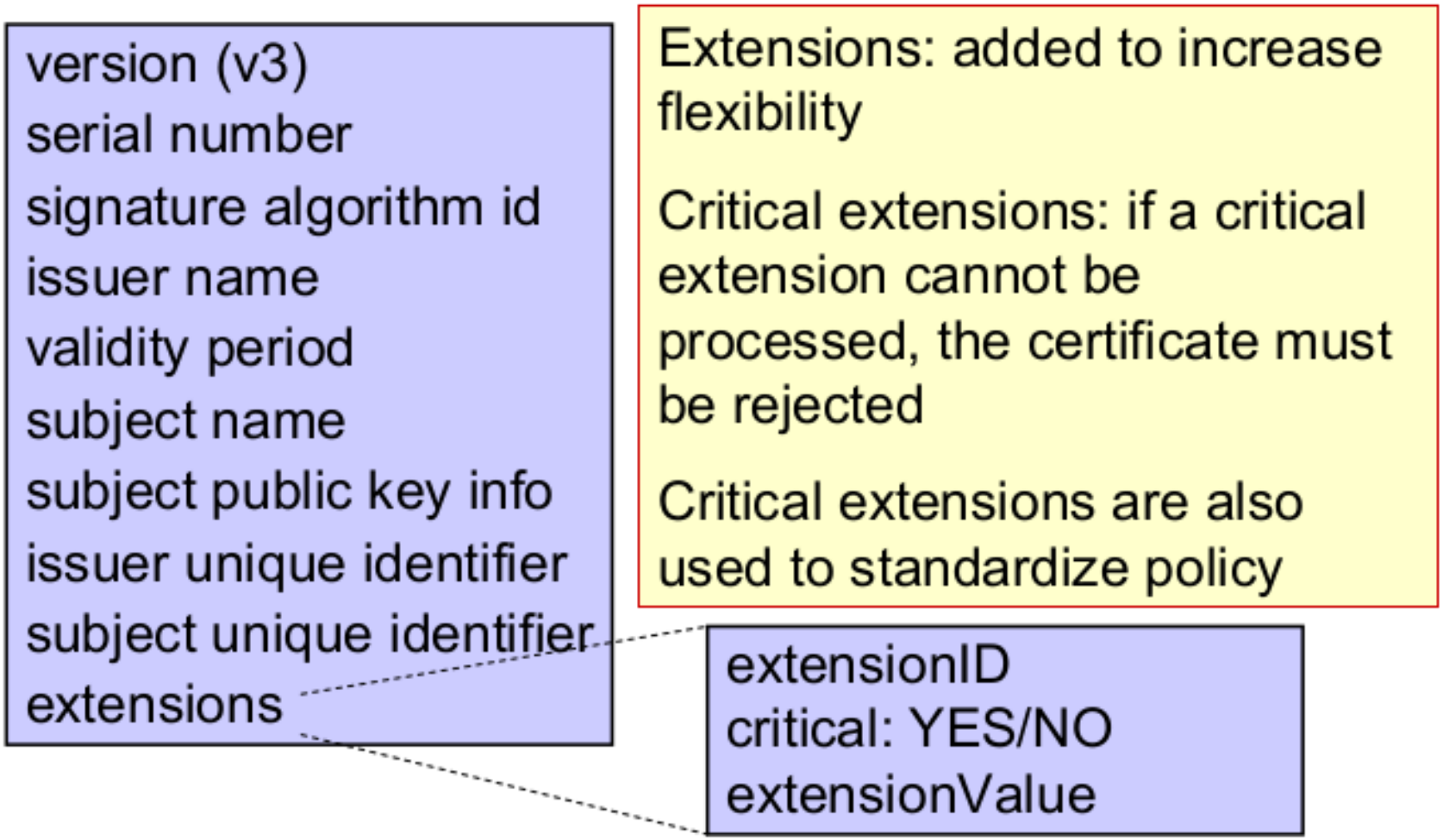
Certificates

- **How do we bind a name to a public key?**
  - Original suggestion: public directory of names and keys
- **Kohnfelder [1978]**
  - digitally signed records containing a name and a public key
  - coined the term certificate for these records.
- **Certificates originally had a single function: associate names and keys.**
- **Today: a certificate is a signed document binding a subject to other information; subjects can be people, keys, names**

X.509

- **X.509: The Directory: Authentication Framework**
  - Geared towards identity based access control
  - Predates web-apps, e-commerce and many modern scenarios
- **Infrastructure Terms**
  - **Certification authority** (CA): an authority **trusted** by one or more users to create and assign certificates.
  - **User certificate** (aka. public key certificate, certificate): public key of a user, rendered unforgeable and signed with the secret key of the certification authority
  - **Attribute certificate**: a set of attributes of a user, digitally signed under the private key of the CA.

