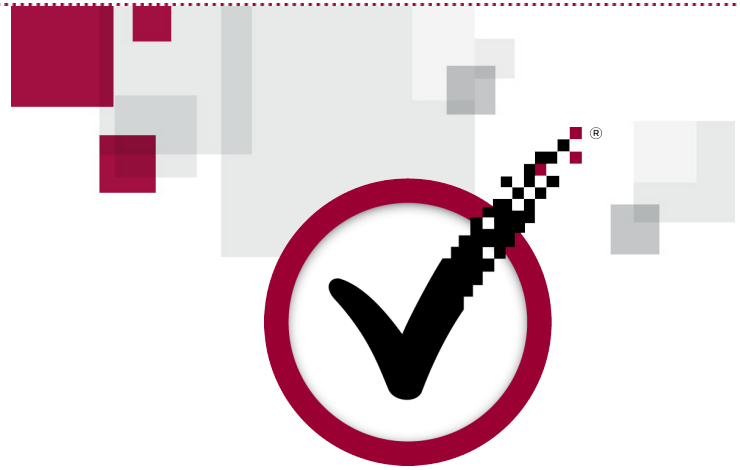
VeriSign®



The Evolution of DNSSEC's Global Rollout

Eric Osterweil

What is DNSSEC Good For?

- The DNS is a core Internet protocol that maps names to other content:
  - IP addresses, mail servers, SPF policies, etc.
- DNSSEC enhances DNS by overlaying crypto
  - Name learning hierarchy is overloaded for crypto key learning
- Crypto verification is recursively performed throughout the hierarchy

- However -

- The overall protections are affected by the operational deployment

Level Setting

- DNSSEC is the first true internet-scale cryptographic system
- Its global rollout has offered a couple of key lessons:
  - The added operational complexities of cryptography have been non-trivial
  - Crypto can cause unforeseen interactions with the network
- DNSSEC has experienced several growth-spurts and also a number of measurable challenges
  - Distributed monitoring has allowed us to learn as we've deployed

Outline

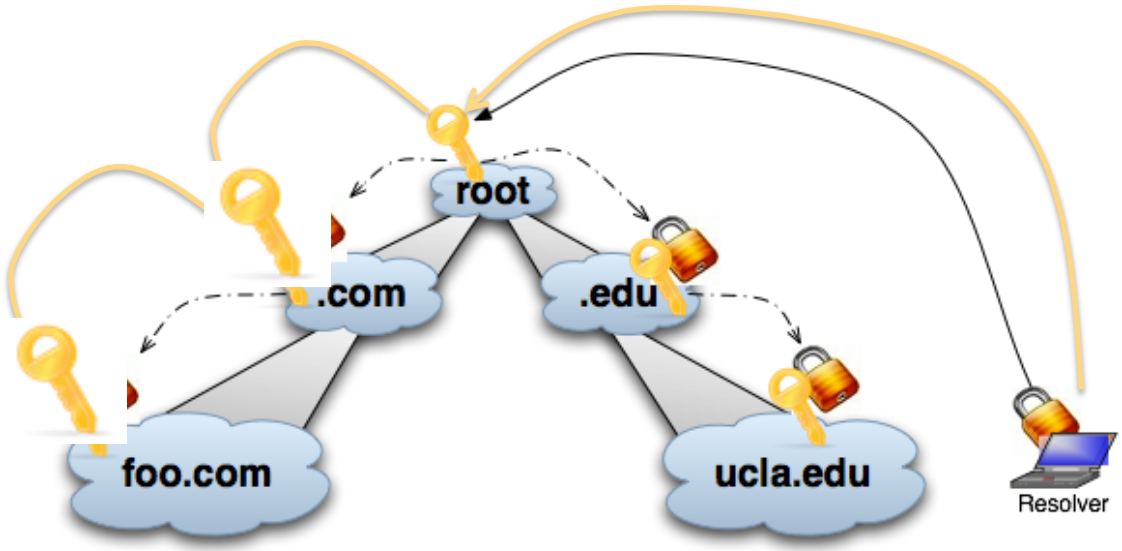
- How DNSSEC works
- State of DNSSEC's Deployment
- Operational Complexities: Managing keys
- Unforeseen Interactions with the Network
- Addressing Challenges
- Summary

DNSSEC

- DNSSEC provides *origin authenticity, data integrity,* and *secure denial of existence* by using public-key cryptography
- Origin authenticity:
  - Resolvers can verify that data has originated from authoritative sources.
- Data integrity
  - Can also verify that responses are not modified in-flight
- Secure denial of existence
  - When there is no data for a query, authoritative servers can provide a response that proves no data exists

How DNSSEC Works

- DNSSEC zones create public/private keys
  - Public key is DNSKEY

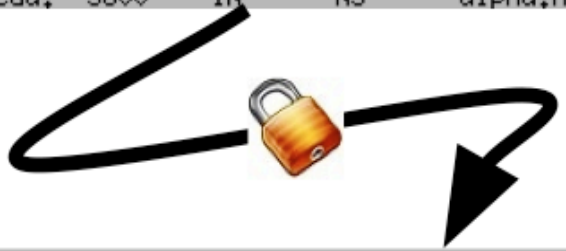


- Zones sign all RRsets and resolvers use DNSKEYs to verify them
  - Each RRset has a signature attached to it: RRSIG
- Resolvers are configured with a single *root* key, and trust flows recursively down the hierarchy

Data Signing Example

Using a zone's key on a standard RRset (the NS)

```
secspider.cs.ucla.edu.  3600    IN      NS      zinc.cs.ucla.edu.
secspider.cs.ucla.edu.  3600    IN      NS      alpha.netsec.colostate.edu.
```

