

Authenticated Bootstrapping of DNSSEC Delegations

ICANN 72 – DNSSEC Workshop
October 27, 2021

Peter Thomassen <peter@desec.io>

Nils Wisiol <nils@desec.io>

[draft-thomassen-dnsop-dnssec-bootstrapping](#)

DNSSEC validation rate

28 %

vs.

secure delegation rate

5 %

- 28% globally
- 50–95% in some places

- 5% globally
- 50–70% in some places
- **even for signed zones:**
< 50%

Sources: deSEC, <https://stats.labs.apnic.net/dnssec>, <https://rick.eng.br/dnssecstat/>,
<https://www.sidn.nl/en/news-and-blogs/dnssec-adoption-heavily-dependent-on-incentives-and-active-promotion>

But why?!

DNSSEC Bootstrapping Today (“How to Turn DNSSEC On”)

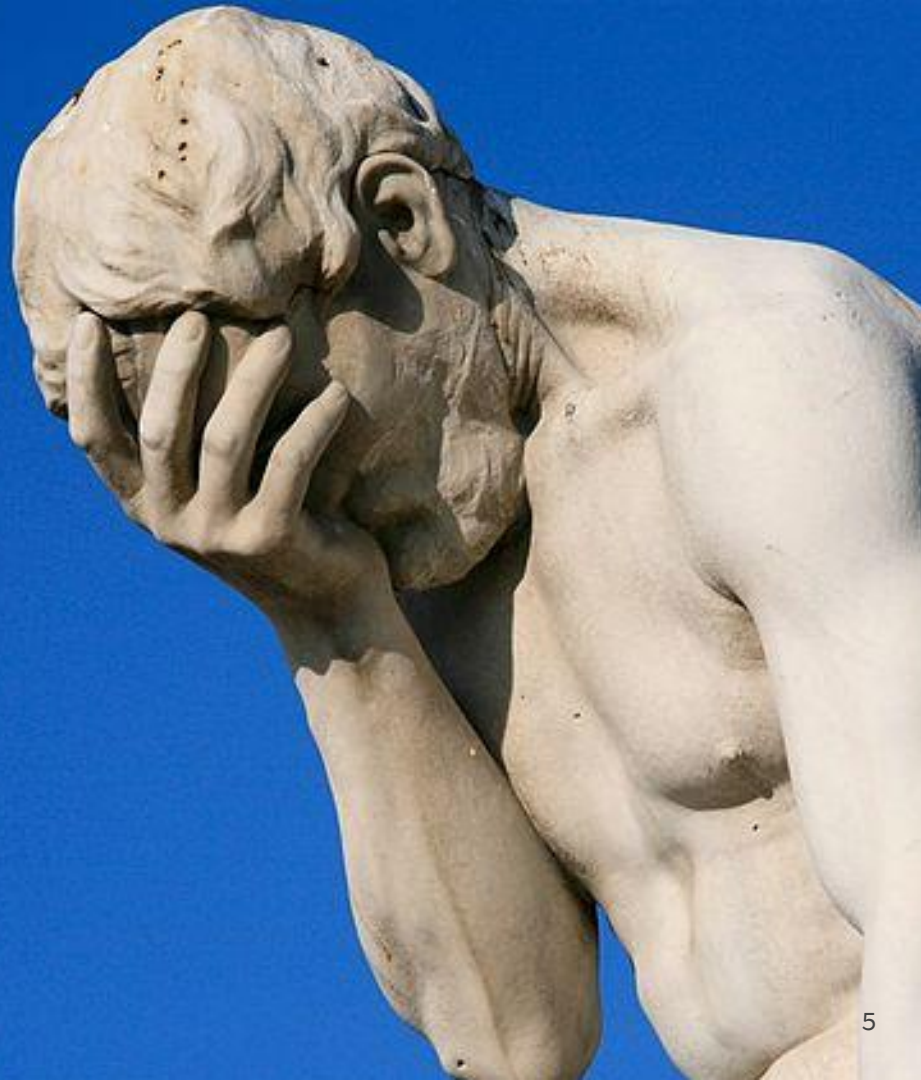
— — —

- Securing delegation requires conveying DS/DNSKEY records to parent
- Several approaches used by registrars / ccTLD registries:
 - trust on first use (TOFU, hope for the best)
 - manual submission by registrant/registrar (common and cumbersome)
 - REST interfaces (seems dead*)
 - CDS/CDNSKEY from insecure child (RFC 8078, requires stateful monitoring)
- Downsides: unauthenticated, out of band, slow, stateful, error-prone, too many parties, no automation / requires trigger, ...

