XFR-over-TLS (XoT)

Planning and Deployment of DNS Zone Transfers-over-TLS

Shivan Kaul Sahib | shivankaulsahib@gmail.com
Allison Mankin | amankin@salesforce.com
Willem Toorop | willem@nlnetlabs.nl
Sara Dickinson | sara@sinodun.com
Pallavi Aras | paras@salesforce.com

XFR-over-TLS (XoT) Background

Why XoT?

- Zone data can be collected via passive monitoring on-the-wire
- Zone owner may desire privacy for personal, organizational, or regulatory/policy reasons
- **The main motivation for XoT is to prevent zone data collection during transfer**

What is XoT?

- Encryption of DNS zone transfer (AXFR & IXFR) using TLS as a transport

XFR-over-TLS (XoT) Background

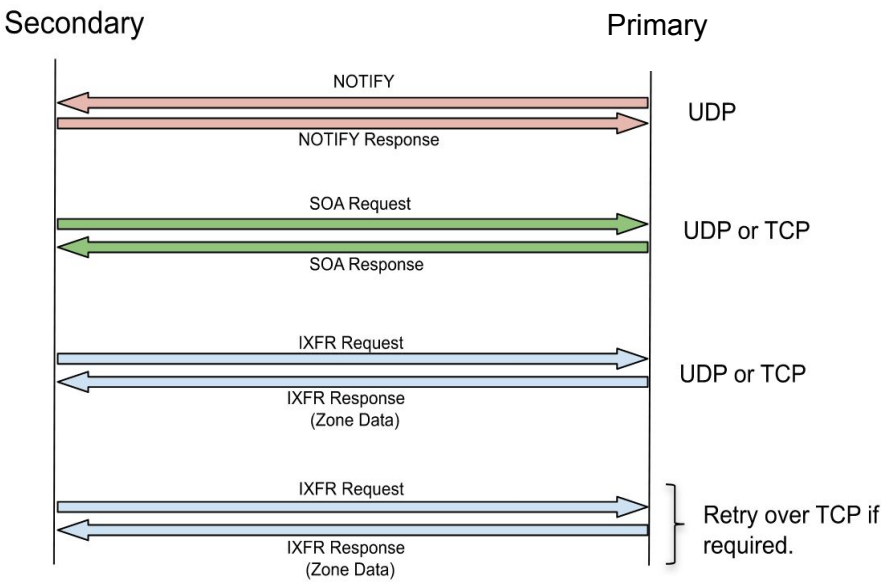
Use Cases

- **Confidentiality**: Encrypting zone transfers will defeat zone content leakage that can occur via passive surveillance
- **Authentication**: Use of single or mutual TLS authentication can complement TSIG/ACLs
- **Performance**: Current usage of TCP for XFR is suboptimal in most cases

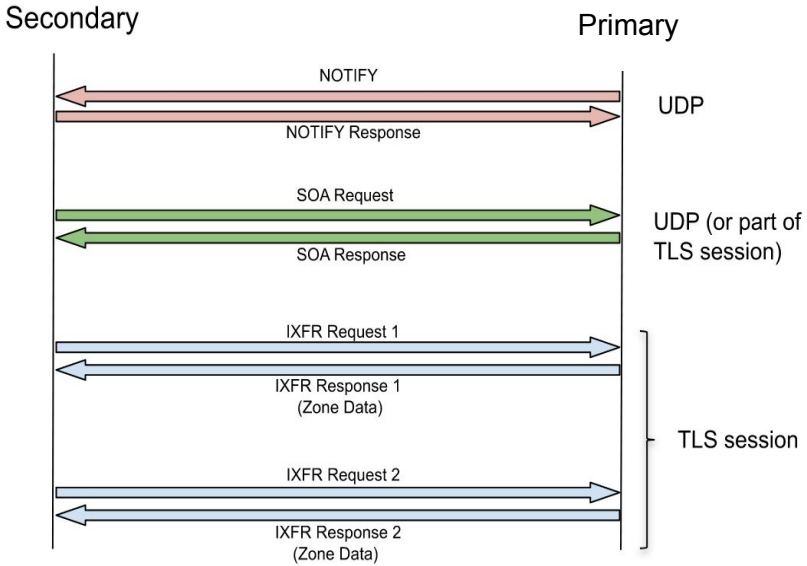
XoT Status

- IETF Draft specifying XFR-over-TLS is now in IESG review
  (final stage before publication)
- Implementations exist, testing and deployment planning in progress