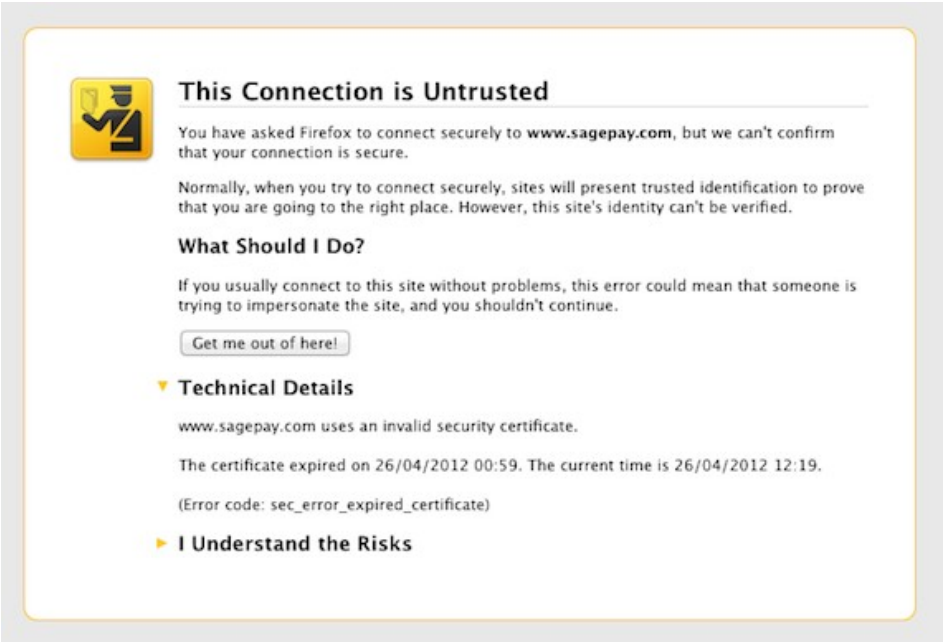
X.509v3 Certificate Format

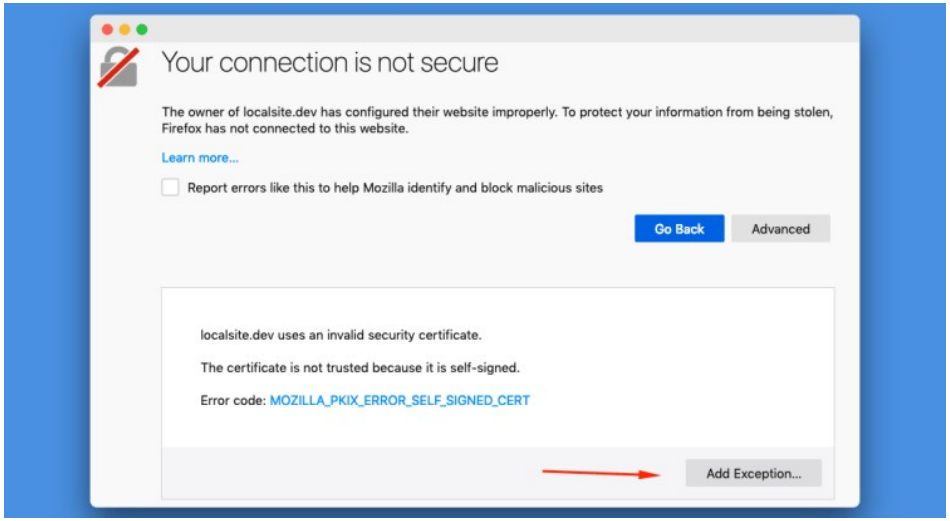


Validity

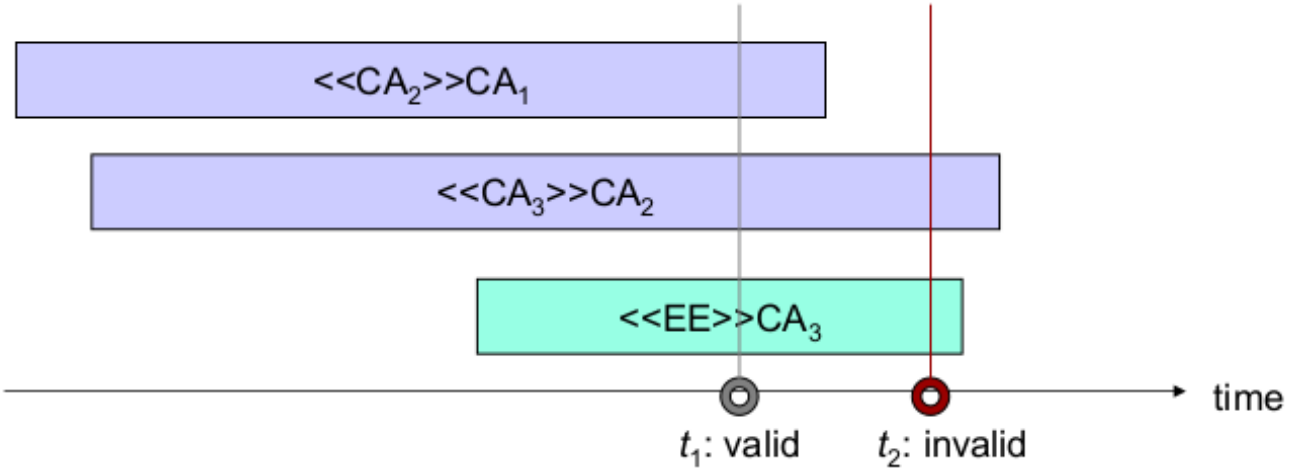
- **Certificates have expiry dates, validity periods.**
- **Misconception: a certificate cannot be used after it has expired.**
  - Dealing with expired certificates is a policy decision.
- **How to evaluate a certificate chain?**
  - Certificates may expire
  - Certificates may be revoked
- **Shell model: all certificates have to be valid at the time of evaluation.**
- **Chain model: issuer’s certificate has to be valid at the time the certificate was issued**

Browser Cert Polices





Shell Model

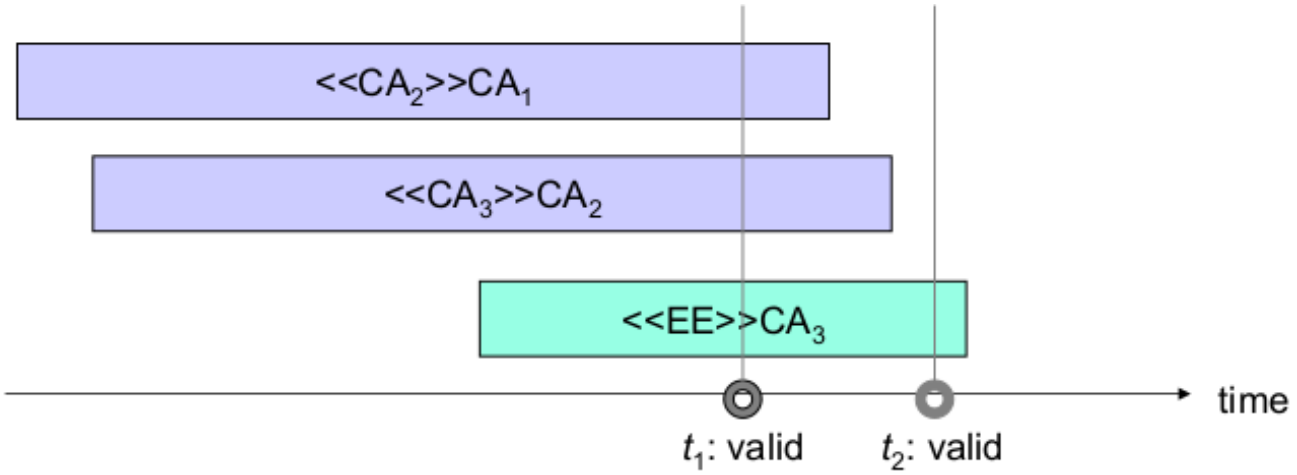


- **Certificate <<EE>>CA_3 valid at time t_1 as all three certificates are valid.**
- **Certificate <<EE>>CA_3 invalid at time t_2 as certificate <<CA_2>>CA_1 has expired.**

Shell Model

- **CAs should only issue certificates that expire before their own certificate.**
- **If a top level certificate expires or is revoked, all certificates signed by the corresponding private key have to be re-issued under a new key.**
- **Appropriate for certificates defining hierarchical address spaces.**

Chain Model



- **Certificate <<EE>>CA_3 is valid at times t_1 and t_2:**
- **<<CA_3>>CA_2 valid when <<EE>>CA_3 was issued**
- **<<CA_2>>CA_1 valid when <<CA_3>>CA_2 was issued**

Chain Model

- **Requires a time-stamping service**
  - Reliably establishing when a certificate was issued
- **If a top level certificate expires or is revoked, certificates signed by the corresponding private key remain valid.**
- **Problem: Business or CA shutdown, but certs remain valid!**