* ICANN 54 (2015), draft-ietf-regext-dnsoperator-to-rrr-protocol (2018)

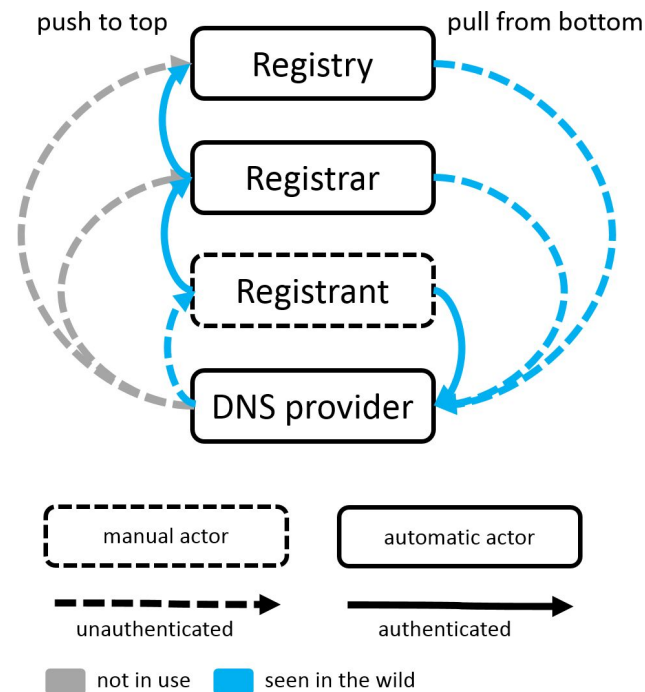
DNSSEC is too hard

and we know it



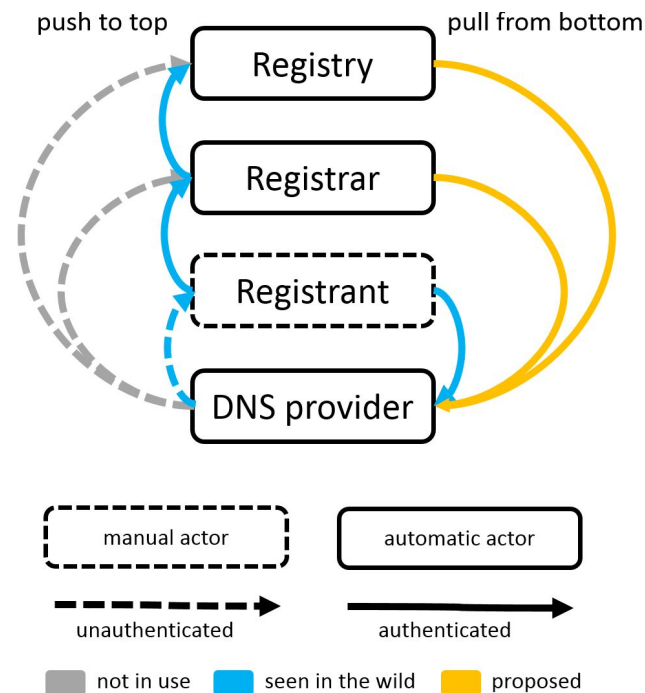
Analysis: DS Signaling Model

- Secure (authenticated) DS signaling currently involves many steps
- Reduce number of steps: make **registries / registrars pull directly from DNS provider**
 - RFC 8078 specifies this (via CDS/CDNSKEY)
 - so far **not secure for DNSSEC bootstrapping**



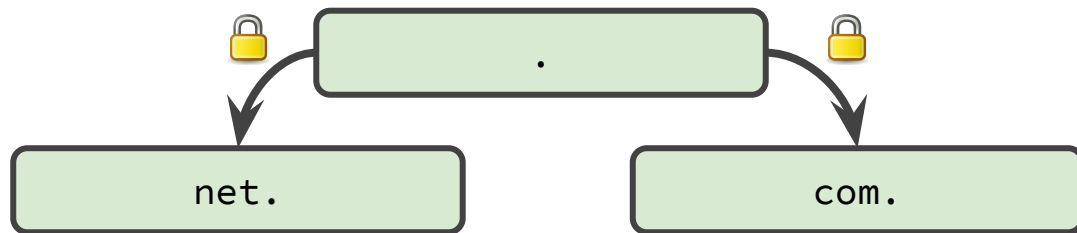
Analysis: DS Signaling Model

- Secure (authenticated) DS signaling currently involves many steps
- Reduce number of steps: make **registries / registrars pull directly from DNS provider**
 - RFC 8078 specifies this (via CDS/CDNSKEY)
 - so far **not secure for DNSSEC bootstrapping**
- **Goal:** authenticate pull from DNS provider
 - add authentication mechanism to CDS/CDNSKEY
 - automated, in-band, immediate, stateless (parent)