IXFR : Existing mechanisms vs IXoT

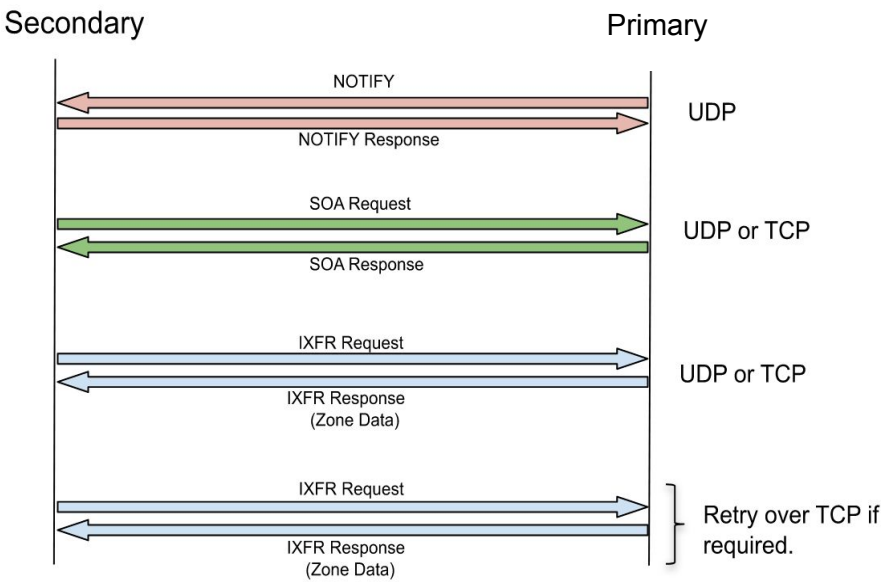


Existing

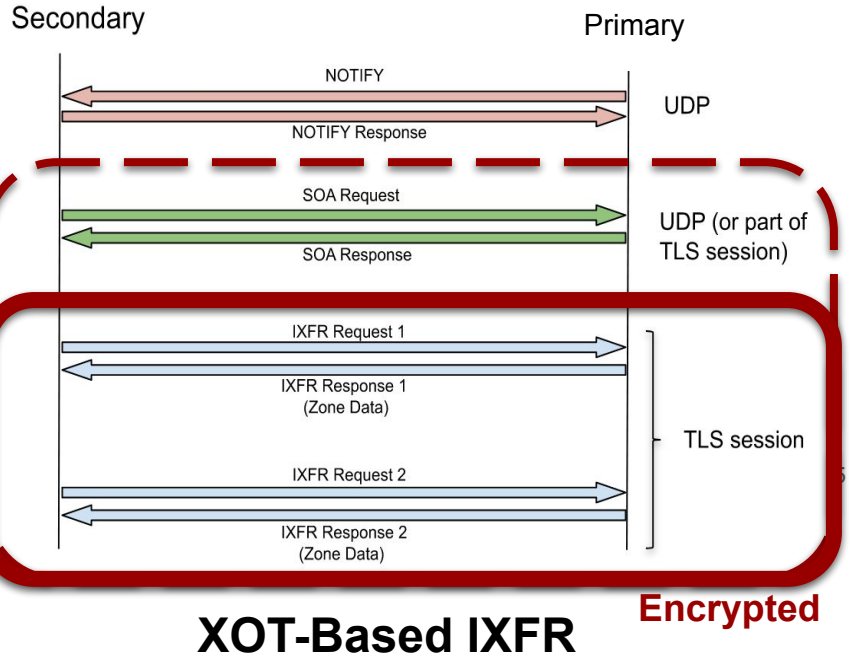


XOT-Based IXFR

IXFR : Existing mechanisms vs IXoT



Existing



XOT-Based IXFR

Key Technical Aspects of XoT

- All XFR messages MUST use TLS 1.3 or later
- Uses port 853 (other port with prior agreement)
- Servers are free to REFUSE other queries on a XoT connection
  i.e. XoT support does *not* require Authoritative DoT support
  - Use Extended DNS Error code (RFC8914) ‘Not Supported’
  - Appendix in draft describes operational and policy options for managing XoT connections

