Automatic DNSSEC Bootstrapping with Authentication

ROW11
June 21, 2022

Peter Thomassen <peter@desec.io>
Nils Wisiol <nils@desec.io>

draft-ietf-dnsop-dnssec-bootstrapping

DNSSEC validation rate

30 %

- globally
- 50–95% in some places

vs.

secure delegation rate

6 %

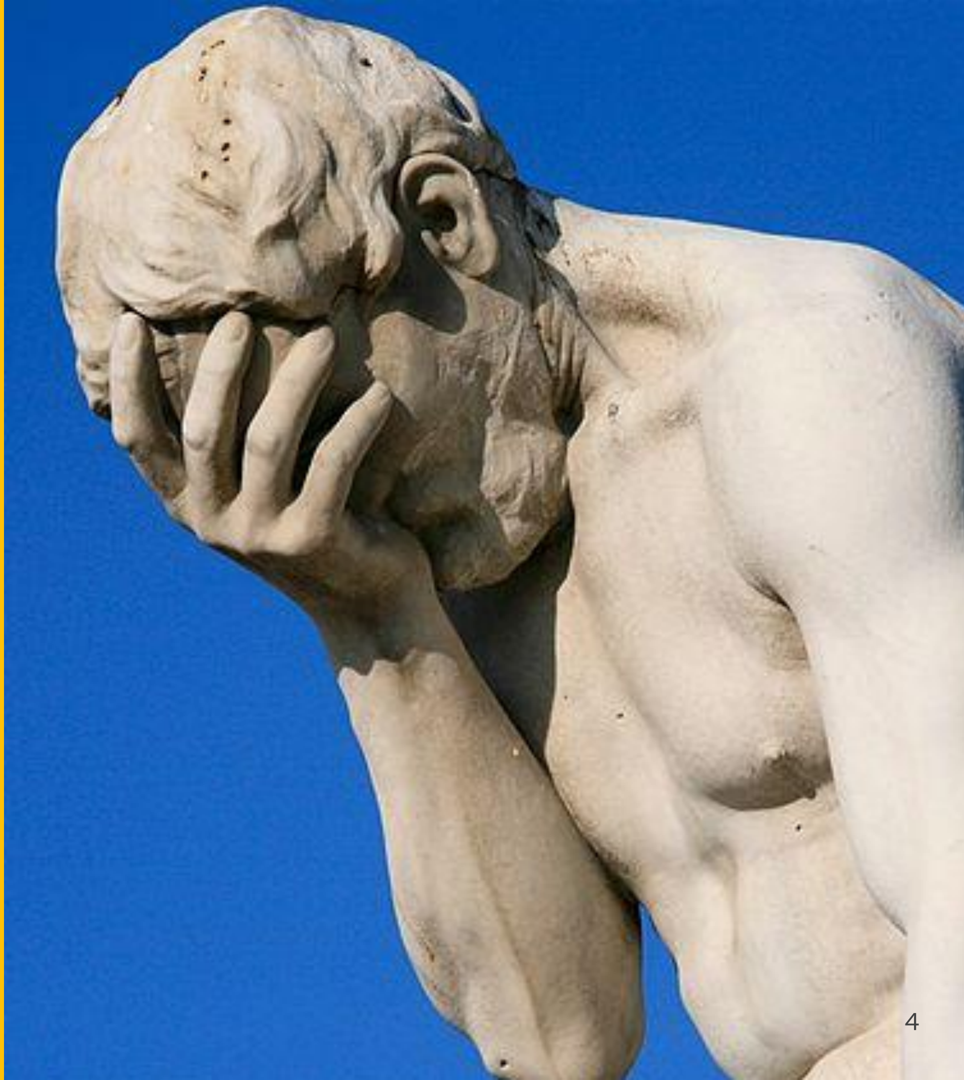
- globally
- 50–70% in some places
- **even for signed zones: < 50%**

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 is too hard

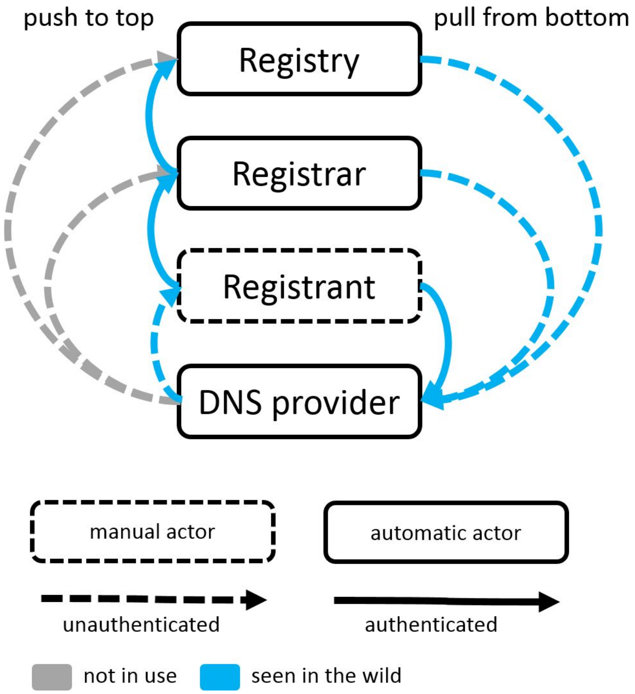
and we know it



The State of DS Bootstrapping

- Various methods available, with downsides
  - TOFU, manual submission, REST interfaces etc.
  - unauthenticated || out of band || slow || stateful || error-prone || too many parties || no automation
  - **Authenticated workflow involves too many steps**

