SecSpider: Distributed DNSSEC Monitoring and Key Learning

Eric Osterweil

UCLA

Joint work with
Dan Massey and **Lixia Zhang**
Colorado State University & UCLA

Who is Deploying DNSSEC?

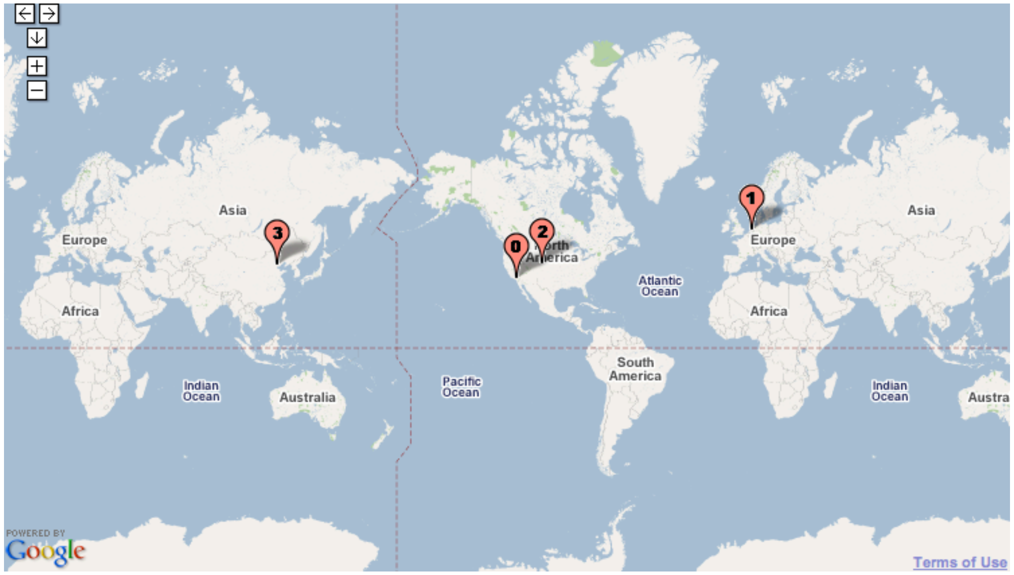
- Monitoring Started From Close to Day One
  - DNSSEC RFCs published in March 2005
  - Monitoring launched in October 2005
- Find Zones Using Crawling and User Submissions
  - Continually crawl DNS looking for secure zones
  - Nightly NSEC walking (until NSEC3 is here)
  - Allow users to submit the names of secure zones

Why Are We Monitoring?

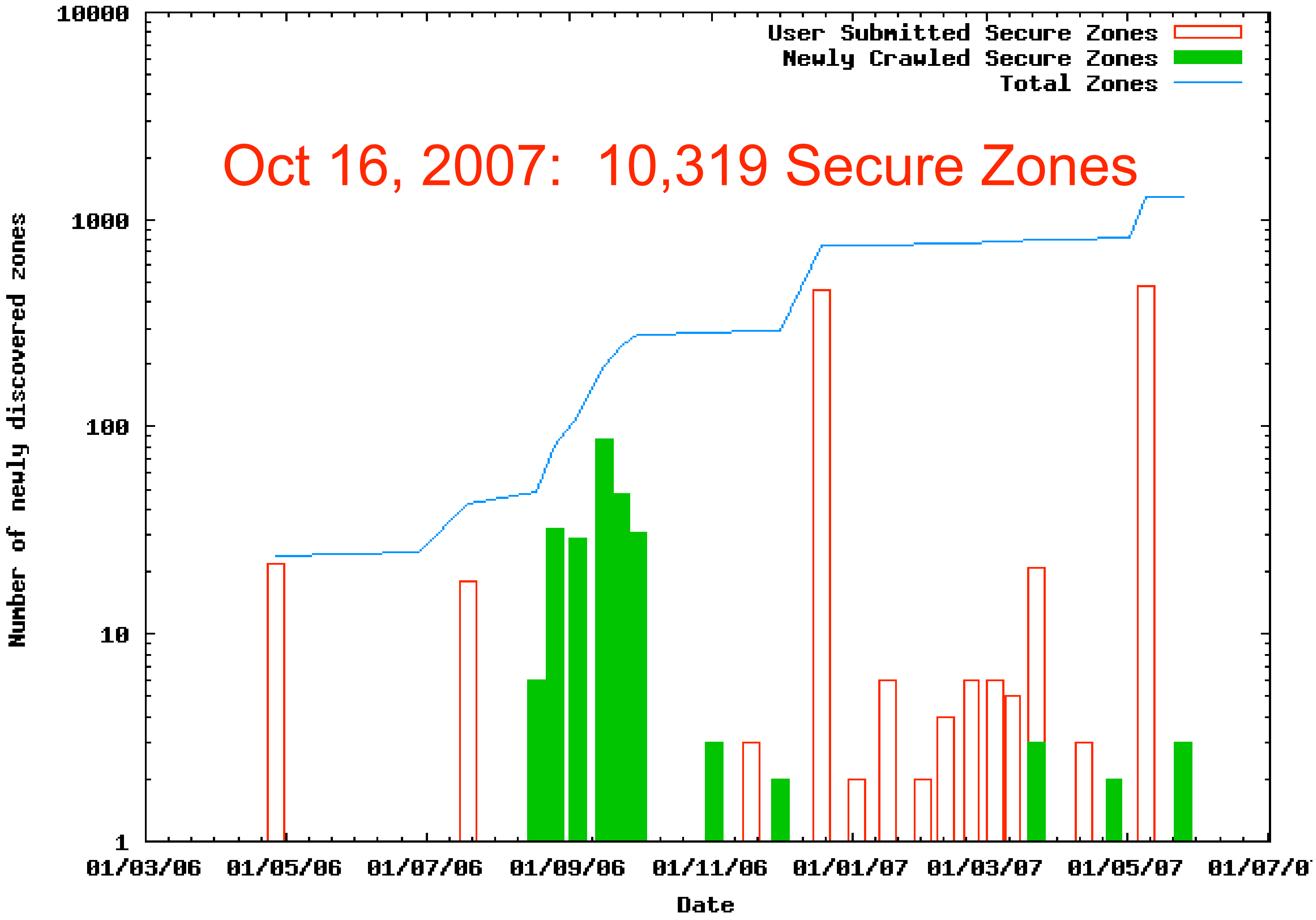
- Keep a historical record of the rollout
  - Tracking the use of crypto, etc
- Analyze behaviors and practices
- Offer a service to the community
  - Feedback always helps with this one ;)

What's New?

