

# DNSSEC

Christoph Egger

21. August 2016

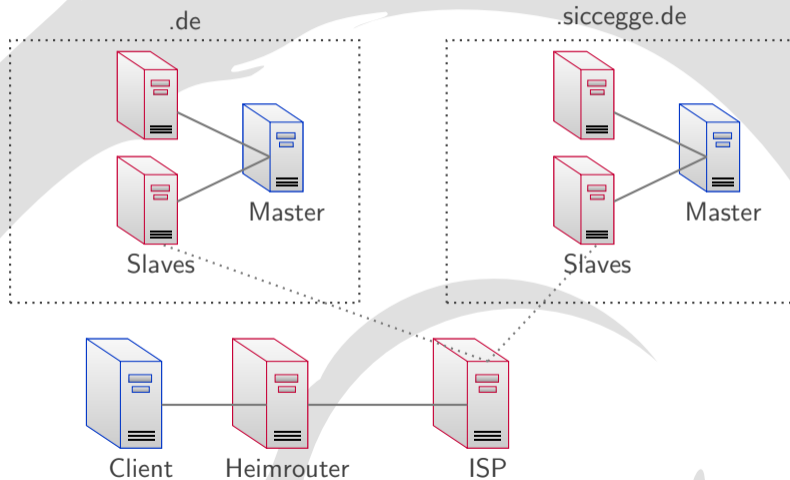


## WIKIPEDIA

The Domain Name System Security Extensions (DNSSEC) is a suite of Internet Engineering Task Force (IETF) specifications for securing certain kinds of information provided by the Domain Name System (DNS) as used on Internet Protocol (IP) networks. It is a set of extensions to DNS which provide to DNS clients (resolvers) origin authentication of DNS data, authenticated denial of existence, and data integrity, but not availability or confidentiality.

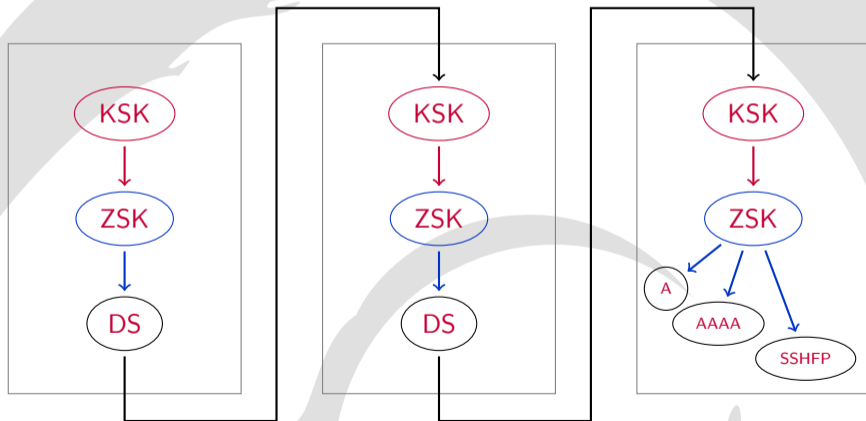


## DNS ANFRAGE



**KSK** "KeySigningKey" – wird in der übergeordneten Zone referenziert und signiert alle Schlüssel *in* der Zone

**ZSK** "ZoneSigningKey" – wird durch den KSK autorisiert und signiert weitere Einträge



## RRSIG

## SICCEGGE.DE

```
siccegge.de. IN A 62.113.200.104
```

```
siccegge.de. IN RRSIG A 8 2 43200 20140908181927 20140809171927 60018 siccegge.de.  
zldkAFJKKV4/gkmZ8DZkV7AT6nIt4mLXjClJwSnGqvrlBWEzc9h3knLMa9iJeEh01ZEZcWi+JRD/vVvNqBg4P1  
vCGsiPDvzBv0+gq0wtXPPpouNZA9r9h9in4sB3Vw/6HpMcqp843mB+B5SGQZkALDsVCcoY4JO/rPWPXYGHQkA=
```