Revocation

- **Certificates may have to be revoked:**
  - if a corresponding private key is compromised,
  - if a fact the certificate vouches for no longer is valid.
- **Certificate Revocation Lists (CRLs):**
  - original solution proposed in X.509
  - distributed in regular intervals or on demand,
  - Delta-CRL: record only the changes since last CRL.
- **CRLs make sense if on-line checks are not possible or too expensive.**

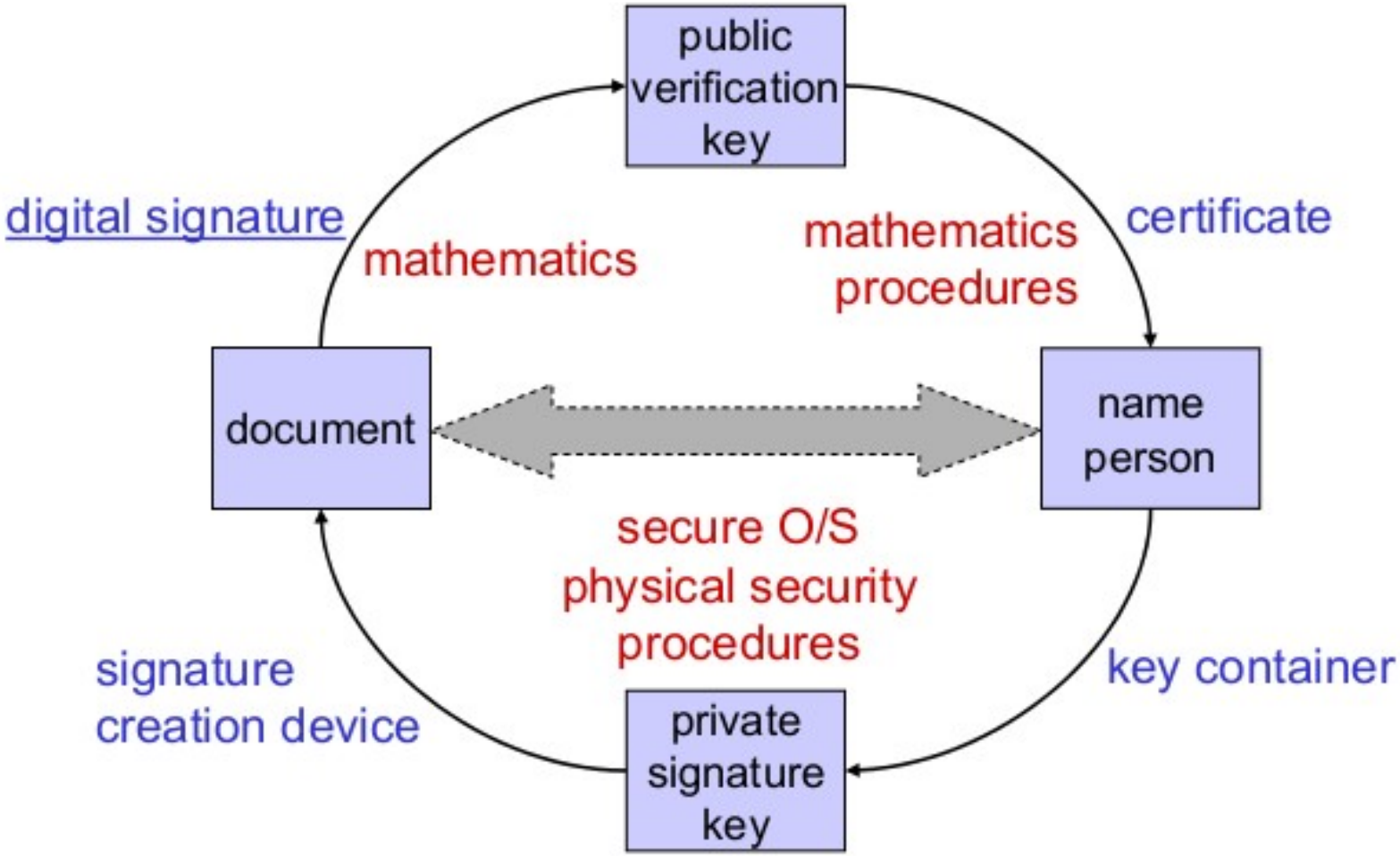
Revocation On-line

- **CRLs can be queried on-line**
  - Online Certificate Status Protocol – OCSP [RFC 2560]
- **A CA issuing certificates for its own use (e.g. for access control) requires only a local CRL.**
- **Alternative to revocation: short-lived certificates**

Electronic Signatures

- **Digital signature: associates documents with keys (cryptographic signing)**
- **Electronic signature: associates documents with persons**
- **Electronic signature services usually use digital signatures,but could be implemented without them**
- **Certificates can record the binding between the name of a person and a key.**

Electronic Signatures



Certificates: Browser Example

- **Browser compares domain name in cert w/ URL**
- **Browser accesses separate cert belonging to issuer**
  - These are hardwired into the browser – and trusted!
- **Browser applies issuer's public key to verify signature S, obtaining hash of what issuer signed**
- **Compares with its own SHA-256 hash of Amazon's cert**
  - Hashes match, now have high confidence it's indeed Amazon

Certificates: Browser Example, cont.

What if browser can’t find a cert for the issuer?

Certificates: Browser Example, cont.

- **If it can’t find the cert, then warns the user that site has not been verified**
  - Can still proceed, just without authentication
- **Which end-to-end security properties do we lose if we incorrectly trust that site?**
- **All of them!**
  - Goodbye confidentiality, integrity, and authentication
  - Man in the middle attacker can read everything, modify, impersonate

Attestation

- **Access request arrives from a remote source.**
- **How can we “trust” claims about application making the request?**
- **A system has to be able to make statements about the software it is running.**
  - Think about secure boot
- **Other systems have to verify such statements.**

Attestation

- **Hardware = “root of trust”.**
- **Trusted Platform Module (TPM)**
  - hardware module that signs statements about the software it is running.
  - Signature (attestation) key installed by hardware manufacturer in TPM.
- **TPM digitally signs the system configuration and the software that is running.**

Unlinkable Attestation

- All attestations from a TPM are signed by the same key
- All transactions from a TPM could be linked
- Full anonymity is not desirable; we must be able to recognize attestations from TPMs known to be compromised.
- Need a from of “group privacy” anyone in group can sign, but don’t know who exactly

Direct Anonymous Attestation

- **Ernie Brickell, Liqun Chen, Jan Camenisch: Direct Anonymous Attestation, http://eprint.iacr.org/2004/205/**
- **Zero-knowledge proof can verify the credential without violating the platform's privacy.**
- **The protocol also supports a blacklisting to identify compromised TPMs**

Group Signatures & Zero-knowledge

- **Based on group signatures:**
  - Only group members can generate valid signatures.
  - Anyone can verify a signature, but not the individual signer.
- **The ‘group’ is all valid TPMs.**
- **Zero-knowledge proofs**
  - Prove knowledge of X without explicitly revealing X.
  - Prove knowledge of a discrete logarithm without revealing any information about its value