- SecSpider v2.0
  - Distributed polling
  - Flat files for DNSKEYs/DS records
  - And more…



DNSSEC Deployment



Oct 16, 2007: 10,319 Secure Zones

Deployment Observations

- (Undirected) Crawling DNS Finds Few Secure Zones
  - Vast DNS + tiny DNSSEC => low (near 0) hit rate for crawler
  - Example: last night's crawl status:

    8,177,214 insecure zones and 187 secure zones
- User Submissions Drive Current Monitoring
  - SecSpider is well publicized => high submission rate
  - Augment secure zones with parent/child and popular sites
- Trend is positive, but still very small deployment overall
  - Some top level domains deploying or deployed (e.g. "se." zone)
  - Not yet at critical mass for DNSSEC

A Closer Look at Secure Zones

- Monitor Closely Tracks All Secure Zones
  - Frequent Queries to Monitor Changes
  - Exploit DNSSEC zone walking
  - Still tractable due to relatively small DNSSEC deployment
- Monitoring Reveals Many Challenges: DNSSEC deployment is not simple after all
  - Challenge in Islands of Security
  - Challenge in Key Management
  - Challenge in Preventing Replays

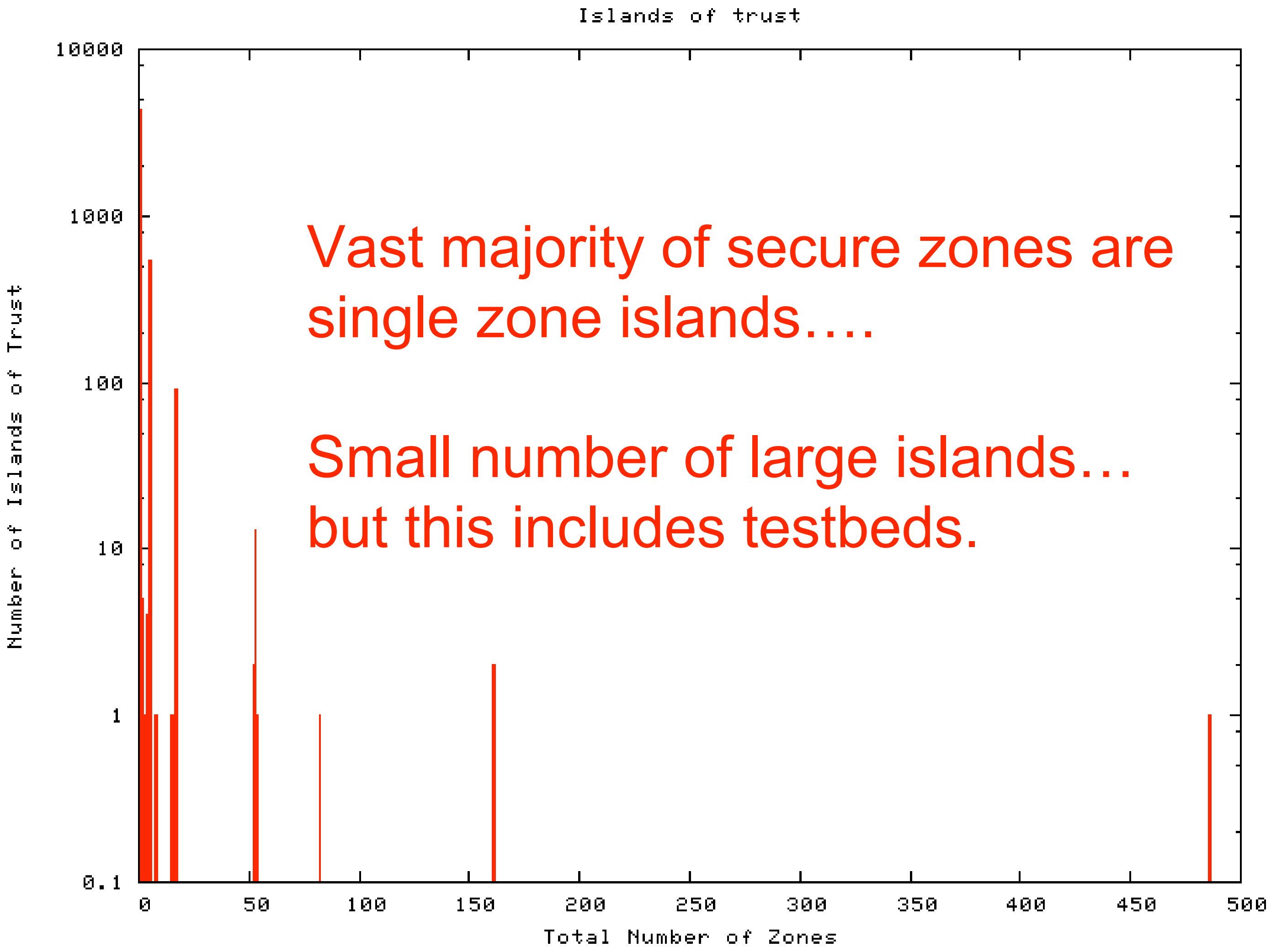
Challenge 1: Islands of Security

- DNS relies on the tree hierarchy to learn public keys
  - Everyone knows root public key
    - But how would this happen and who manages it?
  - Root key used to sign edu public key
    - But neither root or edu have public keys now….
  - edu key used to sign ucla.edu key
    - But no hierarchy leads to the public key?
- How does a resolver learn a secure zone’s public key?

Challenge 1: Islands of Security

- Island of Security: DNS sub-tree where every zone in the sub-tree has deployed DNSSEC
- Design envisioned a single island of security
  - All zones deploy DNSSEC and manually configure the root key
- Monitoring reality shows disconnected deployments
  - DNSSEC deployed in isolated subtrees and must manually configure the public key for each ***island of security***

Islands of Security

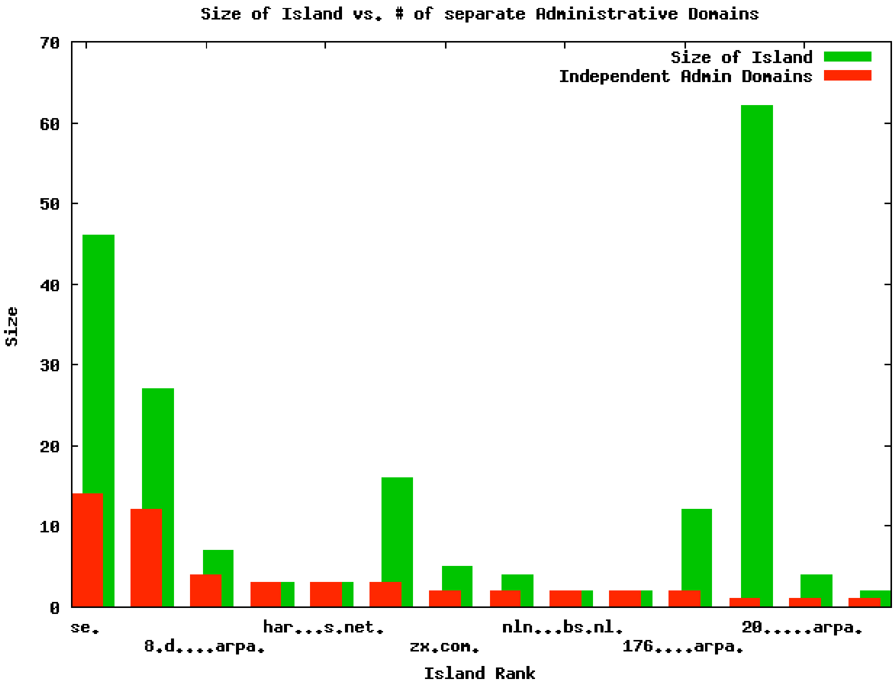


Vast majority of secure zones are single zone islands….