# SCHLÜSSELTAUSCH

## IDEE

Wechsle die Schlüssel regelmäßig. Damit lassen sich auch kleine, effizientere Schlüssel verwenden (DNS verwendet UDP!). Auch in Sachen “Revocation” nützlich



# SCHLÜSSELTAUSCH

## IDEE

Wechsle die Schlüssel regelmäßig. Damit lassen sich auch kleine, effizientere Schlüssel verwenden (DNS verwendet UDP!). Auch in Sachen “Revocation” nützlich

Schlüssel wechseln in DNS ist nicht so einfach:



# SCHLÜSSELTAUSCH

## IDEE

Wechsle die Schlüssel regelmäßig. Damit lassen sich auch kleine, effizientere Schlüssel verwenden (DNS verwendet UDP!). Auch in Sachen “Revocation” nützlich

Schlüssel wechseln in DNS ist nicht so einfach: Stichpunkt TTL





# SCHLÜSSELTAUSCH

## IDEE

Wechsle die Schlüssel regelmäßig. Damit lassen sich auch kleine, effizientere Schlüssel verwenden (DNS verwendet UDP!). Auch in Sachen “Revocation” nützlich

Schlüssel wechseln in DNS ist nicht so einfach: Stichpunkt TTL

2 Methoden:

- Neuen Schlüssel vor der Verwendung veröffentlichen
- Vorübergehend die Daten mit beiden Schlüsseln signieren



# NEGATIVE ANTWORTEN

## PROBLEM

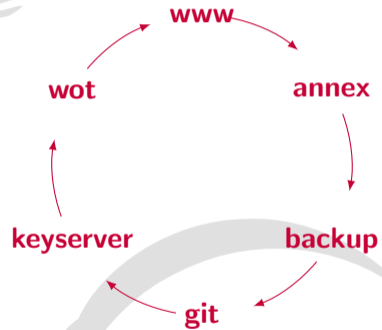
Mit den RRSIGs lassen sich bestehende Einträge im DNS bestätigen. Es ist aber immer noch möglich, Einträge “verschwinden” zu lassen. Was also noch fehlt ist die Möglichkeit, die nicht-Existenz von Einträgen zu signieren.

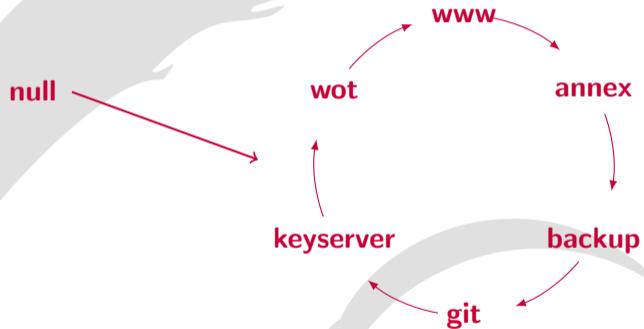


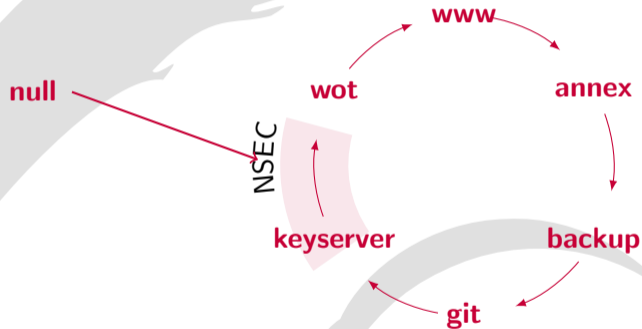
# NSEC (NEXT SECURE)

- Bilde einen Kreis, der alle vorhandenen Einträge umfasst









# NSEC (NEXT SECURE)

- Bilde einen Kreis, der alle vorhandenen Einträge umfasst
- Speichere signierte Feststellung, dass zwischen zwei Namen kein dritter liegt
- Bei negativer Antwort (NXDOMAIN) sende auch den signierten NSEC Eintrag in dessen Interval die Antwort liegen würde



