Lecture 27 – DNS Security (DNSSEC): Key Management, and Authenticated Denial of Existence

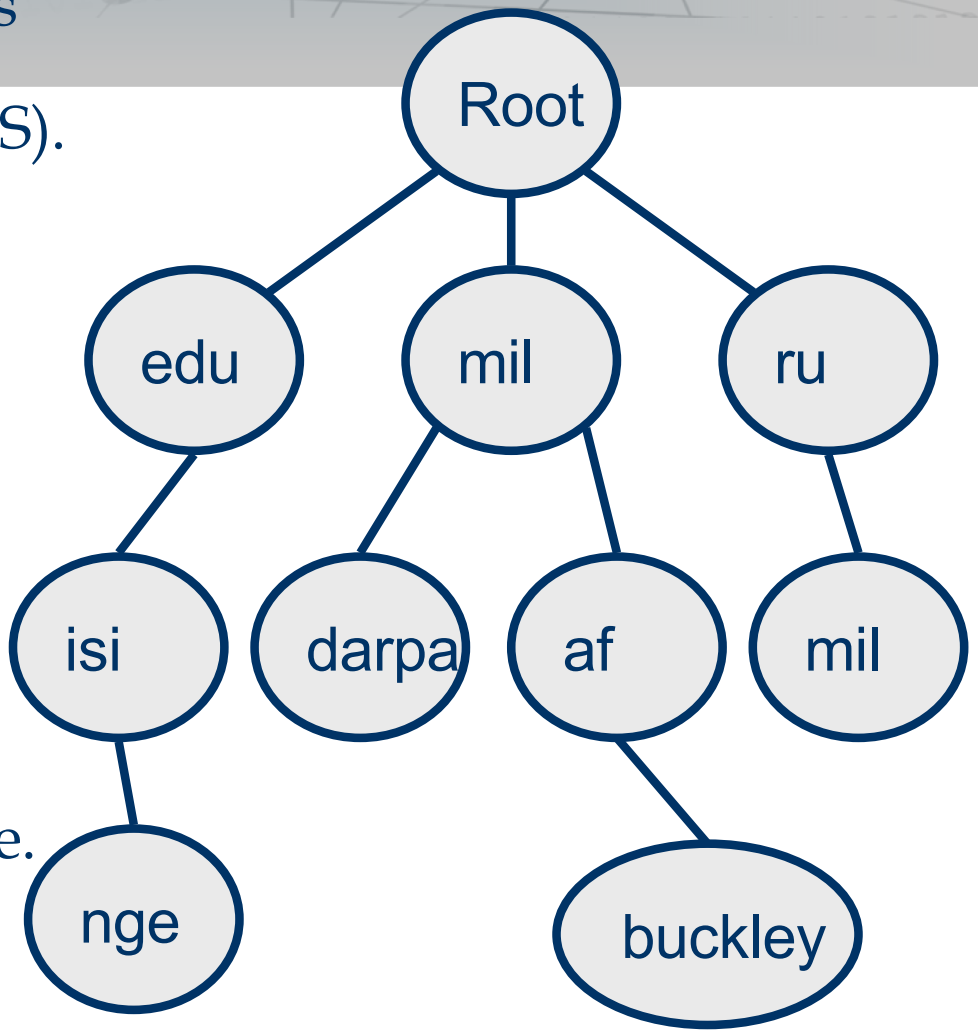
December 4, 2018

Dr. Dan Massey

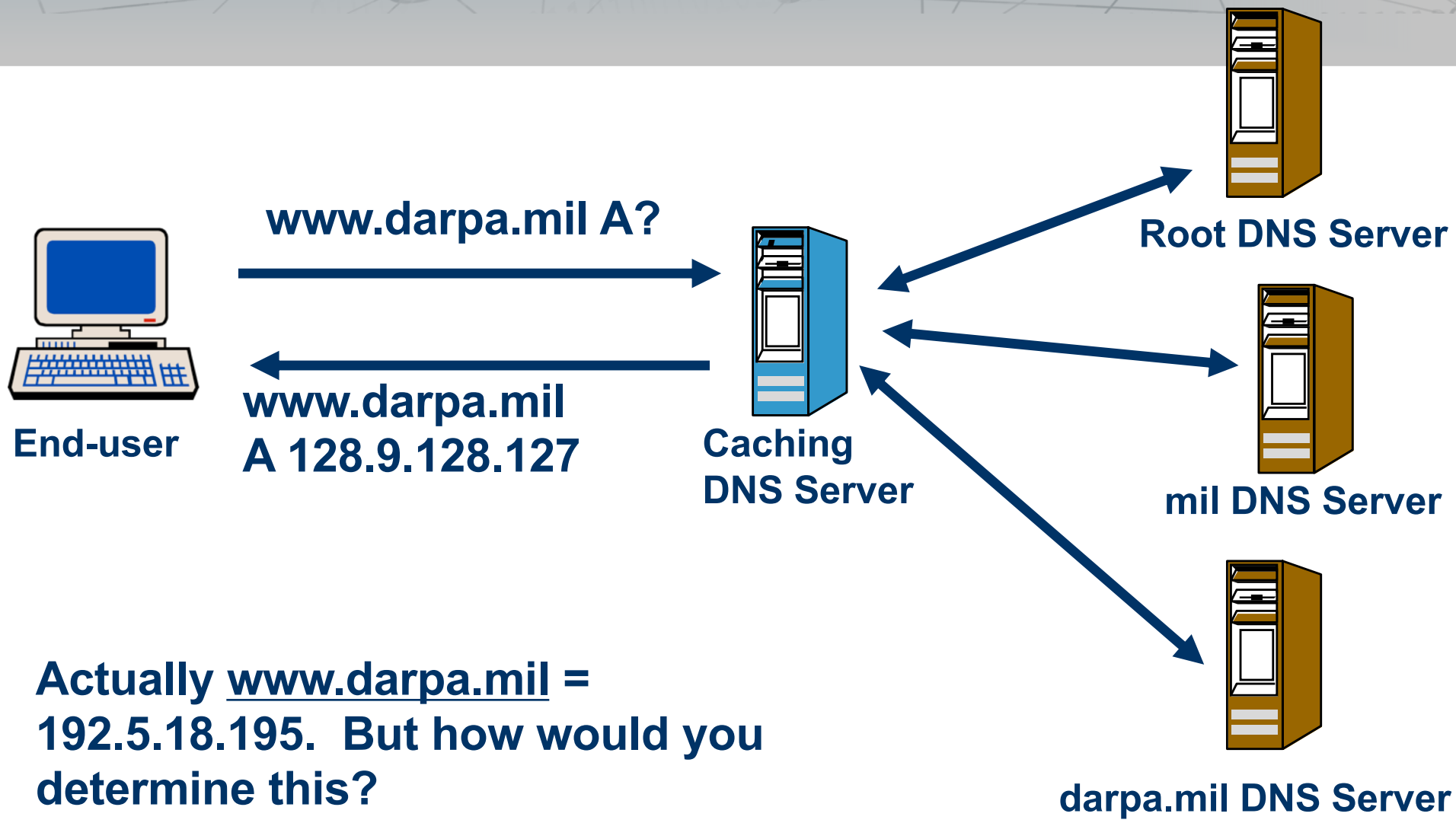
The Domain Name System

- Virtually every application uses the Domain Name System (DNS).
- DNS database maps:
  - Name to IP address

