

# DNSSEC - Keyrollover

Patrick Koetter und Carsten Strotmann, sys4 AG

# Agenda

---

- Gründe für DNSSEC Key-Rollover
- ZSK-Rollover
- KSK-Rollover
- Algorithmus Rollover
- CDS/CDNSKEY Records

# DNSSEC Key-Rollover

---

# Warum Key-Rollover

- Das DNSSEC-Schlüsselmaterial ist öffentlich
  - Einschließlich der Signaturen und des dazugehörigen Klartextes (DNS-Records)
- Der private Schlüssel kann in unbefugte Hände gelangen
  - Versehentlich
  - Durch Angriffe auf den Signierserver/das HSM
  - Bei Verwendung von *online-signing* müssen die privaten Schlüssel auf dem signierenden DNS-Server liegen
- Die Schlüssellänge oder der verwendete Algorithmus ist nicht für eine längere Benutzung geeignet (z.B. 1024bit RSA-Schlüssel)

## Die Herausforderungen

- Das DNS-System ist nicht konsistent
  - DNS-Zonendaten können für einige Zeit zwischen autoritativen Server der gleichen Zone abweichen (Verzögerung bei der Zonenübertragung)
  - DNS-Daten werden in DNS-Resolvern, Betriebssystemen und Anwendungen zwischengespeichert
- Während eines DNSSEC-Schlüsselwechsels muss die Vertrauenskette zu jeder Zeit und von jedem Punkt des Internets ununterbrochen sein

## DNSSEC Keyrollover Dokumentation

- ↪RFC 6781 - "DNSSEC Operational Practices, Version 2"
- ↪RFC 7583 - "DNSSEC Key Rollover Timing Considerations"

# Schlüssel-Rollover, wann und wie oft?

- DNSSEC-Schlüssel haben keine technische Lebensdauer, sie *verfallen* nicht.
- Die *operative* Lebensdauer von DNSSEC-Schlüsseln wird von den zuständigen Administrator(en) festgelegt und kann jederzeit geändert werden.
- Die DNSSEC-Gemeinschaft hat unterschiedliche Auffassungen über die Erneuerung von KSK-Schlüsseln
  - Häufig und regelmäßig
  - Häufig, aber unregelmäßig (um Angreifern keine Informationen zu geben, wann das System aufgrund des Schlüsselwechsels anfälliger sein könnte)
  - Nur wenn es Hinweise gibt, dass der Schlüssel kompromittiert oder gestohlen wurde

# ZSK Rollover

---



## ZSK-Rollover

- Der Zone-Signing-Key (ZSK) hat keine Abhängigkeiten zu externen Ressourcen (wie z.B. die übergeordnete Zone)
- Ein ZSK-Rollover kann zu jeder Zeit gestartet werden.
- Für den ZSK-Rollover wird das 'pre-publication' Rollover-Schema verwendet

## ZSK - Pre-Publication - Schritt 1

- Erzeugen eines neuen ZSK-Schlüsselpaares
- Veröffentlichung des öffentlichen Teils des neuen Schlüssels (DNSKEY) des ZSK in der Zone
- Der aktuelle/alte ZSK wird in der Zone beibehalten
- Die Zone wird mit dem aktuellen/alten (nicht dem neuen) ZSK und KSK signiert

## ZSK - pre-publication - Schritt 2

- Warten, bis die Zone mit dem neuen ZSK auf allen autoritativen DNS-Servern der Zone sichtbar ist (Zonentransfer)
- Warten auf die TTL des DNSKEY RRset (+ etwas Puffer zur Sicherheit)
- Jetzt können wir sicher sein, dass der neue ZSK DNSKEY-Eintrag in allen Caches ist (DNS Resolver, Betriebssysteme, Anwendungen)

### ZSK - pre-publication - Schritt 3

- Signieren der Zone mit dem neuen ZSK
- Der neue ZSK ist jetzt *aktiv*, der alte ZSK ist jetzt *in Rente* (retired)
- Die alte ZSK bleibt vorerst in der Zone erhalten (sie wird benötigt, um alte Signaturen zu validieren, die noch in den Zwischenspeichern des Netzes vorhanden sind)

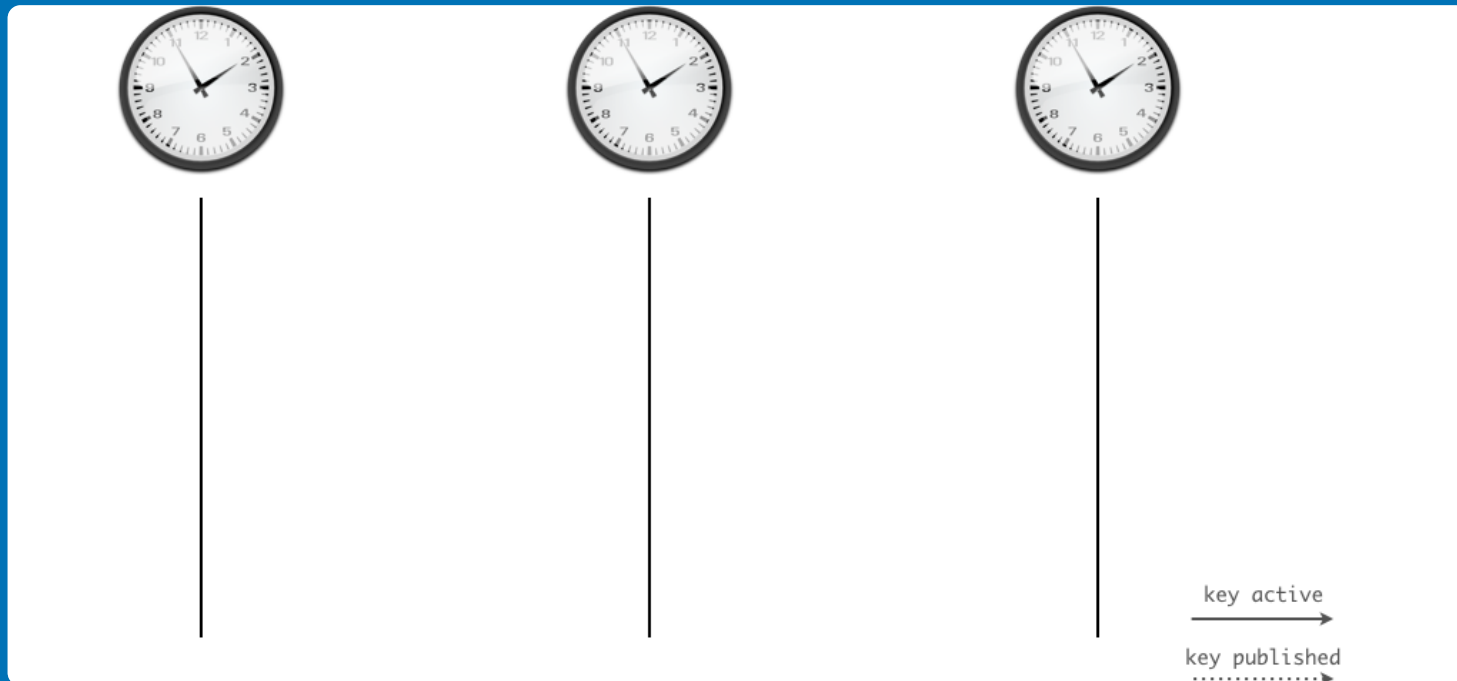
## ZSK - pre-publication - Schritt 4

- Warten Sie, bis die neue Zonenversion mit den Signaturen des neuen ZSK auf allen autoritativen DNS-Servern der Zone sichtbar ist (Zonentransfer)
- Warten auf die größte TTL in der Zone (plus etwas Puffer)
- Jetzt werden die vom alten ZSK erstellten Signaturen aus den Caches gelöscht, und die neuen Signaturen sind verfügbar

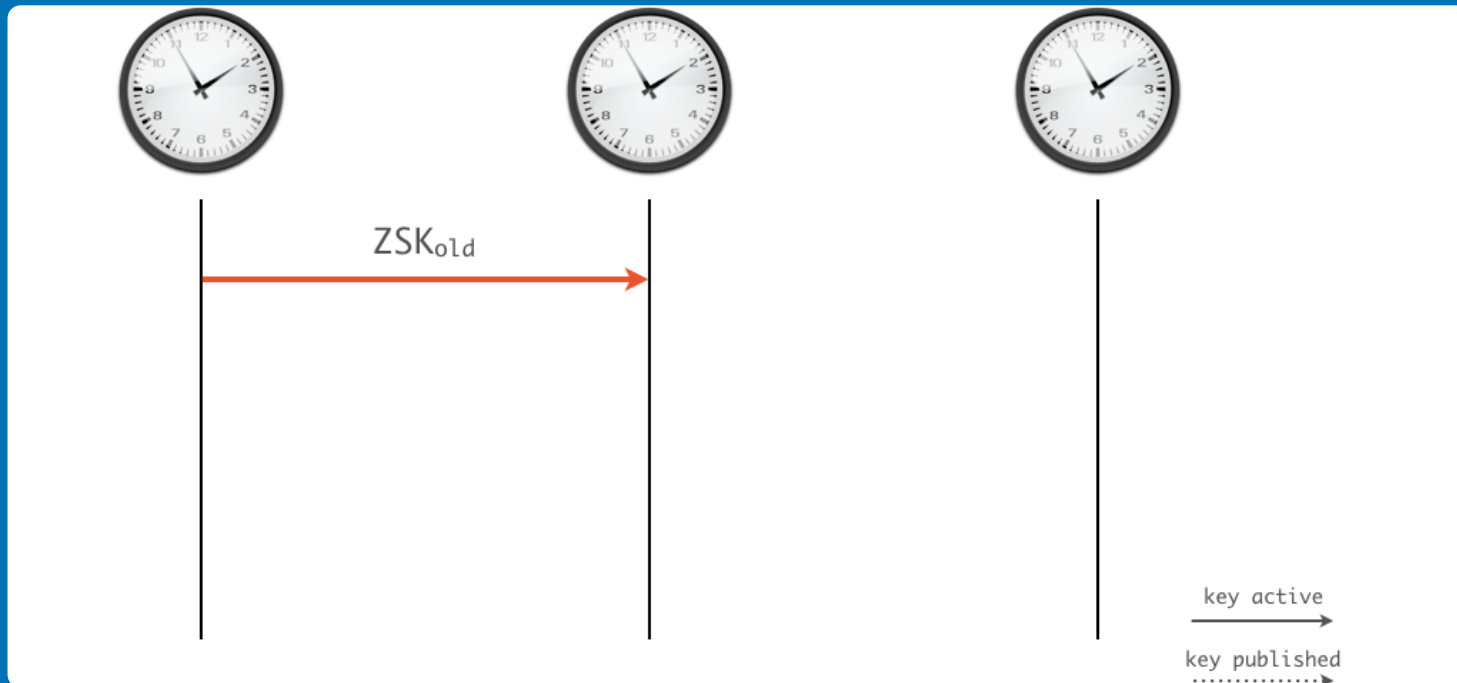
## ZSK - pre-publication - Schritt 5

- Entfernen des alten ZSK aus dem DNSKEY-Recordset der Zone
- Fortsetzen der Signierung der Zone mit dem neuen ZSK