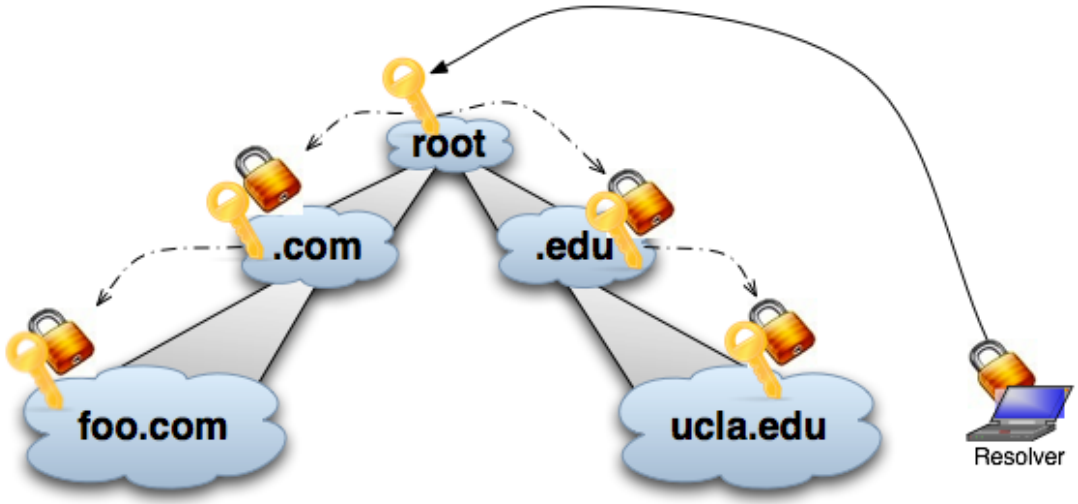


Signature (RRSIG) will only verify with the DNSKEY if *no* data was modified

```
secspider.cs.ucla.edu.  3600 IN NS alpha.netsec.colostate.edu.
secspider.cs.ucla.edu.  3600 IN NS zinc.cs.ucla.edu.
secspider.cs.ucla.edu.  3600 IN RRSIG NS 5 4 3600 20080324024800 (
                                20080322024800 44736 secspider.cs.ucla.edu.
                                E4msde1nzVlfGvwDo2X6jLU5d9Xrk37lrYRCZN6yq5ad
                                mABa3B3KgKl13u2VBXDujJZucHSwPQMBy+JOmotZOggf
                                SgQWUYm86v8G7ABHHcI+YFD3z3eqSoAoBAE5ysafop1u
                                g7tw1J4xd/IADIVeu1HnVIKRSycILXzvCwcaDWwAd610
                                9oJUBSMgWZjGzYeJO9RzOoUUqIqtn9PgVOzdTm+WnRC3
                                LEz50fdoP743QvPhe7RrF9w1KA3MOptTiQA++W8Gg085
                                NhbJ7MD99nEYaEv3+GuDCTkCy5ZOWoI/2Bcjq1NGBDLo
                                71lo6udu72i1tpyRfTEEQUirpInlZ9+IMw== )
```

Getting the Keys

- Until a resolver gets DNSKEY(s), data can be spoofed
- Keys verified by secure delegations from parents to children
  - So resolvers know DNSKEYs are not being spoofed



- DNSSEC's design needs the *full* hierarchy in order to verify keys
  - No middle ground: either a key has a delegation, or you know nothing about it

Growth of DNSSEC's Deployment

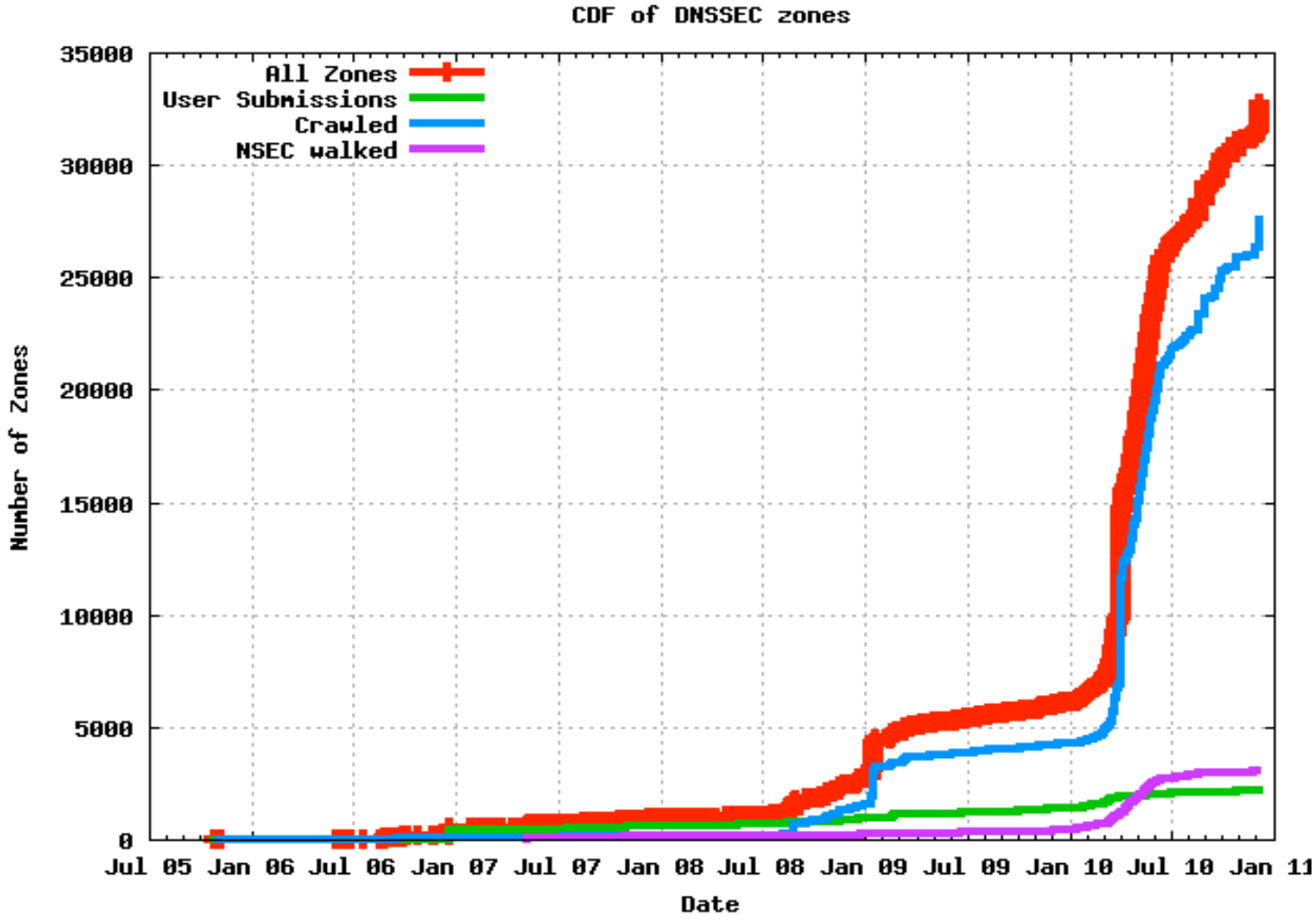


Figure from SecSpider http://secspider.cs.ucla.edu/

Size of the Deployment

- How the 30,102 DNSSEC zones have been found
  - Phase 1: crawling a corpus of DNS zones from a search engine
  - Phase 2: user submissions (thank you!)
  - Phase 3: NSEC walking
- Search engine's crawl size is 35,213,902 which suggests that DNSSEC's deployment is somewhere around 0.1%
- This suggests the deployment challenges and the economic incentives may not be well aligned

Grass-Roots Have Been Essential

- Several Top Level Domains (TLDs) took early initiative
  - .se in 2005, .pr in 2006, .br and .bg in 2007, .cz and .museum and .gov in 2008
  - 19 TLDs deployed in 2009, including .org
  - Today 61 TLDs are signed, including .net!!
- Zones under these TLDs have more incentive to deploy
  - The delegation chain exists
  - In January 2009, over 1,000 .se domains signed
  - In March 2010, several thousand .cz signed
- The deployment's bulk has come from grassroots / early adopters and the *rate* has been affected by large events