XFR Connection reuse

- Historically, implementations of XFR are ‘basic’ wrt TCP usage
    - Earlier specifications (RFC 1034/5, RFC1995) had implications about using one connection for one XFR message, some update for AXFR in RFC5936
    - **XoT draft updates specs for both IXFR and AXFR** to RECOMMEND efficient use of TCP:
        - Multiplexing XFRs of different zones on one connection
        - Pipelining queries and Out-of-Order responses
        - Use of EDNS0 Keepalive to keep connections open for re-use

XoT - Authentication mechanisms

Method		Secondary			Primary		
		Data Origin Auth	Channel Confidentiality	Channel Auth	Data Origin Auth	Channel Confidentiality	Channel Auth
TSIG		■			■		
TLS	Oppo		■			■	
	Strict		■	■		■	
	Mutual		■	■		■	■
ACL on primary							■

XoT - Authentication mechanisms

Method		Secondary			Primary		
		Data Origin Auth	Channel Confidentiality	Channel Auth	Data Origin Auth	Channel Confidentiality	Channel Auth
TSIG		■			■		
TLS	Oppo		■			■	
	Strict		■ ●	■ ●		■	
	Mutual		■	■		■ ●	■ ●
ACL on primary							■ ●

To verify the other party in a XoT transfer the specification says:

- **Secondary MUST use Strict TLS**
- **Primary MUST use either: mTLS (preferred) and/or IP ACL**

Implementation And Deployment Status

Implementations

- **BIND** released support in 9.17
  (likely backport to 9.16 pending successful testing)

- **Unbound** has secondary-side AXFR XoT

- Open PRs against **NSD** (planned for inclusion in 4.3.7 release)
  - Client side
    - XoT support
    - Connection reuse
  - Server side XoT support

- Various **interop testing** at IETF 110 Hackathon with BIND/NSD/PowerDNS

Deployment by Operators

- Managed DNS service operator support is important for XoT

- BIND and NSD were featured in particular because of some operators
  - **UltraDNS (Neustar)** - their TLD-oriented XTLD service is based on BIND servers
  - **PCH** - their anycast service offering receives XFRs with BIND and NSD variously

- Both vendors are open with commitment to offer XoT when it is product-level in the stacks

- PCH have privacy-first policies, and state that when XoT is a product, XoT will become the default offering for secondary service
  - Eventually as customers express their readiness to on the primary side, PCH wants to consider making XoT mandatory

Deployment challenges: Policy Management for XoT

- **‘Transfer Group’** - entire group of servers involved in transfers of a given zone (all primaries, all secondaries)
  - Could be managed by many different operators, many different implementations!

- The entire transfer group SHOULD have the same policy wrt (**no weak point**):
  - TSIG, TLS (Single or Mutual), IP ACLs

- **Challenge**: How to configure, enforce and test policy implementation?
  - Often involves different operators, different software, hidden servers

Deployment challenges

- In practice, there will be **incremental rollout** among transfer group until full XoT is in place
- Require **client side implementation** in **all** secondaries
  (can use TLS proxy with primary)
- **Certificate** management required
- Must consider other paths e.g. dynamic updates should be protected too

Breaking News - ALPN now required!!

- Recent IETF directive: “All new protocols using TLS *MUST* use an ALPN”
  - XoT will use *alpn=dot*
  - Will also apply to any future recursive-authoritative DoT specification (ADoT)
  - (Note: Stub to recursive pre-dates this so currently doesn’t use an ALPN…)

- Since XoT and ADoT are both DNS, need to use the same ALPN but..
  - Hidden primaries won’t have a problem
  - XoT can alway be run on a specific IP address and/or port to avoid overlap with ADoT
  - Servers can answer differently depending on their support, using REFUSED and EDE

Summary

- XoT is soon to be a new IETF Standard
- Implementations now available
- Operators are planning offerings
- Ongoing work to test and deploy
- New work starting up includes padding policy, encryption for DNS update
- Join us!
  - #XoT channel in OARC Mattermost

Backup

Padding Policy?