Small number of large islands… but this includes testbeds.

Production Islands

- When focusing on "production zones"
- Many of the larger zones are served by only a few unique NS+A sets
  - Few organizations serving many zones?
- 14 islands greater than size 1 out of 634 total



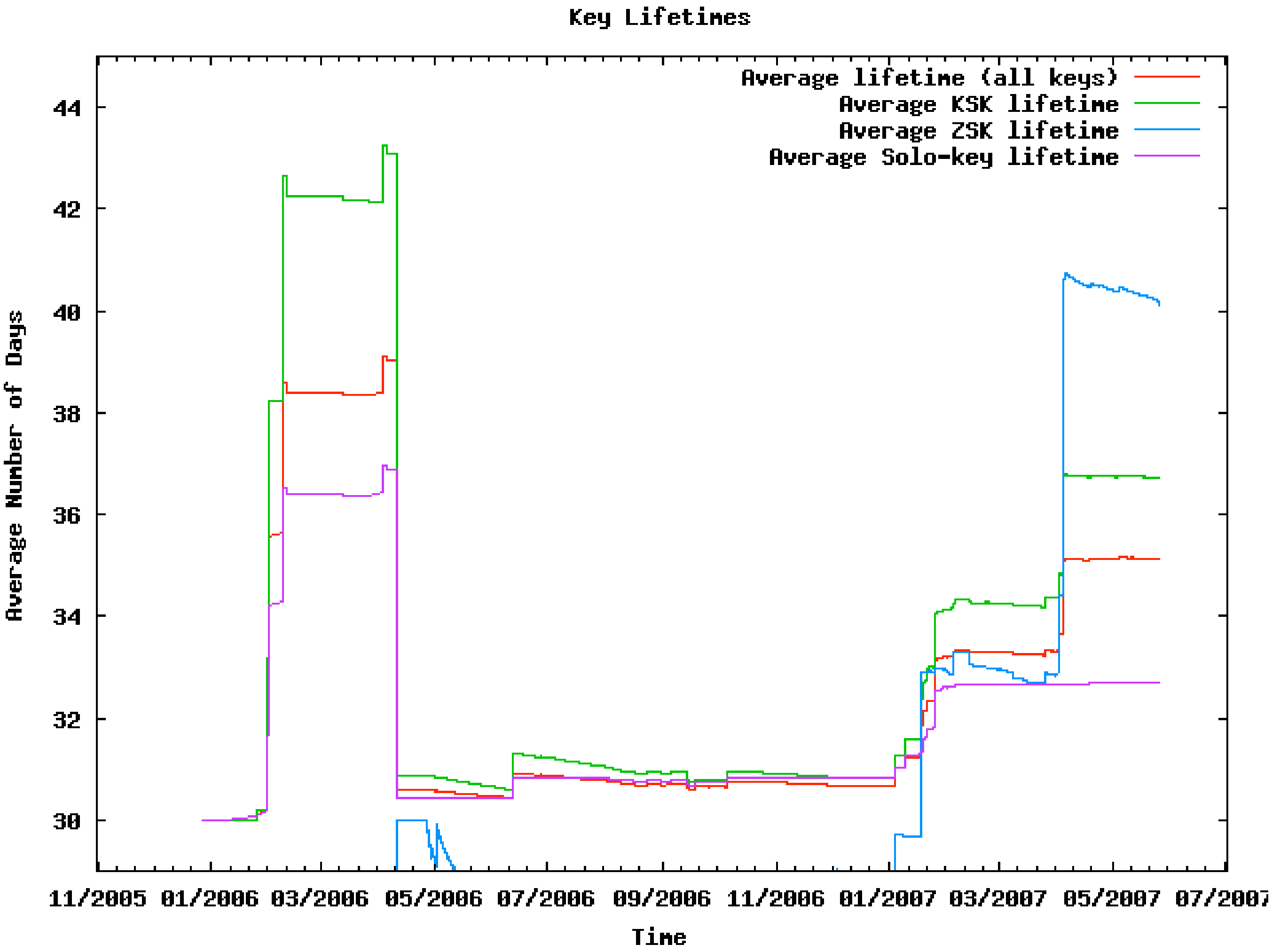
Addressing Islands of Security

- Deploy DNSSEC at all zones or at least from root down
  - Has yet to happen operationally…..
- Develop an Alternative PKI?
  - DLV provides some service to store and report public keys
- Can we trust the public keys visible at the monitor?
  - Must ensure keys came from monitor
  - Must ensure monitor was not tricked…
  - But can rely on distributed services and checking by actual admins….