    www.darpa.mil = 128.9.176.20
  - And many other mappings (mail servers, IPv6, reverse…)
- Data organized as tree structure.
  - Each zone is authoritative for its local data.



DNS Query and Response



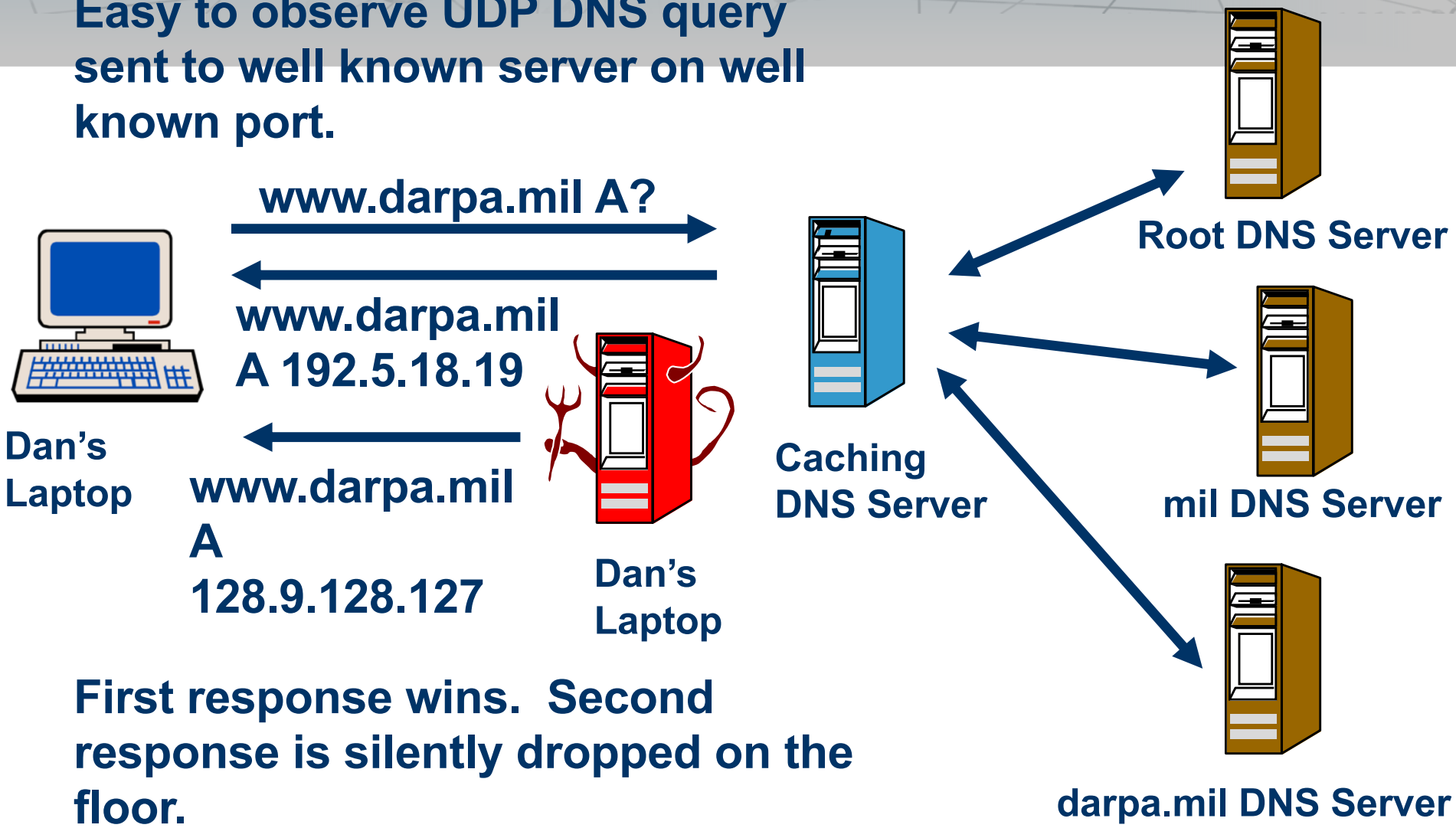
Actually www.darpa.mil = 192.5.18.195. But how would you determine this?