- RFC 8078 brought **parent pulling**
  - **automatic, in-band** (CDS / CDNSKEY)
  - **not secure for bootstrapping** → “accept after delay”

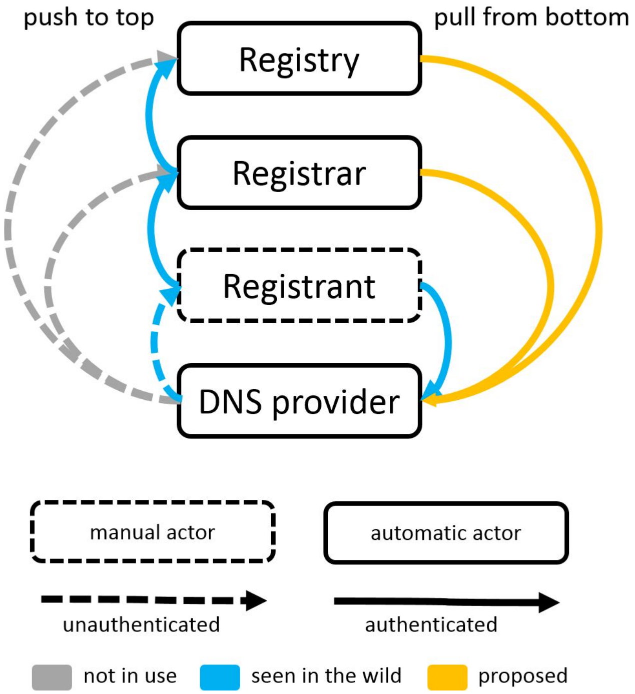


The State of DS Bootstrapping

- Various methods available, with downsides
  - TOFU, manual submission, REST interfaces etc.
  - unauthenticated || out of band || slow || stateful || error-prone || too many parties || no automation
  - **Authenticated workflow involves too many steps**

- RFC 8078 brought **parent pulling**
  - **automatic, in-band** (CDS / CDNSKEY)
  - **not secure for bootstrapping** → "accept after delay"

- **Goal: add authentication** for parent pulling
  - automated, immediate, in-band, stateless



Solution:
Transferring Trust from the DNS Operator

What’s the idea?

1. Create a **signaling mechanism for DNS operators**
   - **What?**
     - allow **publishing arbitrary information** about the zones they are authoritative for
     - in an **authenticated** fashion, **on a per-zone basis**
   - **How?**
     - use namespace **under each nameserver hostname** with **zone-specific subdomains**
     - **require DNSSEC** (requires nameserver domains to be secure)

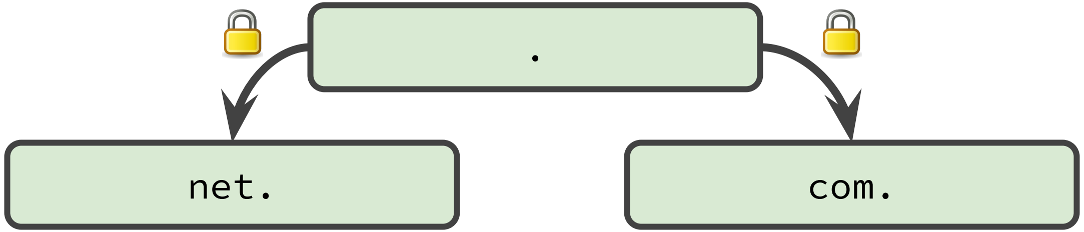
What's the idea?

1. Create a **signaling mechanism for DNS operators**
   - **What?**
     - ➢ allow **publishing arbitrary information** about the zones they are authoritative for
     - ➢ in an **authenticated** fashion, **on a per-zone basis**
   - **How?**
     - ➢ use namespace **under each nameserver hostname** with **zone-specific subdomains**
     - ➢ **require DNSSEC** (requires nameserver domains to be secure)
2. Use this to **publish authentication signal** for CDS/CDNSKEY
   - start with **CDS/CDNSKEY records at the apex** of the target zone (RFC 8078)
   - **co-publish** these records **via signaling mechanism** (signed with NS zone's keys)

