Keeping the DNS Reliable:

DNSSEC and root KSK rollover

June 2026

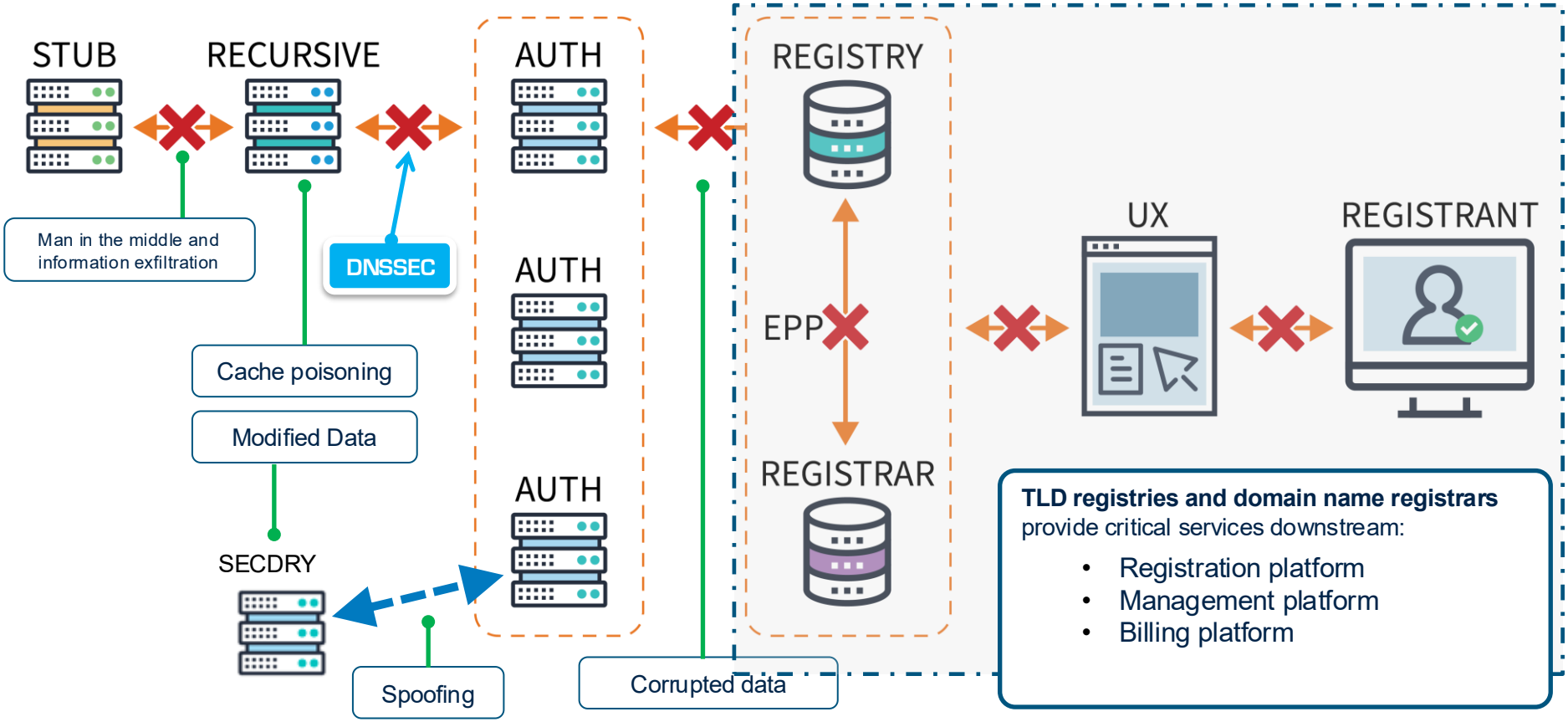
Agenda

Today's Session

- DNSSEC
- Root Signing Process
- Securing the "private" key for the Root signature
- Understanding the KSK Rollover
- Technical Aspects of the KSK Rollover
- Checking my Resolver readiness

DNSSEC

Potential Threats and Attack Vectors in the DNS Ecosystem



DNSSEC: Origin Authentication and Integrity

What DNSSEC does

- It uses public-key cryptography and digital signatures to provide:
  1. Source authentication
  2. Data integrity
- It offers protection against DNS data spoofing
- Prevent Cache Poisoning Attacks