DNS Vulnerabilities

- Original DNS design focused on data availability
  - DNS zone data is replicated at multiple servers.
  - A DNS zone works as long as one server is available.
    - DDoS attacks against the root must take out 13 root servers.
- But the DNS design included no authentication.
  - Any DNS response is generally believed.
  - No attempt to distinguish valid data from invalid.
    - Just one false root server could disrupt the entire DNS.

A Simple DNS Attack

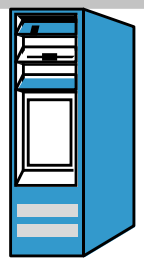
Easy to observe UDP DNS query sent to well known server on well known port.



First response wins. Second response is silently dropped on the floor.

A More Complex Attack

Colorado
Caching Server



www.google.com
= 128.9.128.127

Response

www.attacker.com A 128.9.128.127
attacker.com NS ns.attacker.com
attacker.com NS www.google.com
ns.attacker.com A 128.9.128.2
www.google.com A 128.9.128.127

ns.attacker.com

Query www.google.com

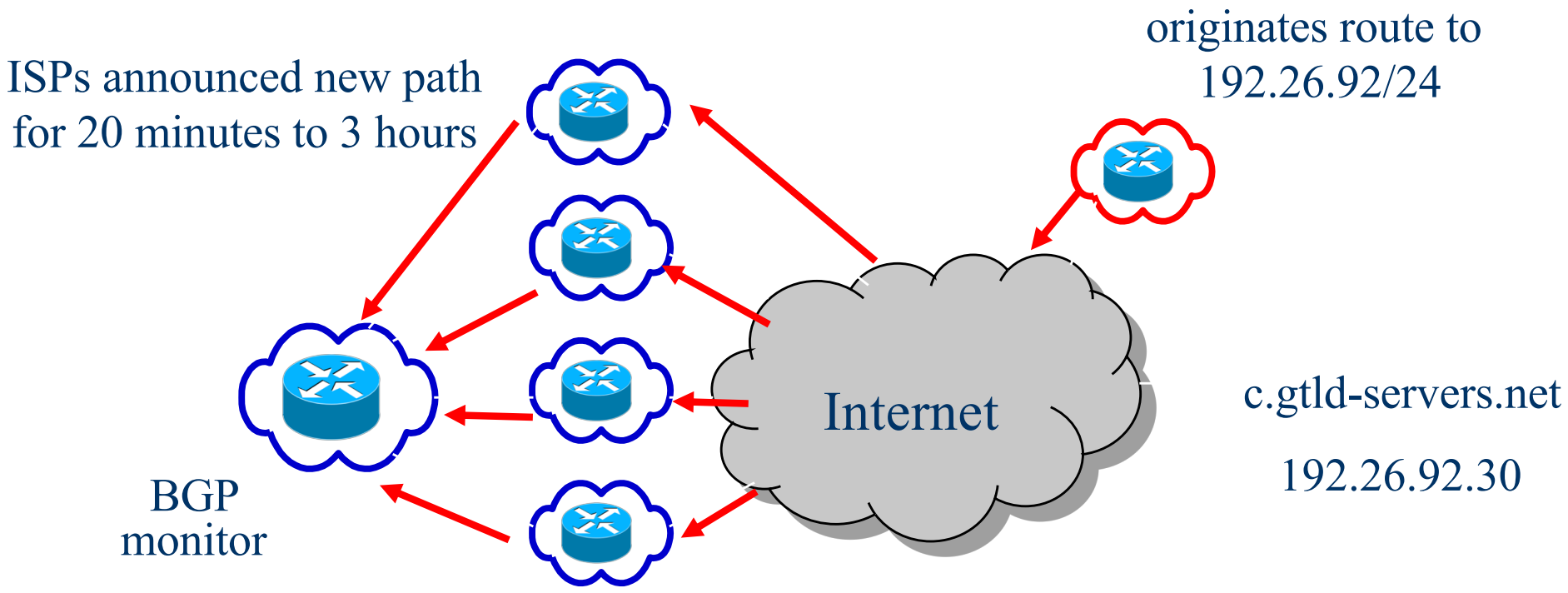
Query
www.attacker.com

Colorado Laptop

Remote attacker

Routing Based DNS Attacks

- BGP Also Provides No Authentication
  - Faults and attacks can mis-direct traffic.