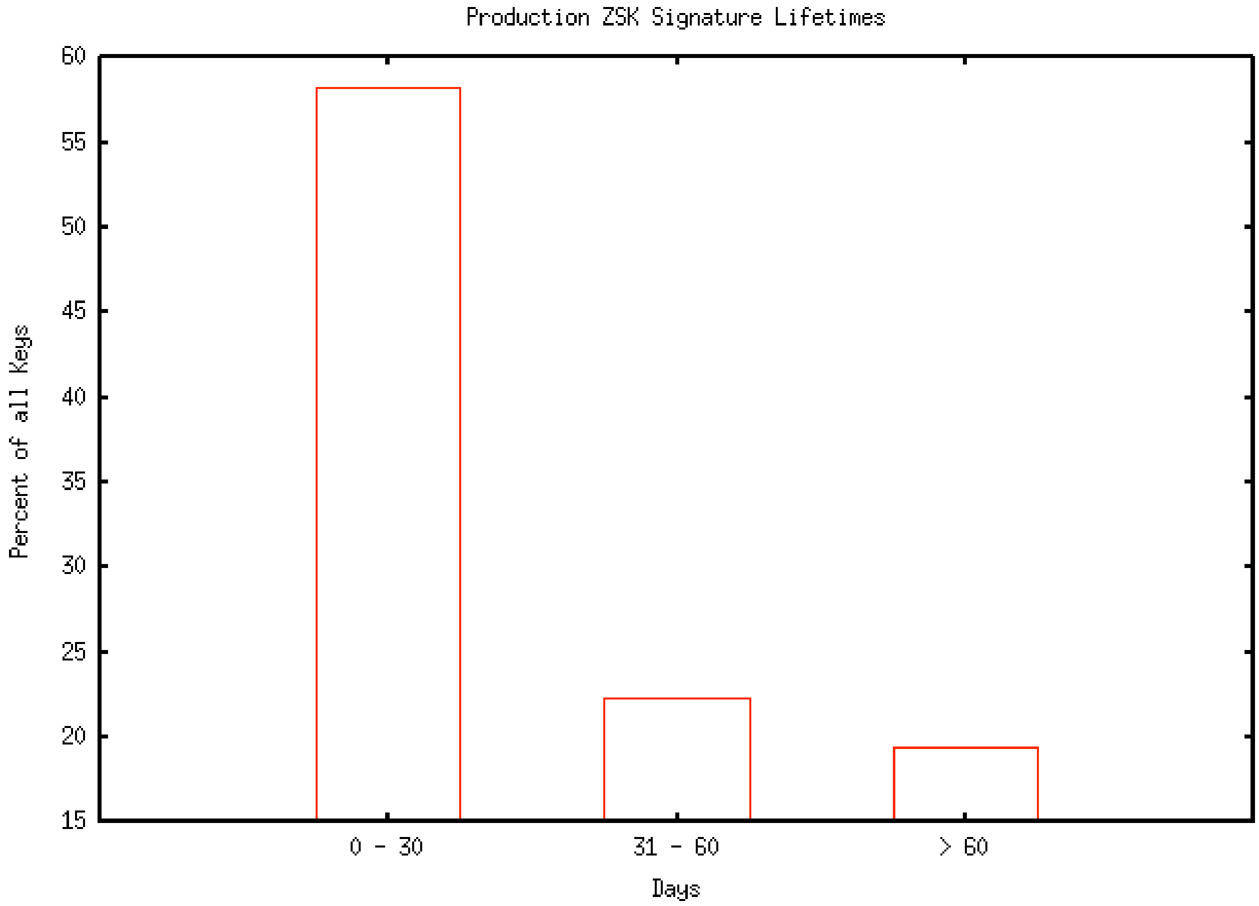
Challenge 2: Key Management

- Design is Relatively Simple, But Operations are complex
  - Establish key pair and sign the zone
    - Relatively straight-forward, but issues below add challenges..
  - Establish an Authentication Chain with a Secure Parent
    - Cross-domain coordination with a different administration
  - Update the key pair periodically
    - Due to planned changes or key compromise
- Simple concept of parent private key signs the child public key….   But many complex details

Average Key Sig Lifetimes



Signature Lifetimes on ZSKs



Key Sig vs Actual Lifetimes

- Sig lifetimes -> Actual average lifetime
  - 0-30 days -> 102.651 days
  - 31-60 days -> 68.9527 days
  - > 60 days -> 395.085
- Pruning keys that have not expired yet
  - 0-30 days -> 83.2043 days
  - 31-60 days -> 209.19 days
  - > 60 days -> 156.762 days

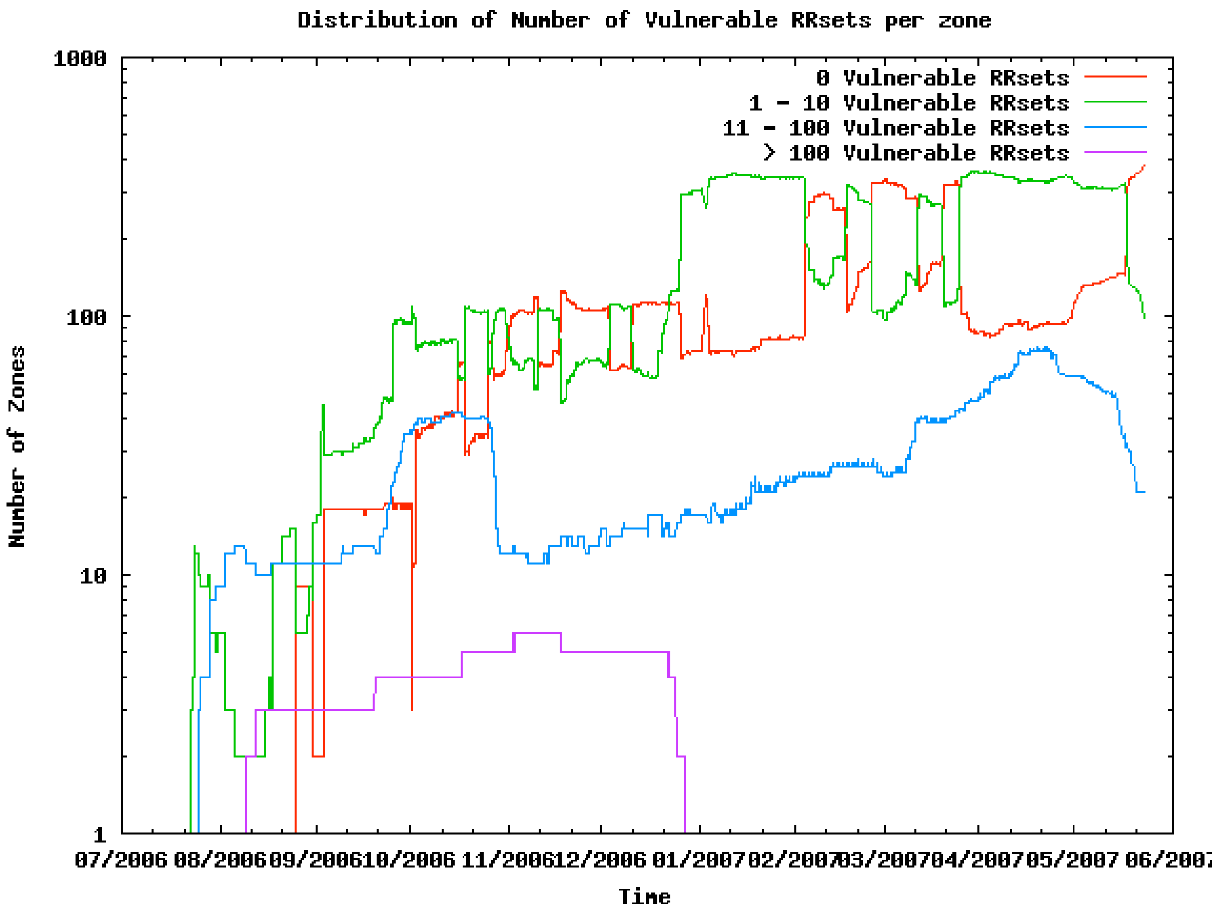
Addressing Key Management

- Manual operation of complex steps is unrealistic
  - Need to improve management tools and increase automation
    - Dnssec-tools.org, hznet.de, etc
  - Also need to overcome off-line key issues
- Match operations with monitoring
  - Must have monitoring to provide external view of zone
  - Must have some form of correctness check
  - Monitoring data can aide in the automation process by checking which steps have been done
    - Ex: detect when the DS record at the parent has changed

Challenge 3: Lifetimes&Replays

- Each cryptographic signature has a fixed lifetime
  - Ex: Signature for www.foo expires on Nov 31.
  - What if the addresses changes today?
- Actions Taken in the DNS
  - Server removes changed record and replaces with new copy
  - But attacker can still replay the old record and signature
- Vulnerable Records: data has changed, but the signature on old copy has not yet expired
  - Vulnerable records can be replayed and resolver will authenticate the old copy

Vulnerable DNS Record Sets



Addressing Lifetimes & Replays

- With sufficient prediction, vulnerable records can be avoided
  - Make signature lifetime match data lifetime
- Dramatic Improvement Coincided With Monitoring
  - Vulnerable records greatly reduced in current data

