

## DNSSEC Key Management

In collaboration with CDAC



Champika Wijayatunga – Regional Technical Engagement Manager - APAC  
26 November 2020

---

---

---

---

---


---

---

---

### Key Lifecycle

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
- Stop using key for signing
- Post-publish key in DNS
- Remove key from DNSKEY set
- Delete the key



12

---

---

---

---

---

---


---

---

### Key Lifecycle

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
- Stop using key for signing
- Post-publish key in DNS
- Remove key from DNSKEY set
- Delete the key

**HSM**  
**(Hardware Security Module) or not?**



13

---

---

---

---

---

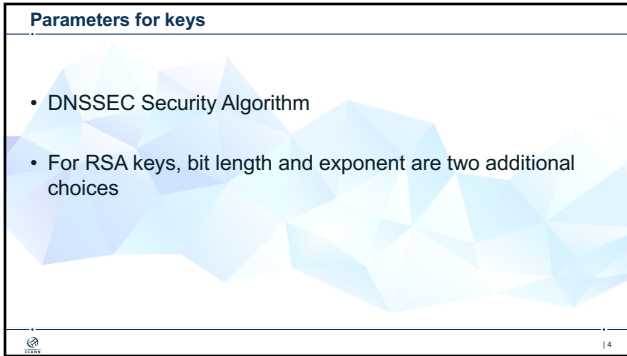
---

---

---

**Parameters for keys**

- DNSSEC Security Algorithm
- For RSA keys, bit length and exponent are two additional choices



14

---

---

---

---

---

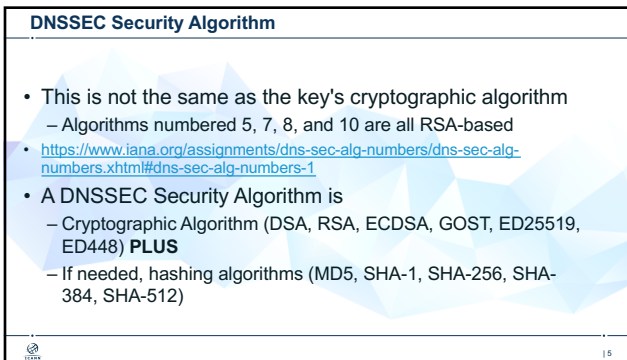
---

---

---

**DNSSEC Security Algorithm**

- This is not the same as the key's cryptographic algorithm
  - Algorithms numbered 5, 7, 8, and 10 are all RSA-based
- <https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml#dns-sec-alg-numbers-1>
- A DNSSEC Security Algorithm is
  - Cryptographic Algorithm (DSA, RSA, ECDSA, GOST, ED25519, ED448) **PLUS**
  - If needed, hashing algorithms (MD5, SHA-1, SHA-256, SHA-384, SHA-512)



15

---

---

---

---

---

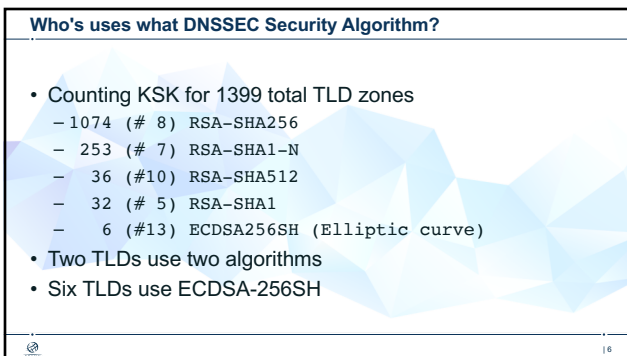
---

---

---

**Who's uses what DNSSEC Security Algorithm?**

- Counting KSK for 1399 total TLD zones
  - 1074 (# 8) RSA-SHA256
  - 253 (# 7) RSA-SHA1-N
  - 36 (#10) RSA-SHA512
  - 32 (# 5) RSA-SHA1
  - 6 (#13) ECDSA256SH (Elliptic curve)
- Two TLDs use two algorithms
- Six TLDs use ECDSA-256SH



16

---

---

---

---

---

---

---

---

**Which to use?**

- The "trendy" thought is to use elliptic curve algorithms
- The downside of elliptic curve algorithms
  - Maybe too new, client software support may not be fully deployed
- The upside of elliptic curve algorithms
  - Smaller messages and conceptually harder to "break"
- Trendy is not always "bad"