## ZSK - pre-publication in Bildern

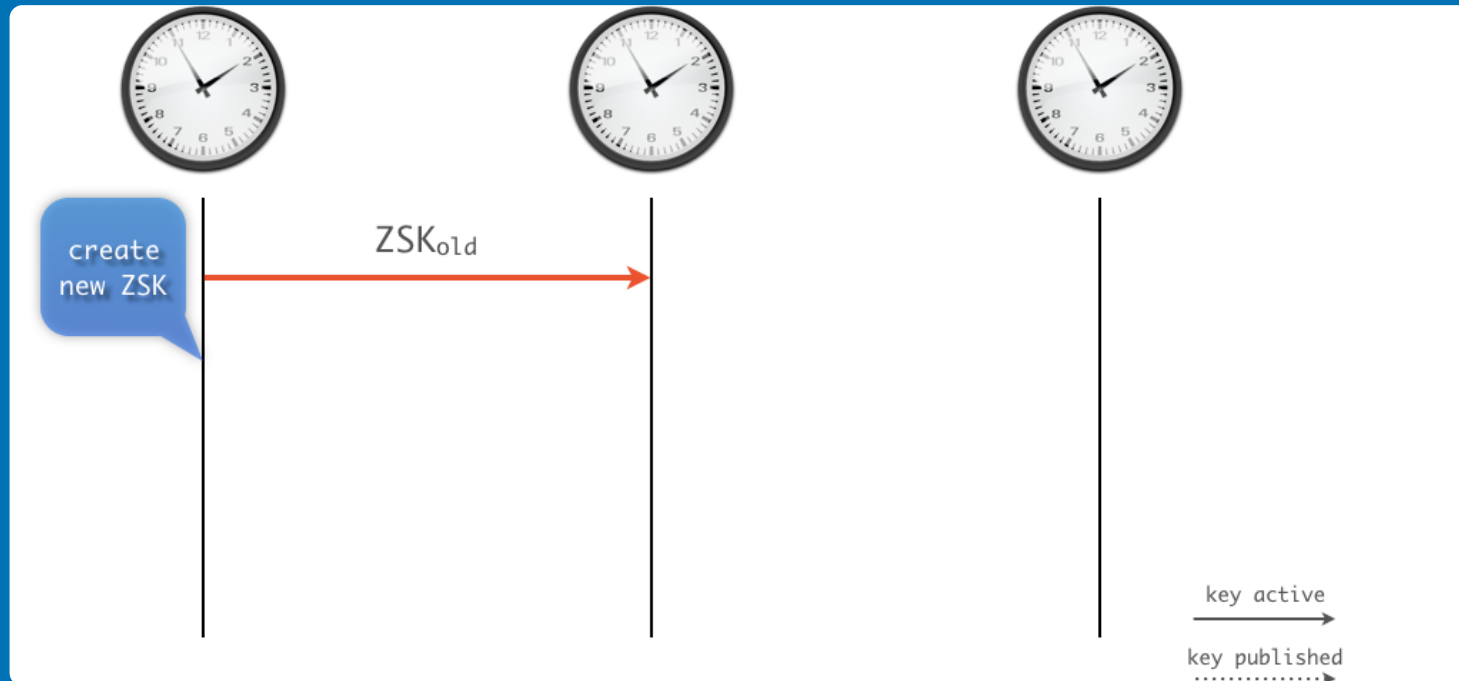


## ZSK - pre-publication in Bildern

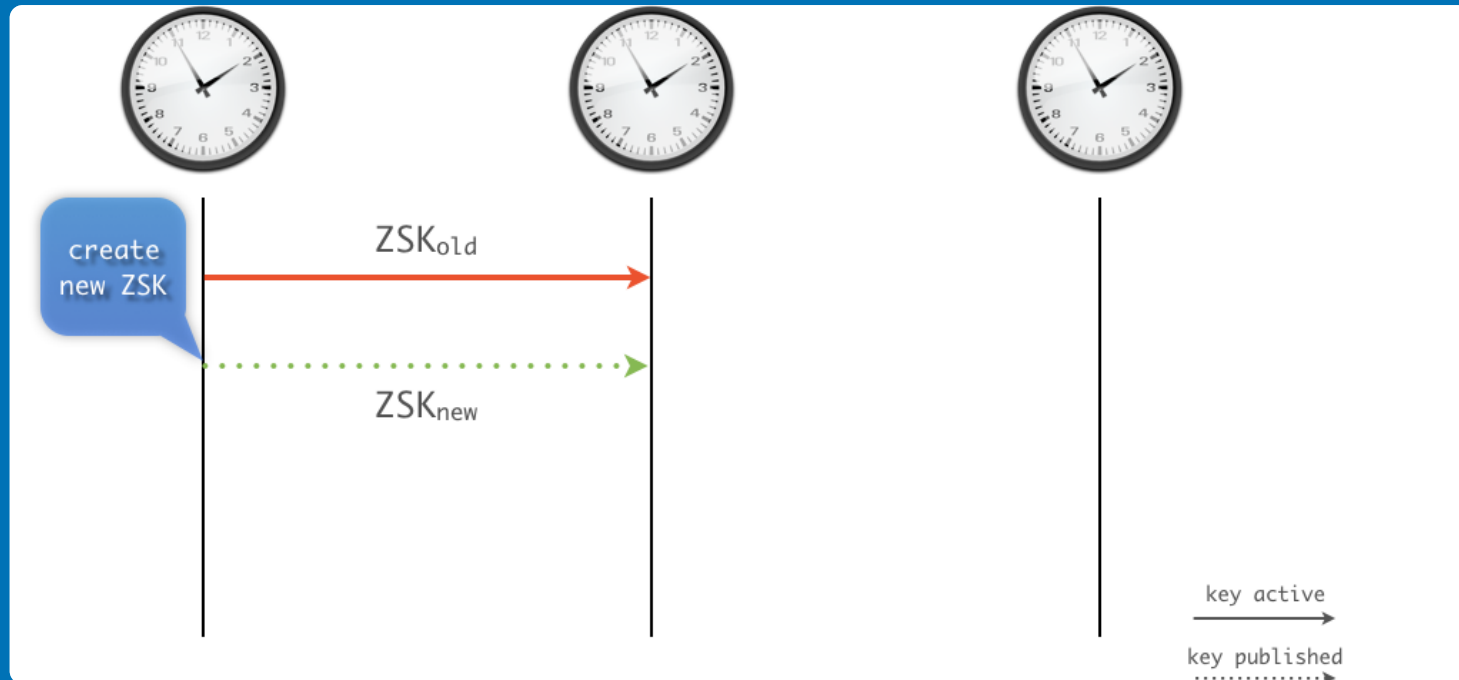




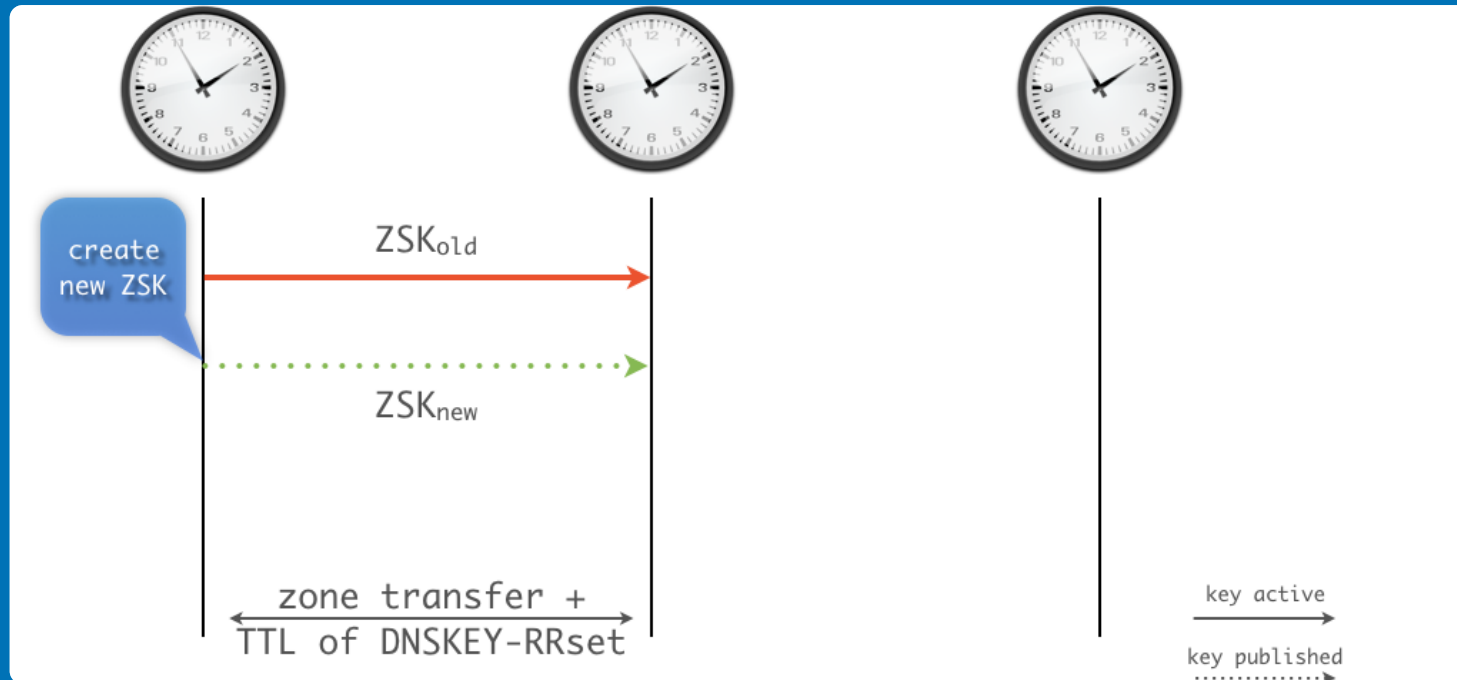
## ZSK - pre-publication in Bildern



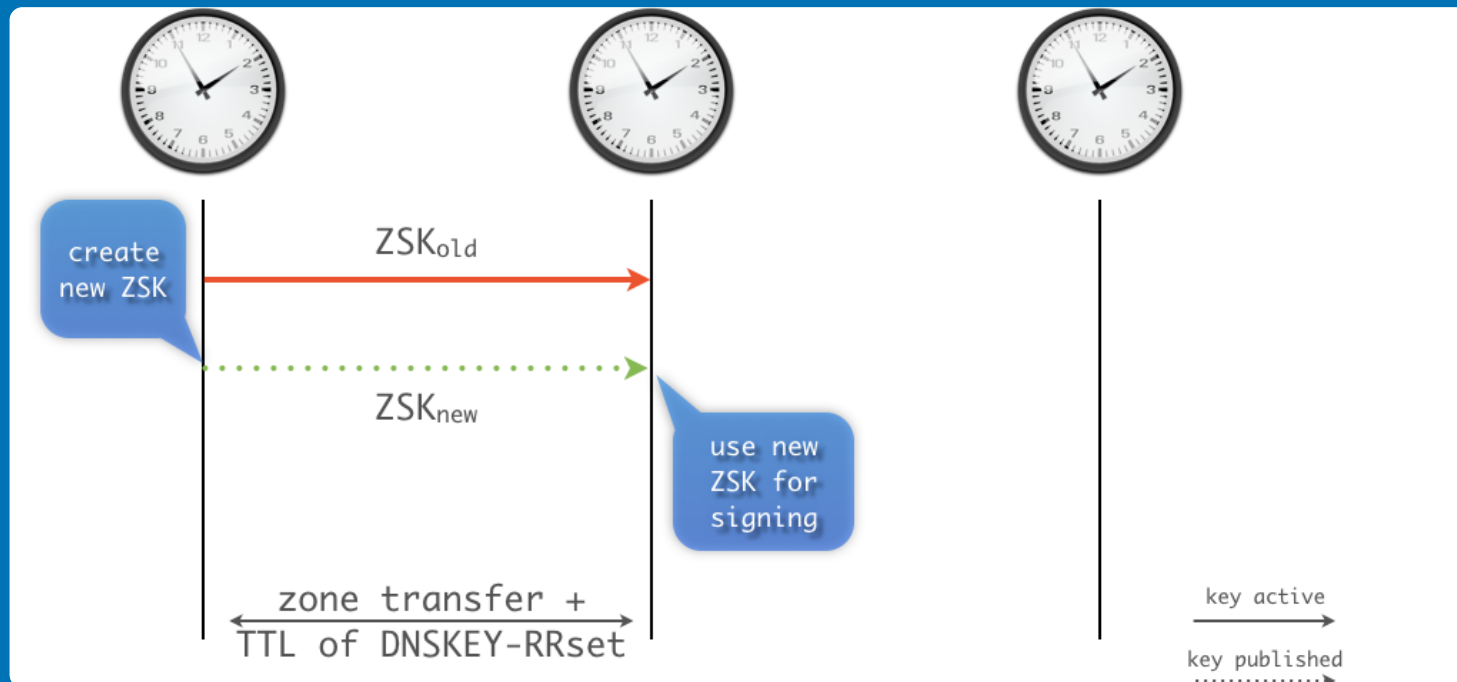
## ZSK - pre-publication in Bildern



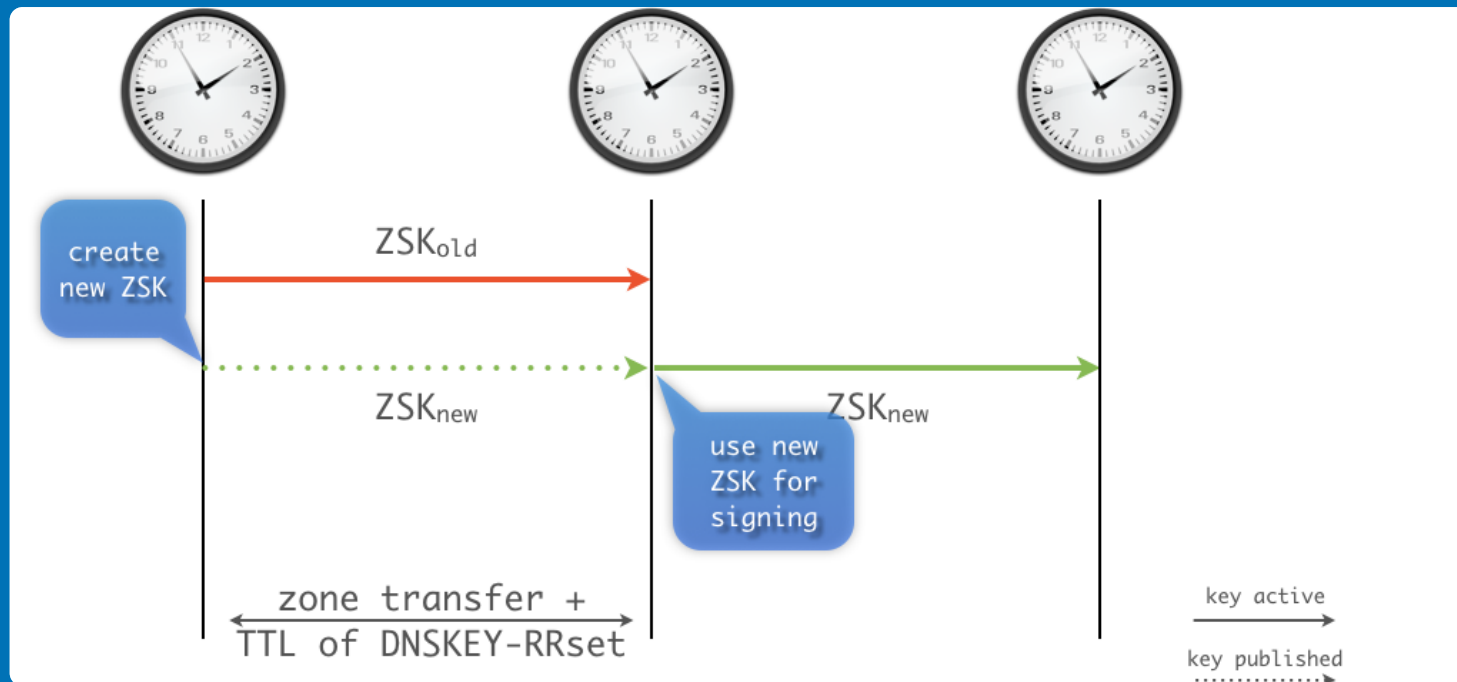
## ZSK - pre-publication in Bildern



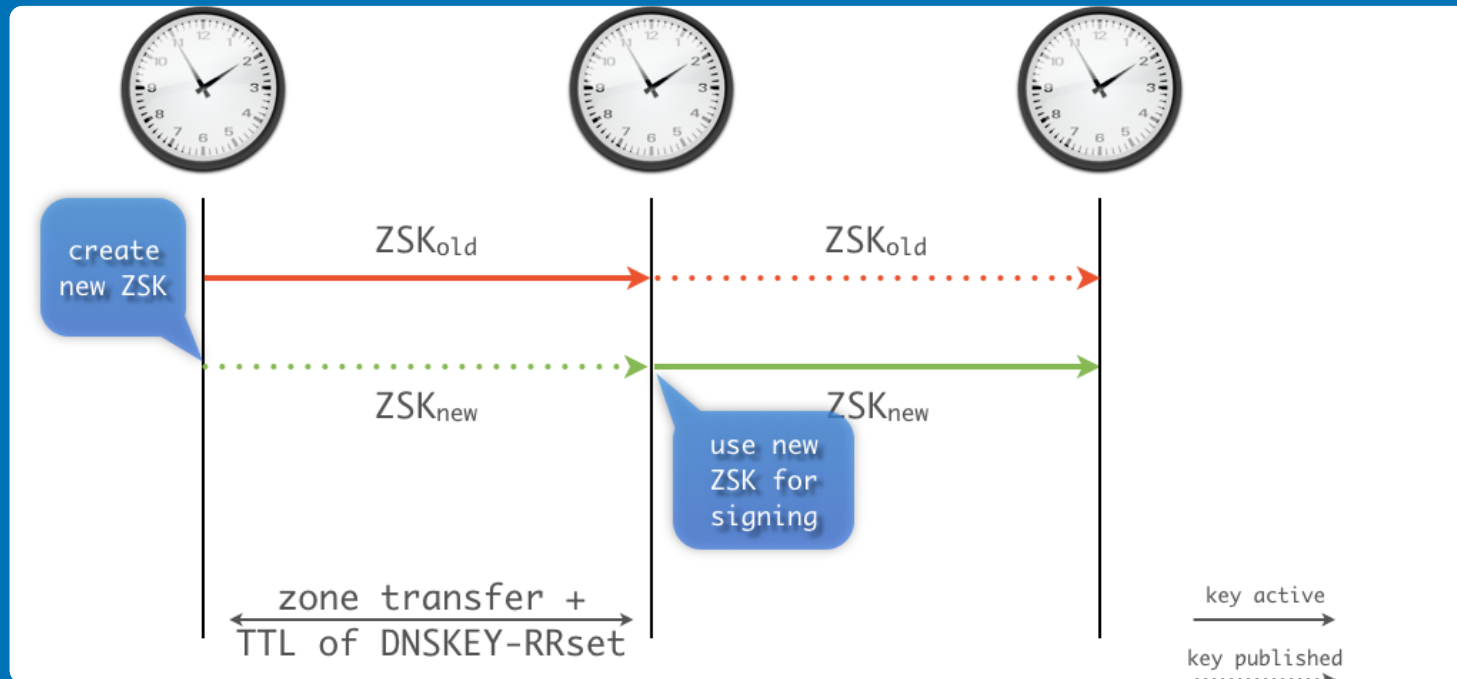
## ZSK - pre-publication in Bildern



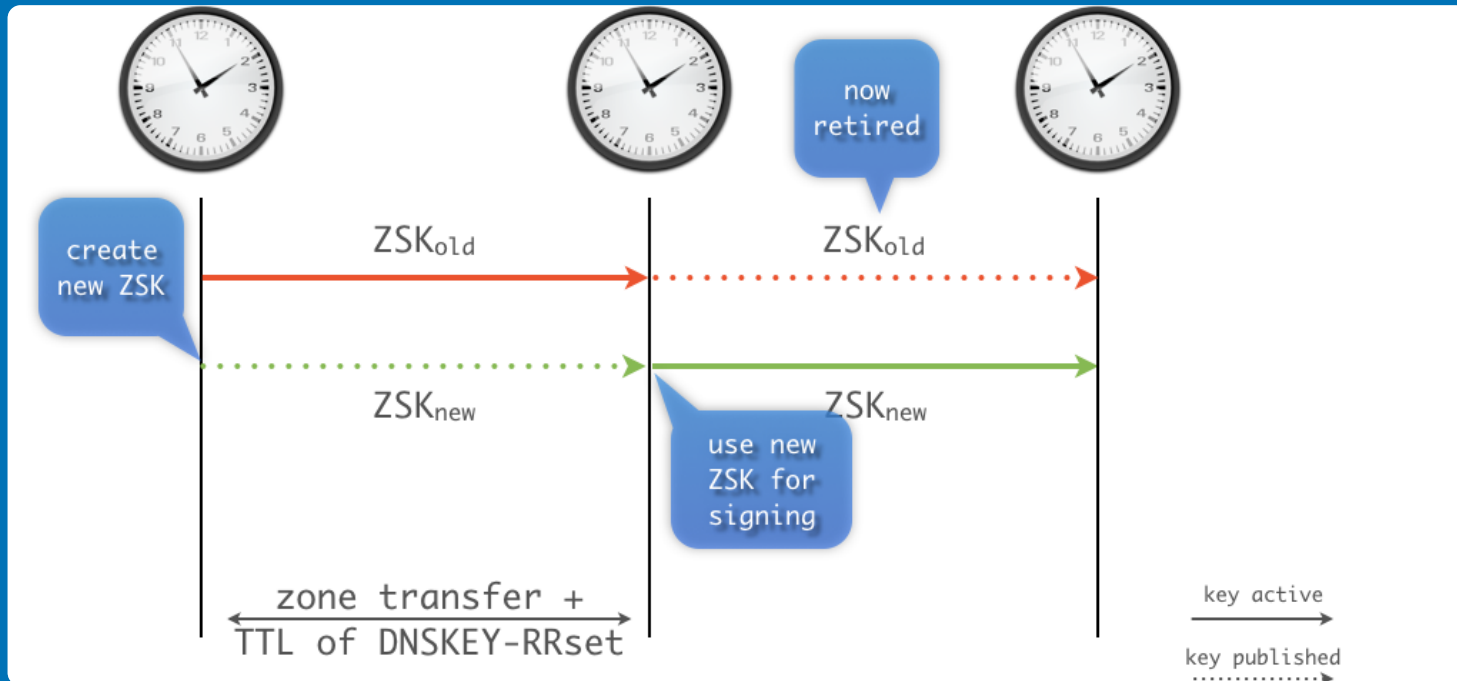
## ZSK - pre-publication in Bildern



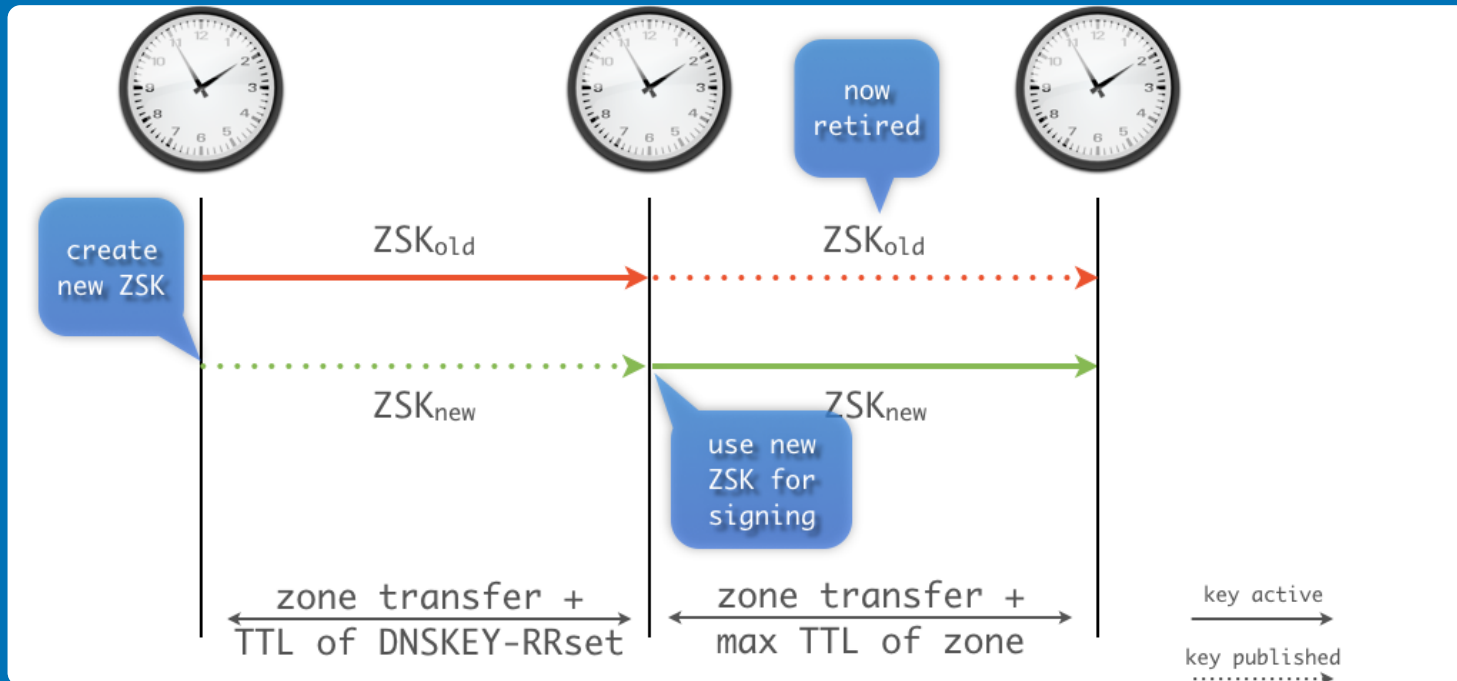
## ZSK - pre-publication in Bildern



## ZSK - pre-publication in Bildern

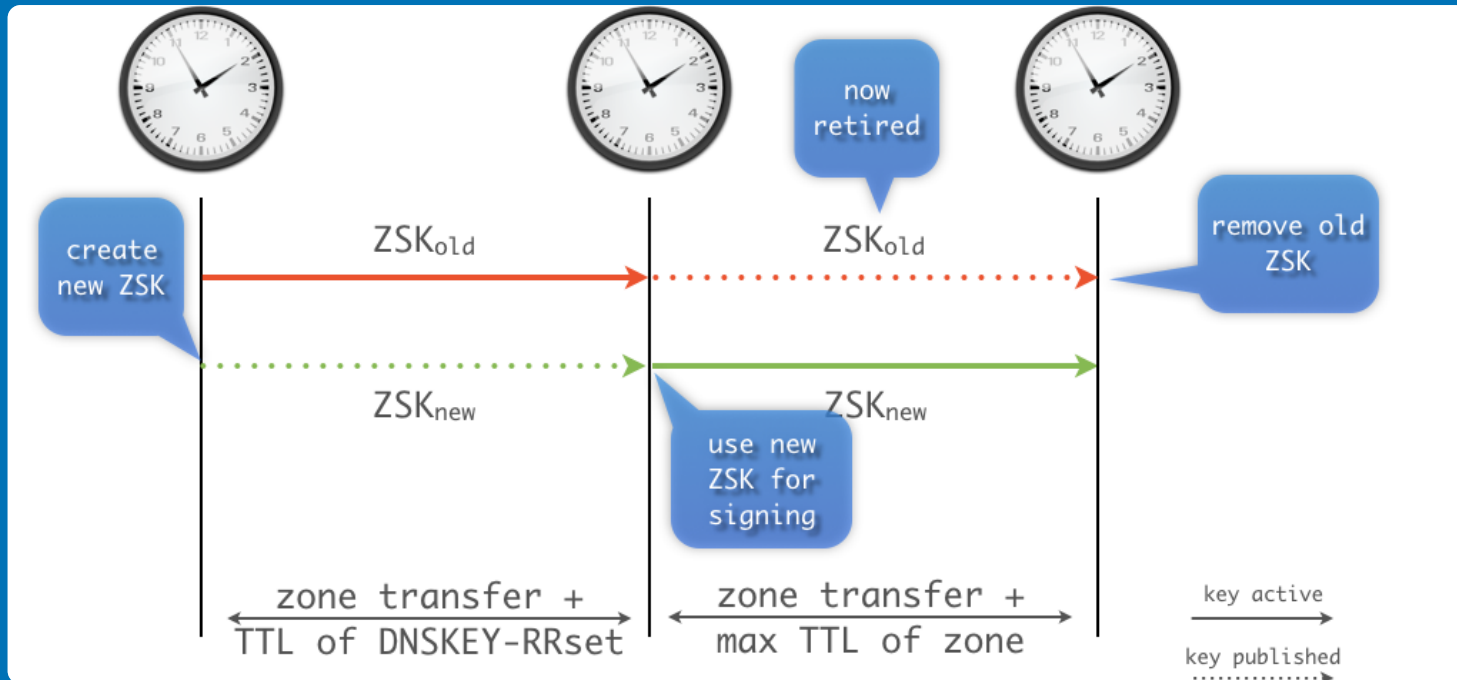


## ZSK - pre-publication in Bildern

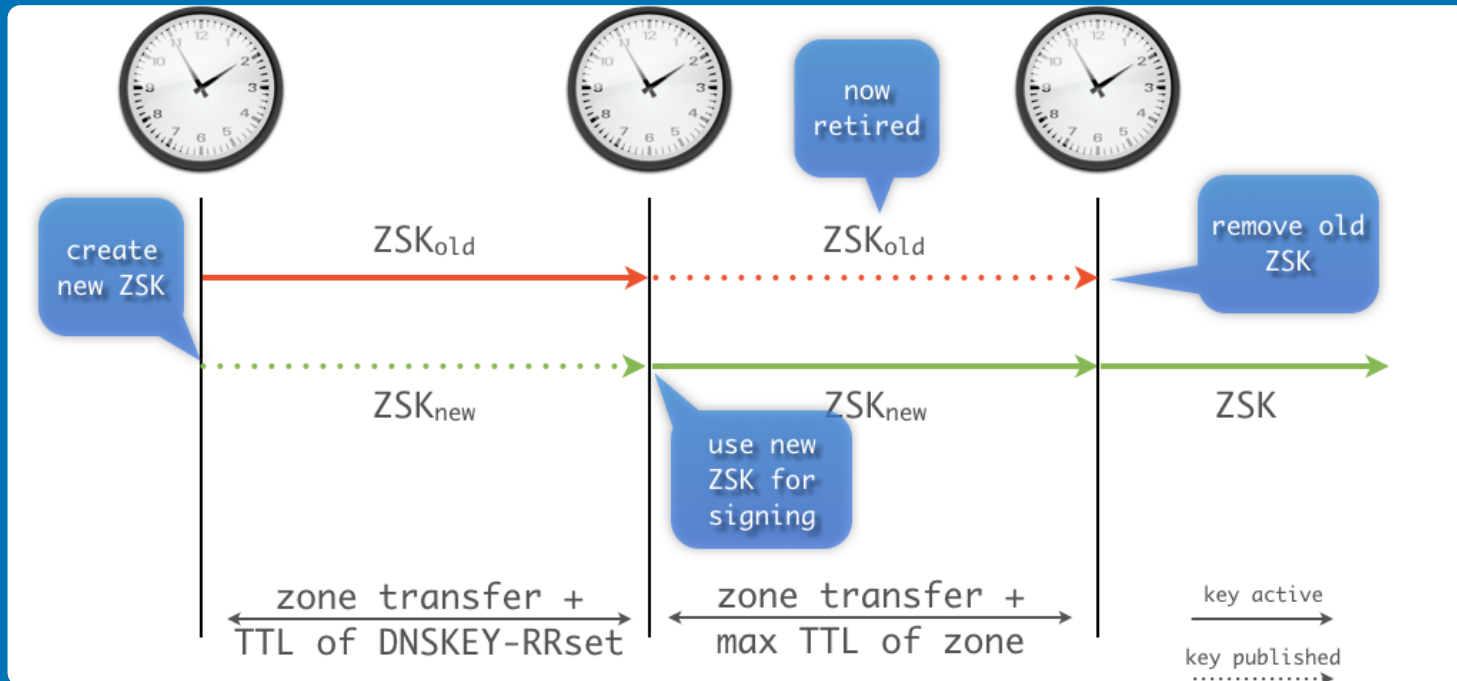