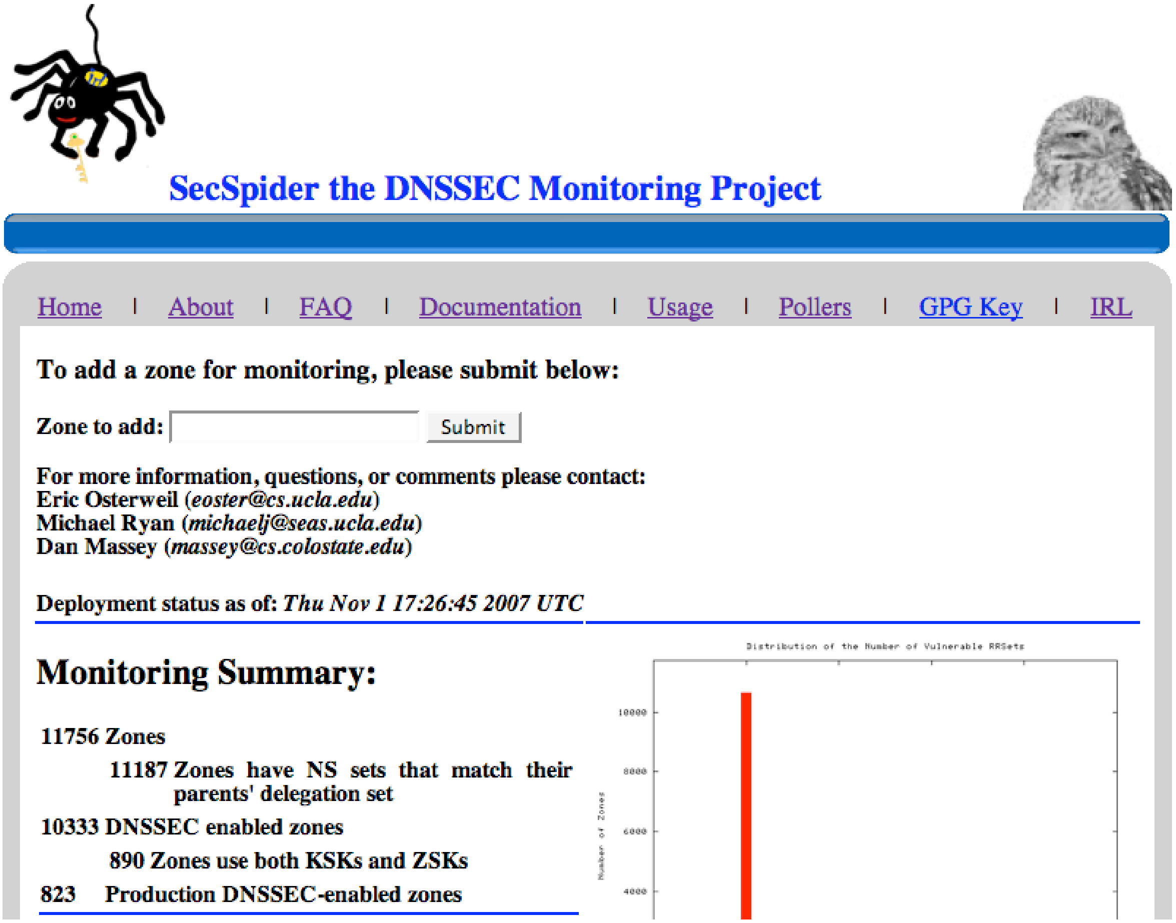
The Role of Monitoring

- Monitoring is essential is large-scale systems
  - Monitoring illustrates extent of known issues in deployment
  - Monitoring identifies new challenges in deployment
- SecSpider Monitoring Benefits DNSSEC
  - Illustrates progress and documents scale of known issues
  - Identifies new challenges
  - Allows zone admins to see how others perceive them
    - Various examples of how monitoring led to changes
- Systems operations don't always match expectations
  - Monitoring has helped us see this with DNSSEC

Monitoring Solutions and Future Directions

- Challenge 1: Islands of Security
  - Distributed monitor can be used to bootstrap public key information
  - Challenge is to authenticate public keys came from monitor and limit chance that all monitors' data is subverted by attacker
- Challenge 2 and 3: Cryptographic Management
  - Given an external view of data, zone admins can adapt
  - Monitoring can verify key management is working
  - Monitoring can aide in automating DNS key management
- Current work is using SecSpider data to identify new challenges and *practically solve existing challenges*

http://secspider.cs.ucla.edu/



SecSpider the DNSSEC Monitoring Project
Home | About | FAQ | Documentation | Usage | Pollers | GPG Key | IRL
Submit
Distribution of the Number of Vulnerable RRSets
Number of Zones
10000
8000
6000
4000

To add a zone for monitoring, please submit below:

Zone to add:

For more information, questions, or comments please contact:
Eric Osterweil (*eoster@cs.ucla.edu*)
Michael Ryan (*michaelj@seas.ucla.edu*)
Dan Massey (*massey@cs.colostate.edu*)

Deployment status as of: ***Thu Nov 1 17:26:45 2007 UTC***

Monitoring Summary:

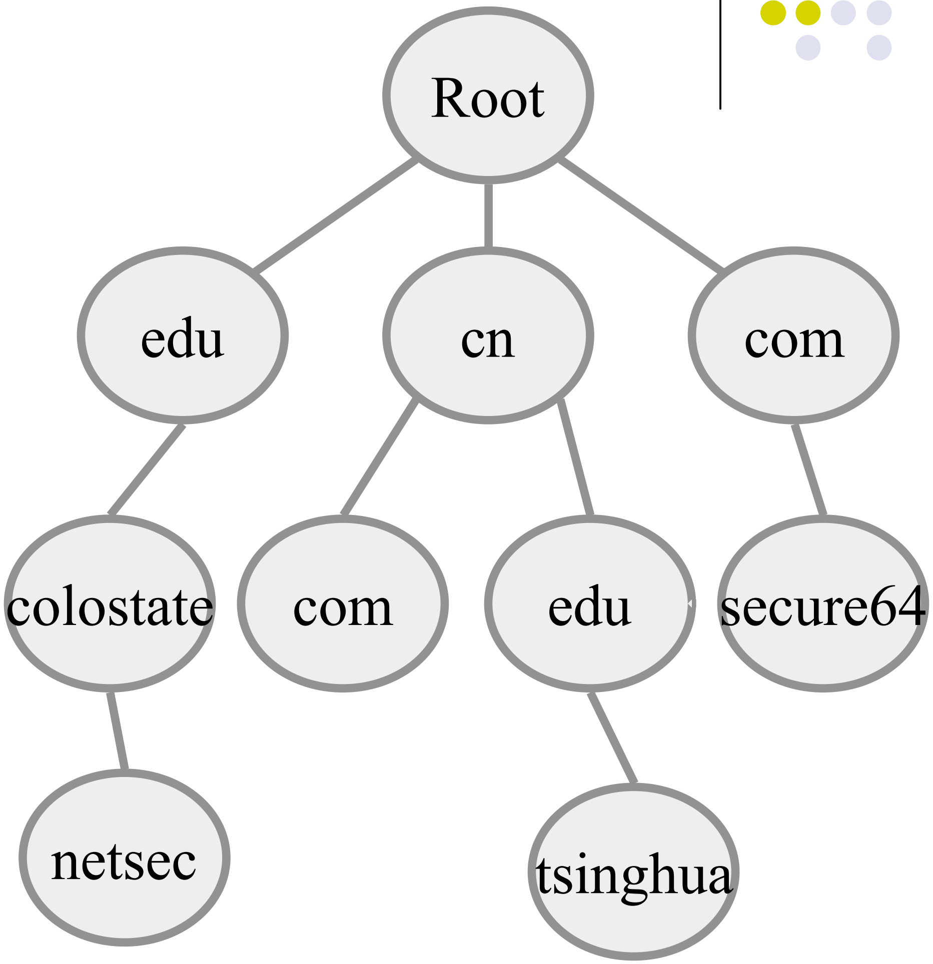
11756 Zones
11187 Zones have NS sets that match their parents' delegation set
10333 DNSSEC enabled zones
890 Zones use both KSKs and ZSKs
823 Production DNSSEC-enabled zones

Thank You!