# NSEC (NEXT SECURE)

- Bilde einen Kreis, der alle vorhandenen Einträge umfasst
- Speichere signierte Feststellung, dass zwischen zwei Namen kein dritter liegt
- Bei negativer Antwort (NXDOMAIN) sende auch den signierten NSEC Eintrag in dessen Intervall die Antwort liegen würde
- “Zonewalking” auflistung aller Einträge in einer Zone





# NSEC3

- Statt Einträge in einem Ring anzuordnen, bilde zuerst eine kryptographische Streusumme



Diagram illustrating a sequence of DNS records:

- null
- wot
- annex
- backup
- git
- keyserver

The records are connected by red arrows in a clockwise cycle: null → wot → annex → backup → git → keyserver → null.

keyserver

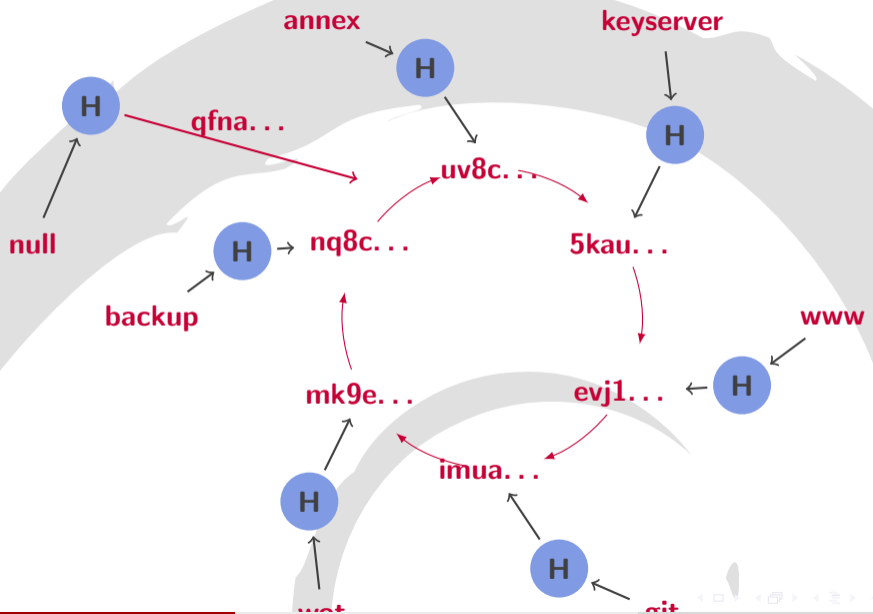
wot

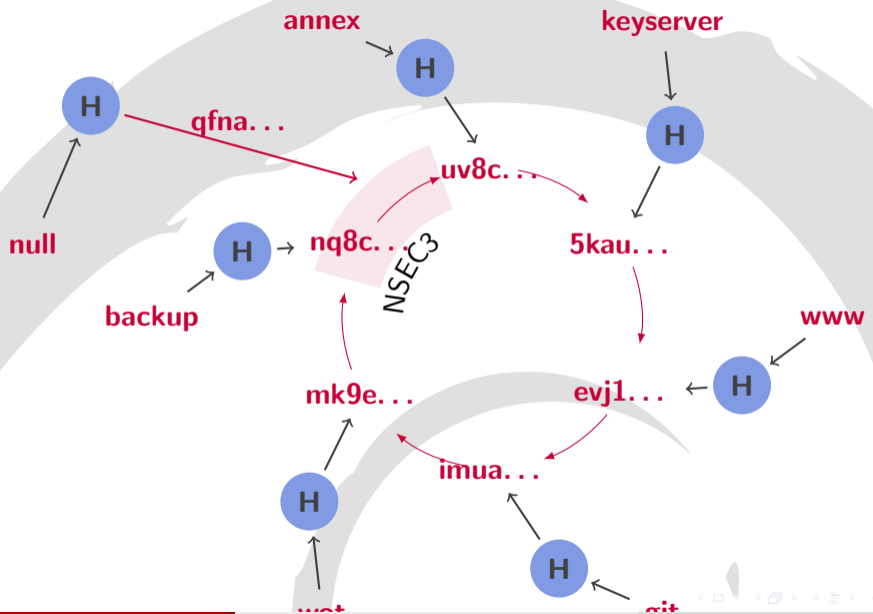
www

annex

backup

git





# NSEC3

- Statt Einträge in einem Ring anzuordnen, bilde zuerst eine kryptographische Streusumme
- Verwende Salz und mehrere Runden der Streufunktion für maximalen Effekt.