## ZSK - pre-publication in Bildern



## ZSK - pre-publication in Bildern



# KSK Rollover

---

## KSK-Rollover

- Der KSK ist von seinem DS-Eintrag in der übergeordneten Zone abhängig.
- Für den KSK-Rollover verwenden wir das "double-signing"-Rollover Schema (der DNSKEY-Recordsatz erhält zwei Signaturen, sowohl vom alten und neuen KSK)

## KSK - double-signing - Schritt 1

- Erstellen eines neuen KSK-Schlüsselpaares
- Veröffentlichung des DNSKEY-Eintrags des neuen Schlüssels in der Zone
- Signieren des DNSKEY RRsets in der Zone mit beiden KSKs (alter und neuer)

## KSK - double-signing - Schritt 2

- Warten, bis der neue Zoneninhalt mit dem neuen KSK auf allen autoritativen DNS-Servern der Zone sichtbar ist
- Warten auf die TTL des DNSKEY RRSet (+ etwas Puffer)
- Der neue KSK ist nun für alle DNS-Clients sichtbar (durch DNS-Resolver Caches)

## KSK - Doppel-Signierung - Schritt 3

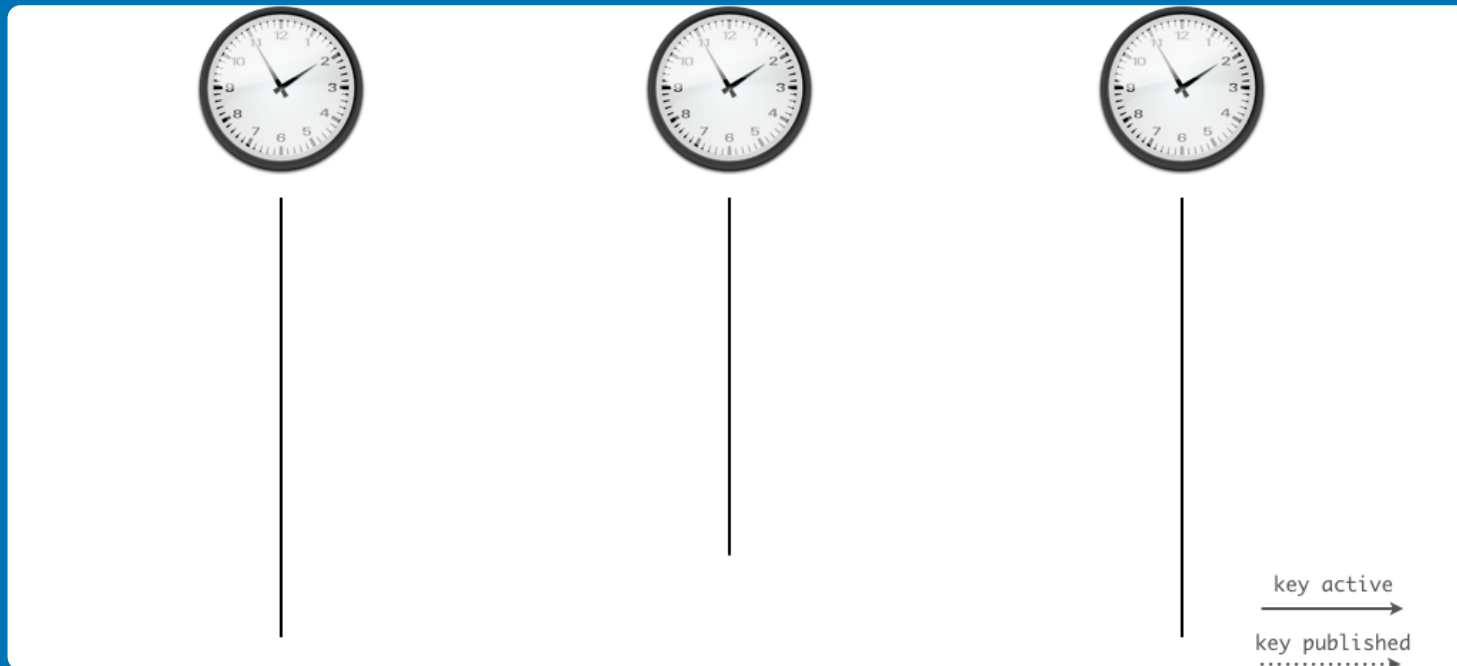
- Senden Sie den neuen DS-Eintrag an den Betreiber der übergeordneten DNS-Zone (in der Regel über eine API oder über ein Web-Interface)
- Warten auf die Aktualisierung des DS-Eintrags in der übergeordneten Zone
- Warten auf die TTL des DS-Eintrags in der übergeordneten Zone (+ ein gewisser Puffer)
- Der neue DS-Eintrag ist nun für alle DNS-Clients sichtbar

## KSK - double-signing - Schritt 4

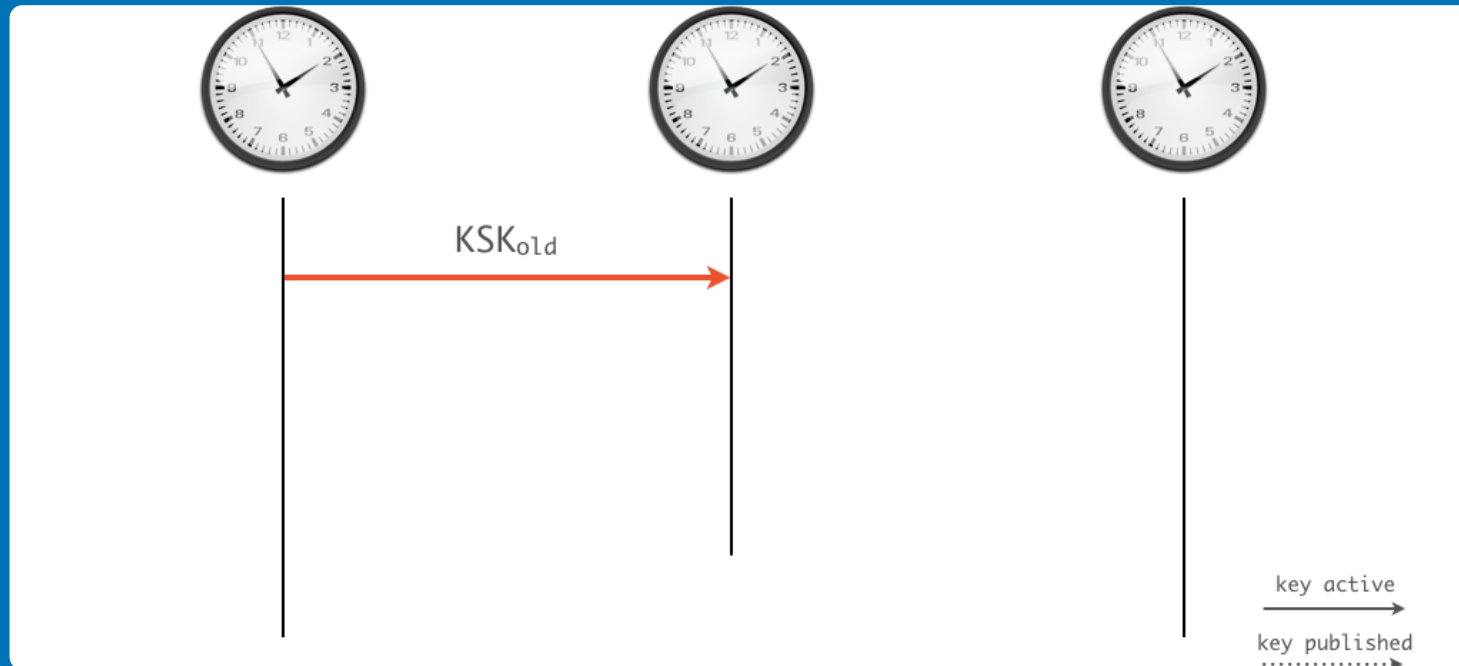
- Entfernen des alten KSK aus dem DNSKEY-Recordset der Zone
- Signiere die Zone nur mit dem neuen KSK
- Warten und den alten DS aus der übergeordneten Zone entfernen