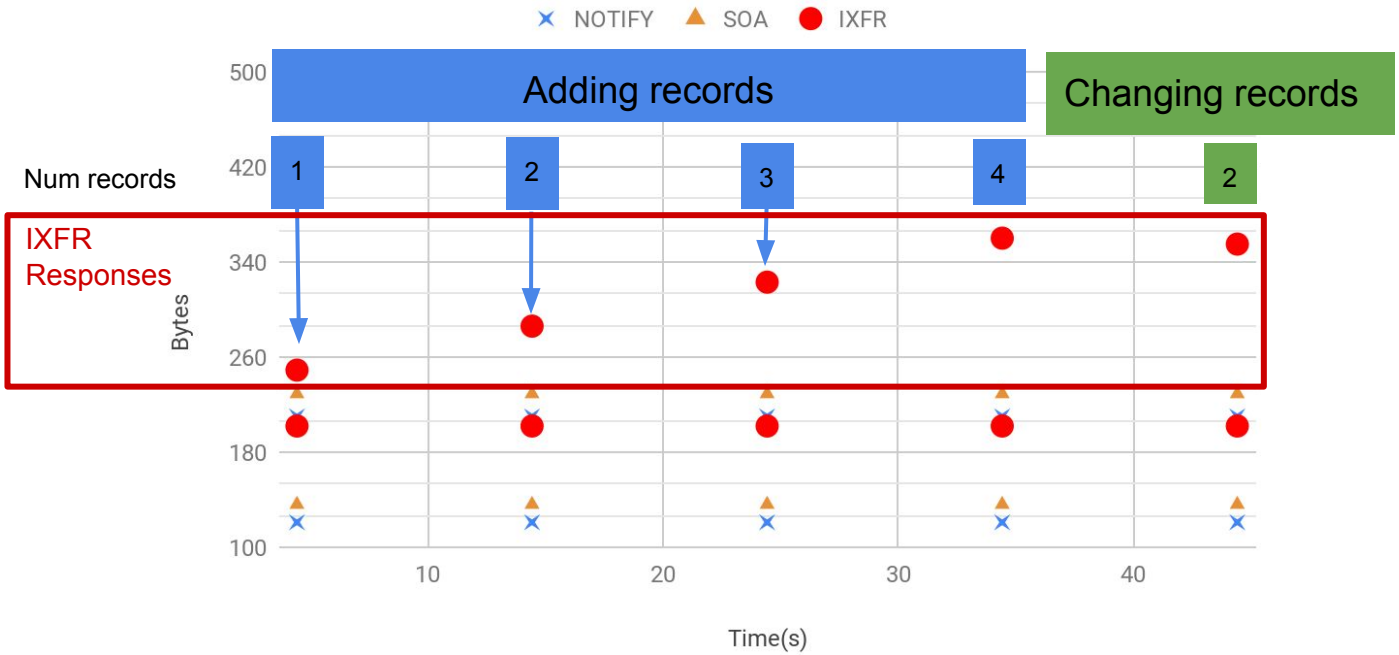
- **Size of transfers could leak information about zone size/activity**

- Specification makes no specific recommendations but notes:
  - AXFR total size needs padding to hide zone size
  - IXFR size patterns vary considerably depending on zone and update characteristics

Update rate	Zone size	DNSSEC	Update frequency	Update size (bytes)	
Low		Y	Low	100s	
Low	Very Large	N	High	1,000s	Jittered resigning
High		N	High	10,000+	RRSIGs significant

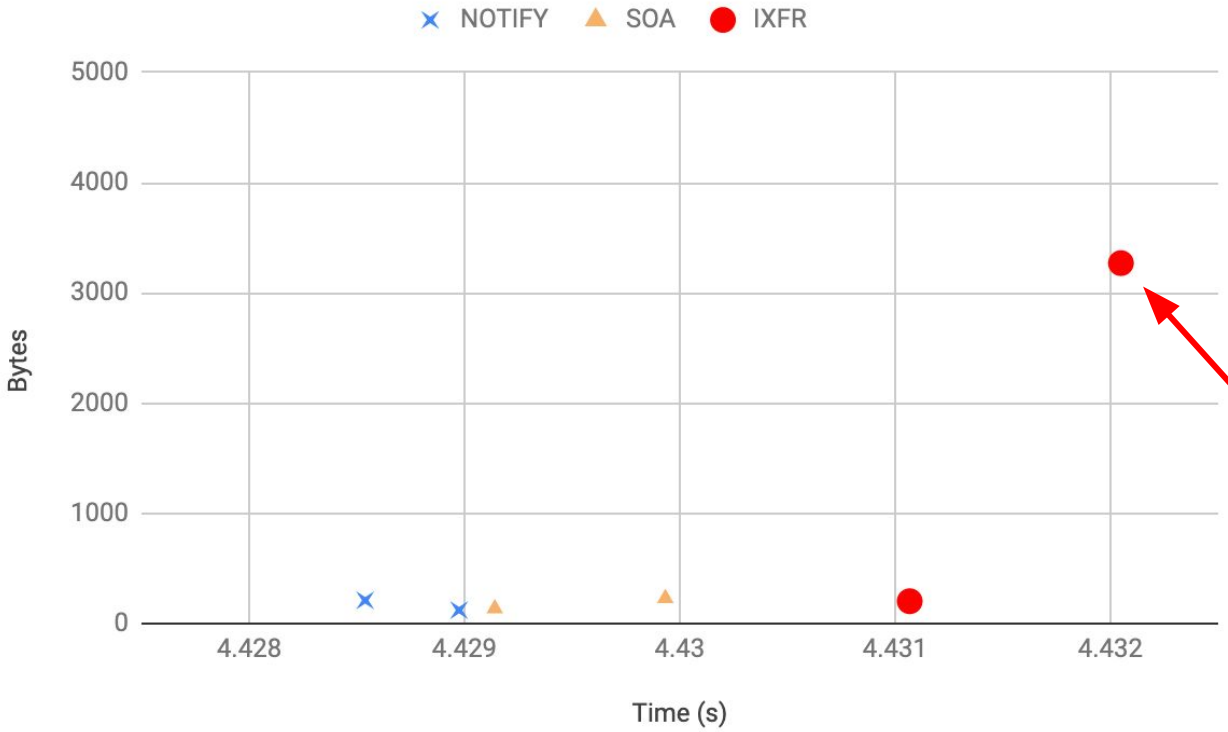
- Expect a future IETF draft to provide details of policies