Backup

The Domain Name System

- Virtually every application uses the Domain Name System (DNS).
- DNS database maps:
  - Name to IP address
    www.netsec.colostate.edu = 129.82.138.2
  - And many other mappings (mail servers, IPv6, reverse…)
- Data organized as tree structure.
  - Each zone is authoritative for its local data.

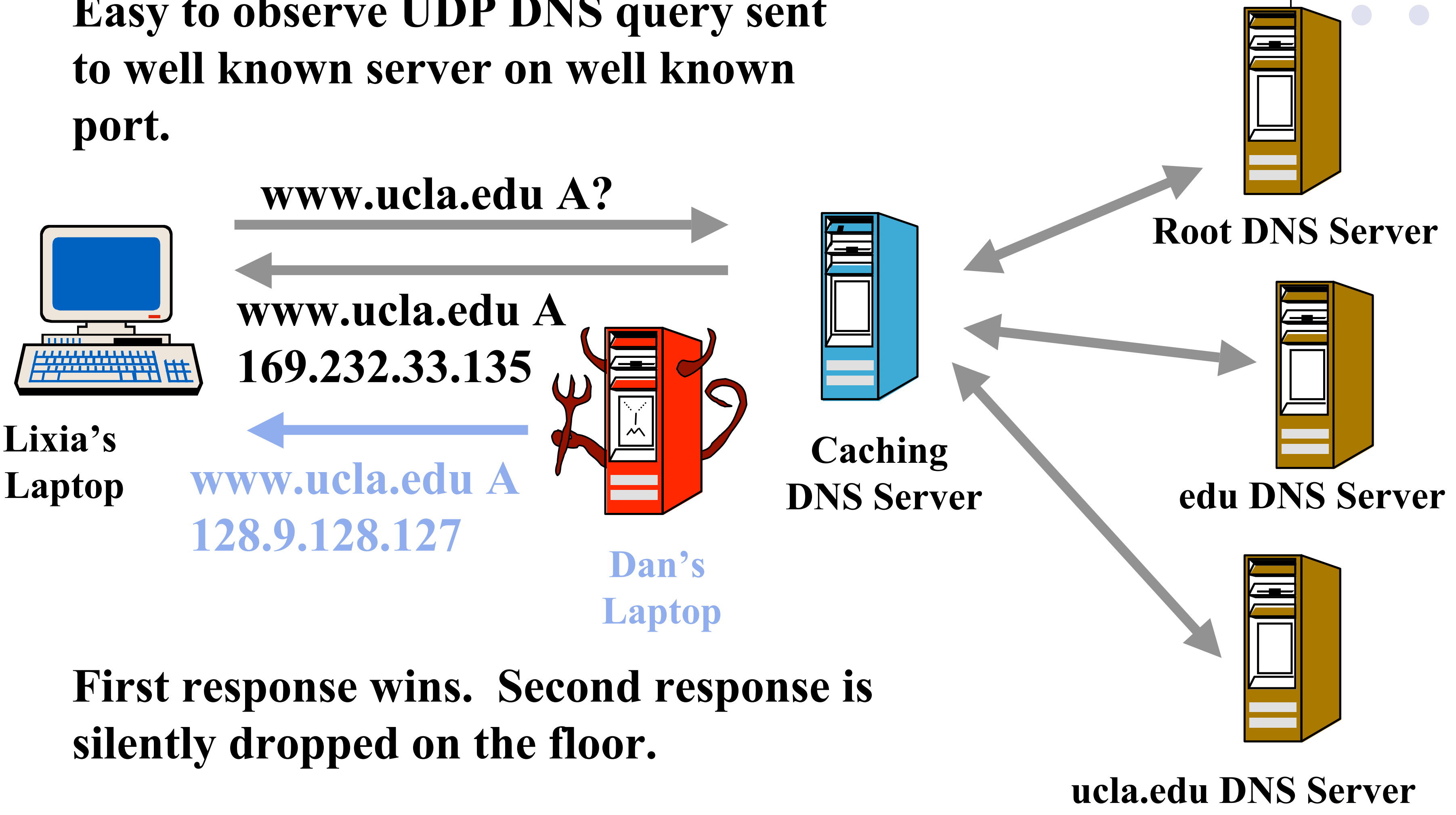


DNS Vulnerabilities

- 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.