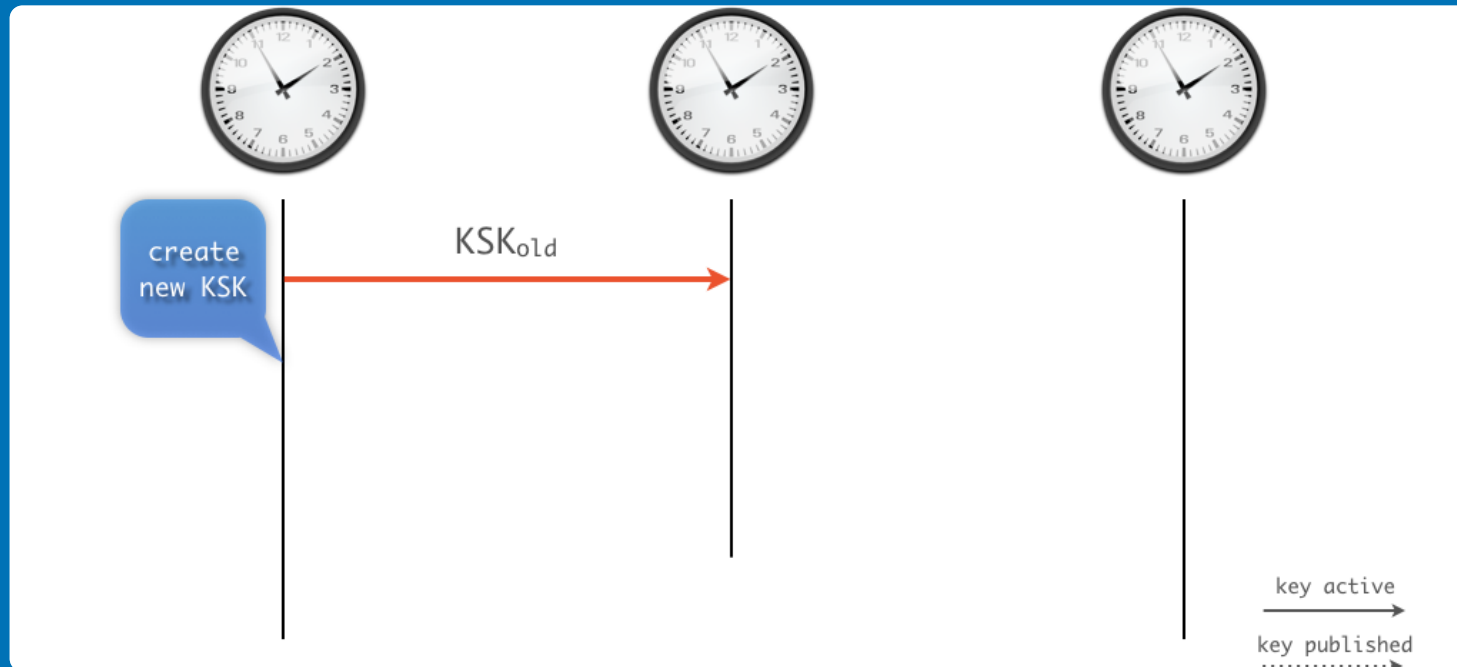
## KSK - double-signing in Bildern



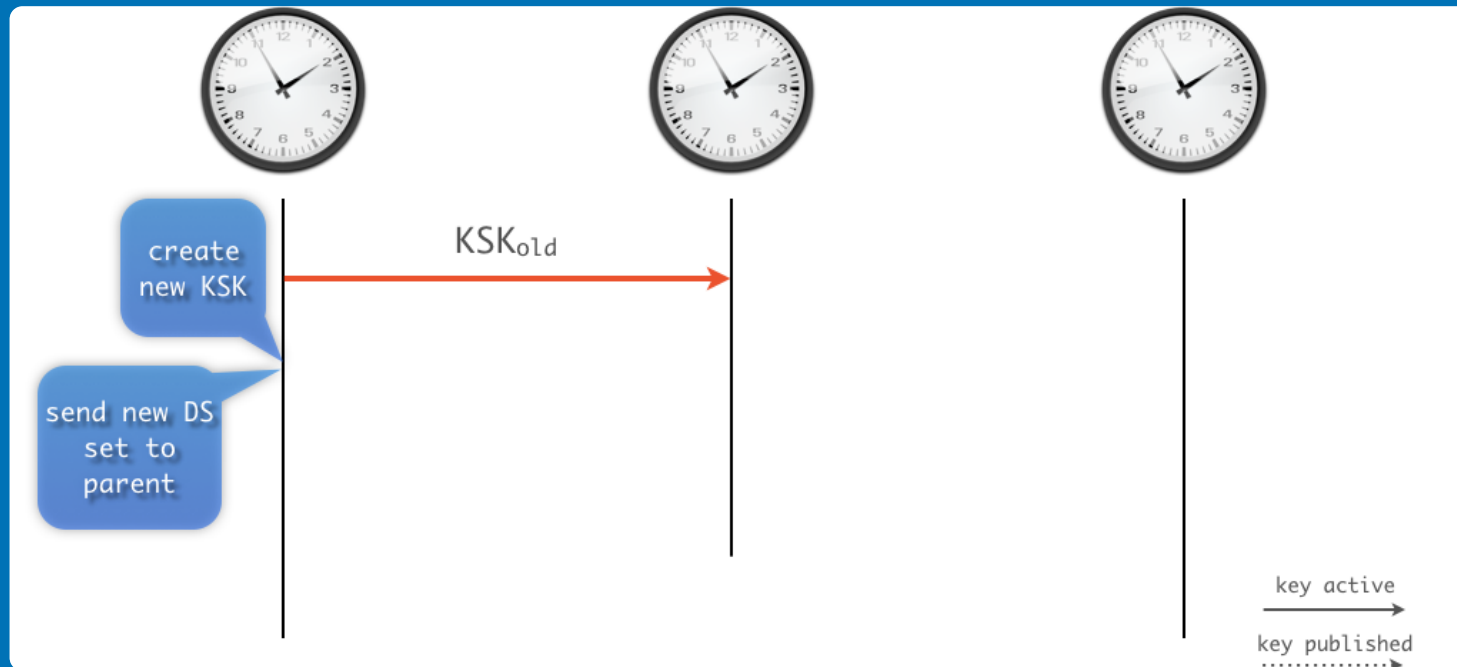
## KSK - double-signing in Bildern



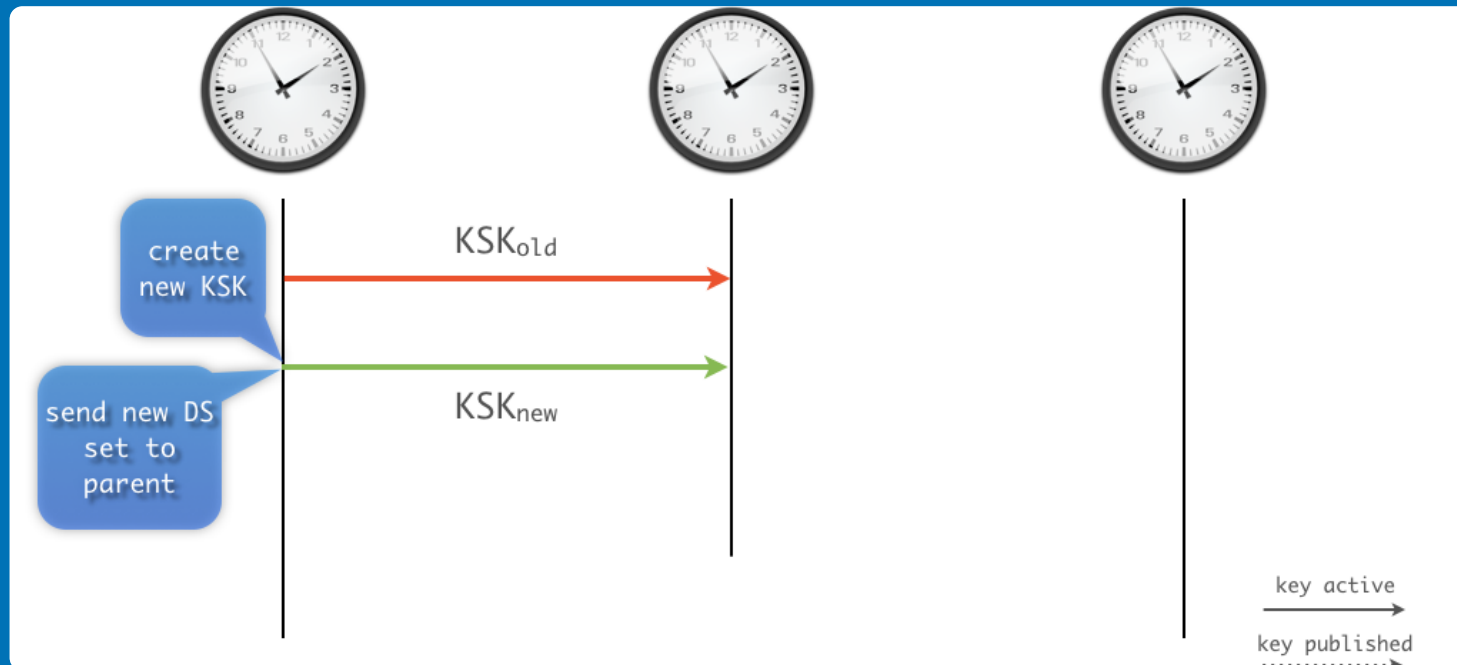
## KSK - double-signing in Bildern



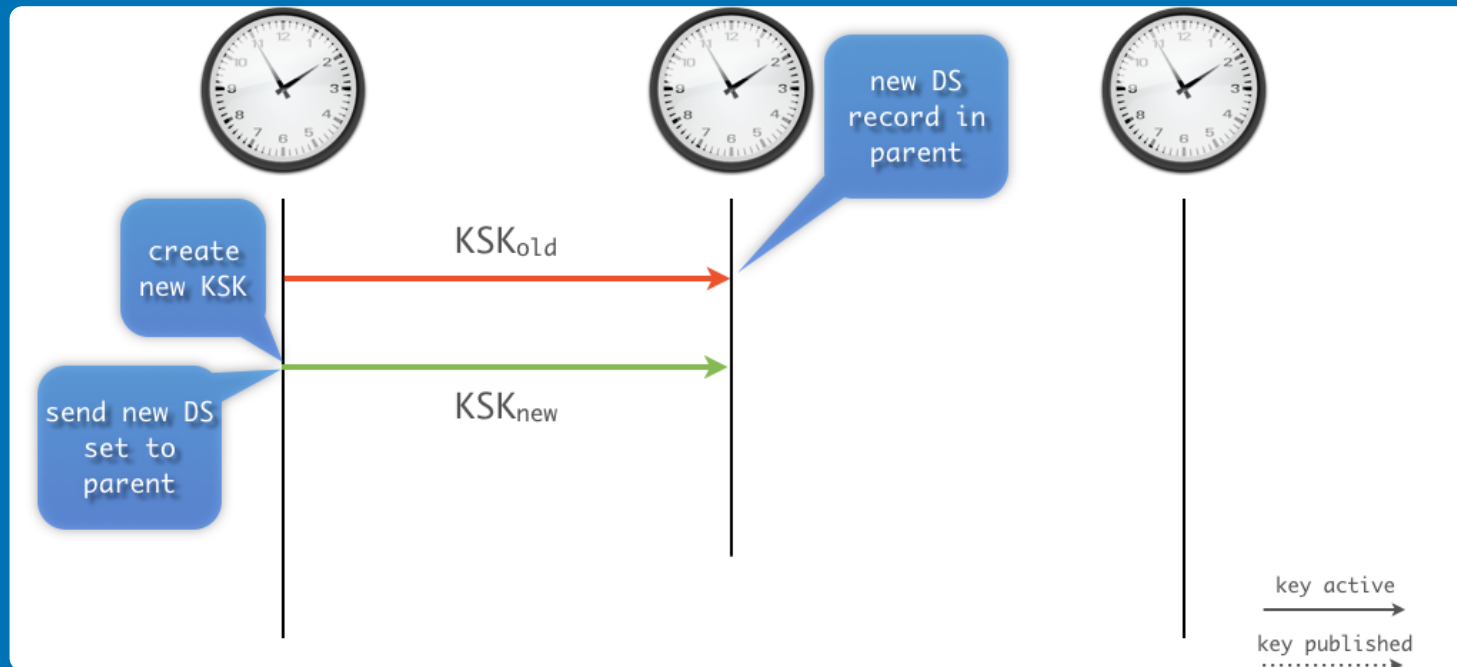
## KSK - double-signing in Bildern



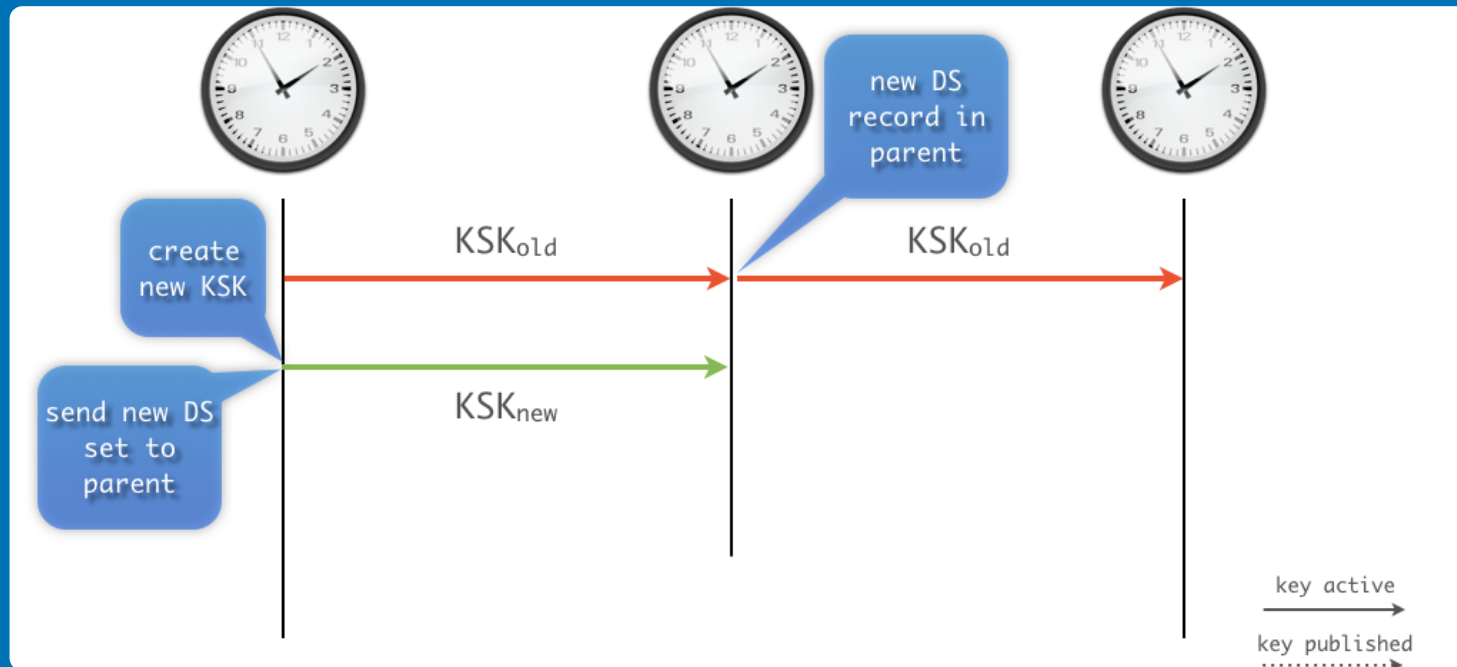
## KSK - double-signing in Bildern



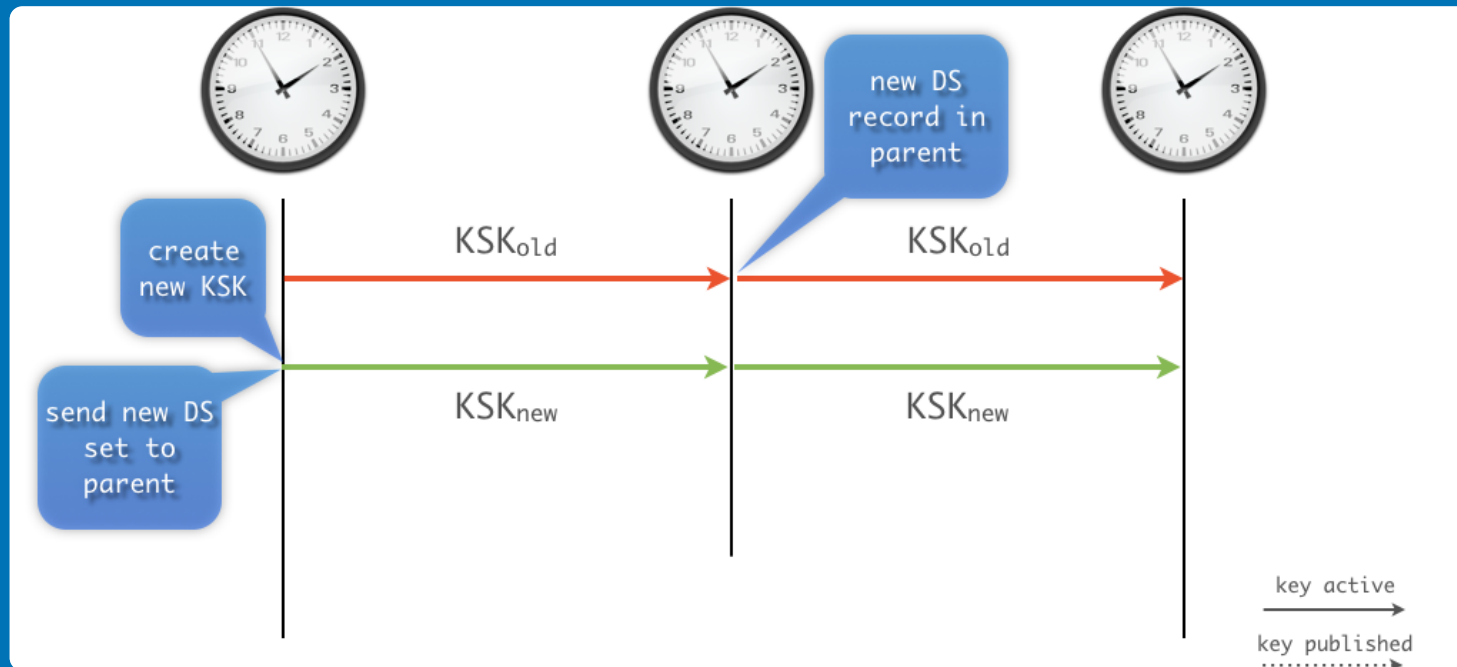
## KSK - double-signing in Bildern



## KSK - double-signing in Bildern

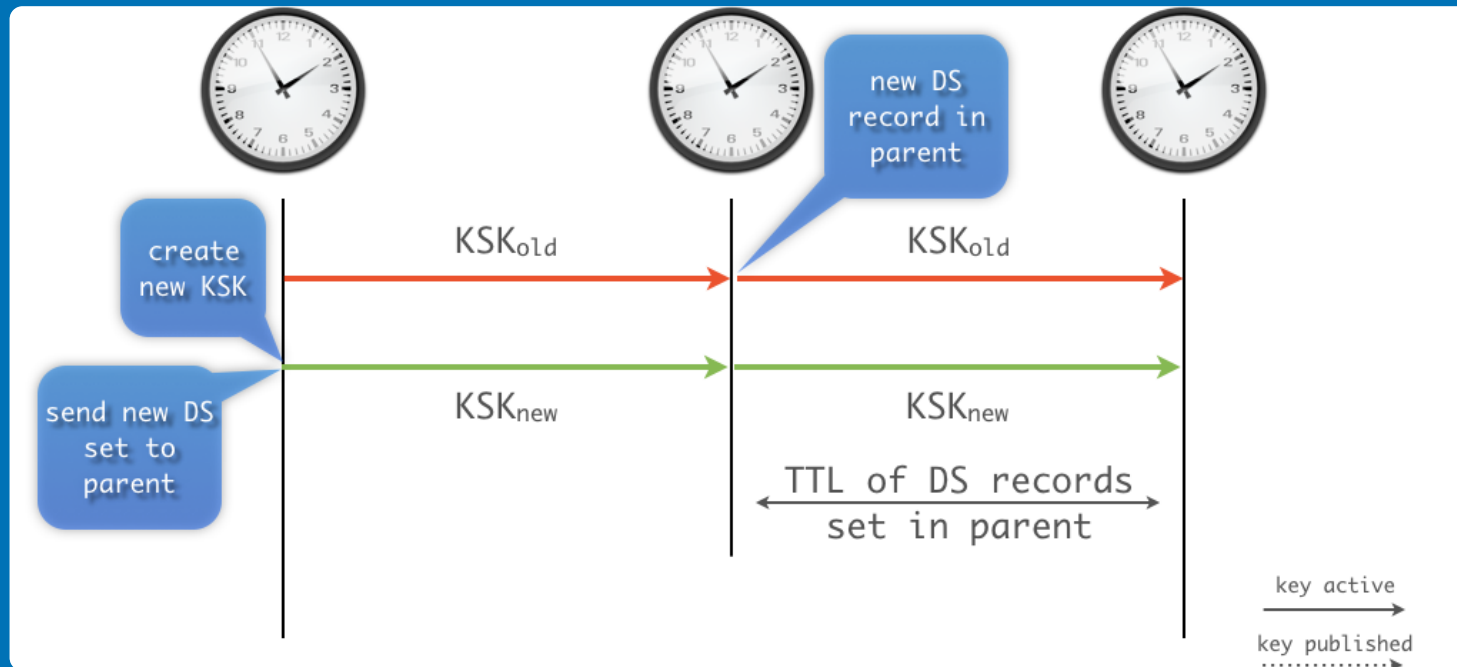


## KSK - double-signing in Bildern

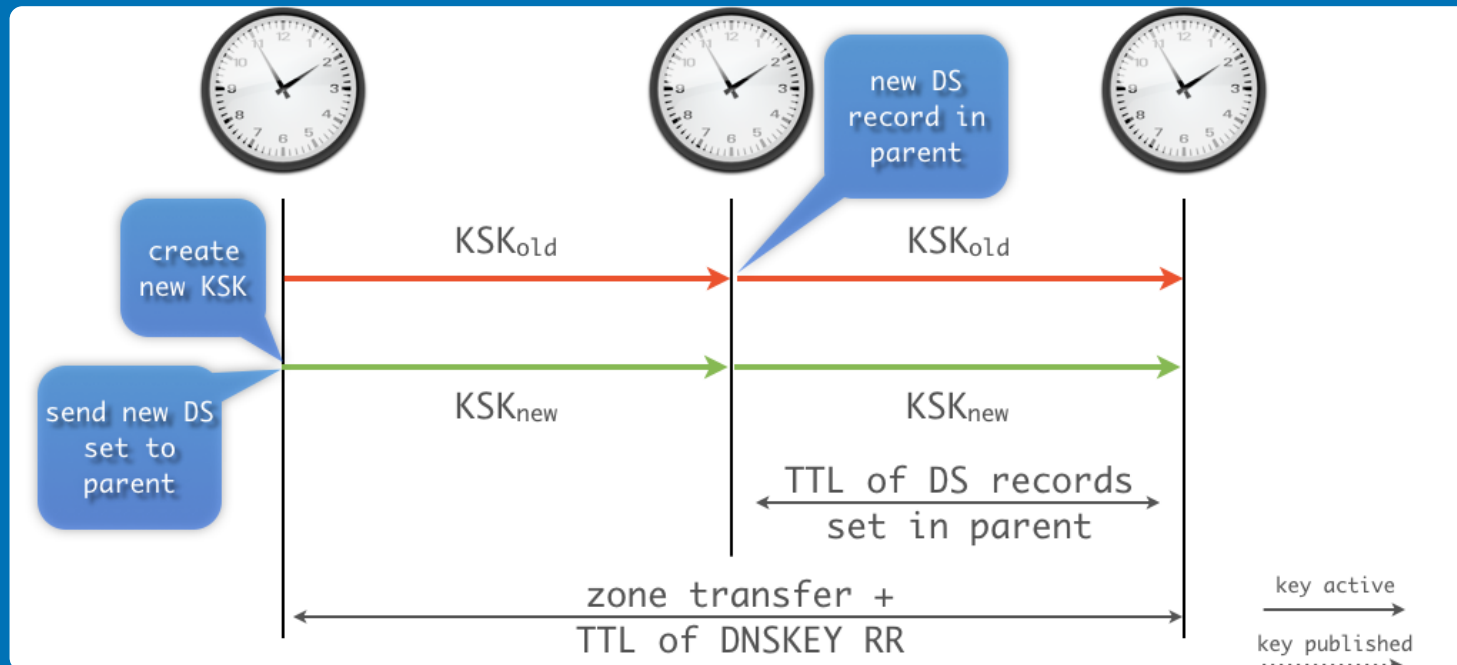




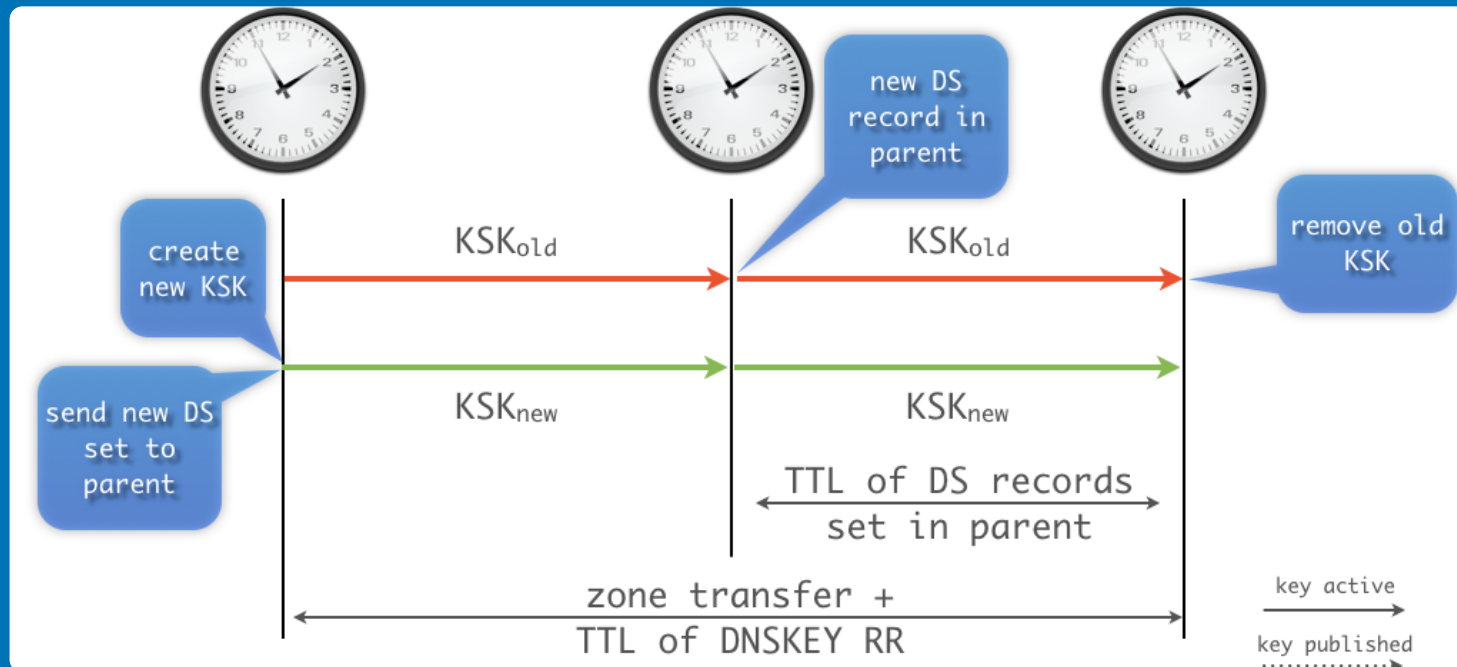
## KSK - double-signing in Bildern



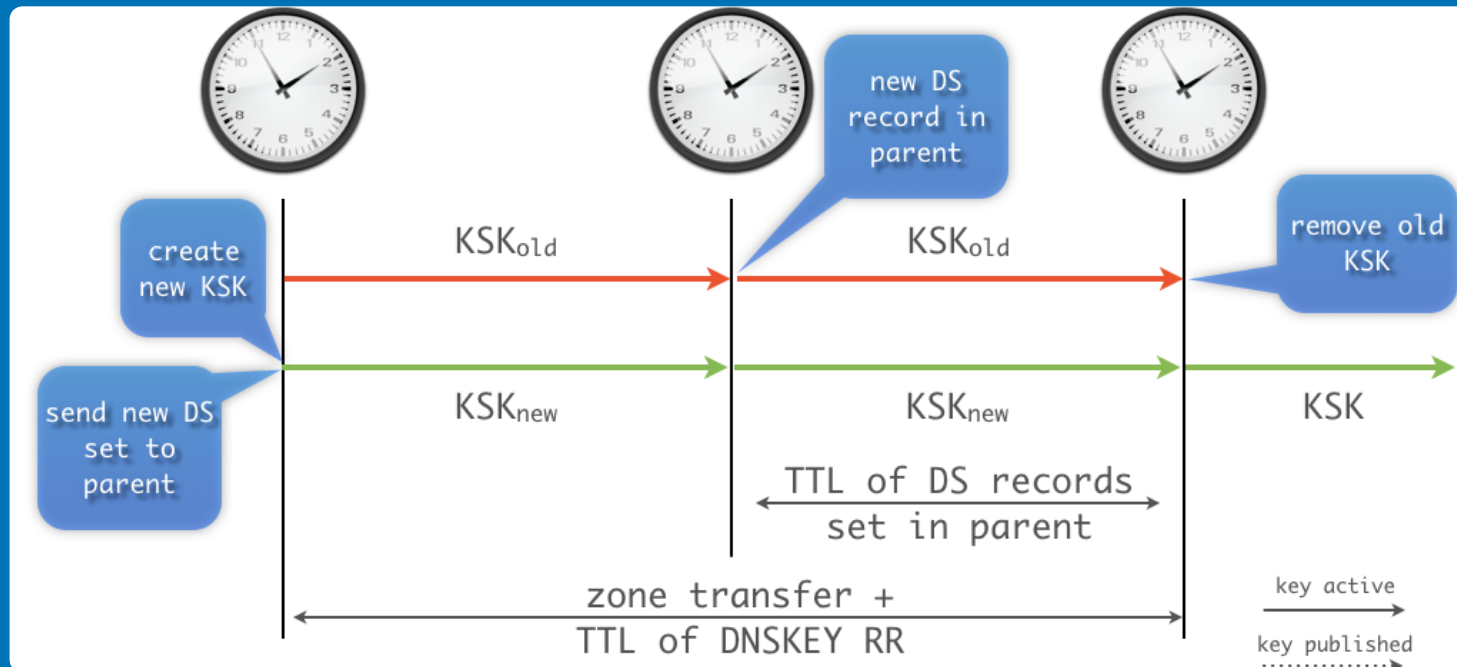
## KSK - double-signing in Bildern



## KSK - double-signing in Bildern



## KSK - double-signing in Bildern



# Weitere DNSSEC-Key-Rollover

---

# Algorithmus-Rollover

- Ein Algorithmus-Rollover wird verwendet, wenn der DNSSEC-Schlüsselalgorithmus der Zone geändert werden muss
  - z.B. beim Wechsel von RSASHA256 zu ECDSASHA256
- Bei einem Algorithmus-Rollover werden der KSK und der ZSK gleichzeitig unter Verwendung eines *double-signing* Rollover-Schemas gewechselt
  - Während eines solchen Rollover sollte die Zone auf Ausfälle und übergroße DNS-Antworten überwacht werden