[GIT.SICCEGGE.DE](https://git.siccegge.de)

```
siccegge.de.  IN  NSEC3PARAM  1  0  5  6D1DAF17E2A6A252
```



# ÜBERPRÜFUNG NEGATIVER ANTWORTEN

## ZIEL

Es ist trivial, in der de-Zone zu zeigen, dass dort `www.siccegge.de` nicht existiert – obwohl der name durchaus vorhanden ist (allerdings nicht in der de-Zone sondern in der `siccegge.de`-Zone). Wir müssen also auch zeigen, dass wir in der “richtigen” Zone operieren.



# ÜBERPRÜFUNG NEGATIVER ANTWORTEN

## ZIEL

Es ist trivial, in der de-Zone zu zeigen, dass dort `www.siccegge.de` nicht existiert – obwohl der name durchaus vorhanden ist (allerdings nicht in der de-Zone sondern in der `siccegge.de`-Zone). Wir müssen also auch zeigen, dass wir in der “richtigen” Zone operieren.

## “CLOSEST ENCLOSER”

Daher 3 NSEC3-Einträge:

- Für die kürzeste, nicht mehr existente Oberdomäne zur Anfrage, den NSEC3-Eintrag, der das Intervall überspannt.
- Den um eine Komponente gekürzten NSEC3-Eintrag, der entweder *keinen* NS-Eintrag oder auch das Flag für SOA enthält.

# ÜBERPRÜFUNG NEGATIVER ANTWORTEN

## ZIEL

Es ist trivial, in der de-Zone zu zeigen, dass dort `www.siccegge.de` nicht existiert – obwohl der name durchaus vorhanden ist (allerdings nicht in der de-Zone sondern in der `siccegge.de`-Zone). Wir müssen also auch zeigen, dass wir in der “richtigen” Zone operieren.

## “CLOSEST ENCLOSER”

Daher 3 NSEC3-Einträge:

- Für die kürzeste, nicht mehr existente Oberdomäne zur Anfrage, den NSEC3-Eintrag, der das Intervall überspannt.
- Den um eine Komponente gekürzten NSEC3-Eintrag, der entweder *keinen* NS-Eintrag oder auch das Flag für SOA enthält.
- Den NSEC3-Eintrag, der das Fehlen eines Wildcard-Eintrags an dieser Stelle nachweist.



## NEGATIVE ANTWORT

## SICCEGGE.DE HAT SOA

```
4ma0fb5t2s6kjtgc6r3qi4o49bn7pc4i.siccegge.de. 3573 IN NSEC3 1 0 5 6D1DAF17E2A6A252
4TRVQLKF545FSK90ED6NCJ7DGM0JB6I8 A NS SOA MX AAAA RRSIG DNSKEY NSEC3PARAM
```

null.siccegge.de hat den Hash-Wert qfna56rlmnlbp3e85m4d6ckonnmpfg1i

## NULL.SICCEGGE.DE EXISTIERT NICHT

```
qd2uevk27c2tdrh6535e0mkiratu1t5h.siccegge.de. 3600 IN NSEC3 1 0 5 6D1DAF17E2A6A252
QLLMC1NCRMN4AU8QCFQ24VAH7JFM6LQ6
```

\*.siccegge.de hat den Hash-Wert 68m2atv971213e67oua61u5hp0v0273a.