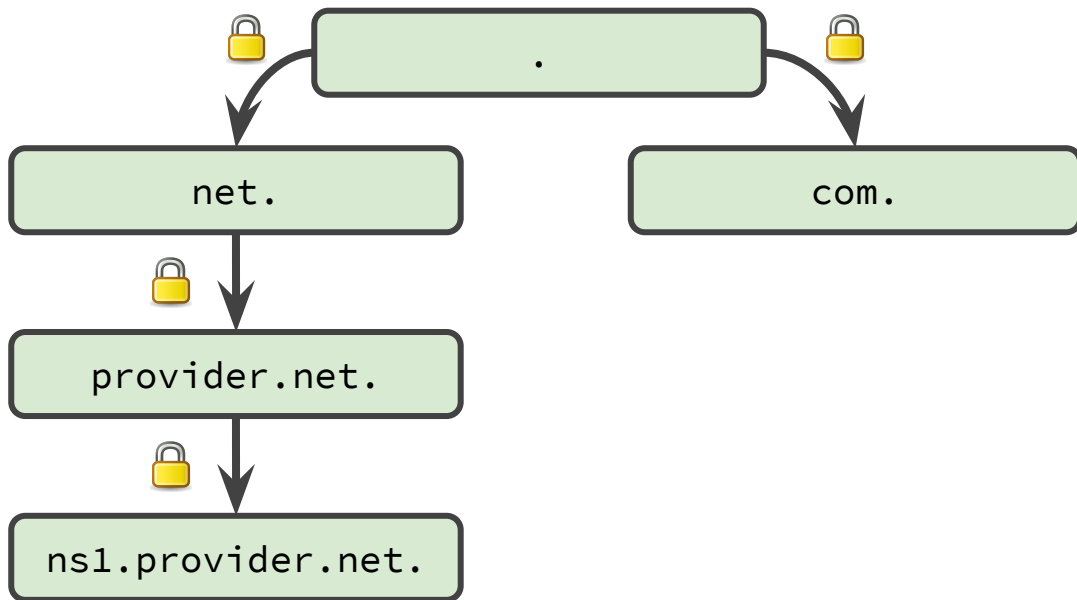


Solution Proposal: Transferring Trust from the DNS Operator

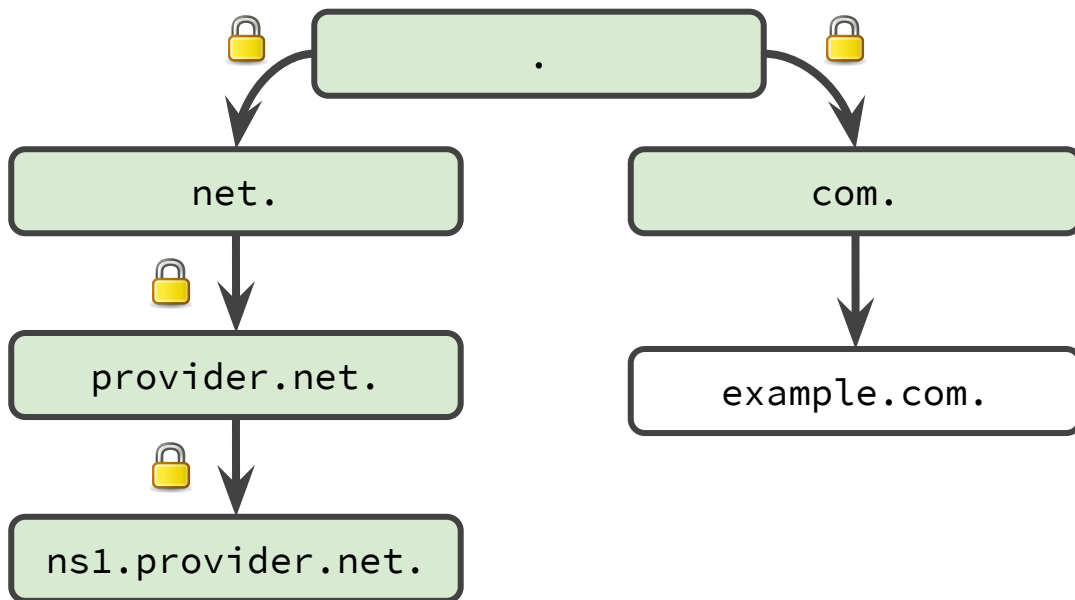
CDS Authentication: Co-Publishing under Trusted Hostname