Timing of DNSSEC's Growth

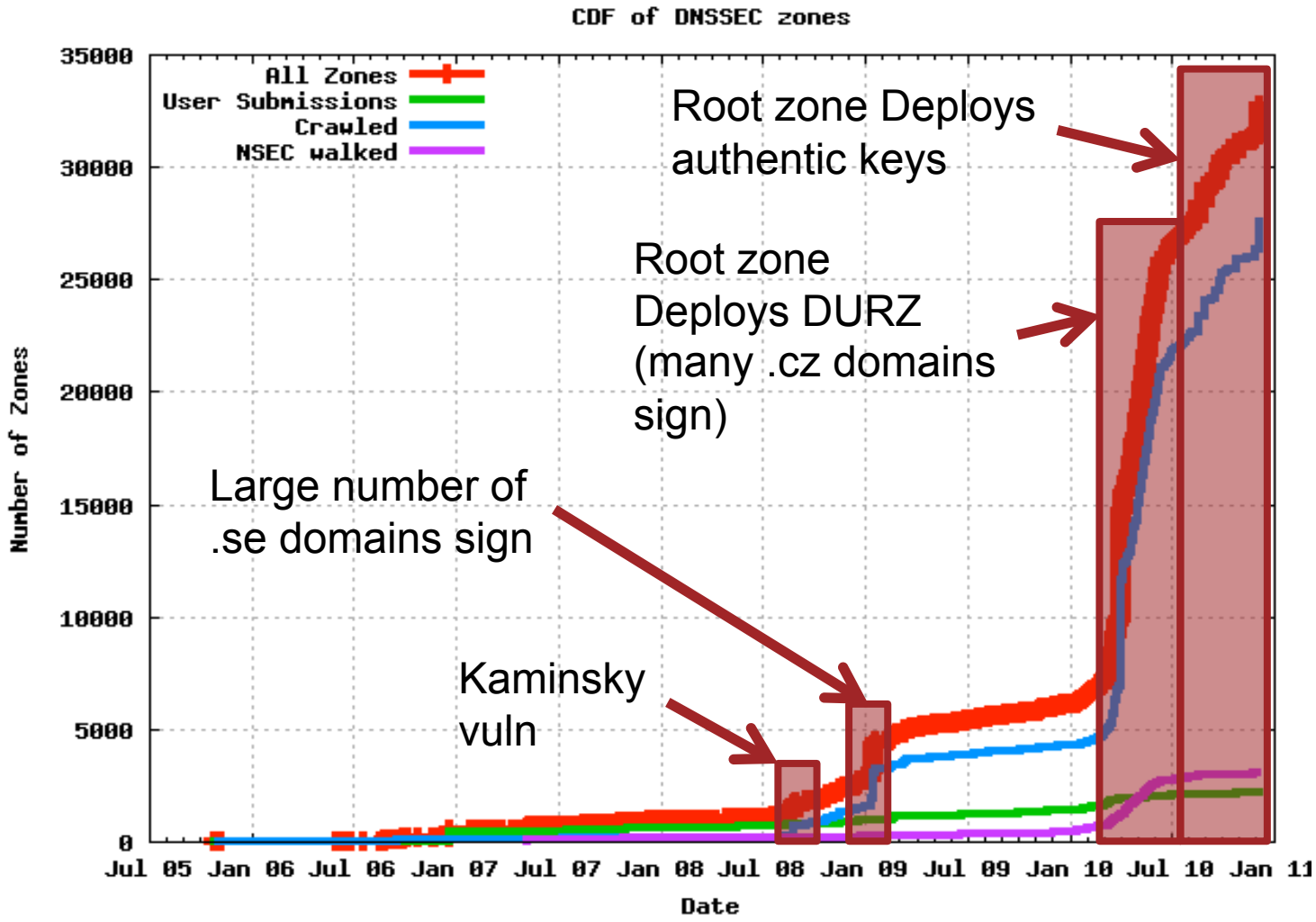
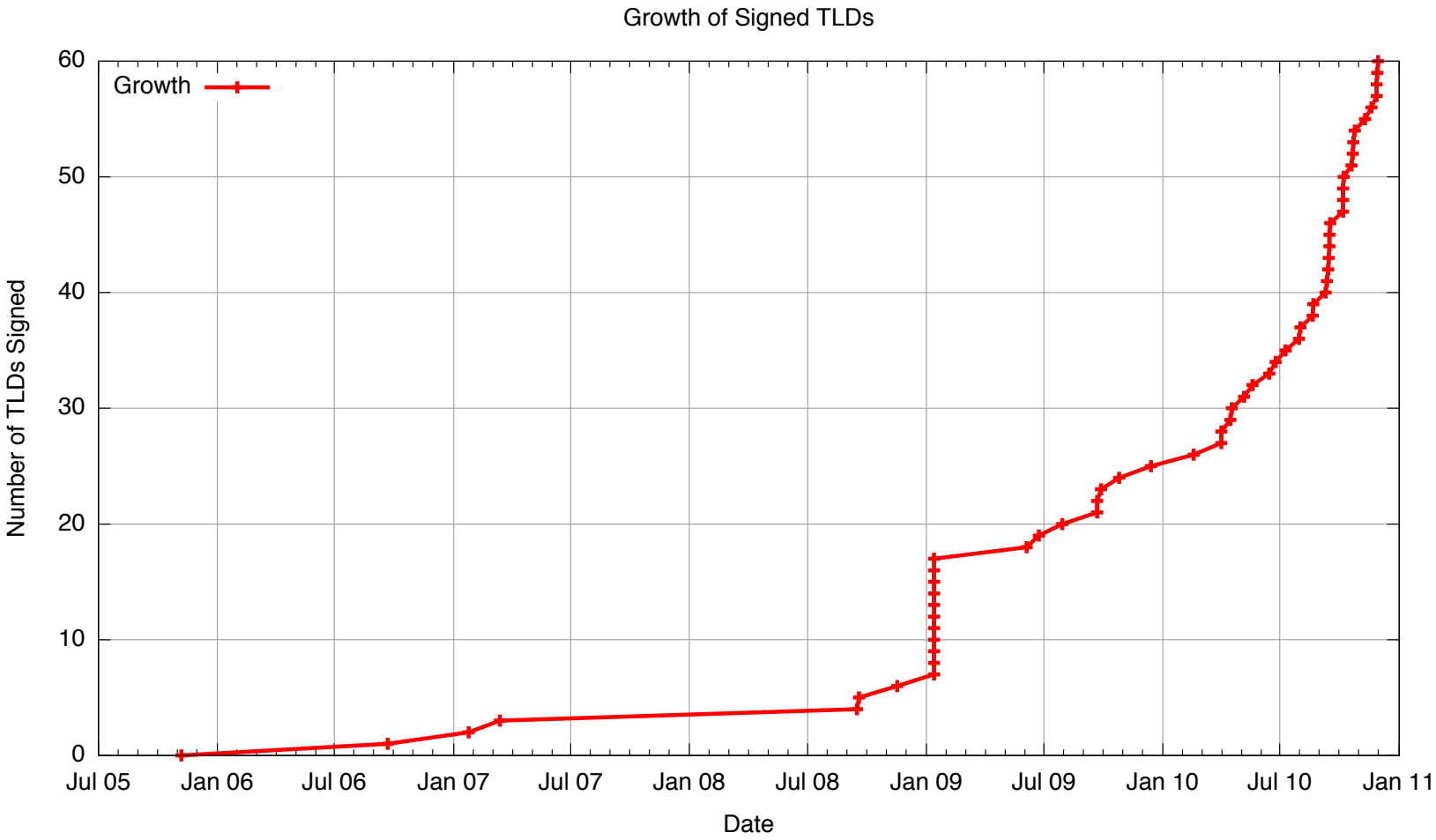
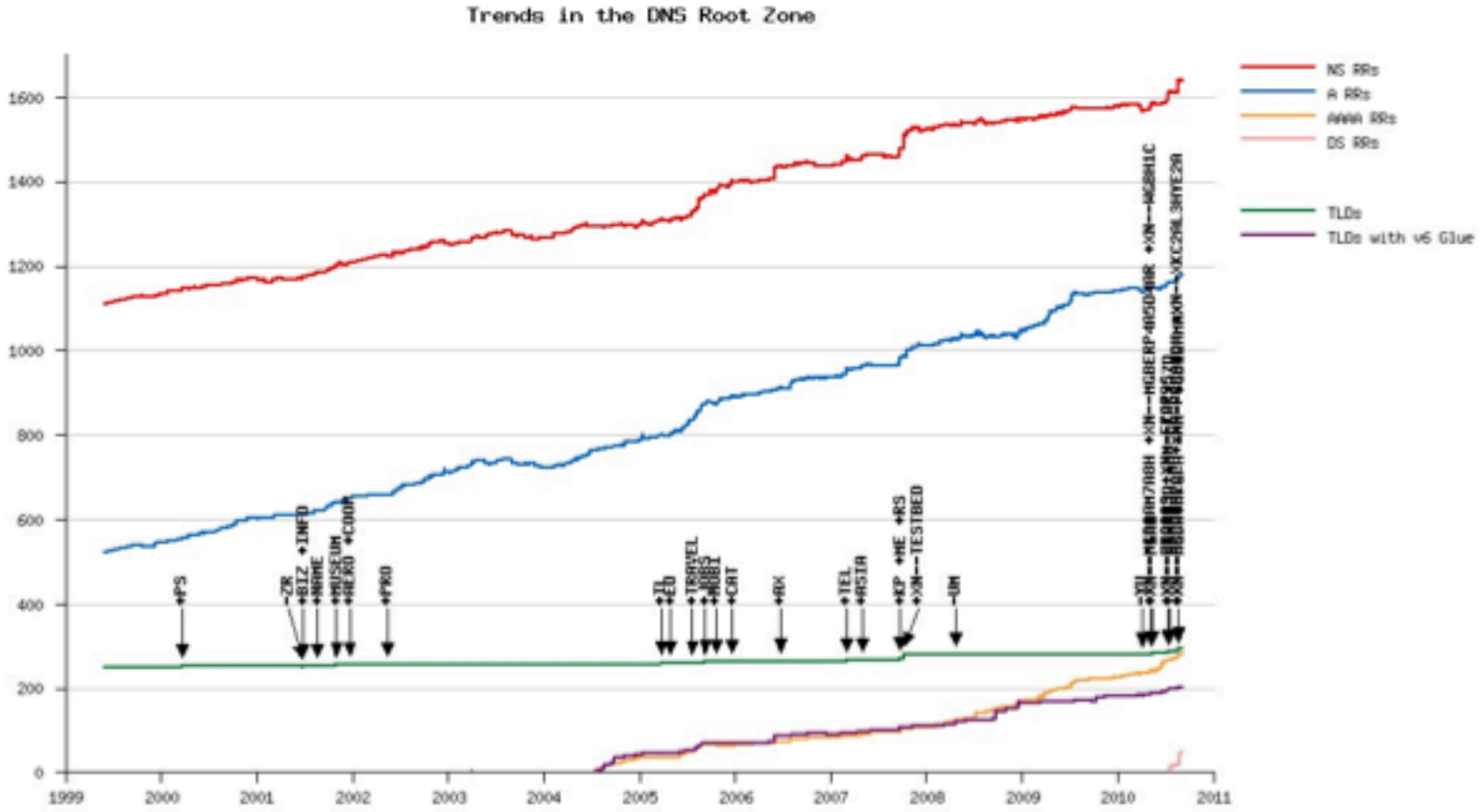