## \*.SICCEGGE.DE EXISTIERT NICHT

```
63r09adu0p1vdmkif5eb4dr6m2a3l5cp.siccegge.de. 3600 IN NSEC3 1 0 5 6D1DAF17E2A6A252
6BJ555D3Q50SL34D50L1PGU887R73DC9 RRSIG TLSA
```

# DANE

Nachdem unser DNS jetzt kryptographisch abgesichert ist (auch nicht schlechter als das CA System) kann man dort jetzt sicher weiteres Schlüsselmaterial ausliefern:

- TLSA für alles was SSL/TLS macht
- SSHFP für SSH Fingerprints
- PGP-Schlüssel-Einträge
- ...



# TLSA

## TLSA

```
.25._tcp.oteiza.siccegge.de.  IN  TLSA  3 1 1 101B5B5CCDC5568CEC385552611FD0355BF15DB293E96F46E29DE4AOC4B2BC3F
.443._tcp.siccegge.de.      IN  TLSA  3 1 1 62BEBD9F2E77CF26A4006A50F69FC3891BF7BEDDAEF8AC96E57C1D9BA2AB1F73
.5222._tcp.xmpp.egger.im    IN  TLSA  3 1 1 9c93fab0d88c911592dedfa7f9385aeee228b0c6d526813ad1182c983677736b
```



# TLSA

## TLSA

```
_25._tcp.oteiza.siccegge.de.  IN  TLSA  3 1 1 101B5B5CCDC5568CEC385552611FD0355BF15DB293E96F46E29DE4AOC4B2BC3F
_443._tcp.siccegge.de.      IN  TLSA  3 1 1 62BEBD9F2E77CF26A4006A50F69FC3891BF7BEDDAEF8AC96E57C1D9BA2AB1F73
_5222._tcp.xmpp.egger.im    IN  TLSA  3 1 1 9c93fab0d88c911592dedfa7f9385aeee228b0c6d526813ad1182c983677736b
```

Achtung! Beim Schlüsseltausch gibt's wieder Spass.



# TLSA

## TLSA

```
_25._tcp.oteiza.siccegge.de.  IN  TLSA  3 1 1 101B5B5CCDC5568CEC385552611FD0355BF15DB293E96F46E29DE4AOC4B2BC3F  
_443._tcp.siccegge.de.      IN  TLSA  3 1 1 62BEBD9F2E77CF26A4006A50F69FC3891BF7BEDDAEF8AC96E57C1D9BA2AB1F73  
_5222._tcp.xmpp.egger.im    IN  TLSA  3 1 1 9c93fab0d88c911592dedfa7f9385aeee228b0c6d526813ad1182c983677736b
```

Achtung! Beim Schlüsseltausch gibt's wieder Spass.

- 3: Bezeichnet ein Service Zertifikat
- 1: Angegeben wird der öffentlich Schlüssel, nicht das Zertifikat
- 1: Angegeben wird eine SHA256-Summe



# SSHFP

## GIT.SICCEGGE.DE

```
git.siccegge.de IN SSHFP 1 1 0E812EE0A3704230F3C415076E1BAA149A5DC75B
git.siccegge.de IN SSHFP 1 2 1CBACAF365040DC1DF841FD07D9186BC343D4AF7DCF689CC8CF4A2F75D7F4B57
git.siccegge.de IN SSHFP 3 1 A2D0495E912DA039EEA51A1593F7F74FB919AAD4
git.siccegge.de IN SSHFP 3 2 9BF73E3654AA65B847054247F85EFB5C88AB7460840B9C922E647B00696661CF
git.siccegge.de IN SSHFP 4 1 2A3EF64AC589193ACFAD783B62E3C193A67F3F46
git.siccegge.de IN SSHFP 4 2 880686195D6C1AAA6791F3A3EF4E7B565DCF9F560F2F1BBB93C56EFD5996F335
```



# SSHFP

## GIT.SICCEGGE.DE

```
git.siccegge.de IN SSHFP 1 1 0E812EE0A3704230F3C415076E1BAA149A5DC75B
git.siccegge.de IN SSHFP 1 2 1CBACAF365040DC1DF841FD07D9186BC343D4AF7DCF689CC8CF4A2F75D7F4B57
git.siccegge.de IN SSHFP 3 1 A2D0495E912DA039EEA51A1593F7F74FB919AAD4
git.siccegge.de IN SSHFP 3 2 9BF73E3654AA65B847054247F85EFB5C88AB7460840B9C922E647B00696661CF
git.siccegge.de IN SSHFP 4 1 2A3EF64AC589193ACFAD783B62E3C193A67F3F46
git.siccegge.de IN SSHFP 4 2 880686195D6C1AAA6791F3A3EF4E7B565DCF9F560F2F1BBB93C56EFD5996F335
```