---

---

---

---

---

---

---

---

**Bit lengths across all RSA-based keys**

<ul style="list-style-type: none"> <li>• KSK lengths:           <ul style="list-style-type: none"> <li>– 17 4096b</li> <li>– <b>1361 2048b</b></li> <li>– 86 2047b (tool bug)</li> <li>– 5 1280b</li> <li>– 1 1024b</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• ZSK lengths:           <ul style="list-style-type: none"> <li>– 1 4096b</li> <li>– <b>160 2048b</b></li> <li>– <b>536 1280b</b></li> <li>– <b>874 1024b</b></li> <li>– 9 1023b (tool bug)</li> <li>– 5 1152b</li> </ul> </li> </ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Why the difference between KSK and ZSK sizes?
  - Assumption that KSK has to be stronger than ZSK
- "Tool bugs" – a commercial product sometimes "shorts" a key by one bit, bit all works

---

---

---

---

---

---

---

---

**Bit length trade off**

- Longer:
  - Cryptographically stronger
  - But more bytes in responses
- We've never had a case were a shorter key was broken

---

---

---

---

---

---

---

---

### Key Lifecycle

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
- Stop using key for signing
- Post-publish key in DNS
- Remove key from DNSKEY set
- Delete the key

This is where the TTL comes in to play

---

---

---

---

---

---

---

---

### DNSKEY life cycle and RRSIG validity durations

The diagram illustrates the timing of key operations and RRSIG validity. Key phases include 'Key Pre-published', 'Key Actively Signs', and 'Key In Retirement'. RRSIG durations are shown as overlapping blocks. At the bottom, three arrows indicate the durations: 'DNSKEY TTL' (blue), 'KEY lifetime' (blue), and 'RRSIG Validity' (brown).

---

---

---

---

---

---

---

---

### Why is pre-publishing needed?

- Cache gets a copy of the DNSKEY set at time t0
- Caches might get a copy of the SOA RR at time t1
- If, at t1, the SOA is signed with a new key, the DNSKEY set must already have it, or validation fails.
  - "Validation fails" is not good.
- A cache won't refresh the key set until t0+the TTL of DNSKEY, so we pre-publish by at least the TTL value

---

---

---

---

---

---

---

---

**Once a new key is ready**

- For the first key, this doesn't matter
- For all new keys after the first, it will be important to preview the new key for some time
- The reason is DNS caching, older signatures will still be around, needing the old key
- The new key ought to be previewed for at least the amount of time in the DNSKEY set's TTL

---

---

---

---

---

---

---

---

**Observed TTLs of key sets**

• 728 1day	• 234 2hour	• 125 900sec's
• 33 2d	• 170 1h	• 46 300s
• 6 4d	• 34 12h	• 3 1800s
• 1 6d	• 9 6h	
	• 6 3h	
	• 3 4h	
	• 1 5h	
	• 1 10h	

- Those in "seconds" may be preparing for changes, i.e., too short

---

---

---

---

---

---

---

---

**Key Lifecycle**

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
  - Stop using key for signing
  - Post-publish key in DNS
  - Remove key from DNSKEY set
  - Delete the key

**"Activate the key" (and stop using the older one)**

---

---

---

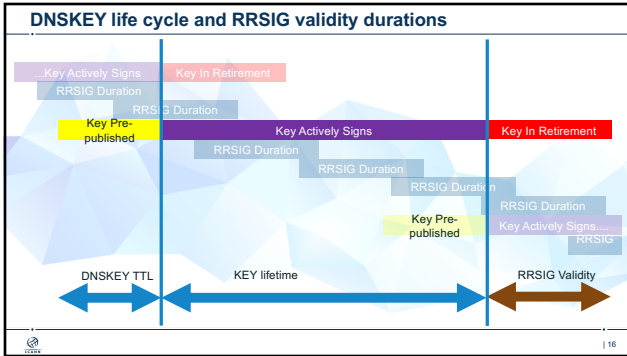
---

---

---

---

---




---

---

---

---

---

---

---

---

### How long should you use a key?

- Truth is, no one knows
- What do ccTLDs do : ZSKs?
  - 1 month or 1 quarter (=3 months) - each popular
  - "forever" – a few
- What do ccTLDs do : KSKs?
  - 1 year seems popular
  - "forever" – a few, but hard to tell from data
- Two with no changes in more than 7 years