Figure from SecSpider http://secspider.cs.ucla.edu/

Growth Rate of Signed TLDs



Growth in the Root Zone



- As the root zone is growing, IPv6 uptic and DNSSEC delegations are showing accelerated adoption

Regarding the Deployment

- DNSSEC's protections come from more than its protocol's correctness
  - Monitoring lets us discover and diagnose its behavior under real operational conditions
- We can see a correlation between large public events and growth spikes
  - Perhaps periodic alignment of costs and incentives
  - But does misalignment of costs and incentives explain the plateaus?
- Posit: the operational complexity often dissuades adoption

Challenges Facing DNSSEC So Far

- To paraphrase Paul Mockapetris:

  "Deploying [cryptography] for a database does not deploy a corresponding amount of expertise"

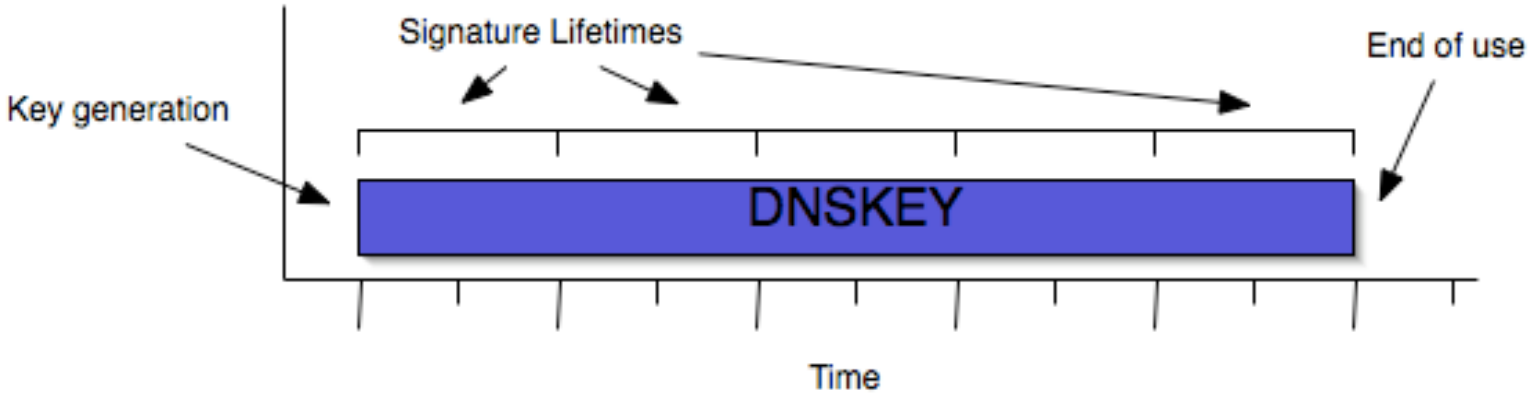
- It adds new crypto-related operational learning curves
  - Managing crypto (key rollovers, algorithm rollovers, etc.)
  - Protecting crypto keys (offline, HSMs, etc)
- Unforeseen interactions at the network layer
  - Large messages / PMTU interactions
  - "Middlebox" interoperability (or lack thereof)
- The road from the protocol's specification to the global rollout has already given us a lot to learn from

Operational Complexity: DNSKEYs

- DNSSEC adds a number of new operational requirements
- Among them is managing cryptographic keys
- DNSKEYs are not used the same way as other records
  - For example, DNS A records aren't subject to crypto analysis or being stolen
- Thus, in DNSSEC, more thought must be invested in managing keys than other record types

Managing Keys: Key Lifetimes

- There is a distinct difference between the signature lifetimes of DNSKEYs and the actual period of use
  - A key with a short signature lifetime can be re-signed indefinitely!