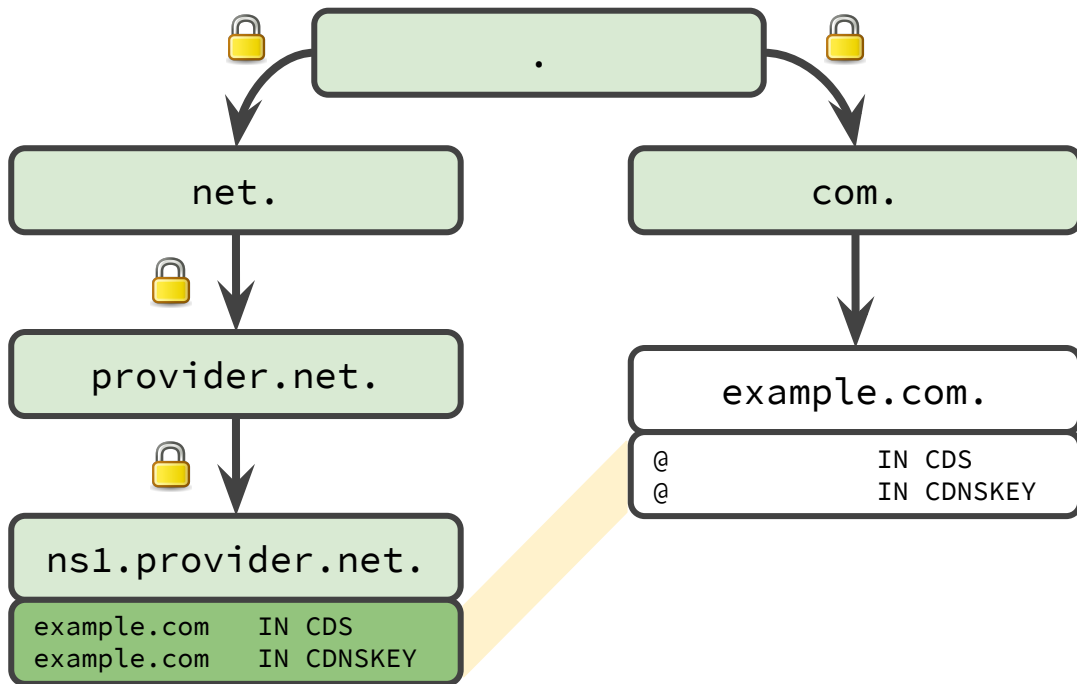
CDS Authentication: Co-Publishing under Trusted Hostname



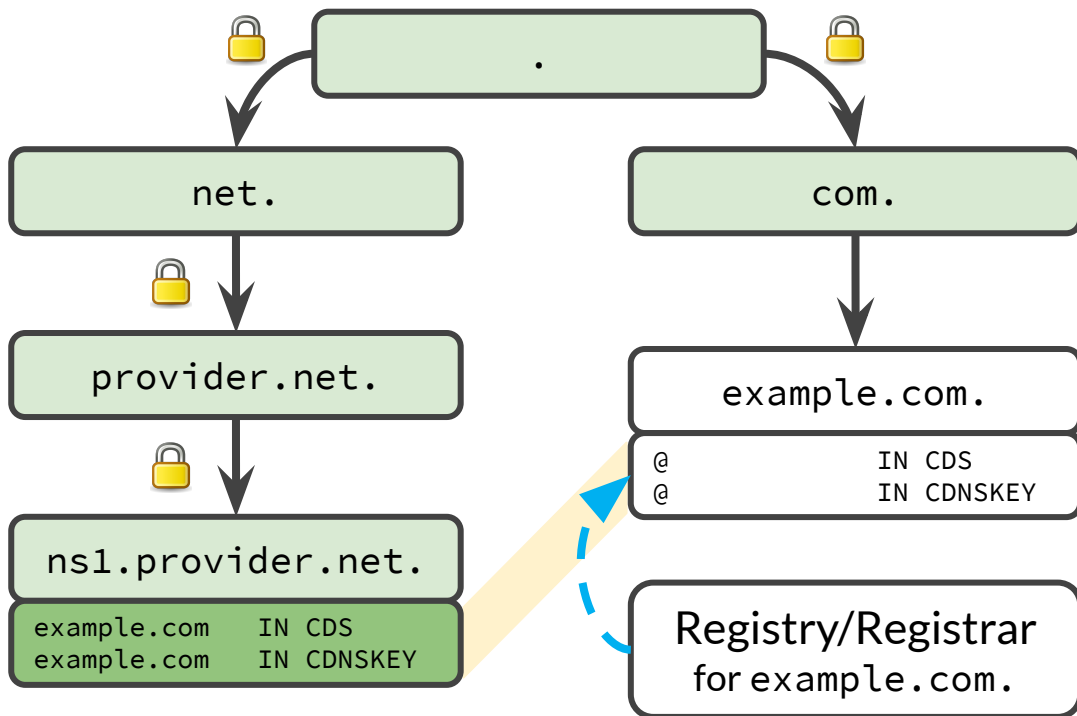
CDS Authentication: Co-Publishing under Trusted Hostname