# Notfall-Rollover-Schlüssel

- In einem Notfall ist Zeit ein entscheidender Faktor
- Um bei einem KSK-Rollover Zeit zu sparen, kann der erste Schritt (Veröffentlichung) im Vorraus erfolgen
  - dieser veröffentlichte zusätzliche Schlüssel wird als *standby*-Schlüssel bezeichnet
  - er spart Zeit im Notfall, macht aber das DNSKEY RRSet größer
- Der *Standby*-Schlüssel wird während des *normalen* DNSSEC-Signierungsvorgangs nicht verwendet.
- Wenn der KSK in einem Notfall geändert werden muss, kann die Zone fast sofort auf den *standby*-Schlüssel umschalten
- Der *Standby*-Schlüssel sollte ersetzt (gerollt) werden, wenn der Produktions-KSK gerollt wird

# Automatisierte Wartung des DNSSEC- Delegationsvertrauens (RFC 7344/8078)

---



## CDS/CDNSKEY

- RFC 7344/8087 definiert einen automatischen Weg zur Aktualisierung der Vertrauenskette gegenüber der übergeordneten Zone bei einem KSK-Schlüssel-Rollover
- Der Betreiber der Zone erstellt einen neuen KSK für die Zone und veröffentlicht den neuen DS-Record oder/und DNSKEY-Record für die übergeordnete Zone in einem CDS- und/oder CDNSKEY-Eintrag in der Zone

# CDS/CDNSKEY

- Der DNS-Server für die übergeordnete Zone fragt regelmäßig die untergeordneten Zonen nach neuen CDS- oder CDNSKEY-Einträgen ab
- Sobald ein neuer CDS-Datensatz gefunden wird, wird er mit den KSK- und DS-Datensätze überprüft, und wenn er gültig ist, wird er in die übergeordnete Zone importiert (und ersetzt damit den DS-Eintrag)