NETWORK SECURITY

Introduction

- Net adversary
- TCP attacks
- DNS attacks
- Firewalls
- Intrusion detection
- Honeypots

Secure End-to-End Channels

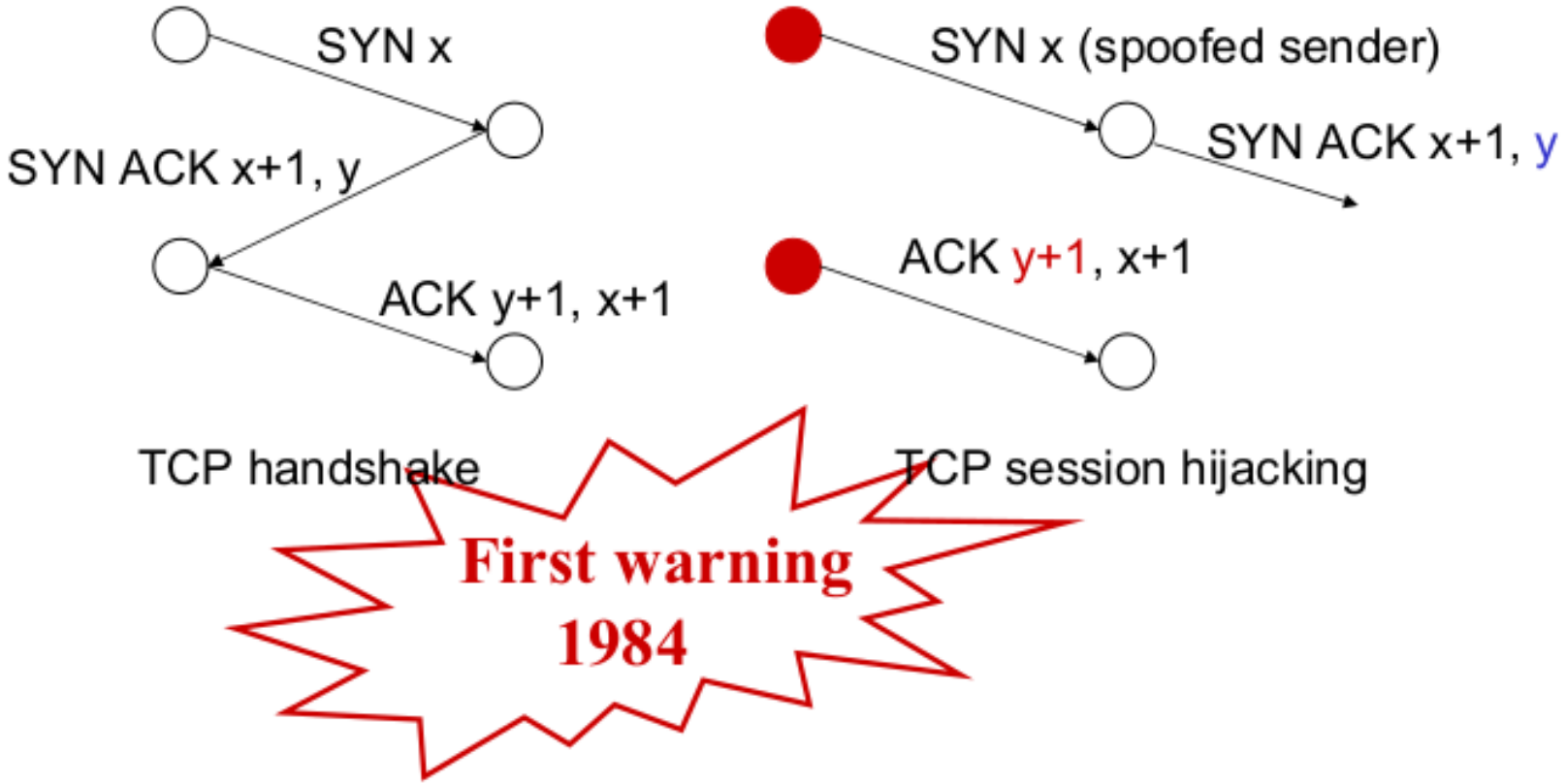
- **End-to-end = protect channel from originating client to intended server, between endpoints**
  - no need to trust intermediaries
- **Dealing with threats:**
  - Eavesdropping?
    - Encryption (including session keys)
  - Manipulation (injection, MITM)?
    - Integrity (use of a MAC); replay protection
  - Impersonation? (someone pretending as you)
    - Signatures
  - Availability?

Net Adversary

- **A botnet consists of bots, programs running on the machines of unwitting Internet users and receiving commands from a bot controller.**
- **Net adversary threat model**
- **A malicious network node able to:**
  - read messages directly addressed to it,
  - spoof arbitrary sender addresses,
  - try to guess fields sent in unseen messages.

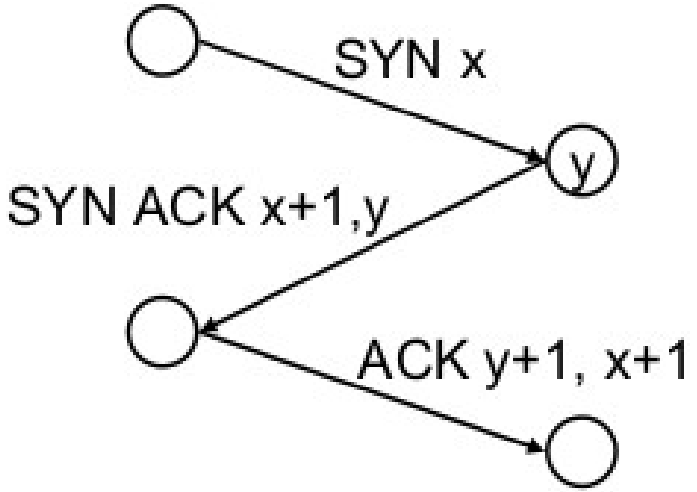
TCP Session Hijacking

- Predict challenge to send messages that appear to come from a trusted host.

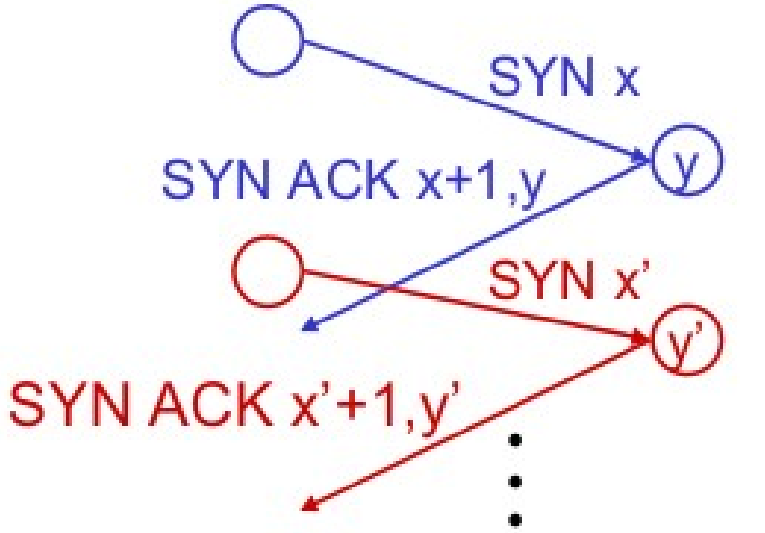


TCP SYN Flooding Attacks

- Exhaust responder's resources by creating half-open TCP connection requests.