- Erste Zahl: Hostkeytyp
- Zweite Zahl: Prüfsummentyp



# ÜBERBLICK

## NAMESERVER

Müssen zusätzliche Einträge ausliefern (RRSIG, NSEC3). Für NSEC3 müssen die richtigen Einträge gefunden werden





# ÜBERBLICK

## NAMESERVER

Müssen zusätzliche Einträge ausliefern (RRSIG, NSEC3). Für NSEC3 müssen die richtigen Einträge gefunden werden

## SIGNATURWERKZEUGE

- Müssen RRSIGs für die vorhandenen Einträge erstellen und gelegentlich erneuern
- Müssen die NSEC3- und NSEC3PARAM-Einträge erstellen und signieren
- Sollten Möglichkeit zum Schlüsseltausch bieten



# ÜBERBLICK

## NAMESERVER

Müssen zusätzliche Einträge ausliefern (RRSIG, NSEC3). Für NSEC3 müssen die richtigen Einträge gefunden werden

## SIGNATURWERKZEUGE

- Müssen RRSIGs für die vorhandenen Einträge erstellen und gelegentlich erneuern
- Müssen die NSEC3- und NSEC3PARAM-Einträge erstellen und signieren
- Sollten Möglichkeit zum Schlüsseltausch bieten

## REGISTRAR

Irgendwie müssen die Schlüssel in die darüberliegende Zone kommen. Wenige Registrare haben das schon im Interface vorgesehen, etliche lassen sich aber per Mail an den Support überreden

# NAMESERVER

## SOFTWARE

Alle nennenswerten Nameserver (nsd, bind, powerdns, knot, ...) können heutzutage DNSSEC ausliefern.



# NAMESERVER

## SOFTWARE

Alle nennenswerten Nameserver (nsd, bind, powerdns, knot, ...) können heutzutage DNSSEC ausliefern.

## SEKUNDÄRSERVER

Kaum ein kostenfreier Sekundärserveranbieter unterstützt DNSSEC – das liegt unter anderem an den deutlich größeren Antworten und dem Rechenbedarf für NSEC3, die signifikant Ressourcen verbrauchen.

⇒ Selber hosten (mit Freunden), beim Registrar schauen oder bezahlen.

# SIGNATURWERKZEUGE

Im Grunde gibt es zwei Typen von Signaturwerkzeugen

IM PRIMÄREN NAMESERVER

BIND, Knot, PowerDNS

**VORTEILE** Keine weiteren Werkzeuge, dynamische Updatesmöglich

**NACHTEILE** Schlüsselmaterial im Netzwerkservers, bestehende Implementierungen unflexibel in Sachen Schlüsselrotation



# SIGNATURWERKZEUGE

Im Grunde gibt es zwei Typen von Signaturwerkzeugen

## IM PRIMÄREN NAMESERVER

BIND, Knot, PowerDNS

**VORTEILE** Keine weiteren Werkzeuge, dynamische Updatesmöglich

**NACHTEILE** Schlüsselmaterial im Netzwerkservers, bestehende Implementierungen unflexibel in Sachen Schlüsselrotation

## SEPARATES SIGNATURWERKZEUG

OpenDNSSEC, dnssec-tools, cron

**VORTEILE** Flexibel, Signaturlösung Nameserver-agnostisch

**NACHTEILE** Softwarequalität . . . , weiteres Element, das kaputt gehen kann

# FRAGEN?

Download: <https://static.siccegge.de/talks/dnssec-FrOSCon-2016-08-21.pdf>  
<https://git.siccegge.de/?p=talk/dnssec.git>

# 42