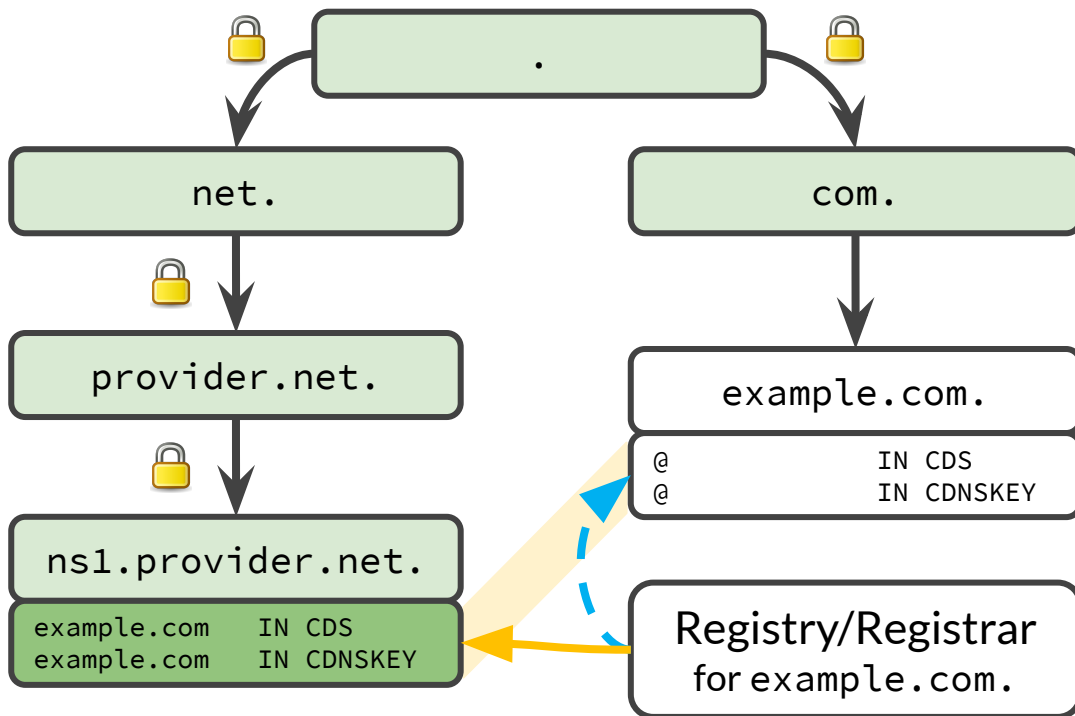
CDS Authentication: Co-Publishing under Trusted Hostname