---

---

---

---

---

---

---

---

### Roll or not?

- Theory people say you must
- Operators show you don't need to
- But you have to know how
- The question of rolling is more about practice than necessity
  - Operations: change of any kind is always risky
- Exercise your contact with IANA
  - I.e., roll the KSK enough so that "in a panic, it won't be an emergency"

---

---

---

---

---

---


---

---

### Key Lifecycle

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
- Stop using key for signing
- Post-publish key in DNS
- Remove key from DNSKEY set
- Delete the key

"Long time later" stop signing with key (but don't remove it)

 | 19

---

---

---

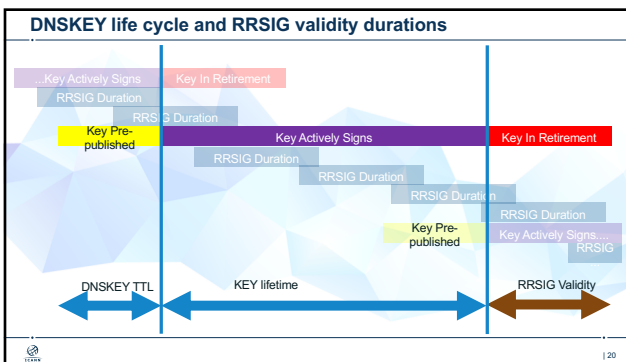
---

---

---

---

---




---

---

---

---

---


---

---

---

### Retirement

- Any signing engine can be told to **"not sign"** with a key that is in the DNSKEY set
  - pre-publish and retirement (post-publish)
- Resolver caches may have older data signed with the key but not have the key set. To validate, the public key is still needed
- Signatures by a key will disappear usually after the TTL expires for data, but TTL can vary
- A better way to end the "lifetime" of signatures to make sure the signature's expiry is managed

 | 21

---

---

---

---

---

---

---

---

### Signature Expiry

```

example. 3600 IN RRSIG (
  SOA                ; type covered by this record
  13                 ; DNSSEC Security Algorithm
  1                  ; Label count
  3600               ; TTL of SOA
  20200911194241    ; expires 2020-09-11@19:42:41UTC
  20200811194241    ; starts 2020-08-11@19:42:41UTC
  6853 example.     ; signed by example's key 6853
  9....3KI....3UBA== ) ; signature value itself
  
```

- Manage the expiry by using a fixed length and knowing when a key is put into retirement

---

---

---

---

---

---

---

---

### Key Lifecycle

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
- Stop using key for signing
- Post-publish key in DNS
- Remove key from DNSKEY set
- Delete the key

Retain key to allow all old signatures to expire

---

---

---

---

---

---

---

---

### Key Lifecycle

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
- Stop using key for signing
- Post-publish key in DNS
- Remove key from DNSKEY set
- Delete the key

Reduces the size of DNSKEY response, many operators forget this step

---

---

---

---

---

---

---

---



**Observation/"Guess"**

- A few times in history a ccTLD will have a large DNSKEY set
  - Filled with retired (unused) keys
- Then the ccTLD will suffer a "failure"
- Never has this been due to the large size of the DNSKEY set
- But it seems like the large set is a symptom of poor monitoring and/or operations processes

---

---

---

---

---

---

---

---

**Key Lifecycle**

- Create a key
- Pre-publish key in a DNSKEY set
- Sign data with the key
- Stop using key for signing
- Post-publish key in DNS
- Remove key from DNSKEY set
- Delete the key