What it doesn't do

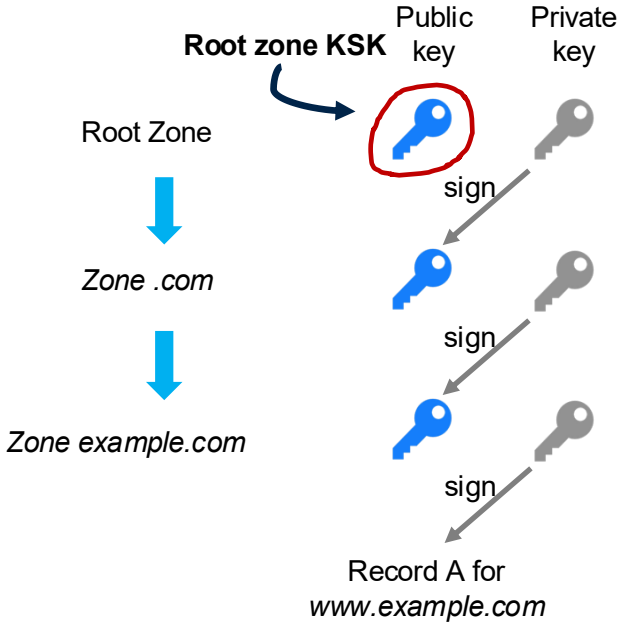
- Provide confidentiality or privacy in DNS data exchange
- Prevent DoS attacks

DNSSEC Records & Trust Anchor

New Records (DNSSEC):

RRSIG	Digital Signature of the records
DNSKEY	Public Key
DS	“Delegation Signer” (Establishes the Chain of Trust)
NSEC	It is used to indicate the "non-existence" of a specific record.

Chain of Trust (DS):



DNSSEC Benefits

Technical Benefits

- Provide source authentication/validation.
- Ensure the integrity and non-tampering of DNS data.
- Authenticated denial of DNS data existence (NSEC).

Impact on the various ecosystem members

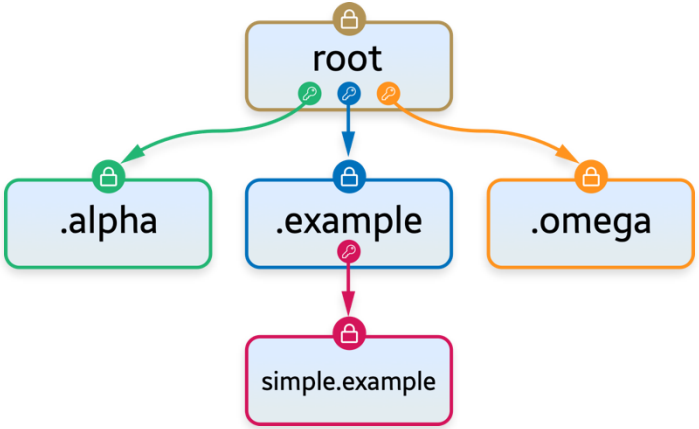
- **End User:** Confidence in reaching the desired/correct website (complement to HTTPS).
- **Registrant:** Fraud mitigation and enhanced brand protection (country code reputation).
- **Registrar:** Comply with industry standards and meet registrant demands for greater security (attract and retain registrants focused on security and reputation).
- **Registry:** Comply with industry best practices and registrar demands for enhanced domain security.

Root Signing Process

There are no “7 keys to the Internet.” That is just marketing. However, there are 14 individuals, each holding a key to a physical safe deposit box. Inside each safe deposit box are credentials that, when used in combination, enable the activation of a device containing the necessary components to generate digital signatures.

Let’s take a closer look....

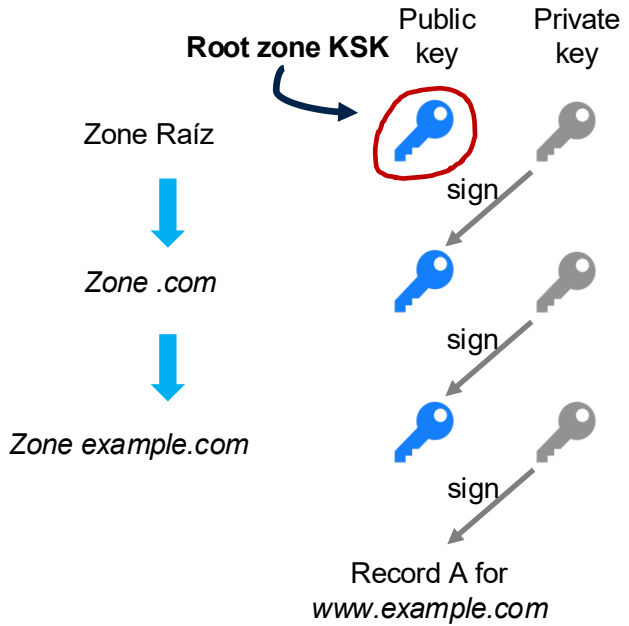
DNSSEC and the Trust Anchor



The mechanism for “trusting” the information necessary to verify the digital signature of DNS data relies on each “parent” zone certifying the authenticity of such information regarding its “children.”