  - Wenn ein neuer CDNSKEY-Eintrag in der Kind-Zone gefunden wird, lädt der autoritative DNS-Server der übergeordneten Zone den Eintrag, berechnet den Hash um einen DS-Record zu erhalten, und ersetzt dann den DS-Record durch den neu berechneten DS-Record

## CDS/CDNSKEY

- BIND 9 unterstützt CDS und CDNSKEY seit Version 9.11
- Das Dienstprogramm `dnssec-cds` kann DS-Einträge basierend auf CDS/CDNSKEY-Einträgen für eine Child-Zone ändern
- CDS und CDNSKEY werden bereits von einigen TLDs unterstützt (tschechische Republik ".cz". Schweiz ".ch", Lichtenstein ".li" ...)

# CDS/CDNSKEY

Parent DNS



tld. IN SOA ...  
tld. IN NS ...  
tld. IN DNSKEY ...

Child DNS



child.tld. IN SOA ...  
child.tld. IN NS ...  
child.tld. IN DNSKEY ...

# CDS/CDNSKEY

Updating DNSSEC Trust chain today

Parent DNS



tld. IN SOA ...  
tld. IN NS ...  
tld. IN DNSKEY ...

Child DNS



child.tld. IN SOA ...  
child.tld. IN NS ...  
child.tld. IN DNSKEY ...

# CDS/CDNSKEY

Updating DNSSEC Trust chain today

Parent DNS



tld. IN SOA ...  
tld. IN NS ...  
tld. IN DNSKEY ...

Child DNS



child.tld. IN SOA ...  
child.tld. IN NS ...  
child.tld. IN DNSKEY ... →

# CDS/CDNSKEY

Updating DNSSEC Trust chain today

Parent DNS



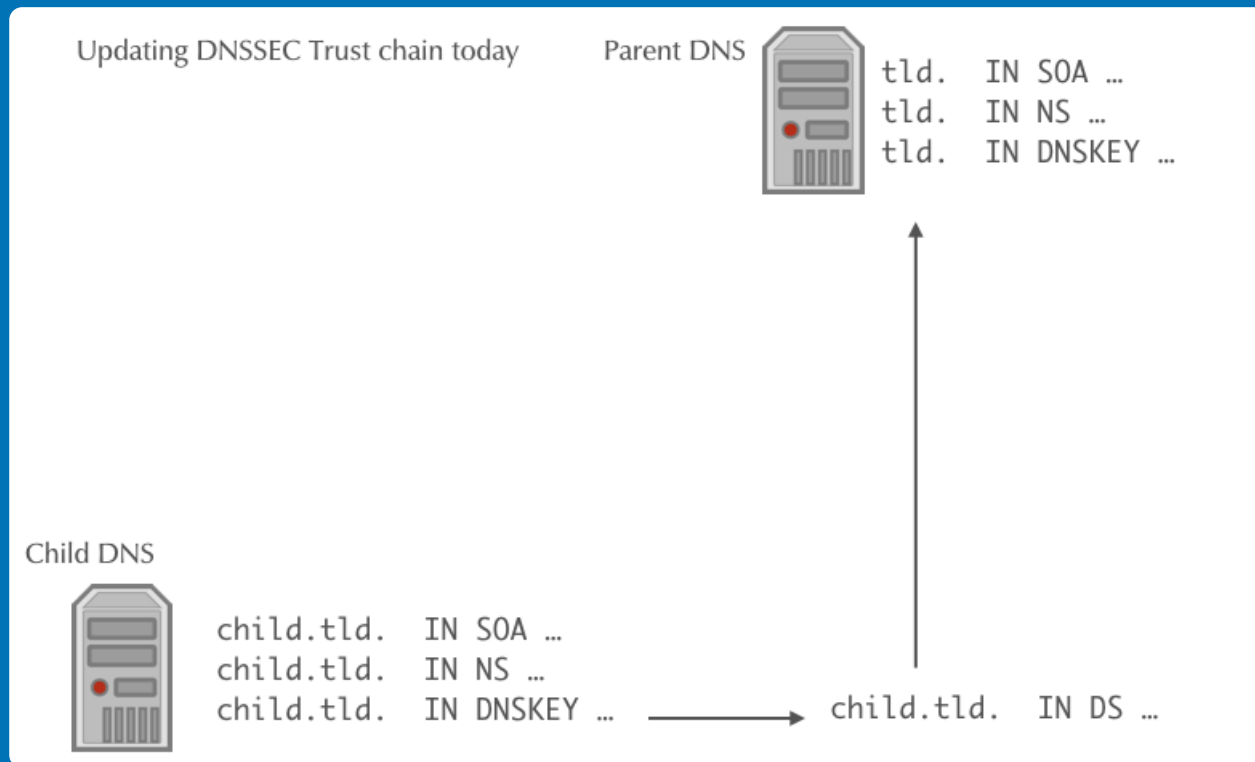
tld. IN SOA ...  
tld. IN NS ...  
tld. IN DNSKEY ...