TCP handshake



SYN flooding attack

DNS ATTACKS

Domain Name System (DNS)

- **Essential infrastructure for the Internet**
  - critical-path for just about everything we do
  - Maps host names to IP addresses (and vice versa)
- **Design only scales if we can minimize lookup traffic**
  - Lots of caching!
  - Pre-fetching additional answers
- **Originally designed for a friendly environment; only basic authentication mechanisms**
- **Directly interacting w/ DNS: dig program on Unix**
  - Allows querying of DNS system
  - Dumps each field in DNS responses

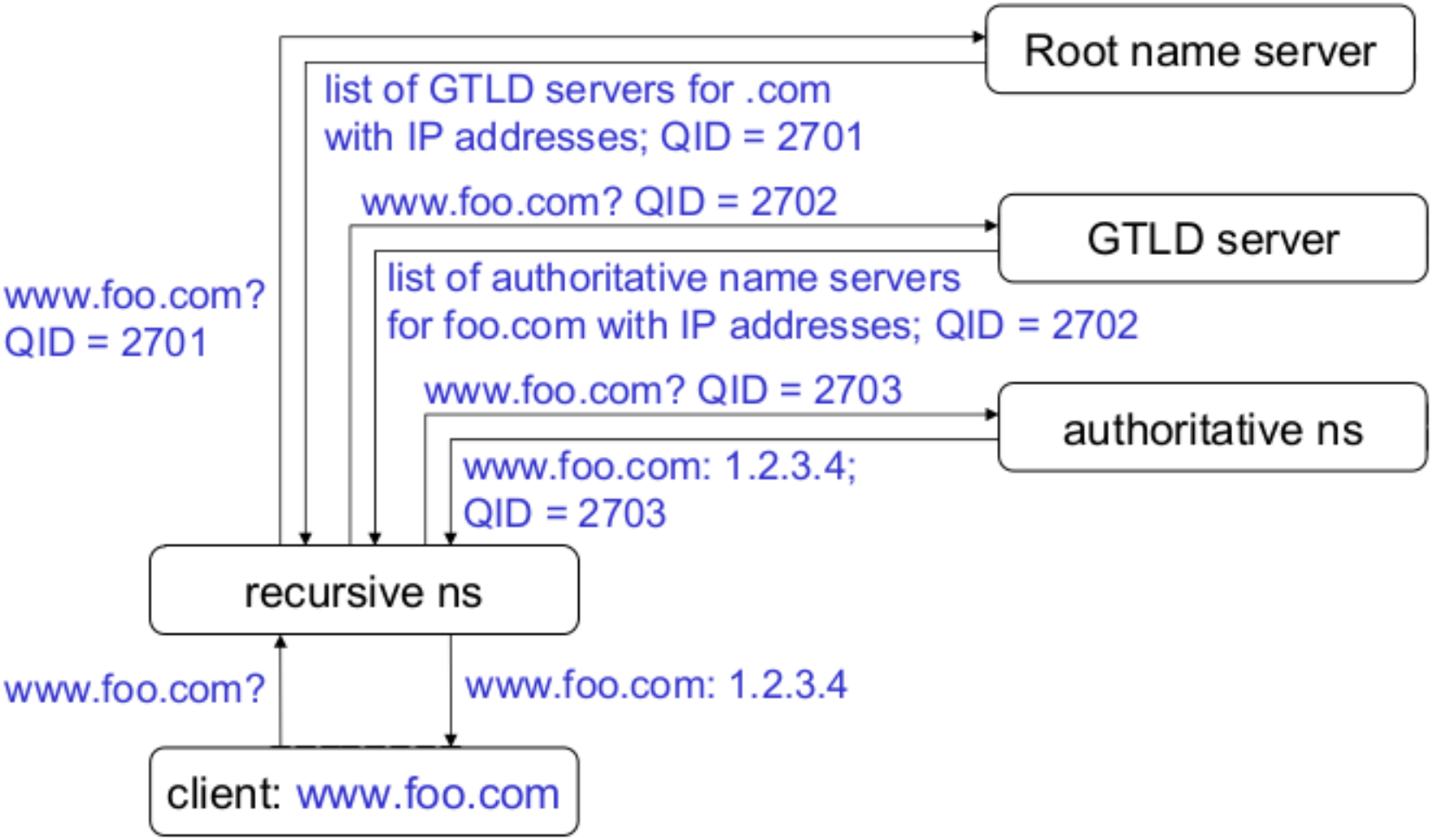
Domain Name System (DNS)

- **Distributed directory service for domain names (host names) used for:**
  - look up IP address for host name, host name for IP address.
  - anti-spam: Sender Policy Framework uses DNS records.
  - basis for same origin policies applied by web browsers.
- **Various types of resource records.**
- **Host names and IP addresses collected in zones managed by authoritative name servers.**

DNS Infrastructure

- **13 root servers; all name servers configured with the IP addresses of these root servers.**
- **Global Top Level Domain (GTLD) servers for top level domains: .com, .net, .org, etc.**
  - There can be more than one GTLD server per TLD.
  - Root servers know about GTLD servers.
- **Authoritative name servers provide mapping between host names and IP addresses for their zone.**
- **GTLD servers know authoritative servers in their TLD**
- **Recursive name servers pass client requests on to other name servers and cache answers received.**

IP Address Lookup – Review



Cache & Time-to-live

- **Performance optimization: stores map in cache**
- **Name server first checks its cache**
- **Answer remains in cache until it expires; time-to-live (TTL) of answer is set by sender.**
- **Design question: reasons for setting TTL by sender, reasons for setting TTL by receiver?**
- **Does Long TTL = high security, low TTL = low security?**

Light-weight Authentication

- **Threat model:**
  - Attacker can only read messages forwarded to her
  - Anybody can pretend to be an authoritative name server for any zone
- **How does a recursive name server know that it has received a reply from an authoritative name server?**
  - Recursive name server includes a 16-bit query ID (QID) in its requests.
  - Responding name server copies QID into reply
  - Recursive name server caches first answer for a given QID and host name; then discards this QID.
  - Drops answers that do not match an active QID.

Authentication – Security?