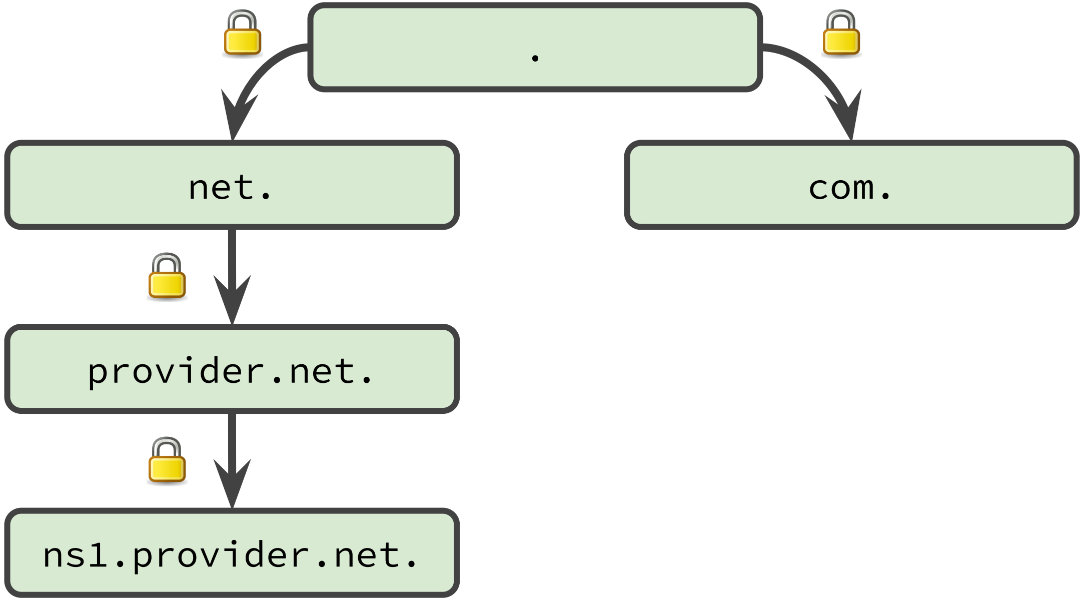
What’s the idea?

1. Create a **signaling mechanism for DNS operators**
   - **What?**
     - ➢ allow **publishing arbitrary information** about the zones they are authoritative for
     - ➢ in an **authenticated** fashion, **on a per-zone basis**
   - **How?**
     - ➢ use namespace **under each nameserver hostname** with **zone-specific subdomains**
     - ➢ **require DNSSEC** (requires nameserver domains to be secure)
2. Use this to **publish authentication signal** for CDS/CDNSKEY
   - start with **CDS/CDNSKEY records at the apex** of the target zone (RFC 8078)
   - **co-publish** these records **via signaling mechanism** (signed with NS zone’s keys)
3. **Validate** the target domain’s CDS/CDNSKEY records **against this signal**
   - if successful: **“transfer trust to the target domain”** → **provision DS records** at parent

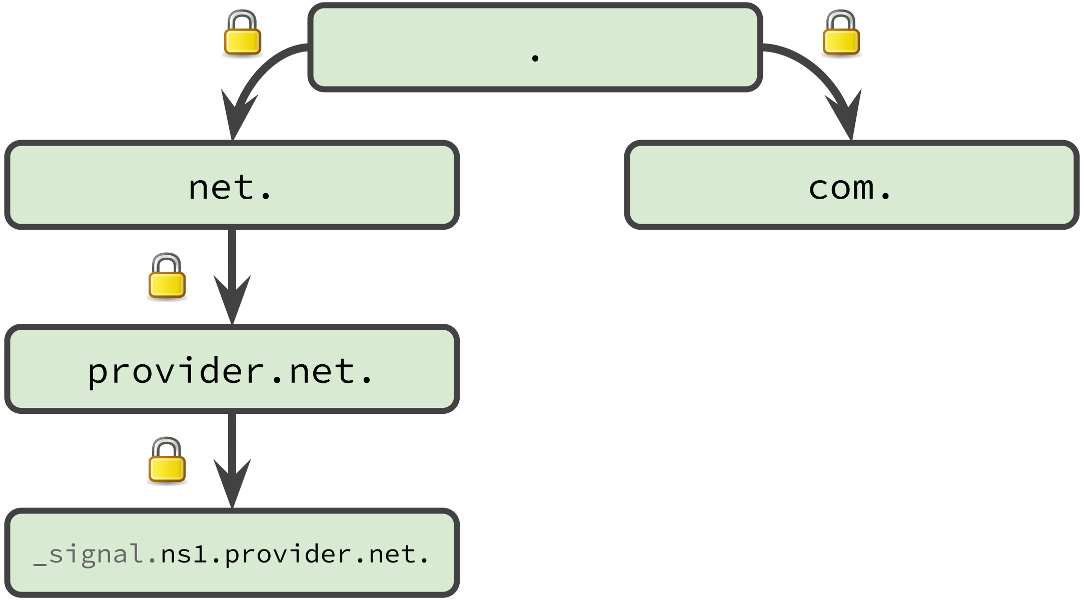
CDS/CDNSKEY Authentication via Nameserver Signaling