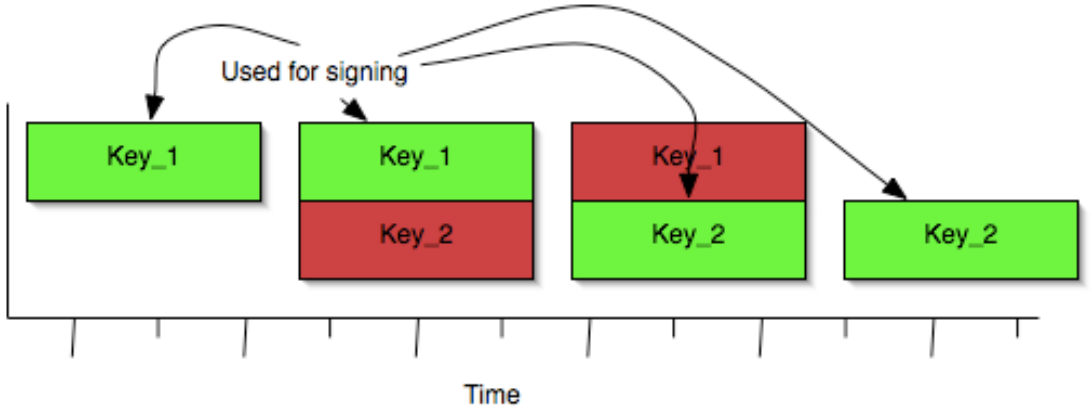


Key Rollovers

- DNSSEC's philosophy is that keys should be replaced periodically
- When keys become compromised, they must be removed/revoked/etc
- Conventional wisdom says that keys should not be used indefinitely
  - RFC 4641 suggests prolonged use of keys increases the probability of "compromise…"

How Keys *Should* be Changed

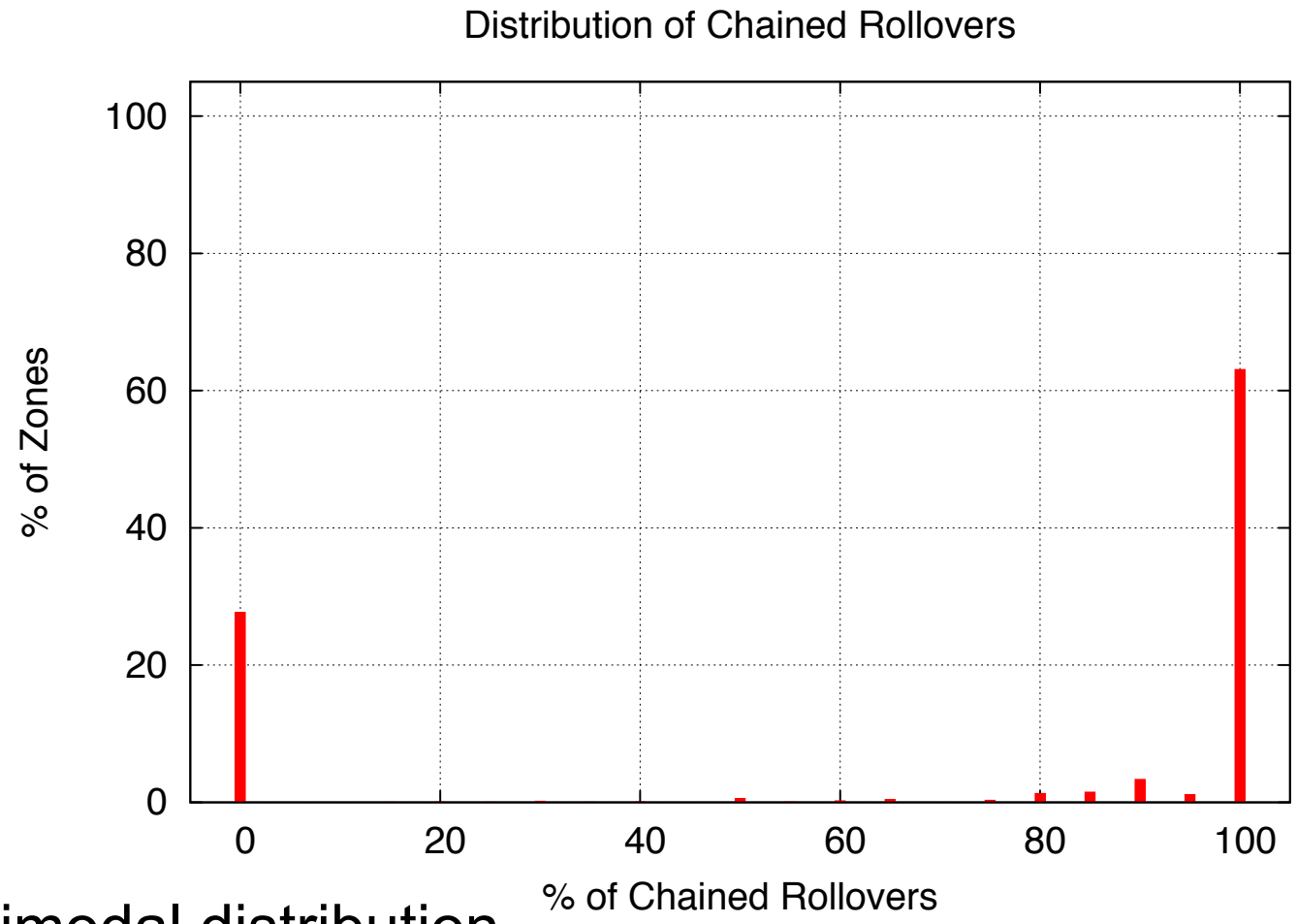
- When key changes are needed, old keys need to overlap with new keys
  - Chained rollovers
- Sign with Key_1 ➔ add Key_2 ➔ Sign with Key_2 ➔ Stop serving Key_1



Otherwise…

- Removing a key too soon can cause validation failures
  - If caches only have signatures from a recently removed key, resolvers may not be able to verify data
- Key changes must be chained
  - Until all signatures from a key have expired, a zone must serve that key
  - Otherwise resolvers may encounter data that seems false
- So, how often are people chaining?

Managing Key Rollovers



- Bimodal distribution
  - Some operations seem to be struggling with rollovers

Addressing this Type of Challenge

- One of the main problems is the new operational complexity
  - Key rollovers are a new type of operational requirement
  - No clear analog in operating a plain old DNS zone
- Other operational challenges also exist:
  - Algorithm rollovers
  - Managing offline keys
  - DS record synchronization / rollover
- Tool suites like: BIND-tools, ldns utilities, Vantages have arisen to address and ease these problems

But What Happens to DNSSEC on the Wire?

- Operational / configuration complexity are not the only problems posed

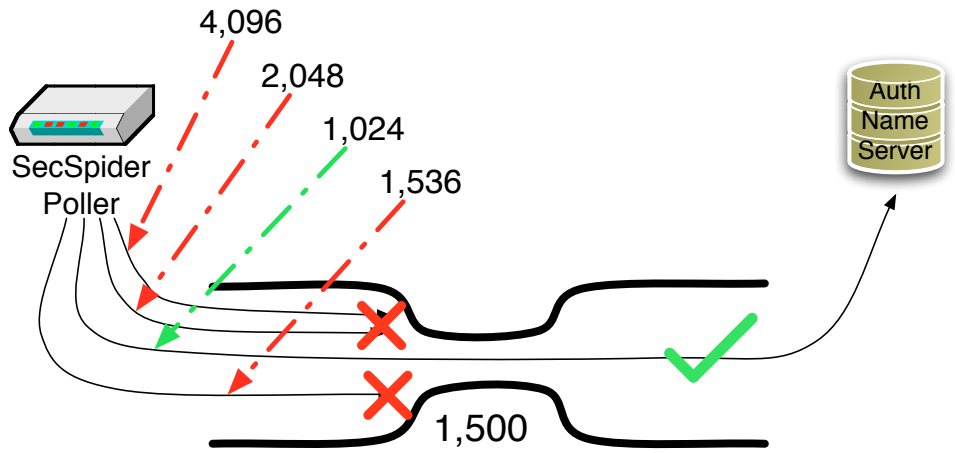
- DNSSEC's have significant differences from plain old DNS'