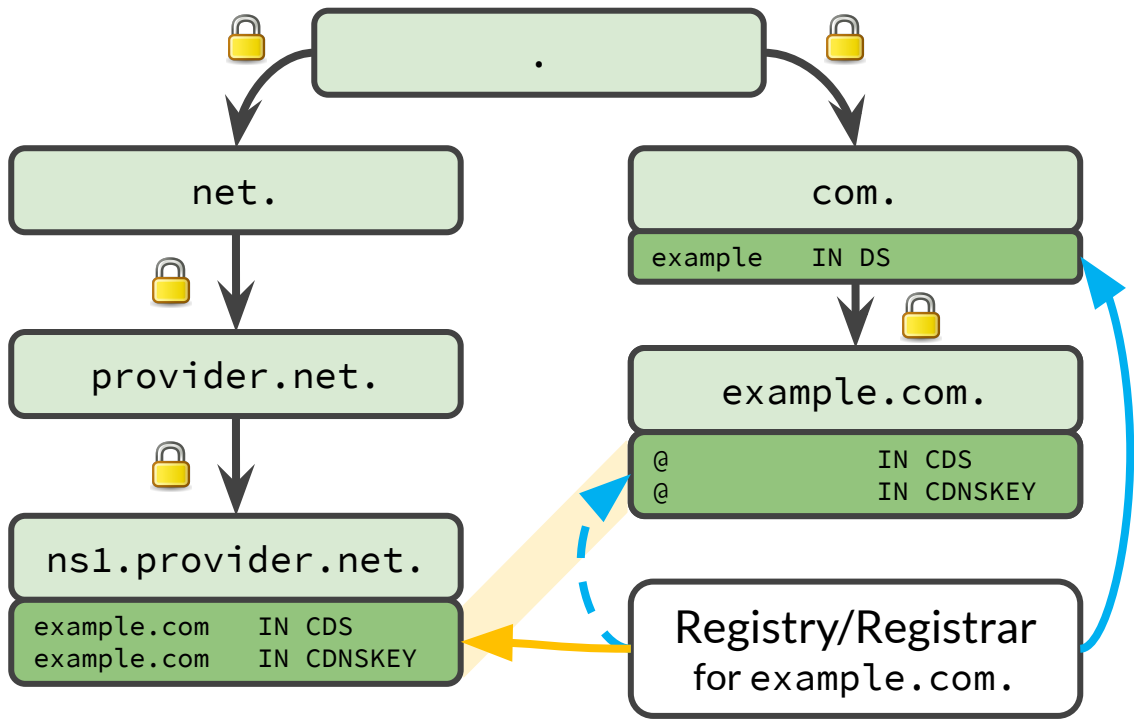
CDS Authentication: Co-Publishing under Trusted Hostname



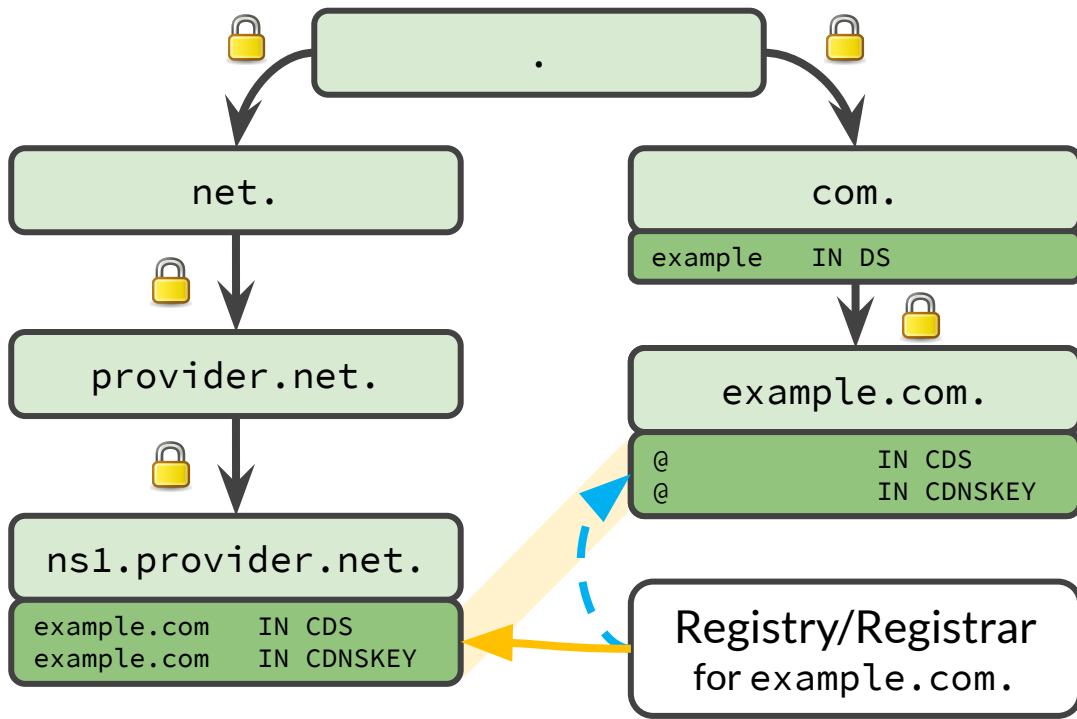
CDS Authentication: Co-Publishing under Trusted Hostname



CDS Authentication: Co-Publishing under Trusted Hostname



CDS Authentication: Co-Publishing under Trusted Hostname



💡 Use an **established chain of trust** (left) to take a detour

- authenticated, immediate
- no active on-wire attacker

Technical Considerations

- No collision with primary use of CDS/CDNSKEY (those are apex-only)
- Replace ancestor labels with hash: **example.h(com).ns1.provider.net**
 - to avoid hitting length constraints, and to allow per-parent handling
- Add extra label: **example.h(...)._boot.ns1.provider.net**
 - to enable delegation of signaling data to separate zone
- Advantages:
 - removes risk of accidentally modifying the nameserver's A/AAAA records
 - reduces churn on nameserver zone
 - allows splitting off DNS operations (e.g. online-signing with different key; delegate by parent)
 - allows parent to discover bootstrappable domains under **h(parent)._boot** (XFR, NSEC walk)

How about some numbers?