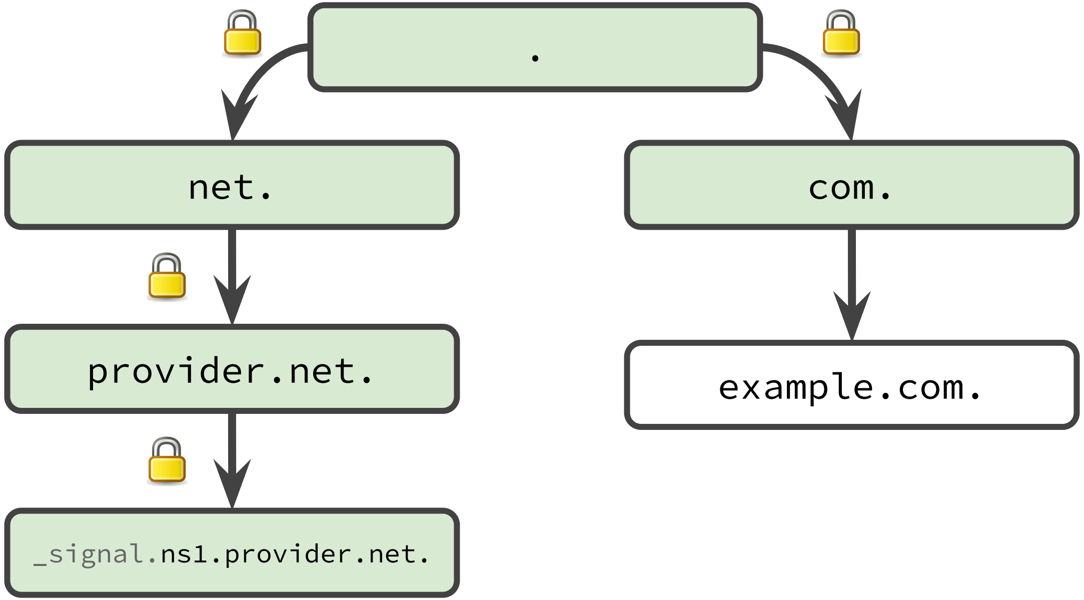
CDS/CDNSKEY Authentication via Nameserver Signaling



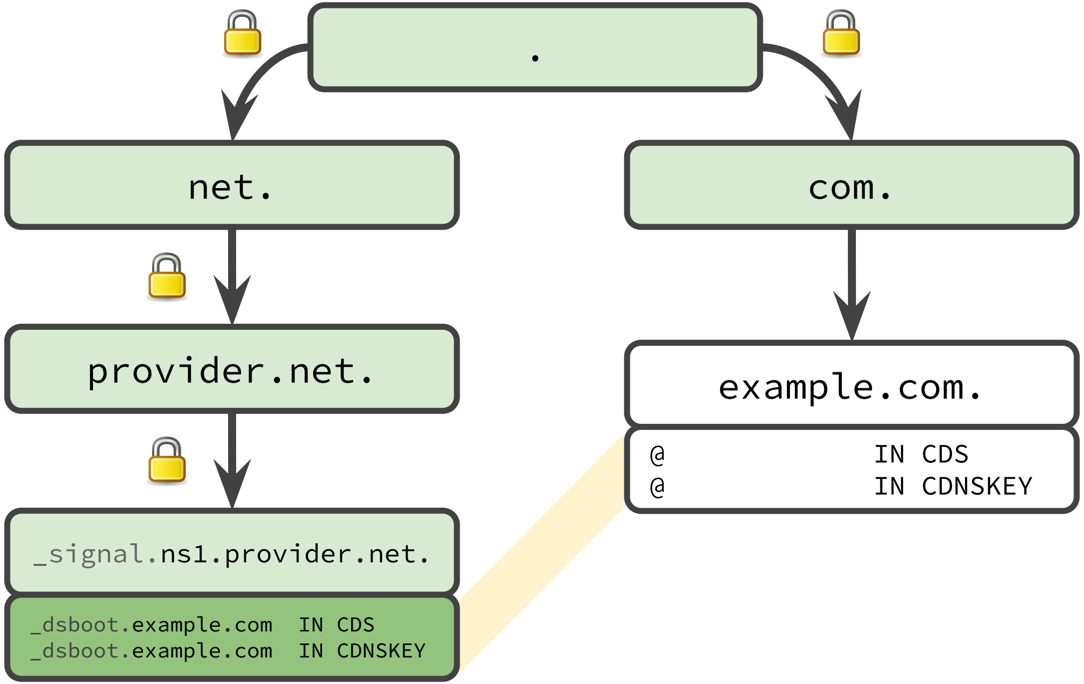
CDS/CDNSKEY Authentication via Nameserver Signaling