DNSSEC's Interaction with the Network

- DNSSEC Overstressed the DNS
  - We added crypto keys (DNSKEYs), anywhere up to 4,096 bits each
  - Zones should have at least 2 (ZSK + KSK) and maybe more
- We added crypto signatures (RRSIGs)
  - At least one in each RRset and sometimes one for each DNSKEY
  - Varying in size, based on DNSKEY sizes
- Large messages have lead to a prominent availability problem in DNSSEC's deployment
  - Resolvers request large response messages, even if they wont fit
- Sometimes these larger messages are *too* large to fit over the network path
  - The Path Maximum Transmission Unit (PMTU) is too small

The Network Path and PMTU

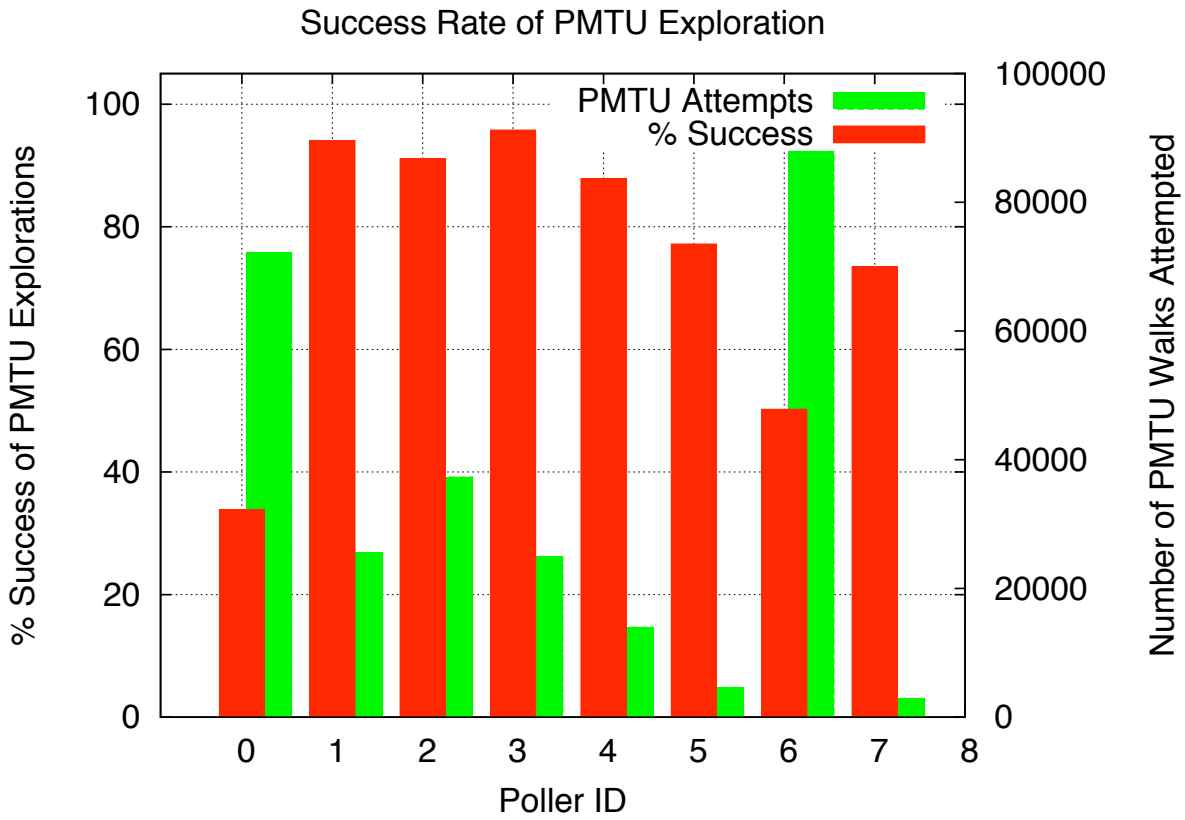
- A network path is a sequence of links
- Each link can only support packets of a certain size (MTU)
- The smallest MTU for a network path is its bottleneck, or its *Path Maximum Transmission Unit* (PMTU)



Studying the PMTU Problem

- A recent study showed that roughly 60% of queries seen at one root server ask for response sizes of 4,096 https://www.dns-oarc.net/node/146
- In 2009, SecSpider used its distributed pollers to illustrate:
  - http://irl.cs.ucla.edu/talks/2009-05-RIPE-PMTU.pdf
  - How often does the default behavior of using 4,096 byte buffers work for DNSSEC
  - When it fails, is it possible to advertise smaller buffer sizes that will work
  - How often are key sets just too large to fit over paths

As Seen From SecSpider's Pollers



- Green bars indicate the number of times a poller needed to do a PMTU walk
- Red bars indicate the percentage of times a PMTU was was able to find a buffer size the allowed DNSKEYs to be received

A Correlated Jump in Walks

- In September of 2008, roughly 100 zones began serving DNSKEYs that didn't "fit" their PMTUs
- In November of 2008 the availability was restored, but only with PMTU walks
- Zones can always check their statuses at:

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

Addressing Challenges

- Monitoring discovers deployment problems
  - This is an ongoing need: new problems will arise
- Before and during DURZ, extensive measurements were taken to ensure stability
- Tool suites have been critical in easing pains
  - But there is still plenty to do
- Approaches like SecSpider have taken additional proactive steps
  - Quantifies the operational protection of DNSSEC

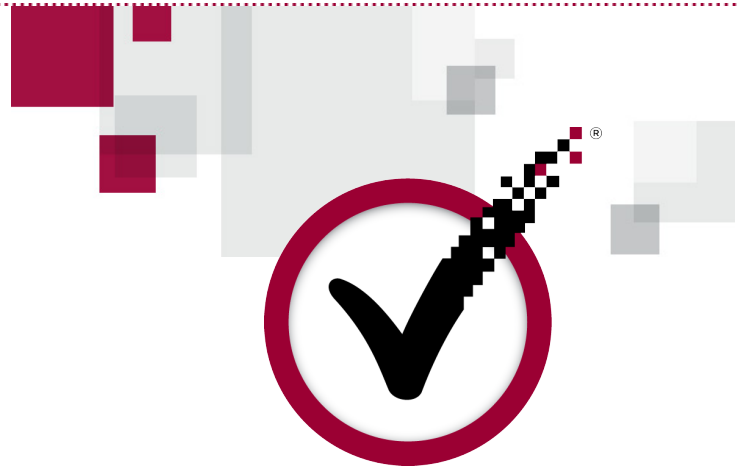
SecSpider's View of DNSSEC's Deployment

- Quantifying DNSSEC lets us to concisely describe its status
  - We define 3 measures to quantify the overall system status
- **Availability:** resolvers must be able to get data
  - Quantify the dispersion of the PMTU problem from different vantages
- **Verifiability:** resolvers must verify cryptographic keys and signatures
  - As opposed to spoofed by an attacker
- **Validity:** The data covered by cryptographic protections must be valid
  - A verifiable RSA signature does not mean an RRset's data is correct