What's needed for deployment?

- Secure signaling **requires that NS targets are in securely delegated zones**
 - if already the case: simplifies deployment for DNS operators
 - if not: overhead for DNS operator seems manageable
- DS bootstrapping **requires that NS targets are not part of the same zone**
 - **mostly the case:** > 99% of NS targets are out of bailiwick
in bailiwick: < 0.33% for .com, < 0.72% for .net (thanks to John R. Levine)
- ... and obviously, the zone itself needs to be signed.
- Survey time!

Survey on Deployment Requirements

- Analyze **top 1M sites** (Tranco dataset)
- For each domain in the dataset, extract
 - a. whether the domain itself is **secure** (has validation path),
 - b. whether there zone itself is **signed** (has RRSIGs),
 - c. **all NS targets** in the delegation,
 - d. which NS targets are **secure** (if any),

... and compute things like

Bootstrappability: What fraction of domains have $a == \text{false}$, but $c == d$?

Survey on Deployment Requirements: General Results

Failure rate	3.80%
Remaining sample size	962012
Proportion of secure zones	4.47%
Proportion of signed zones	5.87%
Proportion of zones with all nameserver targets secure:	24.14%
Proportion of zones with ≥ 1 nameserver targets secure:	25.36%

bootstrappable:

domain is not secure and NS targets have validation path → signaling possible

Proportion of bootstrappable zones (all NS)	21.77%
Proportion of bootstrappable zones (≥ 1 NS)	22.66%

Survey on Deployment Requirements: by TLD and Provider

tld	zones	bootstrappable	
	total count	rel.	abs.
com	493152	23.6%	116343
org	68720	18.0%	12396
net	43894	23.6%	10371
ru	31435	13.8%	4327
uk	20102	18.9%	3798
in	9208	28.7%	2645
io	7134	34.4%	2452
co	7089	30.3%	2146
de	27158	7.3%	1978
au	7964	24.3%	1934

ns_name	zones	bootstrappable	
	total count	rel.	abs.
dns.cloudflare.com.	247146	76.4%	188746
dns.hostinger.com.	3958	86.8%	3436
hostmaster.nsome.net.	19804	12.5%	2470
nan	54313	3.6%	1959
hostmaster.cscdns.net.	6026	23.1%	1393
postmaster.ijj.ad.jp.	949	97.7%	927
root.v1.wpxhosting.com.	641	99.7%	639
nsadmin.nic.in.	813	69.2%	563
dns.ds.network.	637	83.2%	530
hostmaster.infomaniak.ch.	719	63.1%	454

Recap: We got ...

Signaling

- of zone-specific information
- from the NS operator
- to the public (e.g. the parent)

... which is

- authenticated,
- in-band,
- immediate,
- requires no third parties.



Recap: We got ...

Signaling

- of zone-specific information
- from the NS operator
- to the public (e.g. the parent)

... which is

- authenticated,
- in-band,
- immediate,
- requires no third parties.

What else
can be done
with it?



Multisigner Key Exchange (in a Nutshell)

Multisigner Goals (RFC 8901):

- **Redundancy:** multi-homed zones with full validation of responses
- **Integrity:** smooth transition during provider transfer (w/o going insecure)

How it works:

- Operators advertise each others' ZSKs via the DNSKEY set that they sign;
- Parent advertises all of the KSKs via its DS records.

How can operators learn each other's ZSKs?

- Publish them in a signaling DNSKEY RRset below ns1.other.net
- **Same signaling mechanism** as for DS bootstrapping

Thank you!

... also to our sponsors:



Questions?