  - One (of many) examples observed from BGP logs.
  - Server could have replied with false DNS data.



The Problem in a Nutshell

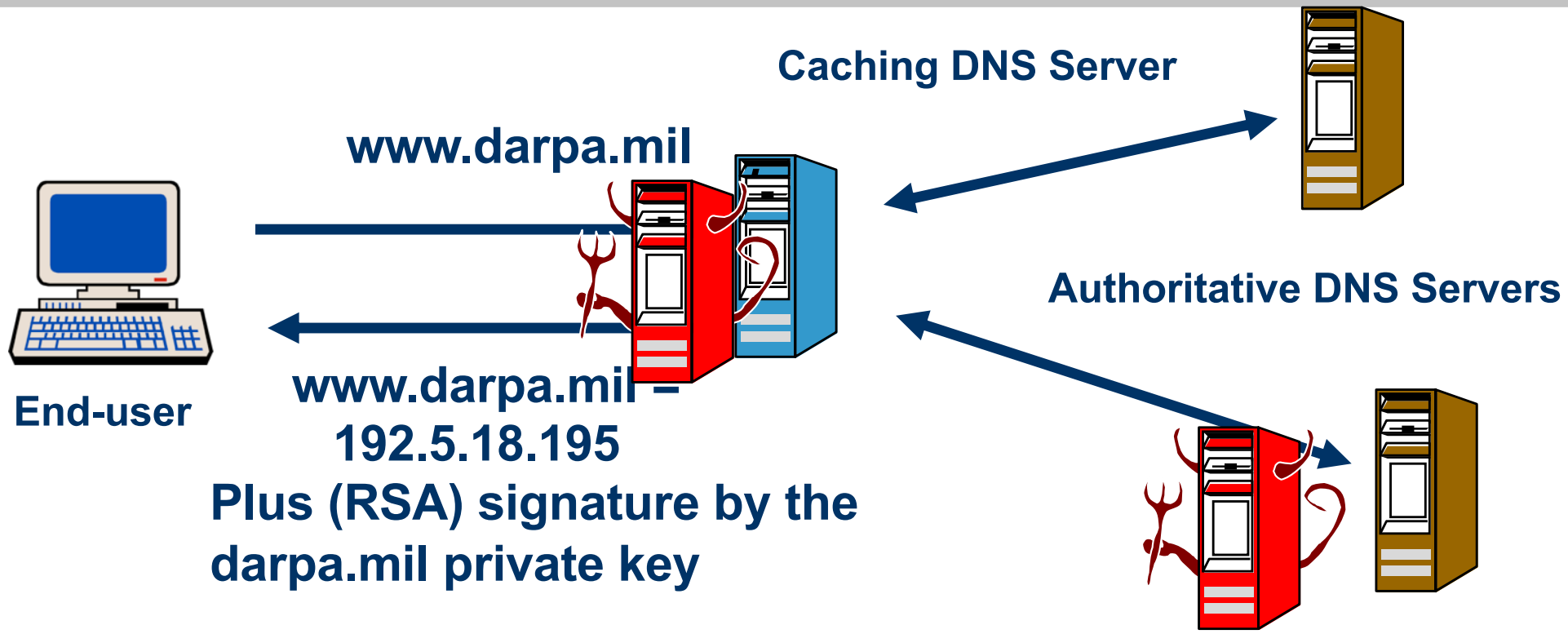
- Resolver can not distinguish between ***valid and invalid*** data in a response.
- Idea is to add source authentication
  - Verify the data received in a response is equal to the data entered by the zone administrator.
  - Must work across caches and views.
  - Must maintain a working DNS for old clients.

DNS Security Extensions

Cryptography is like magic fairy dust, we just sprinkle it on our protocols and its makes everything secure

- IEEE Security and Privacy Magazine, Jan 2003

Secure DNS Query and Response



Attacker can not forge this answer without the darpa.mil private key.

Authenticated DNS Responses

- Each zone signs its data using a private key.
  - Recommend signing done offline in advance
- Query for a particular record returns:
  - The requested resource record set.
  - A signature (SIG) of the requested resource record set.
- Resolver authenticates response using public key.
  - Public key is pre-configured or learned via a sequence of key records in the DNS heirarchy.