Summary

- Today's problems may indicate that DNSSEC's operational complexity is a disincentive to deploy it
  - Many tools and services have arisen to deal with this: SecSpider, BIND-tools, ldns utils, Vantages, etc.
- DNSSEC's network interactions have also spawned a great deal of discussion and best practice work
  - Operational guidelines now include consideration for large message sizes, DU[R]Z, etc.
  - Tools like dnsfunnel and OARC's reply-size test for this
- With major milestones like the root being signed, incentives should finally arriving

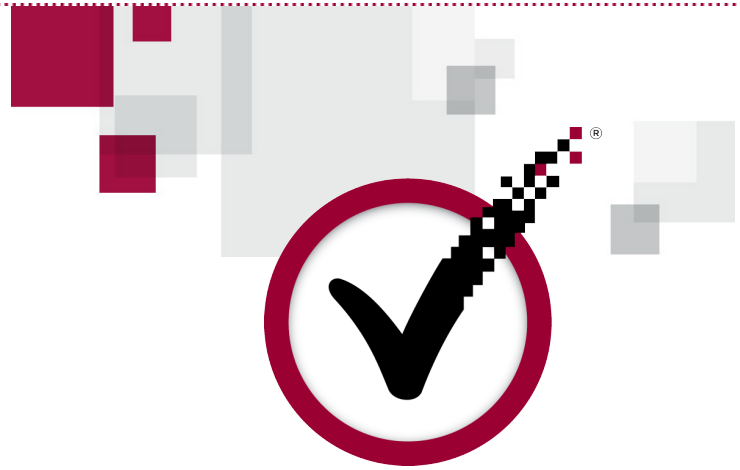
VeriSign®



Thank you

Questions?

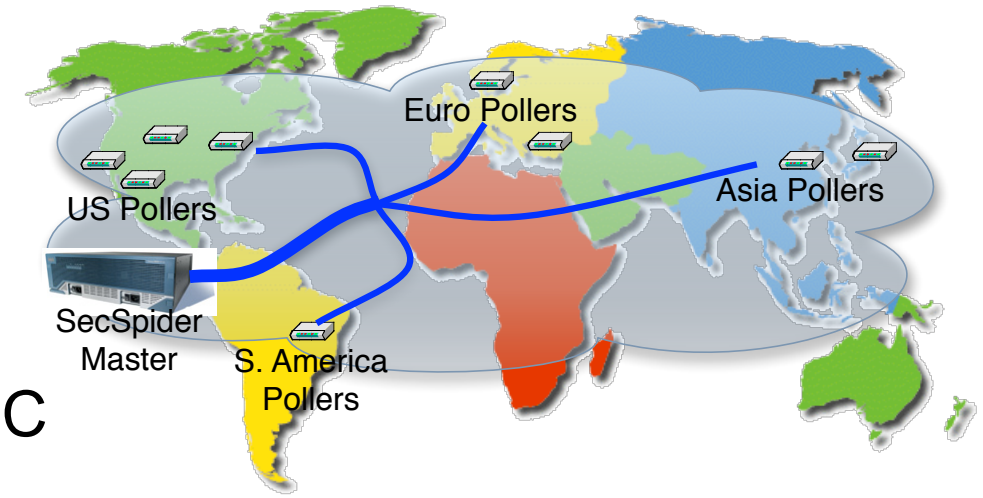
VeriSign®



Backup

SecSpider's Distributed Polling

- 9 pollers in:
  Asia,
  North America,
  South America,
  and Europe

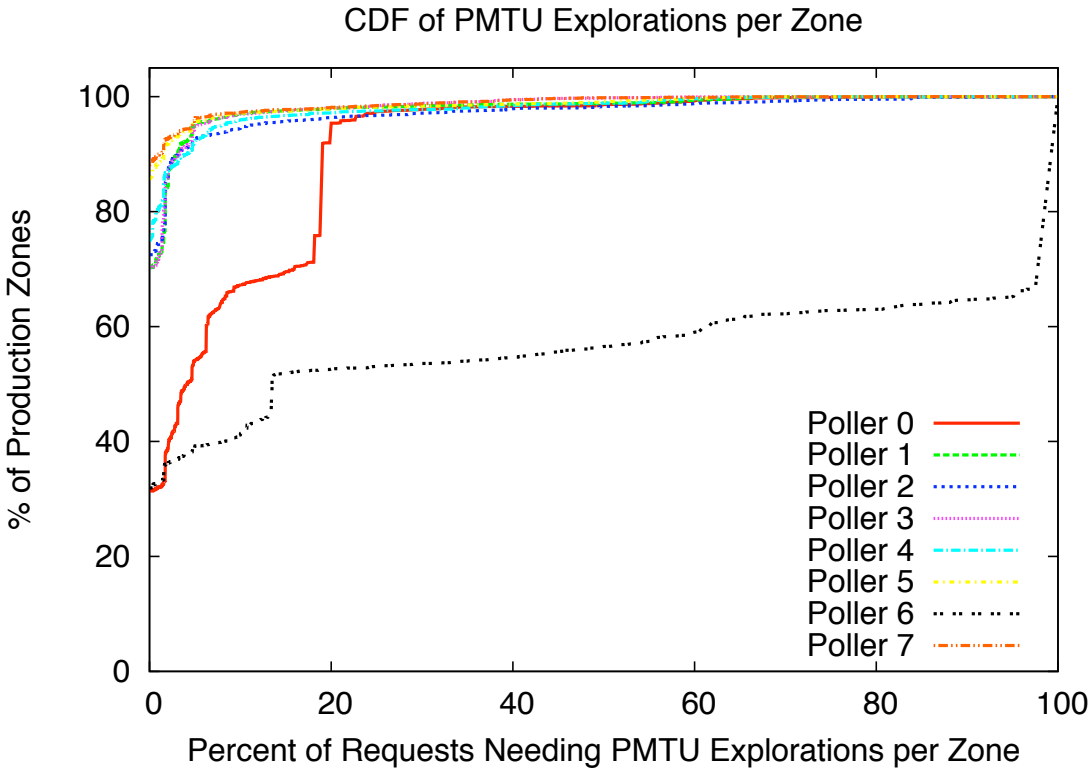


- Pollers are lightweight C daemons called rdnsD
- Communications between master coordinator and pollers is secured using TIG
  - Symmetric key crypto