However... the Root zone has no parent. Therefore, we require a “trustworthy” mechanism to guarantee the authenticity of the Root’s signature.

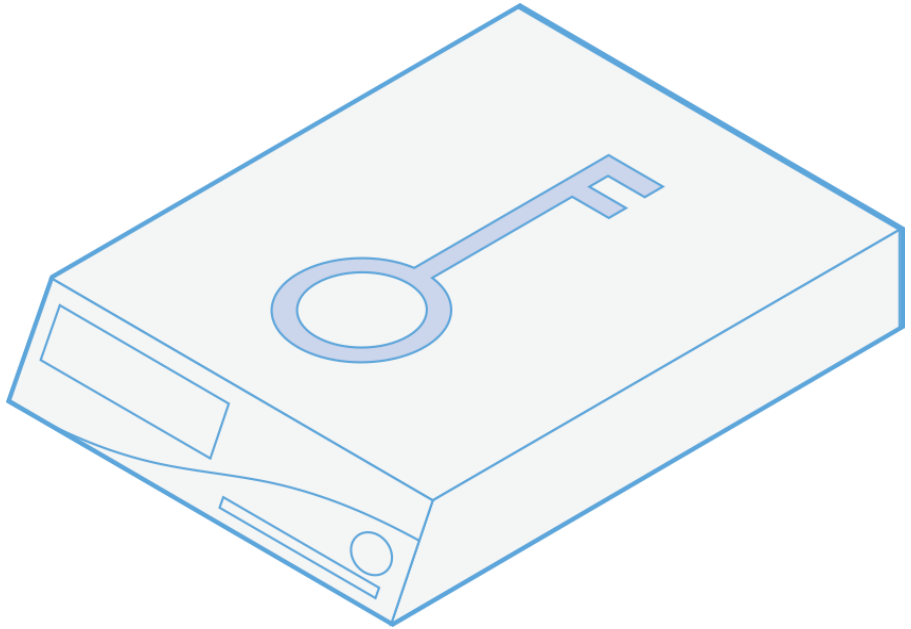
Trust anchor (DS):



Securing the “private” key for the Root signature

Root Signing Key Security

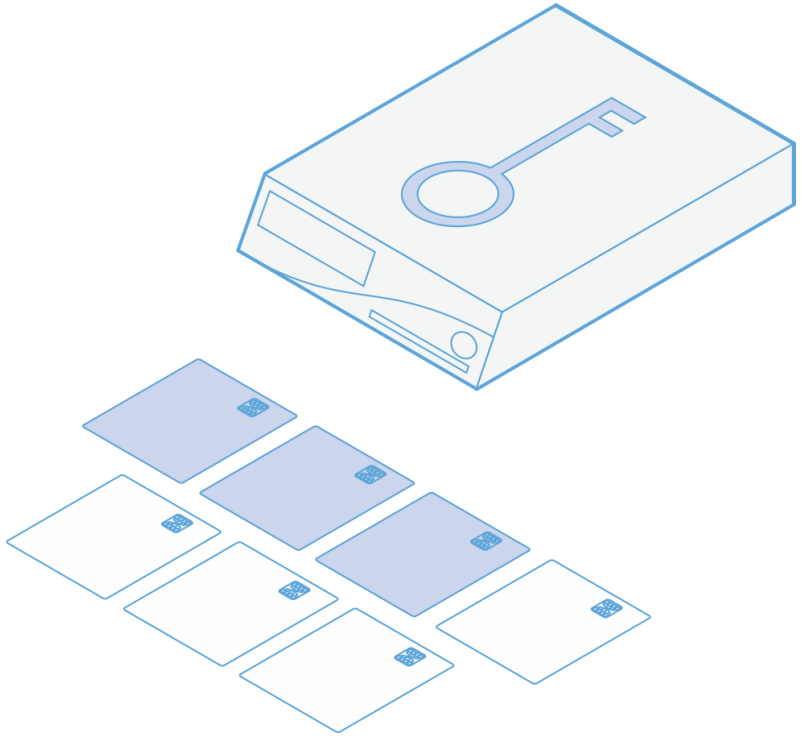
The Root signing key is stored in a device called a “Hardware Security Module” (HSM), whose sole purpose is to securely store cryptographic keys. The device is designed to be tamper-proof. If an attempt is made to open it, its contents will self-destruct.