Backup

Open Questions

- Should we support sharding, by splitting Signaling Names into several labels?
 - How exactly would that work? Should that be configurable? (How to store configuration?)
- Should the hash (ancestor) label have a PTR record pointing to ancestor?
 - This would allow full enumeration of bootstrappable domains
- For an operator supporting the protocol: is it REQUIRED for all domains?
 - Probably no, as it won't work with secondary providers?
- When NS RRset is received at registration, zone may not yet be operational
 - What else would be a good trigger for the registry/registrar? Perhaps a nightly NSEC walk?
- Should the proposal be rephrased as a new mode of operation for RFC 8078?
 - cf. RFC 8078 Section 3.1

Closed Questions (I)

— — —

- If a DNS operator deploys DS bootstrapping, parents may like bulk processing. How is that best achieved?
 - allow NSEC walking of signaling zone (thanks to Brian Dickson)
 - allow public AXFR of signaling zone (thanks to John R. Levine)
- Should an extra layer be inserted in the Signaling Name to allow parent-specific bulk processing? (thanks to John R. Levine)
 - Yes
 - compatible with both NSEC walking
 - also compatible with AXFR (but benefit gained only when using subzones for large parents)
- Do we need hash collision mitigation (salt) and/or hash algo upgrade path?
 - No: due to child apex check, collisions don't affect key integrity
 - In case of collision, bootstrapping fails (for this parent) → fallback to conventional DS init

Closed Questions (II)

- Drop requirement that CDS/CDNSKEY within the target zone must match?
 - No. Prevents synchronization mismatch when Child rolls key and signaling zone is stale. Prevents hash collisions. Allow straightforward opt-out. Also, implies all RFC 8078 guarantees.
- Drop requirement that all NS responses must agree?
 - No. Otherwise, multihoming with different signers will break the zone.
 - Deployment effort is manageable: 95% of delegations with at least one securely delegated NS target in fact have *all* NS targets securely delegated. Also, dropping this requirement would be inconsistent with requiring records at the child apex to match. It's also unclear what should happen in case of contradictory signaling records, if they are not required to agree.
- Registries/registrars can select which TLDs to trust in the chain. Desirable?
 - No (at least in the spec). One could say that you can't trust a DNS operator anyway if its NS hostnames are not trusted. (That doesn't prevent parents from deciding locally to ignore or reject certain signaling names.)

Securing the `example.com` delegation (no existing DS)

Assumption: The NS targets (e.g. `ns1.provider.net`) live in securely delegated zones (e.g. `provider.net`).


(I) On the DNS provider side:

Publish `example.com`'s CDS/CDNSKEY records at a “**signaling name**” under the nameserver zone:

`example.com.ns1.provider.net`

(II) On the registrar / ccTLD registry side:

When receiving a new NS record set,

1. **query** CDS/CDNSKEY records from **DNS provider** (using all NS names):
 - `example.com.ns1.provider.net,...`
2. **validate**
 - **DNSSEC signatures** of responses,
 - **sanity check** (consistency with target zone);
3. **publish** `example.com`'s **DS records** in the parent zone → **done!** 

Security Model

— — —

- We use an established chain of trust to take a detour
 - authenticated, immediate
 - no active on-wire attacker
- Actors in the chain of trust can undermine the protocol
 - can also undermine CDS / CDNSKEY from insecure
 - but: known point in time / window of opportunity much smaller
- Further mitigations exist, e.g:
 - monitor delegation
 - diversify NS TLDs
 - multiple vantage points

	MANUAL	BOOTSTRAPPING METHOD CDS/CDNSKEY	PROPOSED
BOOTSTRAPPING INVOLVES			
zone operator Z	✓ ¹	✓	✓
domain owner	✓	✗	✗
registrar	✓	✗	✗
registry	✓	✓	✓
ACTORS WHO CAN INITIALIZE KEYS			
<i>Required parties (trusted)</i>			
registrar	✓	✓ ²	✓ ²
NS zone operator	✗	(✓)	(✓) ³
NS zone ancestors	✗	(✓)	(✓)
NS zone owner	✗	(✓)	(✓)
<i>Others parties (untrusted)</i>			
active on-wire attacker	depends	✓ ⁴	✗
social engineering attacker [1]	✓	✗	✗
PROPERTIES			
Prerequisites	out-of-band channel	MITM attack mitigation	suitable NS zone configuration
Authentication	bad in practice [1]	none	cryptographically
Duration	varies	days	minutes

Table 1: Comparison of methods for establishing a new secure delegation, displaying a) entities involved in the bootstrapping of an individual insecure zone, b) attack surface towards trusted and untrusted third parties, and c) prerequisites, key material authentication, and bootstrapping duration. Key initialization within parentheses (✓) requires collusion across all NS zones. ¹ For offline signing, only the signing key holder is involved. ² Registry could refuse deployment through registrar. ³ Requires knowledge of private key. ⁴ Several vantage points and long time must be covered.