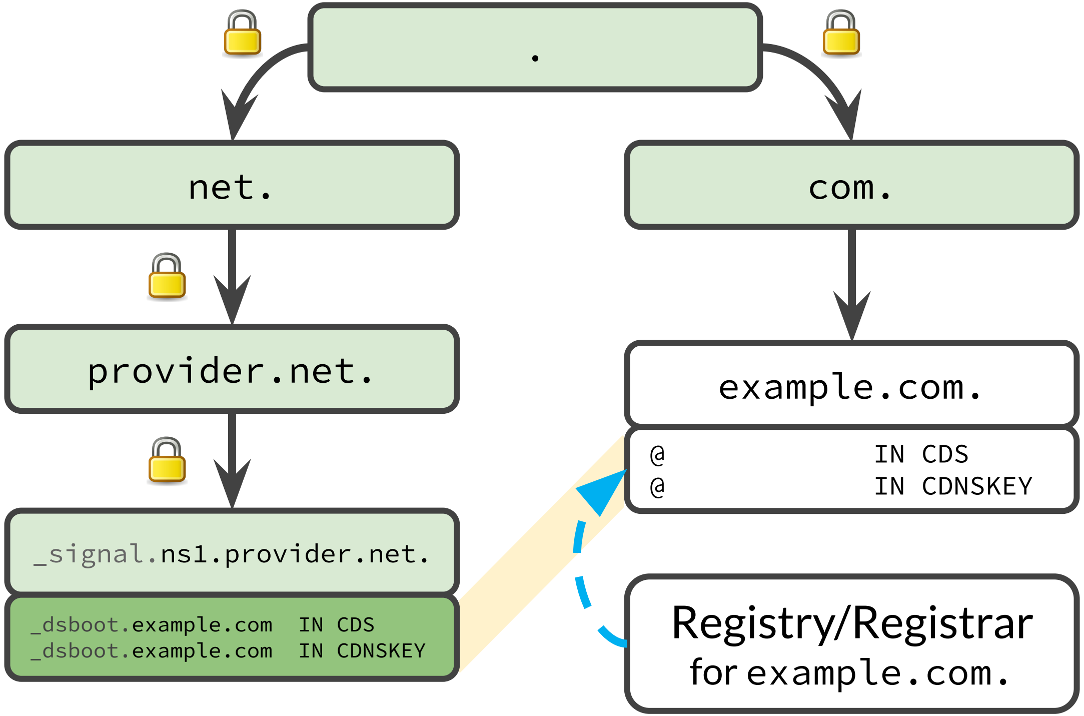
CDS/CDNSKEY Authentication via Nameserver Signaling



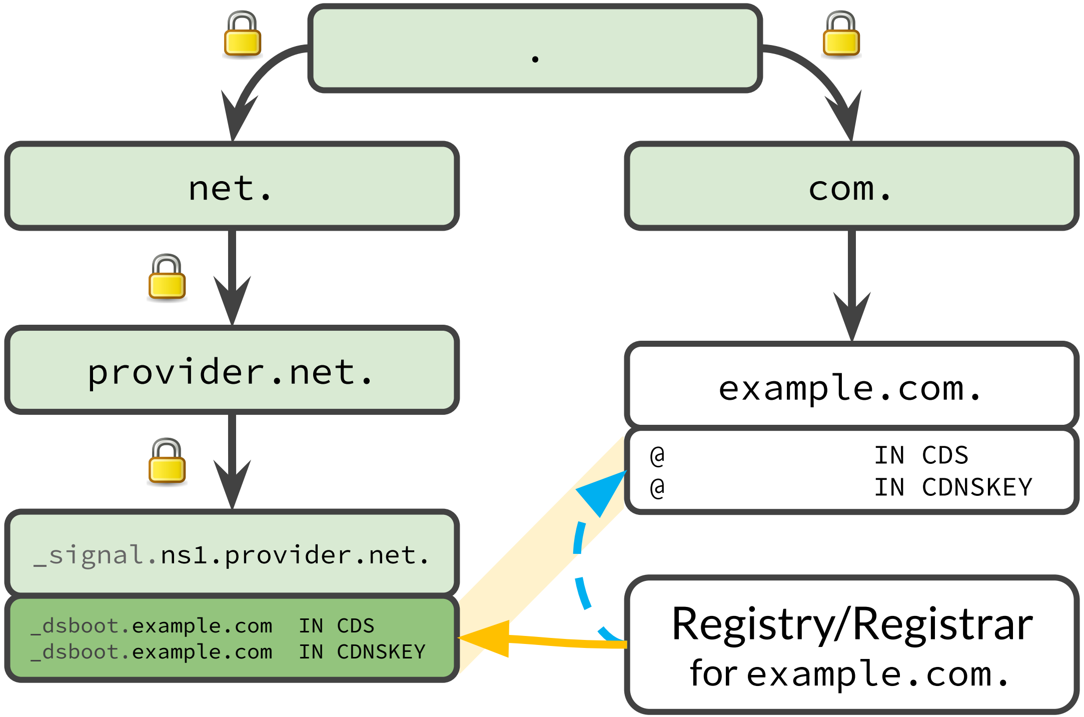
CDS/CDNSKEY Authentication via Nameserver Signaling