Learning DNS Public Keys

- Public key is required to verify signature
  - RRSIG record identified the key name and key tag.
  - If you are pre-configured with key, then done.
    - UCLA resolver is configured with the ucla.edu key
- Typical resolver does not have all the public keys.
  - Configure root key and perhaps some local keys
  - Query zone for the desired public
  - Query returns DNSKEY record and a signature from the parent zone.

Example DNSSEC Records

name TTL class RRSIG type_covered Algorithm labels TTL expiration (
inception dates key_tag key_name signature)

www.darpa.mil. 82310 IN A 192.5.18.195
www.darpa.mil. 82310 IN RRSIG A 1
20040127023910 468 darpa.mil.
Base 64 encoding of signature)

name TTL class DNSKEY FLAGS PROTOCOL Algorithm public key

darpa.mil. 81020 IN DNSKEY 256 1 (*Base64 encoding of pub key*)
DNSKEY 1
20040127023910 569 mil.
Base 64 encoding of signature)

Note the darpa.mil DNSKEY is signed by the mil private key
(*We later show why this doesn't work)*

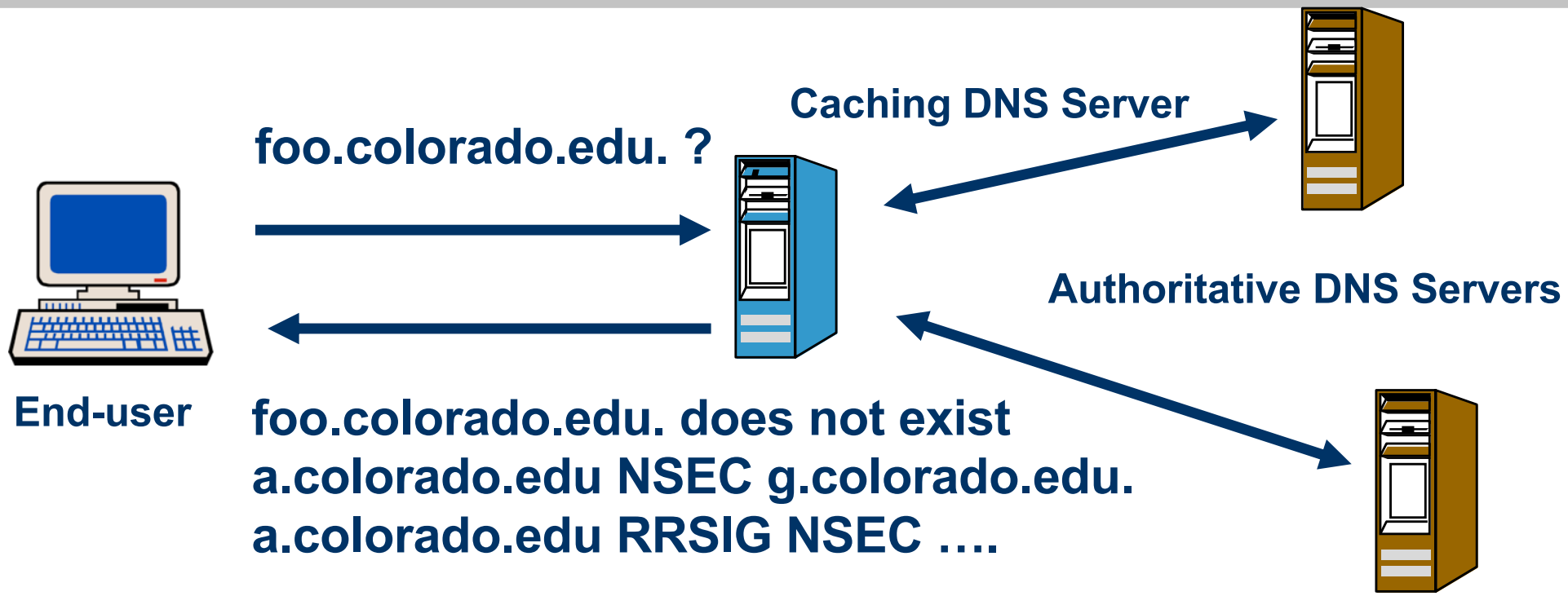
Authenticated Denial of Existence

- What if the requested record doesn't exist?
  - Query for foo.colorado.edu returns "No such name"
  - How do you authenticate this?
- Must meet a variety of operational constraints
  - Some zones refuse to store any keying information online.
  - Some zones don't trust (all) of their secondary servers.
    - Can't control which server a resolver contacts.
  - Some zones don't have computational resources to sign on the fly
  - Can't predict user would ask for "foo.colorado.edu"

NSEC Records

Solution:

sign “next name after a.colorado.edu. is g.colorado.edu.”