Simplest IXFR pattern (unsigned zone with regular updates)



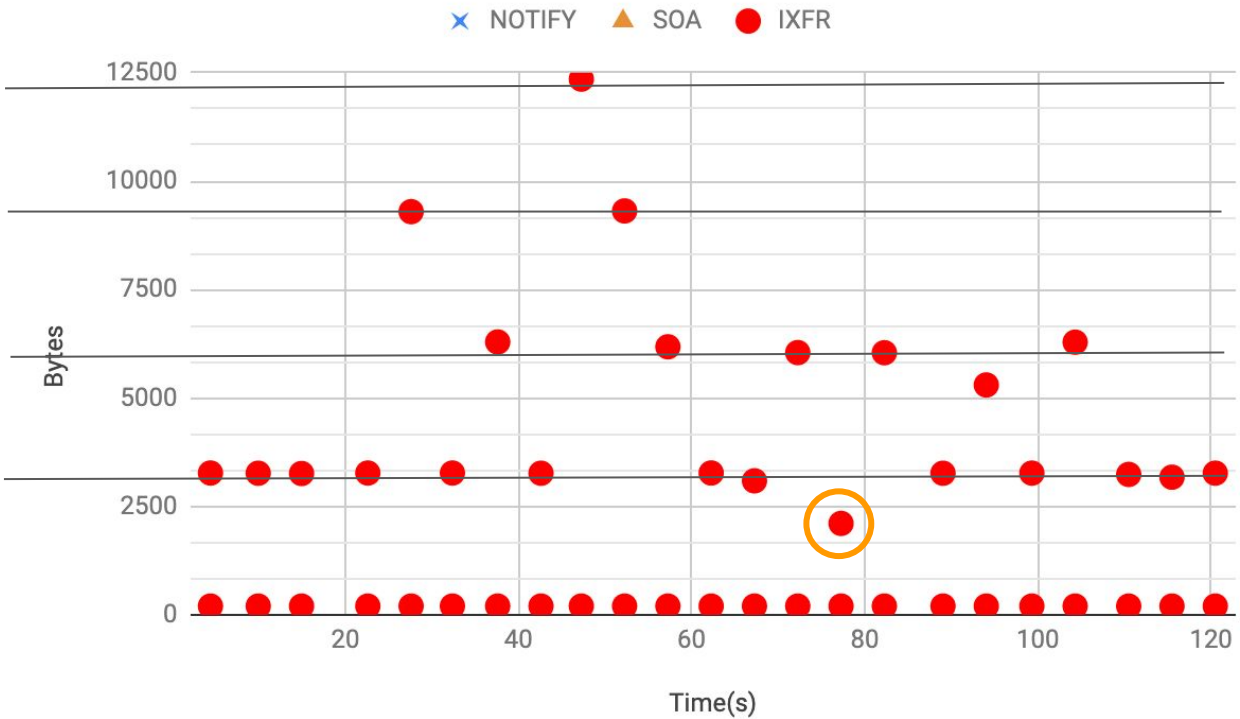
- Unsigned zone with records added every 10 seconds
- **Smallest XFR response packet possible** would be 5 records:
  - 1 new record
  - 4 SOAs
- Order of few hundred bytes (**~250 in this case**)
- Packet size can indicate record changes but adding and changing are hard to distinguish (and name compression happens)