- **Attack method: guess QID to subvert cache entries.**
- **If query is not passed by mistake to the attacker, her chance of generate faking a answer is 2^-16**
- **Security relies on correct routing from local name to authoritative name server.**

DNS Cache Poisioning

- **Ask recursive name server to resolve host name in attacker's domain.**
- **Request to attacker's name server contains current QID.**
- **Attackers asks recursive name server to resolve victim host name**
- **Attacker sends answer that includes next QID and maps victim host name to chosen IP address**
- **If attacker's answer arrives first; the correct answer is dropped and cache is poisoned**

Predictable Challenges

- **Do not use predictable challenges (e.g. QID)**
- **Attacker can improve chances:**
  - Send answers with QIDs from a small window.
  - Slow down authoritative name server with a DoS attack.
  - Prevent that a new query from restoring the correct binding, set a long time to live.

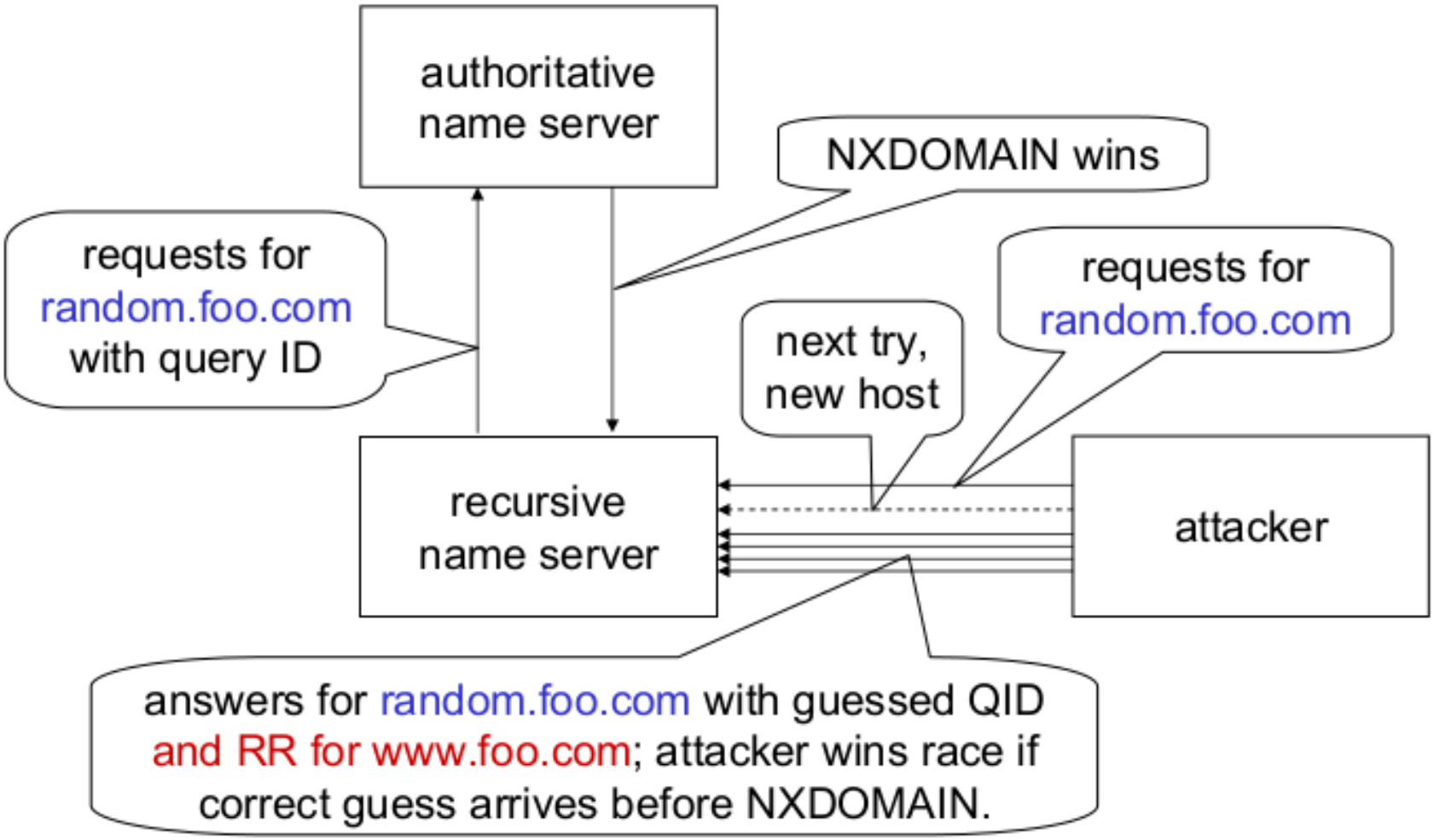
Bailiwick Checking

- **Bailiwick: an area of jurisdiction**
- **Optimize perf:**
  - Name servers send additional resource records
  - Might save round trips
  - Assumes benign servers
  - Malicious name server sends records for other domains
- **Bailiwick checking rejects records outside of the queried domain (i.e. out of jurisdiction)**

Dan Kaminsky's Attack (2008)

- Attacker requests random.foo.com from name server
- Recursive name server refers request to authoritative name server for foo.com
- Attacker sends answers for random.foo.com with guessed QIDs and additional resource record for www.foo.com (in bailiwick)
- If guessed QID is correct and attacker wins race with NXDOMAIN, poison entry is cached with a TTL set by attacker
- Recursive name server will now direct all queries for www.foo.com to attacker's IP address

Dan Kaminsky's Attack



Countermeasures

- **Run queries on random ports**
  - Attacker now must guess QID & port number
- **Restrict access to local recursive name server: split name server**
- **Access control for records prevent unauthorized overwriting**
- **DNSSec: authentication using digital signatures**
- **Server does not reply to malformed queries??**

Split-split Name Server

- Local users who want to connect to the outside world
- Remote users who want to connect to local hosts
- Recursive name server for internal queries to resolve (external) host names
- Non-recursive authoritative name server for zone to resolve external queries for host names in zone
- DNS server facing external users does not cache resource records so there is no cache to poison
- No defense against local attackers