There is no magic fairy dust

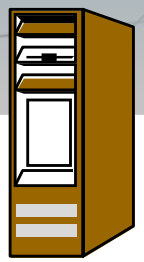
Challenges in System Security

- Challenges in Key Management
  - Original RFC 2535 design did not scale and did not work in realistic operations.
- Limitations in the Original DNS Design
  - Wildcards were not well defined and they complicate non-existence proofs.
  - NS and glue records appeared in both parent and child, who signs them?
- Security Deployment and Usage Issues
  - Added cost of NSEC RRs (without added revenue)
  - Configuring and changing the apex keys (e.g. root keys)
  - Resolver rules for using the DNSSEC
    - Attempts to overspecify end behavior & dictate local policy

Obsolete Key Management (RFC 2535)

- A zone's DNSKEY is signed by its parent.
  - ISI sends "isi.edu DNSKEY" to "edu"
  - "edu" signs the DNSKEY with edu private key.
  - Resulting RRSIG sent to ISI and stored in the isi.edu zone.
- But Existing Design Adds Complications
  - Smallest item in DNS is *(name, record type)*
  - RRSIG covers ***all*** (isi.edu. DNSKEY) records
    - edu signs the ***set*** of isi.edu DNSKEYS
    - ISI can't modify the set without a new RRSIG

The Obsolete Key Infrastructure



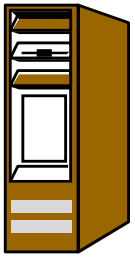
Root KEY record
SIG by root

Store the public keys in DNS
Parent key signs child key

Root DNS Server

Now try to change this KEY

Adds communication and synchronization with parent

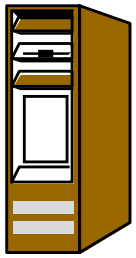


com KEY record
SIG by root

Com DNS Server

Adds communication and synchronization with child

children varies between 0 and 22 million

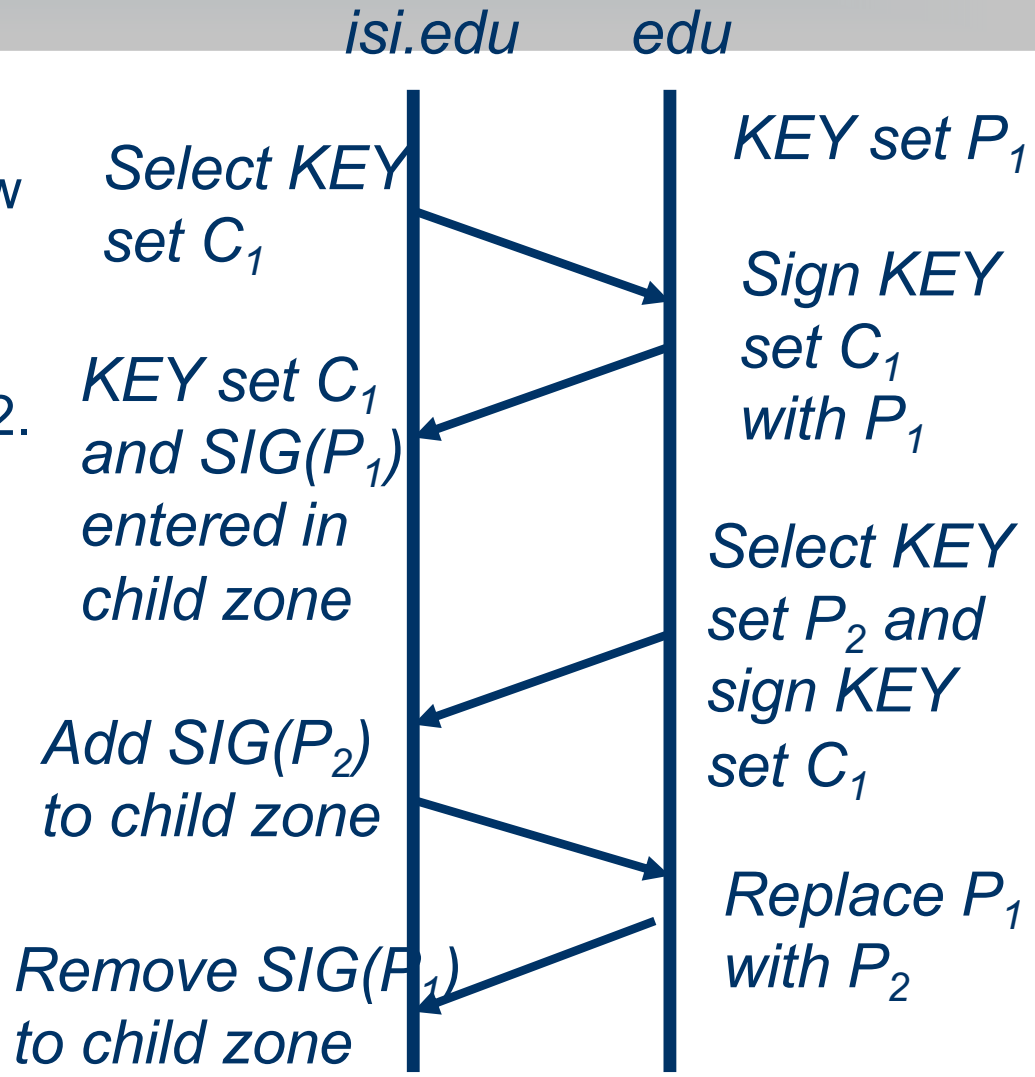


foo.com KEY record
SIG by com

foo.com DNS Server

An Attempt to Change the "edu" Key

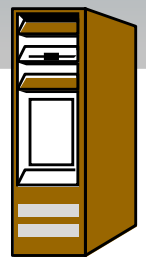
- Edu seeks to change from key P1 to key P2.
- Can't add P2 until getting a new RRSIG from root
- Can't remove P1 until all child zones have new RRSIGs by P2.
- Requires coordination with parent and all children.