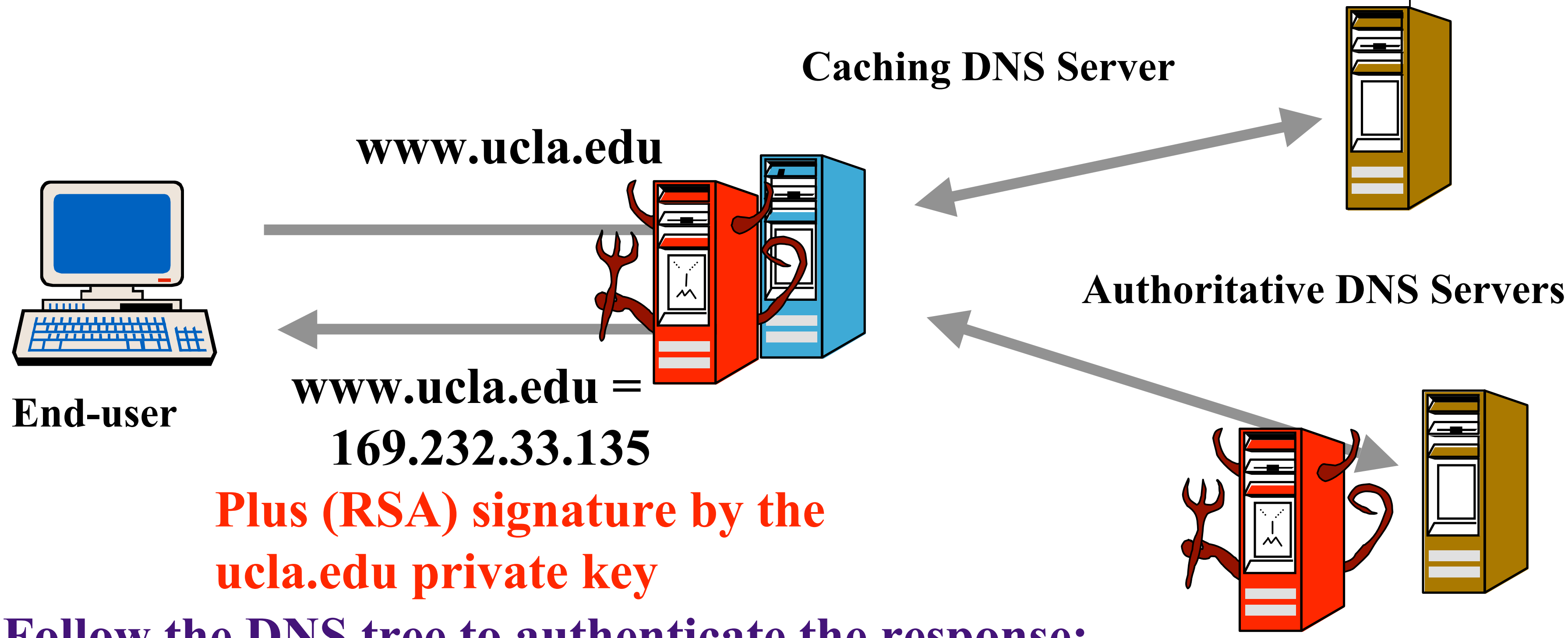
A Simple DNS Attack

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.

Secure DNS Query and Response



Caching DNS Server

www.ucla.edu

End-user

www.ucla.edu = 169.232.33.135

Authoritative DNS Servers

Plus (RSA) signature by the ucla.edu private key

Follow the DNS tree to authenticate the response:

1) Assume root public key is well known
2) Root key signs edu key
3) edu key signs ucla.edu key
4) ucla.edu key signs the data

The Overall DNSSEC Design

- Simple Combination of DNS and public key cryptography
- Each zone manages its own key pair
- DNS Tree Hierarchy leveraged to form a PKI
- Standardized in RFC 4033, 4034, and 4035
  - Currently supported by most DNS implementations

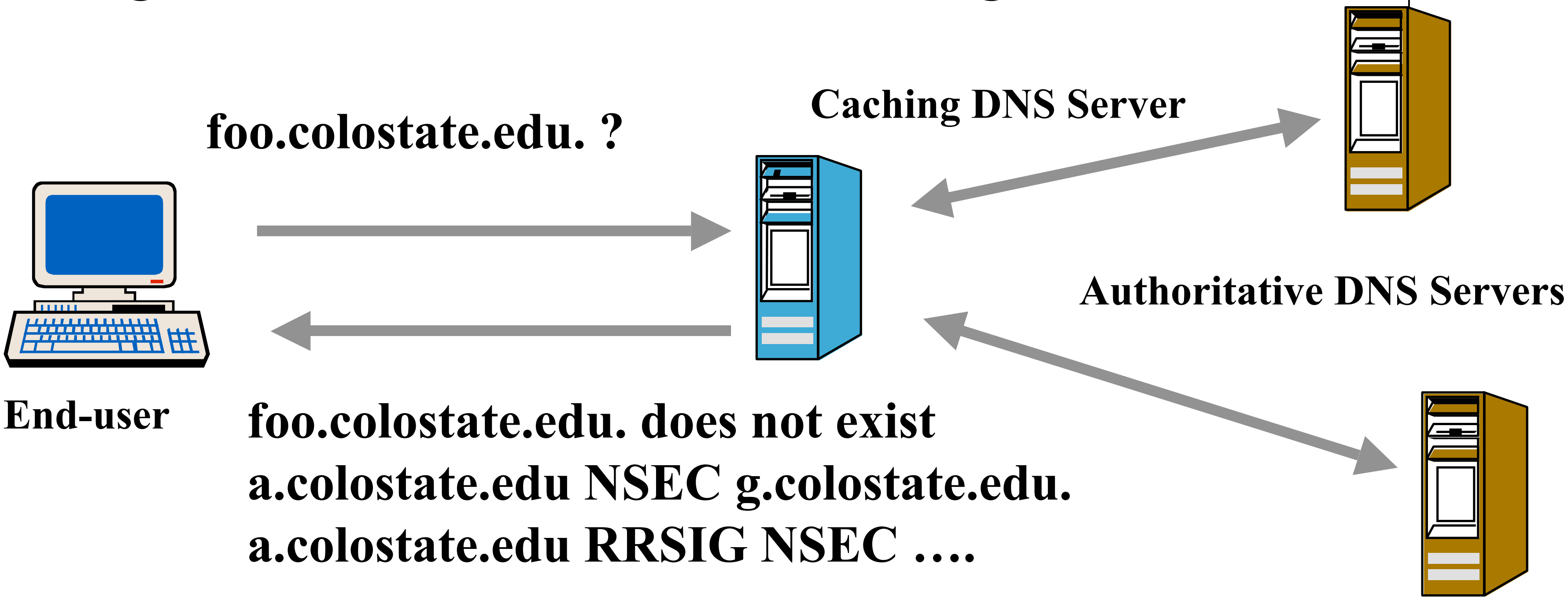
Authenticated Denial of Existence

- What if the requested record doesn't exist?
  - Query for foo.colostate.edu returns "No such name"
  - How do you authenticate this?
- Must return message that proves a name does not exist….
  - But cannot predict what ***non-existent*** names will be queried.
  - And cannot sign message for specific non-existent name since private key off-line

Zone Walking and Monitoring

Solution:

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



Caching DNS Server

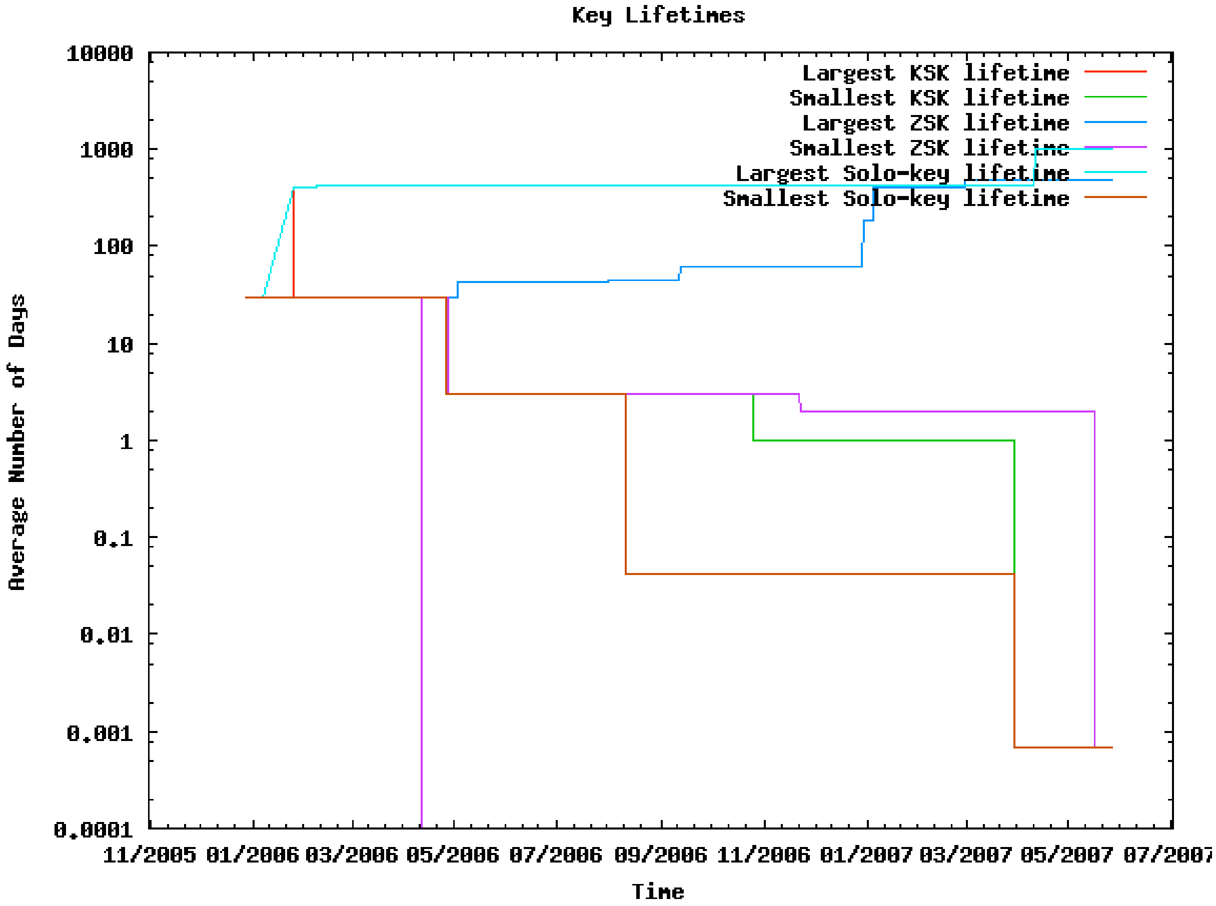
foo.colostate.edu. ?