Child DNS



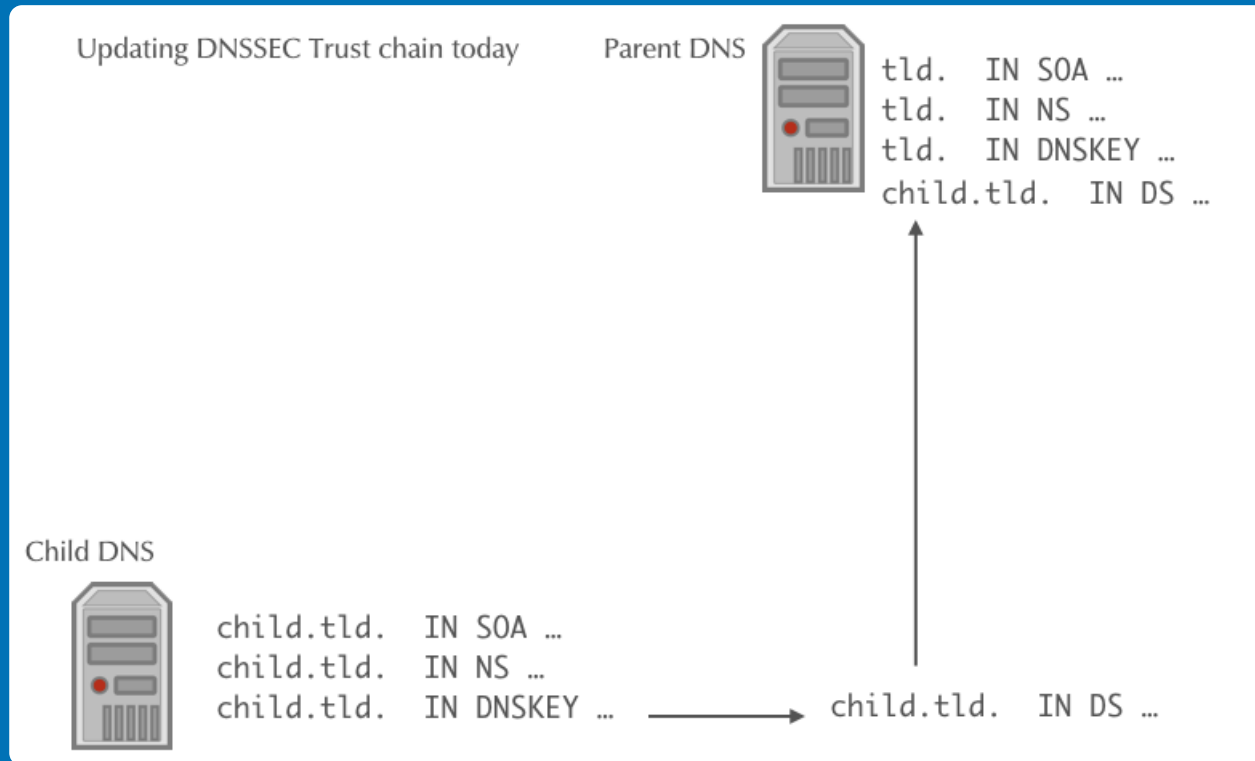
child.tld. IN SOA ...  
child.tld. IN NS ...  
child.tld. IN DNSKEY ... → child.tld. IN DS ...

# CDS/CDNSKEY





# CDS/CDNSKEY



# CDS/CDNSKEY

Updating DNSSEC Trust chain  
with CDS / CDNSKEY

Parent DNS



tld. IN SOA ...  
tld. IN NS ...  
tld. IN DNSKEY ...

Child DNS



child.tld. IN SOA ...  
child.tld. IN NS ...  
child.tld. IN DNSKEY ...

# CDS/CDNSKEY

Updating DNSSEC Trust chain  
with CDS / CDNSKEY

Parent DNS



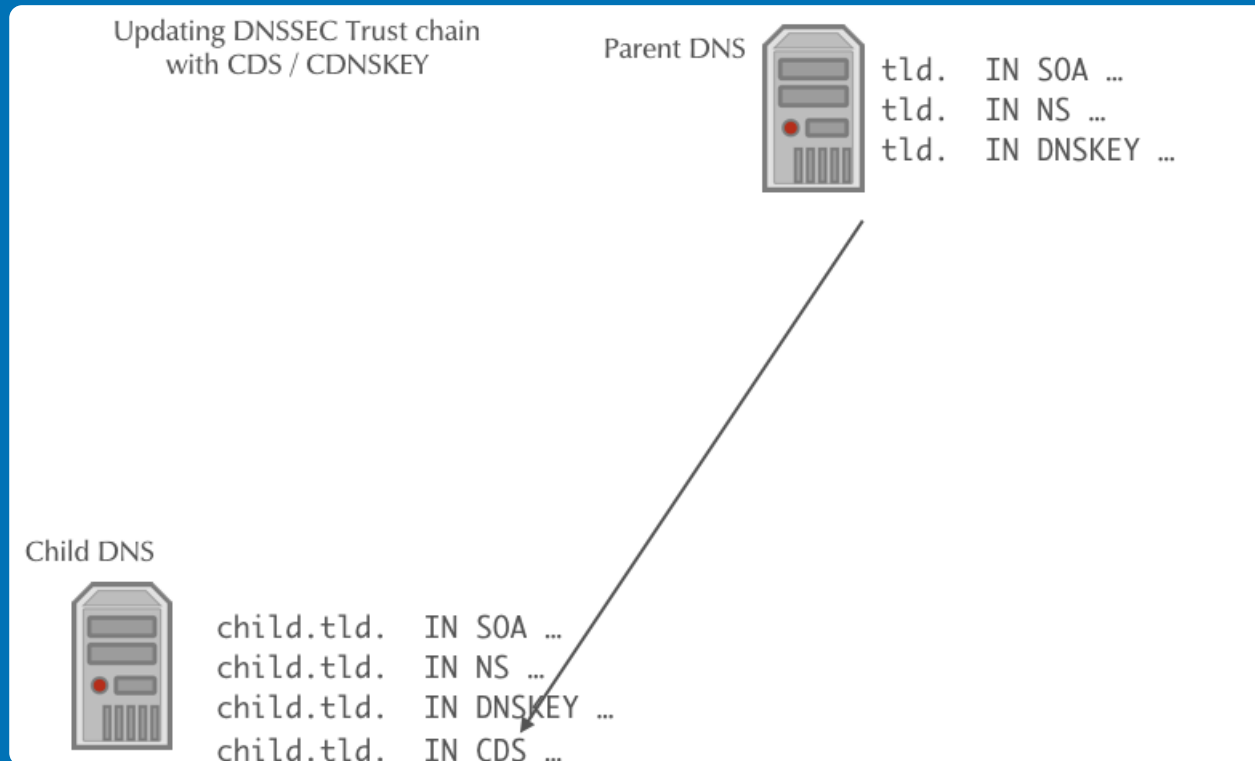
tld. IN SOA ...  
tld. IN NS ...  
tld. IN DNSKEY ...

Child DNS

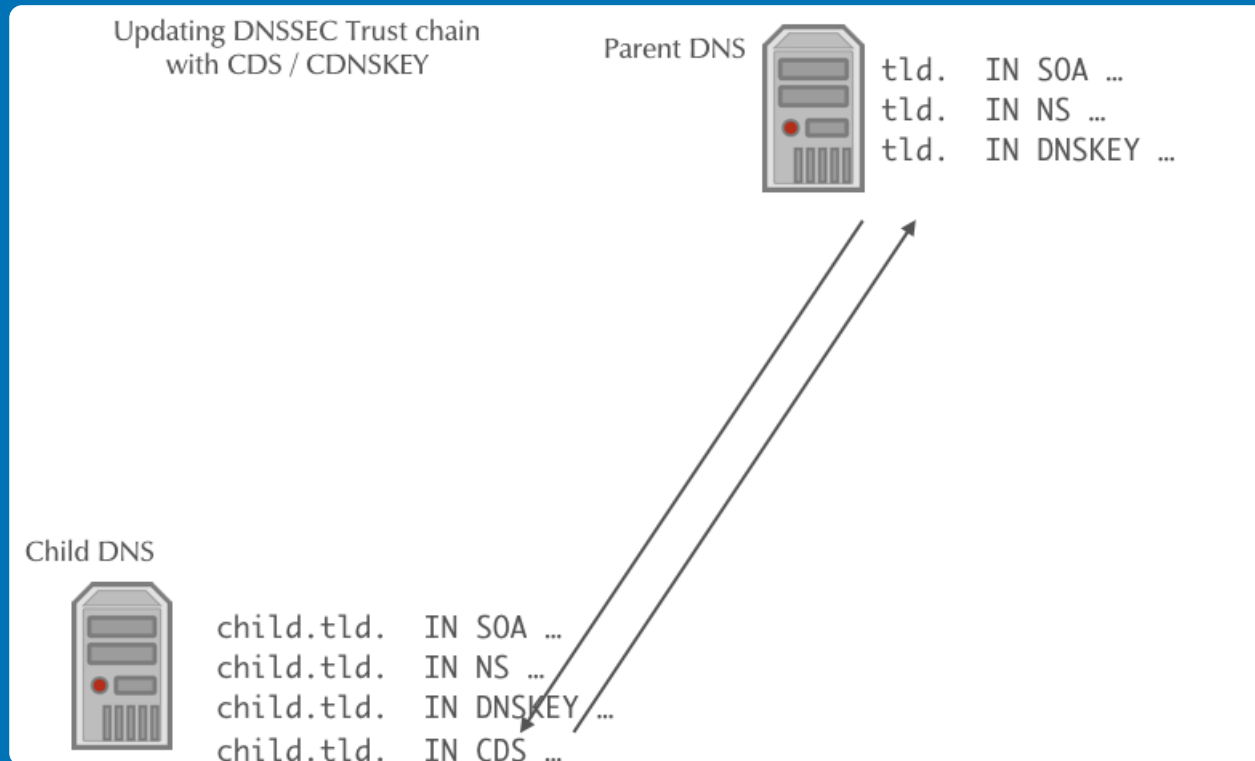


child.tld. IN SOA ...  
child.tld. IN NS ...  
child.tld. IN DNSKEY ...  
child.tld. IN CDS ...

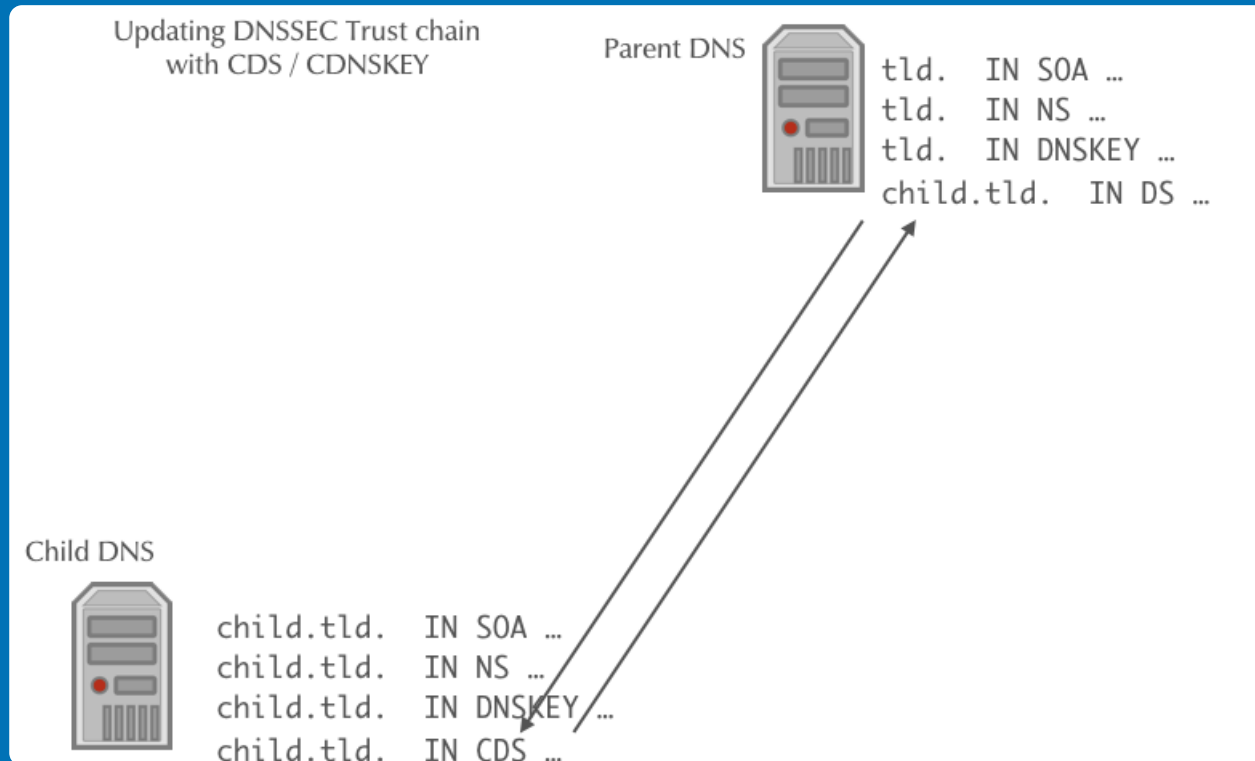
# CDS/CDNSKEY



# CDS/CDNSKEY



# CDS/CDNSKEY



Ende des Kapitels "DNSSEC Key-Rollover" - Fragen?

---