Revising DNS Key Managment

- Operational Problems in RFC 2535 DNSSEC
  - USC/ISI led one of the first ***multi-administion*** testbeds.
  - Identified fundamental key management & scaling issues.
  - Revision now at the end of the standards process
    - Provide a coherent design that meets needs of vendors/operators
    - Currently co-editor of the IETF revision [1].
- Basic Principles Behind the DNS Revision
  - The DNS succeeded by de-coupling zones.
    - But authentication chains require coordination.
  - Store a hash (copy) of the child key at the parent.
    - Overcomes the DNS atomic RRset problem.
    - Manages authentication chains using operations similar to those currently used for maintaining server chains.

Revised DNS Key Management



darpa.mil NS records

Can Change mil key without notifying darpa.mil

darpa.mil DS record (hash of pubkey 1)

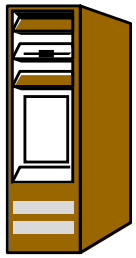
darpa.mil SIG(DS) by mil private key

mil DNS Server

---

darpa.mil DNS Server

darpa.mil DNSKEY (pub key 1)



darpa.mil DNSKEY (pub key 2)

darpa.mil RRSIG(A) by key 1

Can Change key 2 without notifying .mil

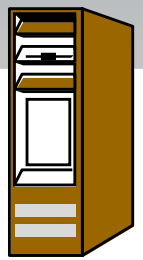
www.darpa.mil A record

www.darpa.mil RRSIG(A) by key 2

DNS "Key Signing" Key Rollover

darpa.mil DS record (hash of pubkey 3)

darpa.mil RRSIG(DS) by mil private key



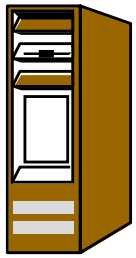
darpa.mil DS record (hash of pubkey 1)

darpa.mil RRSIG(DS) by mil private key

mil DNS Server

---

darpa.mil DNS Server



darpa.mil DNSKEY (pub key 1)

darpa.mil DNSKEY (pub key 2)

darpa.mil DNSKEY (pub key 3)