.
net.
com.
provider.net.
example.com.
@ IN CDS
@ IN CDNSKEY
_signal.ns1.provider.net.
_dsboot.example.com IN CDS
_dsboot.example.com IN CDNSKEY

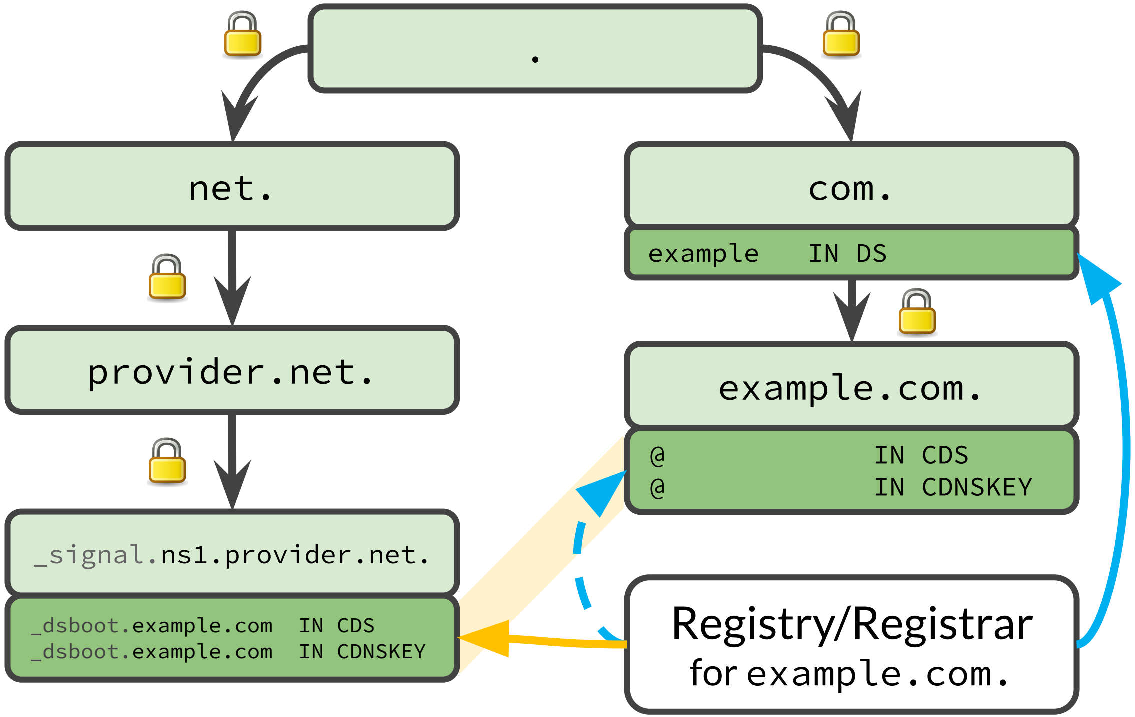
CDS/CDNSKEY Authentication via Nameserver Signaling



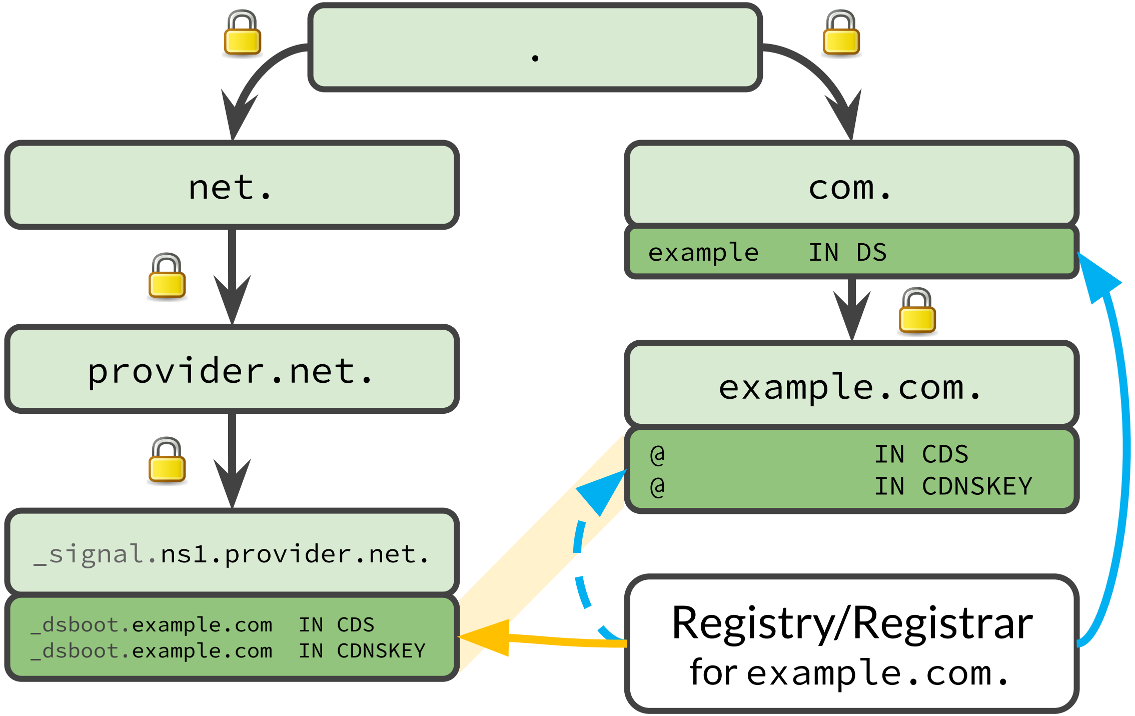
CDS/CDNSKEY Authentication via Nameserver Signaling