Always clean up, no value in old key, could be misused

---

---

---

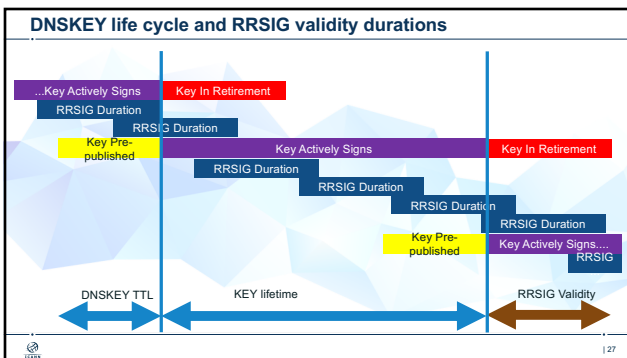
---

---

---

---

---




---

---

---

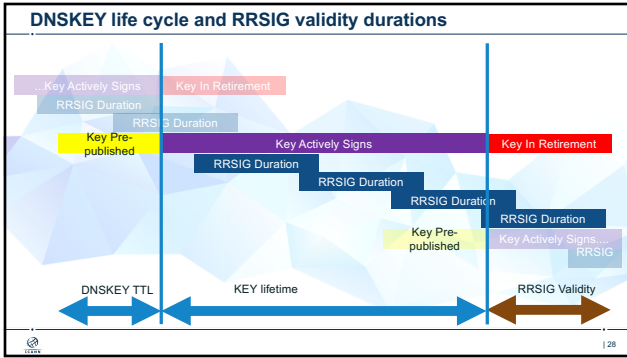
---

---

---

---

---



---

---

---

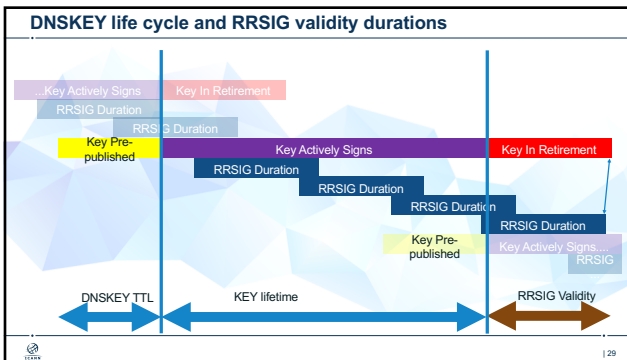
---

---

---

---

---



---

---

---

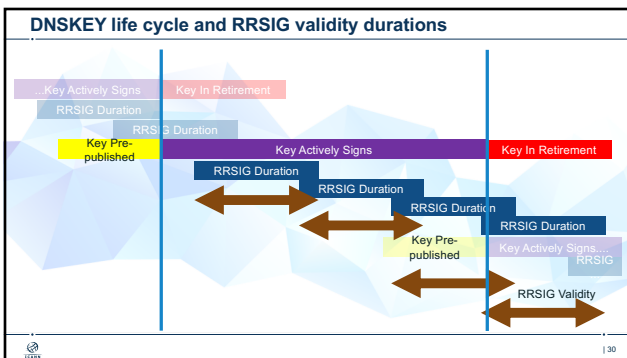
---

---

---

---

---



---

---

---

---

---

---

---

---

**How does IANA manage the KSK?**

- Warning
  - IANA manages the root zone KSK and that key is very special
  - TLD keys may not require the same level of protection
    - In the technical sense
    - There may be other accountability factors
- Using the same methods as IANA is **not** a recommendation, but useful to know

---

---

---

---

---

---

---

---

**How does IANA manage the KSK?**

Tier 1 – Data Facility – Access Control by Facility Operator  
 Tier 2 – Data Facility – Access Control by Facility Operator  
 Tier 3 – Man Trap – Access Control by ICANN  
 Tier 4 – Ceremony Room – Access Control by ICANN  
 Tier 5 – Safe Room – Access Control by ICANN  
 Tier 6 – Safe #1  
 Tier 6 – Safe #2  
 Tier 7 - HSM  
 Tier 7 - Post Boxes  
 Private Keys      Key Ceremony Computer      Crypto Officer's Credentials

---

---

---

---

---

---

---

---

**How does IANA manage the KSK?**

Tier 6 Safes  
 Tier 5 Safe Room  
 Tier 4 Ceremony Room  
 Tier 3 Man Trap

---

---

---

---

---


---

---

---

**How does IANA manage the KSK?**

- Accessing the keys through these tiers is done in "ceremonies"
  - Four times a year
- Multiple people (roles) are needed to access tiers
- External (to IANA) participants are required for public accountability

 | 34

---

---

---

---

---


---

---

---

**How does IANA manage the KSK?**

- For a ccTLD operation, all of this is probably "overkill" (too much)
  - There may be other considerations than pure technical
- Be more flexible

 | 35

---

---

---

---

---

---

---

---

**Engage with ICANN – Thank You and Questions**

 One World, One Internet

Visit us at [icann.org](http://icann.org) Email: [champika.wijayatunga@icann.org](mailto:champika.wijayatunga@icann.org)

-  @icann
-  [facebook.com/icannorg](https://facebook.com/icannorg)
-  [youtube.com/icannnews](https://youtube.com/icannnews)
-  [flickr.com/icann](https://flickr.com/icann)
-  [linkedin/company/icann](https://linkedin/company/icann)
-  [slideshare/icannpresentations](https://slideshare/icannpresentations)
-  [soundcloud/icann](https://soundcloud/icann)

 | 36

---

---

---

---

---

---

---

---