} DNSKEY 1 with new DNSKEY 3

darpa.mil RRSIG(A) by key 1

darpa.mil RRSIG(A) by key 3 }

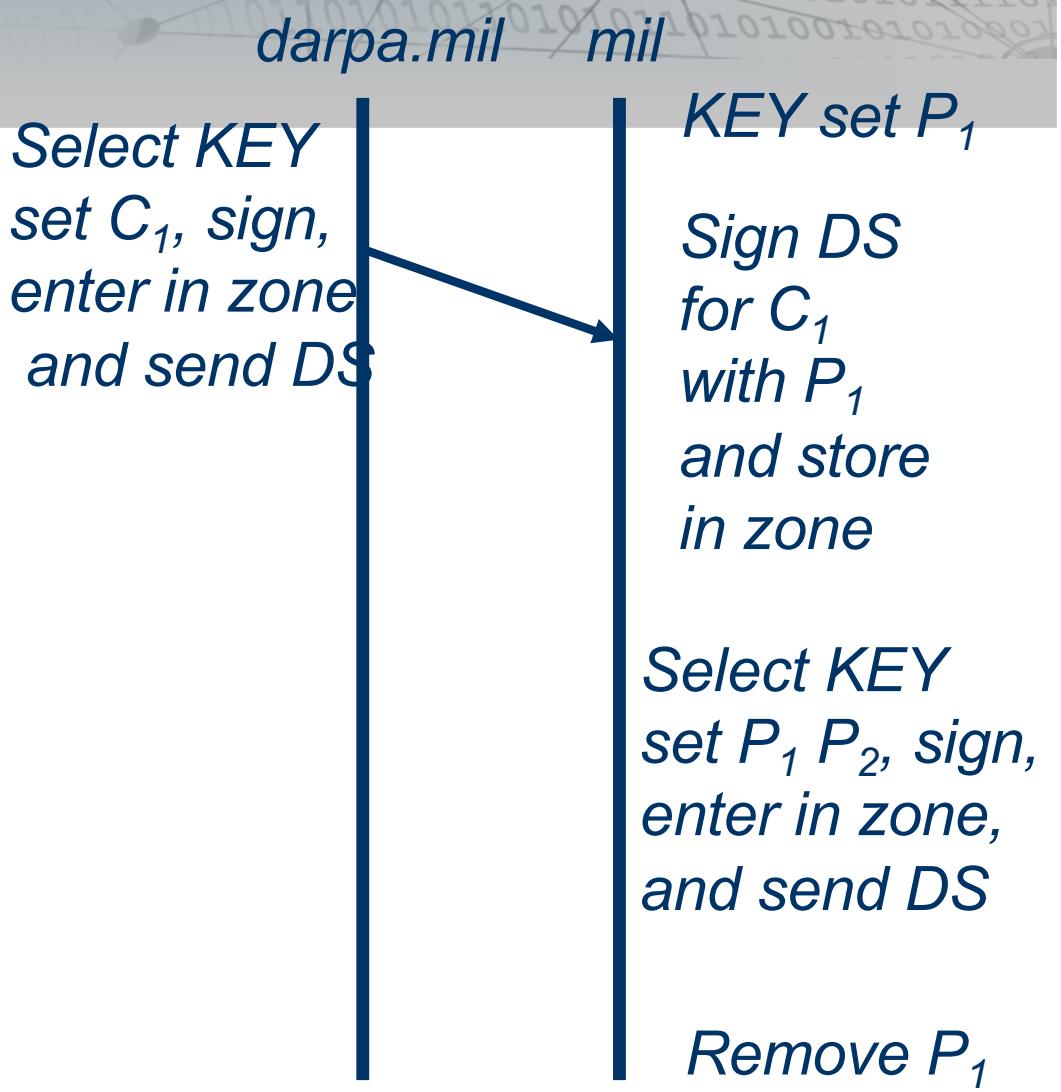
darpa.mil RRSIG(A) by key 3

Minimal Requirements

- Parent must indicate how to reach the child.
  - NS records at parent MUST identify at least one valid name server for child.
- Parent must identify a trusted key at child.
  - DS record at parent MUST match a valid KEY stored at the child.

Revised Key Change Details

- Figure shows process of a key change at mil.
- Figure assumes
  - secure communication
- Figure allows for:
  - darpa.mil key change
  - mil has any number of other child zones
  - mil parent (root) may or may not DNSSEC
- Single exchange process.
  - “Send DS along with NS”



Protocol Complications

- Building on an existing system
  - Objective is to strengthen the system.
  - But additions also add stress to weak points.
- Some example cases:
  - Denial of service added by the DS record.
  - NS records stored at the parent.
  - Over use of the KEY record.

Hidden Denial of Service

- DS Record is Stored Only at the Parent.
  - All DS records should be sent to parent.
  - What if you ask ucla.edu for the ucla.edu DS?
- Early BIND Implementation Choice:
  - Server says “I’m not authoritative for this data”.
  - Resolver hears “Not authortitative for ucla.edu”
- The Resulting Under-Specification Attack
  - Ask resolver to send DS query to each ucla.edu server.
  - Each server declared not authortiative for ucla.edu!
  - Query for www.ucla.edu now returns:
    cache says all ucla.edu servers are broken

Flaws in DNS Design (glue)

- Parent (*edu*) stores NS records for child (*isi.edu*)
  - Provides a ***hint*** on how to reach the child
- Child also stores a copy of the same NS RRset.
  - Child (*isi.edu*) is the authoritative source.
- Implications for Authentication:
  - Parent is not the authoritative source of the data.
  - Child data is not available if parent is wrong.
  - Resolver/cache can't distinguish between the set stored at the child and the set stored at the parent.
- DNSSEC Solution:

  Only the child signs its NS RRset

NS Records Stored at the Parent

- Edu server stores NS records for Colorado.edu
  - Tells a resolver where to find Colorado.edu servers.
- Colorado.edu server also stores Colorado.edu NS set.
  - Colorado.edu is the authoritative source.
  - Parent and child differ due to various reasons
- Loose coordination works since only requires some overlap in parent/child NS.
  - Security assumes identical.
  - Security also says parent can’t sign NS set.
    - Parent is not the authoritative source of the data.

Over Use of the KEY Record

- Original KEY record used for DNSSEC and IPSEC, email, TLS, etc.
- Resolver looking for DNSSEC key gets all possible keys.
  - Data (ipsec key) should be separate from control (DNSSEC) key.
  - Fixed in RFC 3449 that limits KEY to DNSSEC.

Wild Cards

- Authenticated denial further complicated by wildcards.
  - Must prove “b.darpa.mil” does not exist
  - Must also prove no wildcard would match.
- Algorithm for computing necessary wildcards now available in revised specification.
  - Pathological cases require many NXT records.
  - Motivates one further change…
- Proposal to add bit indicating:
  next (signed?) name after “a” is “c”
  and no wildcards apply in between

Moving DNSSEC to the End Host

- DNS security design worked with DNS resolvers and servers.
  - Default was to send DNSSEC data.
  - Verified old resolvers and servers ignore it.
- But DNS interacts with many other pieces.
  - Firewalls did not ignore the extra records.
    - Our subcontract was blackholed by DARPA.
    - DARPA firewall thought the SIG records were an attack!
- DNSSEC OK bit added to protocol
  - Resolver sets this bit indicated DNSSEC data is okay.

Heilmeier Questions

- What are you trying to do? Articulate your objectives using absolutely no jargon.
- How is it done today, and what are the limits of current practice?
- What is new in your approach and why do you think it will be successful?
- Who cares? If you succeed, what difference will it make?
- What are the risks?
- How much will it cost?
- How long will it take?
- What are the mid-term and final “exams” to check for success?