CDS/CDNSKEY Authentication via Nameserver Signaling



CDS/CDNSKEY Authentication via Nameserver Signaling



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

- identically co-published
- authenticated, immediate
- no active on-wire attacker

Extends RFC 8078 to add authentication for initial DS

Protocol Details

Algorithm

- **Co-publish CDS/CDNSKEY records** under a subdomain of the NS hostnames:
  → `CDS/CDNSKEY IN _dsboot.example.com._signal.ns1.provider.net`
- Use **DNSSEC to validate** these records, under **each NS hostname**

Protocol Details

Algorithm

- **Co-publish CDS/CDNSKEY records** under a subdomain of the NS hostnames:
  → `CDS/CDNSKEY IN _dsboot.example.com._signal.ns1.provider.net`
- Use **DNSSEC to validate** these records, under **each NS hostname**

Technical Considerations

- Naming scheme with `_signal` label allows delegating to separate zone
  - 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)
- prefix allows different types of signals (e.g. for multi-signer p2p key exchange)

Status & Implementations

- Adopted by IETF DNSOP WG in April 2022
  - Internet Draft: draft-ietf-dnsop-dnssec-bootstrapping
  - Blog: https://blog.apnic.net/2022/03/08/authenticated-bootstrapping-of-dnssec-delegations/

- **Child-side**
  - **Cloudflare: in production**, for all signed domains (announced @ ICANN74)
  - working on (1) **native support at deSEC**, (2) **native support in authoritative servers**

- **Parent-side**
  - PoC for authenticated CDS/CDNSKEY scanning: https://github.com/desec-io/dsbootstrap
  - ccTLDs: .cl close to roll-out; 59 ccTLDs (via CoCCA) and others under way
  - Registrars: GoDaddy has implementation planned

What’s the impact?

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 (.com/.net)

- … and obviously, the zone itself needs to be signed.

- Survey time!

Deployability Survey (Top 1M)

- 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 == false, but c == d?

Deployability Survey (Top 1M): General Results (06/2022)

```
Failure rate .........................................:   1.83%
Remaining sample size ................................:  981747

Proportion of secure zones ...........................:   4.79%
Proportion of signed zones ...........................:   6.36%

Proportion of zones with all nameserver targets secure:  28.65%
Proportion of zones with ≥ 1 nameserver targets secure:  30.01%
```

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

```
Proportion of bootstrappable zones (all NS) ..........:  26.08%
Proportion of bootstrappable zones (≥ 1 NS) ..........:  27.15%
```

Deployability Survey (Top 1M): by TLD and Provider (06/2022)

TLD	Total	Bootstrappable	
com	470,054	25.7%	120,905
ru	58,037	21.8%	12,631
net	59,680	20.9%	12,471
org	48,675	22.1%	10,736
xyz	15,461	63.6%	9,838
top	7,946	63.1%	5,011
quest	3,779	99.0%	3,743
uk	16,020	22.8%	3,649
monster	3,298	98.4%	3,245
io	8,520	33.2%	2,827
Σ	**691,470**		**185,056**

Number of bootstrappable domains by top 10 TLDs.

NS SOA RNAME	Total	Bootstrappable	
dns.cloudflare.com	291,087	80.4%	233,988
dns.hostinger.com	3,655	88.8%	3,245
hostmaster.nsone.net	6,358	39.5%	2,512
noc.dns.icann.org	1,923	99.5%	1,914
(multiple)	78,399	2.0%	1,600
hostmaster.cscdns.net	5,289	20.9%	1,103
dns.openprovider.eu	1,065	94.4%	1,005
postmaster.iij.ad.jp	839	97.7%	820
nstld.versign-grs.com	6,808	11.1%	755
dnstech.comaude.com	591	92.9%	549
Σ	**396,014**		**247,491**

Number of bootstrappable domains by top 10 DNS providers (as inferred from RNAME of the SOA record of name server names, if consistent across all name servers).

Outlook

Document Status

- Authors not aware of any remaining open issues, implementation proceeding
- Going to ask for WG Last Call

What now?