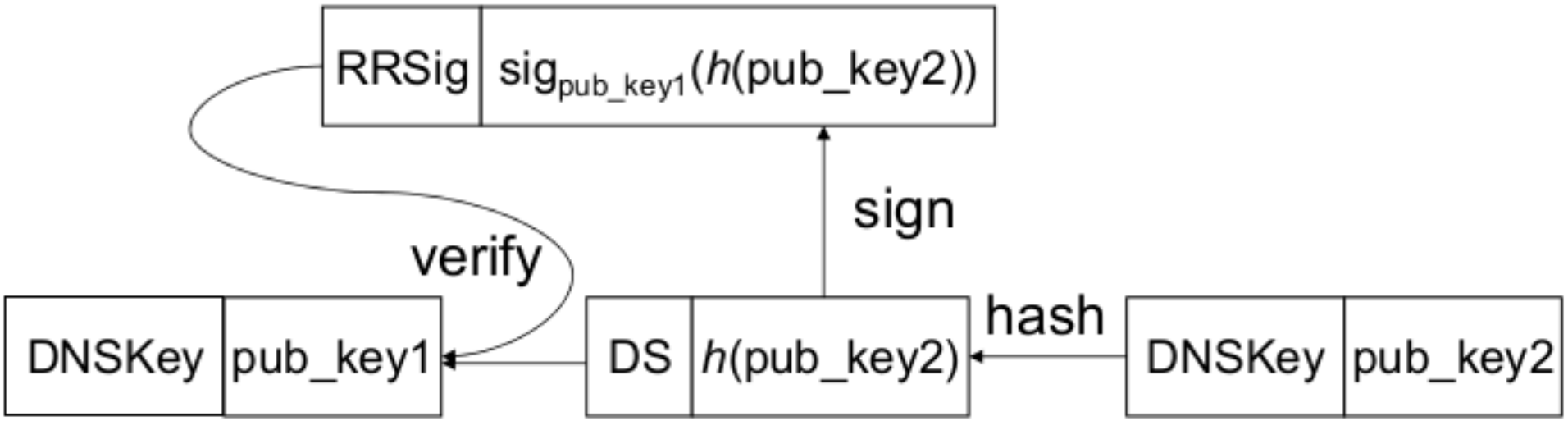
DNSSec

- **DNS Security Extensions, protect resource records with digital signatures**
- **Several new resource record types introduced:**
  - RRSIG resource records contain digital signatures of other resource records.
  - DNSKEY resource records contain the public keys of zones.
  - DS (Delegation Signer) resource records contain hashes of DNSKEY research records.

DNSSec

- **Build chain by alternating DNSKEY and DS records.**
- **Key in DNSKEY record verifies the signature on the next DS record**
- **Hash in the DS record links to next DNSKEY record, and so on.**
- **Verification in the resolver has to find a trust anchor for the chain (root verification key).**

DNSSec – Chain



DNS Rebinding

- **Same origin policy: script in a web page can only connect back to the server it was downloaded from.**
- **To make a connection, the client's browser needs the IP address of the server.**
- **Authoritative DNS server resolves 'abstract' DNS names in its domain to 'concrete' IP addresses.**
- **The client's browser 'trusts' the DNS server when enforcing the same origin policy.**
- **Trust is Bad for Security!**

DNS Rebinding Attack

- **“Abuse trust”: Attacker creates attacker.org domain and name server**
- **Evil names server binds attacker.org IP and then switches to victim IP address**
- **Client downloads applet from attacker.org; script connects to target; permitted by same origin policy.**
- **Defense: Same origin policy with IP address.**
  - D. Dean, E.W. Felten, D.S. Wallach: Java security: from HotJava to Netscape and beyond, 1996 IEEE Symposium on Security & Privacy.

DNS Rebinding Attack

- **Client visits attacker.org; attacker's DNS server resolves this name to attacker's IP address with short time-to-live.**
- **Attack script waits before connecting to attacker.org.**
- **Binding at browser has expired; new request for IP address of attacker.org, now bound to target address.**
- **Defense: Don't trust the DNS server on time-to-live; pin host name to original IP address**

DNS Rebinding Attack

- **Attacker shuts down its web server after the page has been loaded.**
- **Malicious script sends delayed request to attacker.org.**
- **Browser’s connection attempt fails and pin is dropped.**
- **Browser performs a new DNS lookup and is now given the target’s IP address.**
- **Error handling procedures has security implications!**

DNS Rebinding Attack

- **Next round – browser plug-ins, e.g. Flash.**
- **Plug-ins may do their own pinning.**
- **Dangerous combinations:**
  - Communication path between plug-ins.
  - Each plug-in has its own pinning database.
- **Attacker may use the client's browser as a proxy to attack the target.**
  - DDOS, send spam, etc.

FIREWALLS

Introduction

- **Cryptographic mechanisms protect data in transit**
- **Authentication protocols verify the source of data.**
- **Control which traffic is allowed to enter or leave our system**
- **Access control decisions based on information like addresses, port numbers, protocol, etc.**

Firewall

- **Firewall: a network security device controlling traffic flow between two parts of a network.**
- **Often installed between an organization's network and the Internet**
- **All traffic has to go through the firewall for protection to be effective.**
  - Wireless LANs, USB devices!?

Purpose

- **Firewalls control network traffic to and from the protected network.**
- **Can allow or block access to services (both internal and external).**
- **Can enforce authentication before allowing access to services.**
- **Can monitor traffic in/out of network.**

Types of Firewalls

- **Packet filter**
- **Stateful packet filter**
- **Circuit-level proxy**
- **Application-level proxy**

Packet Filter

- Inspect headers of IP packets, TCP and UDP ports
- Rules specify which packets are allowed through the firewall, and which are dropped.
- Actions: bypass, drop, protect
- Rules may specify source / destination IP addresses, and source / destination TCP / UDP port numbers.
- Rules for traffic in both directions.
- Certain common protocols are difficult to support securely (e.g. FTP).

Example

- **TCP/IP packet filtering router.**
  - Router which can throw packets away.
- **Examines TCP/IP headers of every packet going through the Firewall, in either direction.**
- **Packets can be allowed or blocked based on:**
  - IP source & destination addresses
  - TCP / UDP source & destination ports
- **Implementation on router for high throughput.**

Stateful Packet Filter

- **Packet filter that understands requests and replies**
  - e.g. for TCP: SYN, SYN-ACK, ACK
- **Rules need only specify packets in one direction**
  - from client to server – the direction of the first packet in a connection
- **Replies and further packets in the connection are automatically processed.**
- **Supports wider range of protocols than simple packet filter (FTP, IRC).**

Stateful Packet Filter & FTP

- **Client sends ftp-request to server**
- **Firewall stores connection state**
  - FTP-Server Address
  - state of connection (SYN, ACK, ...)
- **If correct FTP-server tries to establish data connection, packets are not blocked.**

Circuit-level proxy

- Similar to a packet filter, except that packets are not routed.
- Incoming TCP/IP packets accepted by proxy.
- Rules determine which connections will be allowed and which blocked.
- Allowed connections generate new connection from firewall to server.
- Similar specification of rules as packet filter.

Application-level Proxy

- Layer-7 proxy server.
- “Client and server in one box”.
- For every supported application protocol.
- SMTP, POP3, HTTP, SSH, FTP, NNTP...
- Packets received and processed by server.
- New packets generated by client.
- MITM?

Application-level Proxy

- **Complete server & client implementation in one box for every protocol the firewall should handle.**
  - Client connects to firewall.
  - Firewall validates request.
  - Firewall connects to server.
- **Response comes back through firewall and is also processed through client/server.**
- **Large amount of processing per connection.**
- **Can enforce application-specific policies.**

Firewall Policies

- **Permissive: allow by default, block some.**
  - Easy to make mistakes.
  - If you forget something you should block, it’s allowed, and you might not realize for a while.
  - If somebody finds find a protocol is allowed, they might not tell you ….
- **Restrictive: block by default, allow some.**
  - Much more secure.
  - If you forget something, someone will complain and you can allow the protocol.