Single IXFR exchange for large DNSSEC NSEC3 signed zone (no



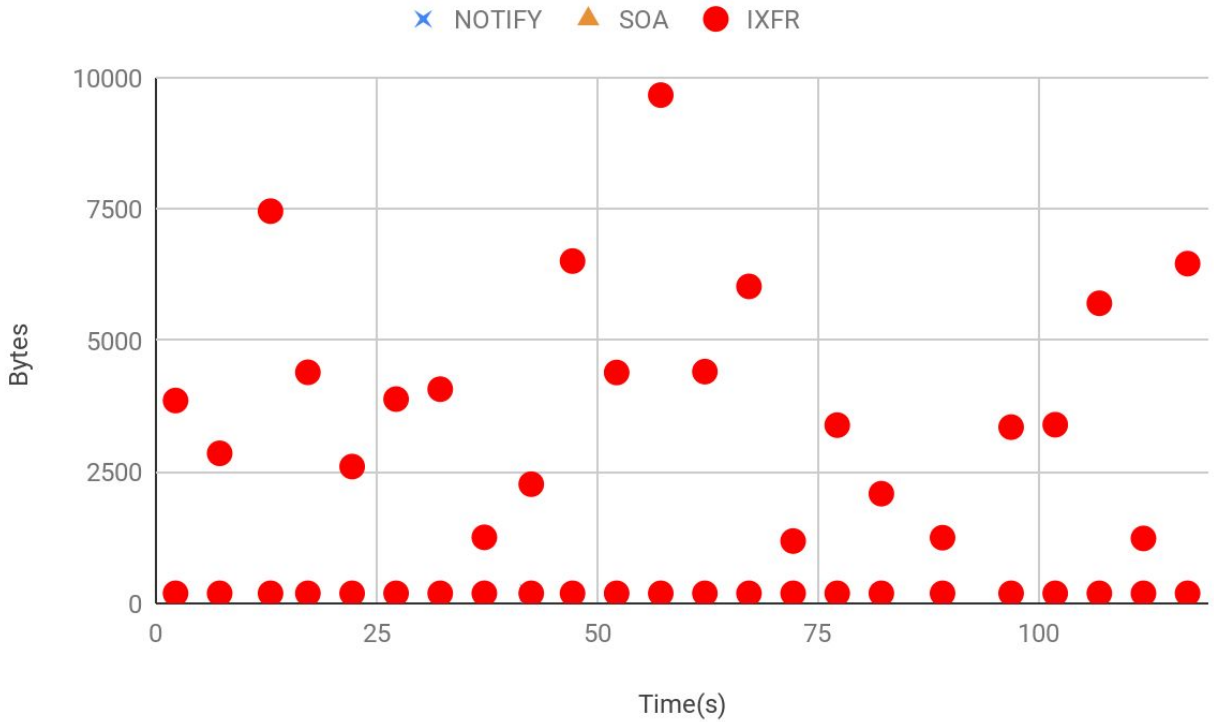
- **Update triggered purely by resigning of signatures** (zone signed with jitter)
- 1 SOA change -> 12 RRSIGs regenerated
- 28 records in response
  - 12 removes
  - 12 adds
  - 4 SOA records
- Each record averages just over 100 bytes, **response is ~3000 bytes**

Multiple IXFRs for large DNSSEC NSEC3 signed zone (one update shown)



- **Periodic resigning dominates**
- Transfers every 5s, on a **separate TCP connection**
- Responses clustered around **multiples of 3k** bytes (1 SOA change) - note no condensation of changes
- Anomaly at 77s is caused by a **single record update to the zone**

Multiple IXFRs - large dynamic DNSSEC NSEC3 signed zone (many updates)



- **Updates to zone every few seconds**
- If updates are frequent, size pattern is more complex
- **But answers still dominated by RRSIG records**
- Still see 5s intervals

Takeaways

- **Padding specifics**
  - Unsigned zones can directly leak number of record updates even when encrypted
  - Re-using a single connection for multiple zones would disguise the update pattern (as well as being a performance gain)
  - DNSSEC signing with jitter disguises the actual updates, but pattern varies with zone size and signing details