Further Complications with DNS' Large Packets

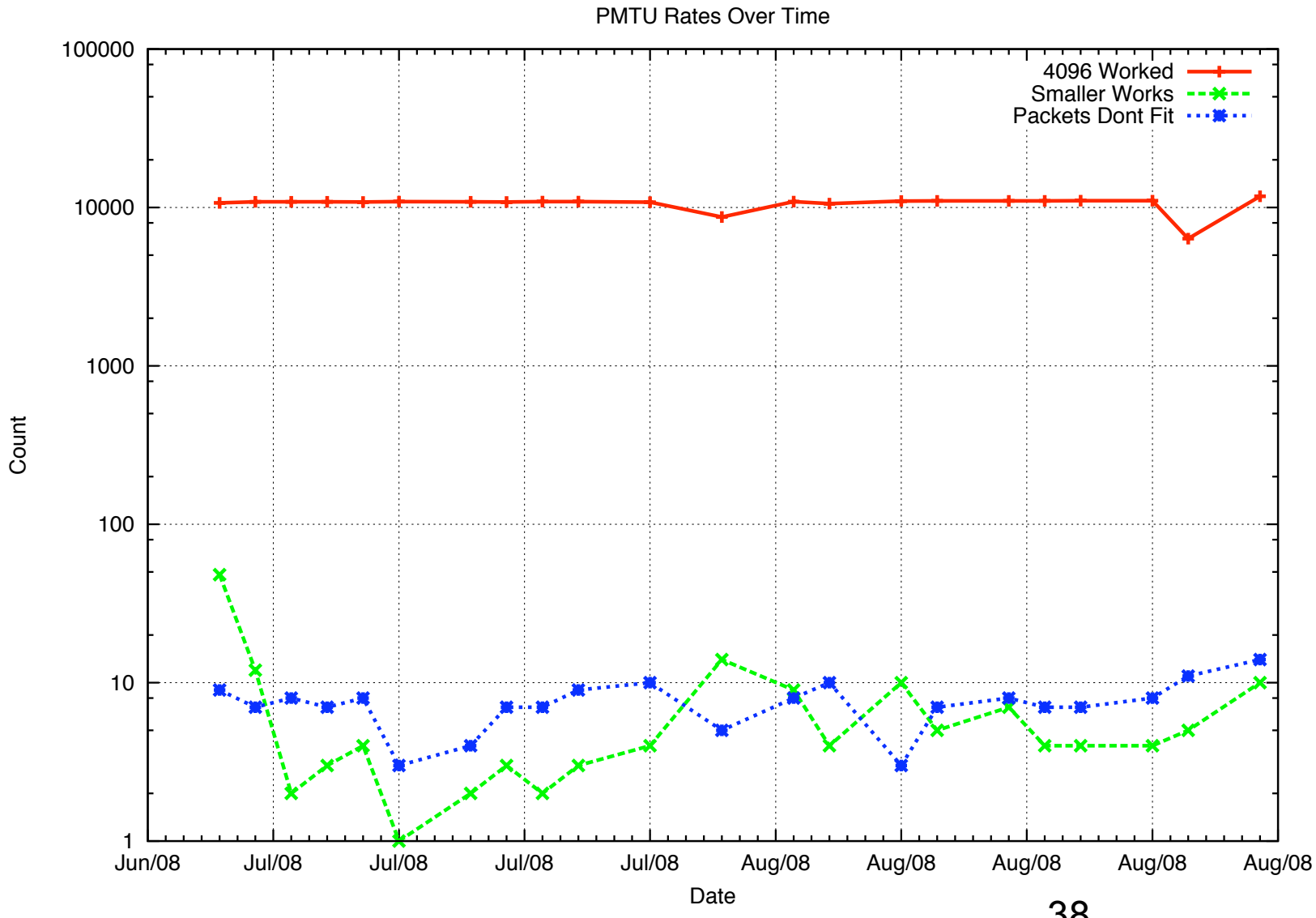
- DNS messages are further limited by "middle boxes" (firewalls, NAT, etc.)
  - Some firewalls drop "suspicious" DNS traffic
  - A recent study found this was quite common in SOHO routers http://download.nominet.org.uk/dnssec-cpe/DNSSEC-CPE-Report.pdf
- Because of middle boxes, network paths that may support large packets may fail to deliver large DNS messages
- We overload the term PMTU to apply in these cases too

How Many Zones Have Trouble?



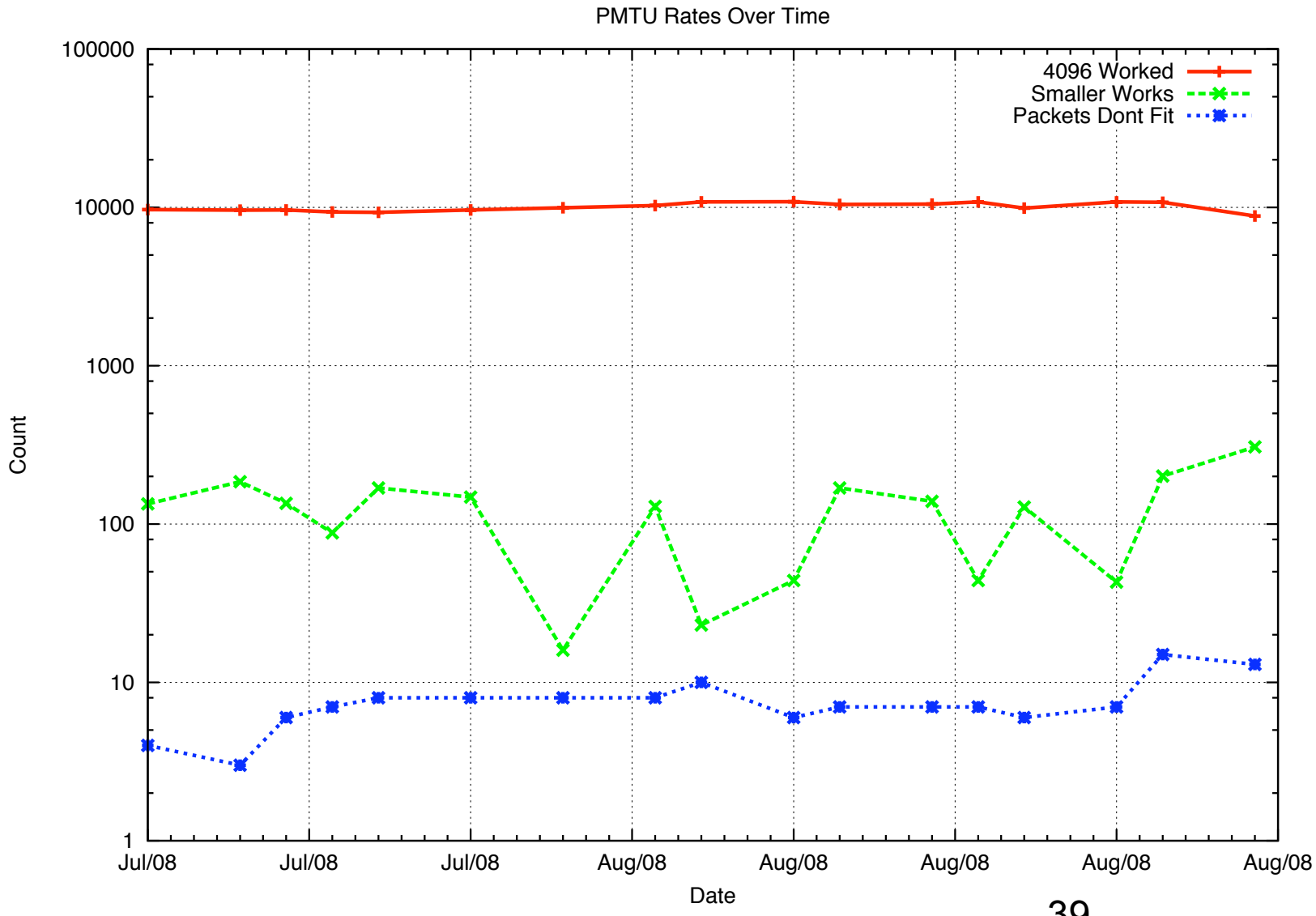
- Fraction of queries (x-axis) that cause PMTU exploration (y-axis)
- For Ex: from poller 0: ~70% of the production zones only need PMTU walks ~20% of the time (or less)
- Poller 6: ~60% of the zones need PMTU walks up to 90% of the time

NL NetLabs Poller



38

SOHO Router in Los Angeles



Something Interesting…