Firewall Policies – Examples

- **Permissive policies: Allow all traffic, but block …**
  - IRC
  - telnet
- **Restrictive policies: block all traffic, but allow …**
  - http
  - POP3
  - SMTP
  - ssh

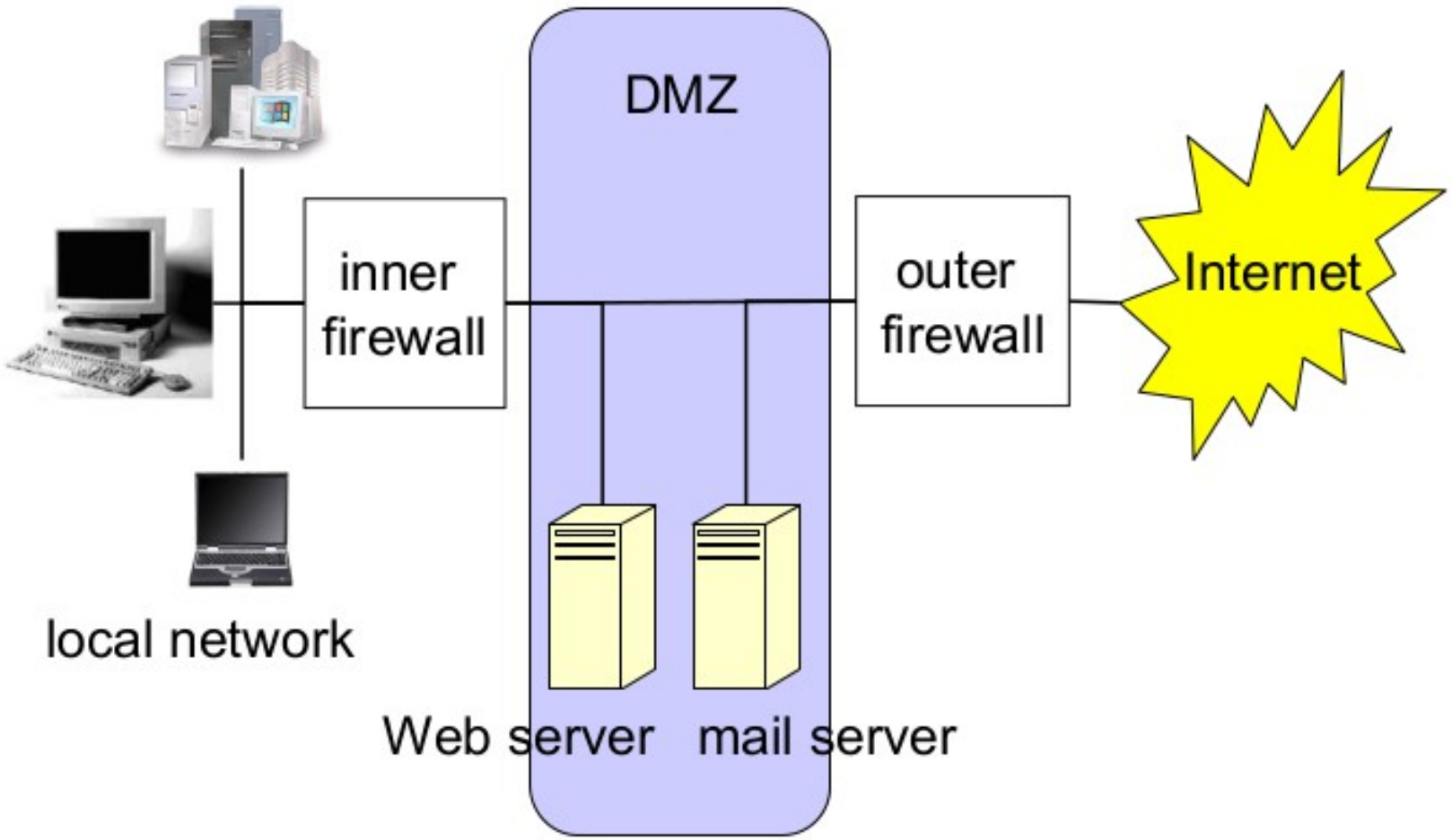
Typical Firewall Ruleset

- **Allow from internal network to Internet:**
  - HTTP, FTP, HTTPS, SSH, DNS
- **Allow reply packets**
- **Allow from anywhere to Mail server:**
  - TCP port 25 (SMTP) only
- **Allow from Mail server to Internet:**
  - SMTP, DNS
- **Allow from inside to Mail server:**
  - SMTP, POP3
- **Block everything else**

Firewall Location

- **Firewall can only filter traffic which goes through it.**
- **Where should we put a mail server?**
  - Requires external access to receive mail from the Internet.
    - Should be on the inside of the firewall
  - Requires internal access to receive mail from the internal network.
    - Should be on the outside of the firewall
- **Solution: “a perimeter network” (aka DMZ).**

DMZ


DMZ
inner
firewall
outer
firewall
Internet
local network
Web server
mail server

Firewalls – Limitations

- **Firewalls do not protect against insider threats.**
- **Blocking services may create inconveniences for users.**
- **Network diagnostics may be harder.**
- **Some protocols are hard to support.**
- **Protocol tunneling: sending data for one protocol through another protocol circumvents the firewall.**
  - More and more protocols are tunneled through http to get through the firewall
- **Encrypted traffic cannot be examined and filtered**
  - Some solutions can! HTTPS proxy

INTRUSION DETECTION SYSTEMS

Reminder: Security Strategies

- **Prevention**: take measures that prevent your assets from being damaged.
- **Detection**: take measures so that you can detect when, how, and by whom an asset has been damaged.
- **Reaction**: take measures so that you can recover your assets or to recover from a damage to your assets.

Security Strategies

- Cryptographic mechanisms and protocols are fielded to prevent attacks.
- Perimeter security devices (e.g. firewalls) mainly prevent attacks by outsiders.
- Although it would be nice to prevent all attacks, in reality this is rarely possible.
- New types of attacks occur: denial-of-service (where crypto may make the problem worse).
- How to we detect network attacks?

Vulnerability Assessment

- **Examines the “security state” of a network:**
  - Open ports
  - Software packages running (which version, patched?)
  - Network topology
  - Returns prioritized lists of vulnerabilities
- **Only as good as the knowledge base used.**
  - Have to be updated to handle new threats
- **Vulnerability Assessment Methods.**
  - Software solutions (ISS Scanner, Stat, Nessus etc.)
  - Audit Services (manual Penetration tests etc)
  - Web based commercial (Qualys, Security Point etc)

Intrusion Detection Systems (IDS)

- **Passive supervision of network, analogue to intruder alarms.**
  - Creates more work for personnel.
  - Provides security personnel with volumes of reports that can be presented to management
- **Approaches to Intrusion Detection:**
  - Knowledge-based IDS – Misuse detection
  - Behavior-based IDS – Anomaly detection
- **IDS can also be used as response tool.**