Authoritative DNS Servers

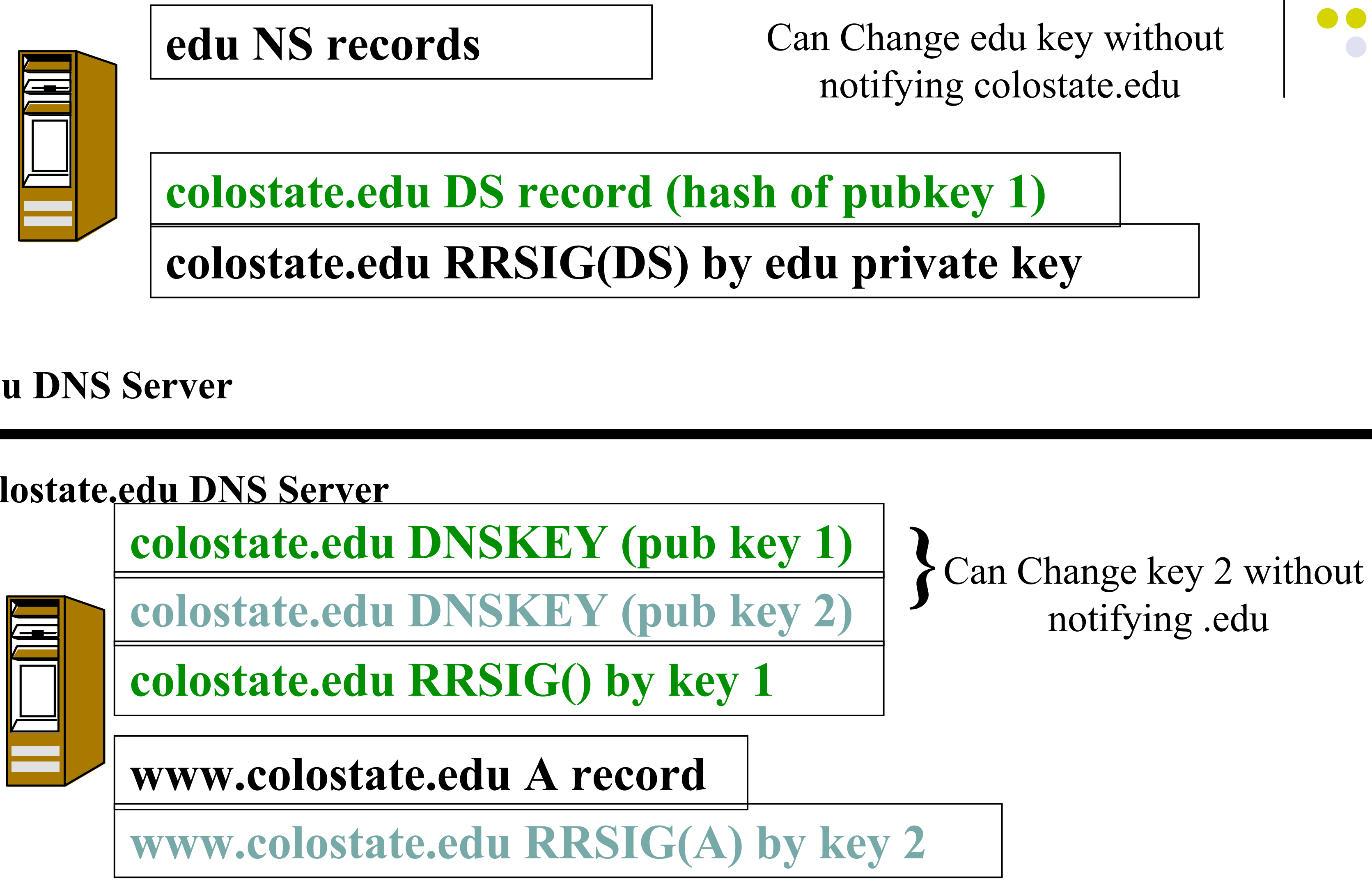
End-user

foo.colostate.edu. does not exist
a.colostate.edu NSEC g.colostate.edu.
a.colostate.edu RRSIG NSEC ….

Minimum and Maximum Values



DNS Key Management



edu NS records

Can Change edu key without notifying colostate.edu

colostate.edu DS record (hash of pubkey 1)

colostate.edu RRSIG(DS) by edu private key

edu DNS Server

colostate.edu DNS Server

colostate.edu DNSKEY (pub key 1)

colostate.edu DNSKEY (pub key 2)

colostate.edu RRSIG() by key 1

Can Change key 2 without notifying .edu

www.colostate.edu A record

www.colostate.edu RRSIG(A) by key 2

DNS Key Signing Key Roll-Over

colostate.edu DS record (hash of pubkey 3)

colostate.edu RRSIG(DS) by edu private key

colostate.edu DS record (hash of pubkey 1)

colostate.edu RRSIG(DS) by edu private key

edu DNS Server

colostate.edu DNS Server

colostate.edu DNSKEY (pub key 1)

colostate.edu DNSKEY (pub key 2)

colostate.edu RRSIG(A) by key 1

colostate.edu DNSKEY (pub key 3)

colostate.edu RRSIG(A) by key 1

colostate.edu RRSIG(A) by key 3

colostate.edu RRSIG(A) by key 3

Objective: Replace DNSKEY 1 with new DNSKEY 3