- Document review / suggestions for improvement
  - https://datatracker.ietf.org/doc/draft-ietf-dnsop-dnssec-bootstrapping/
- Registrars / ccTLD registries → **Implementations!** 🤩
- **Let's make DNSSEC easy.**

Thank you!

… also to our sponsor:

SSE

Questions?

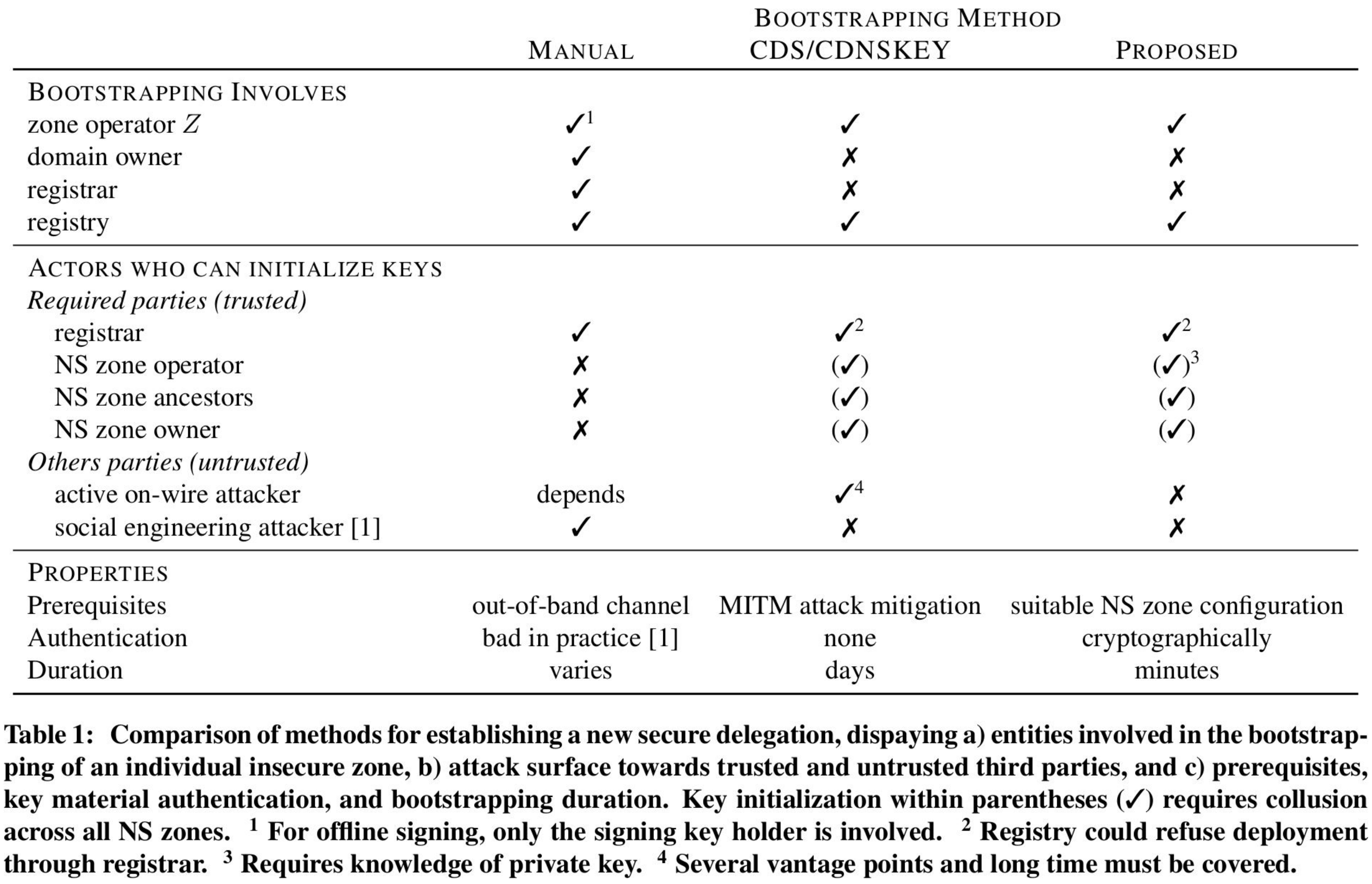
Backup

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

- Mitigations exist, e.g:
    - monitor delegation
    - diversify NS TLDs
    - multiple vantage points



		Bootstrapping Method	
	Manual	CDS/CDNSKEY	Proposed
Bootstrapping Involves			
zone operator Z	✓[1]	✓	✓
domain owner	✓	✗	✗
registrar	✓	✗	✗
registry	✓	✓	✓
Actors who can initialize keys			
Required parties (trusted)			
registrar	✓	✓[2]	✓[2]
NS zone operator	✗	(✓)	(✓)[3]
NS zone ancestors	✗	(✓)	(✓)
NS zone owner	✗	(✓)	(✓)
Others parties (untrusted)			
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, dispaying 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.