Root Signing Key Security

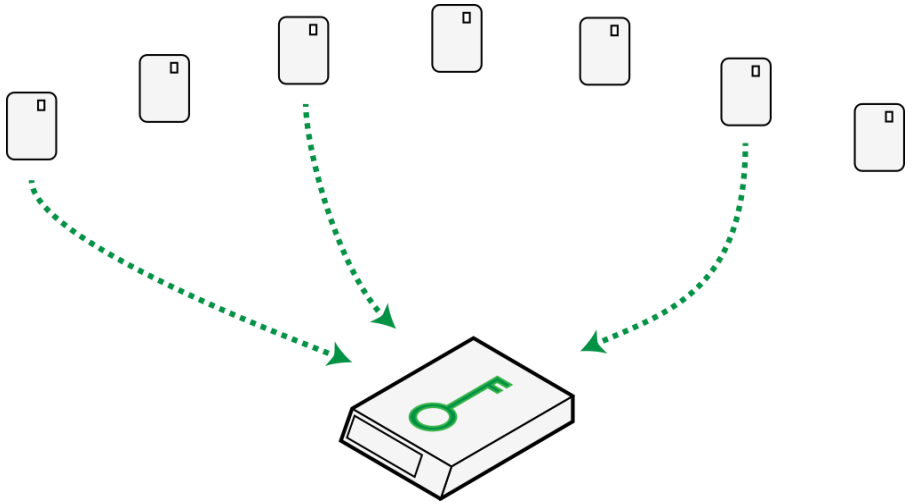
There are seven smart cards that can activate each device. The device is configured such that three of the seven smart cards must be present for it to be usable.

In other words, if I do not have at least three of the seven cards, I will not be able to access the device's contents.



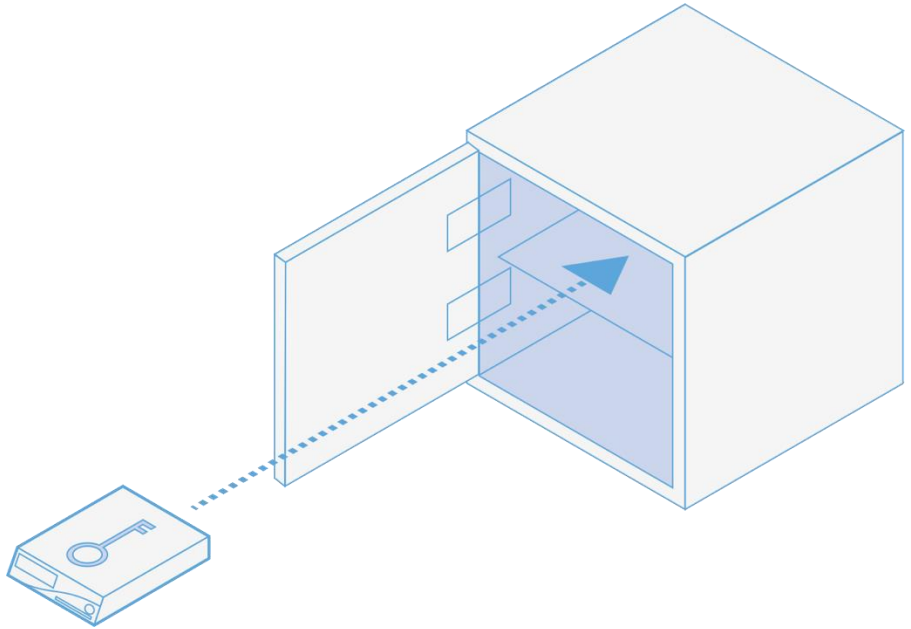
Root Signing Key Security

Each smart card is assigned to a different member of the ICANN community, known as a “Trusted Community Representative” (TCR). Therefore, to access the signing key, at least three of these TCRs must meet in person. These planned events are called key signing ceremonies.



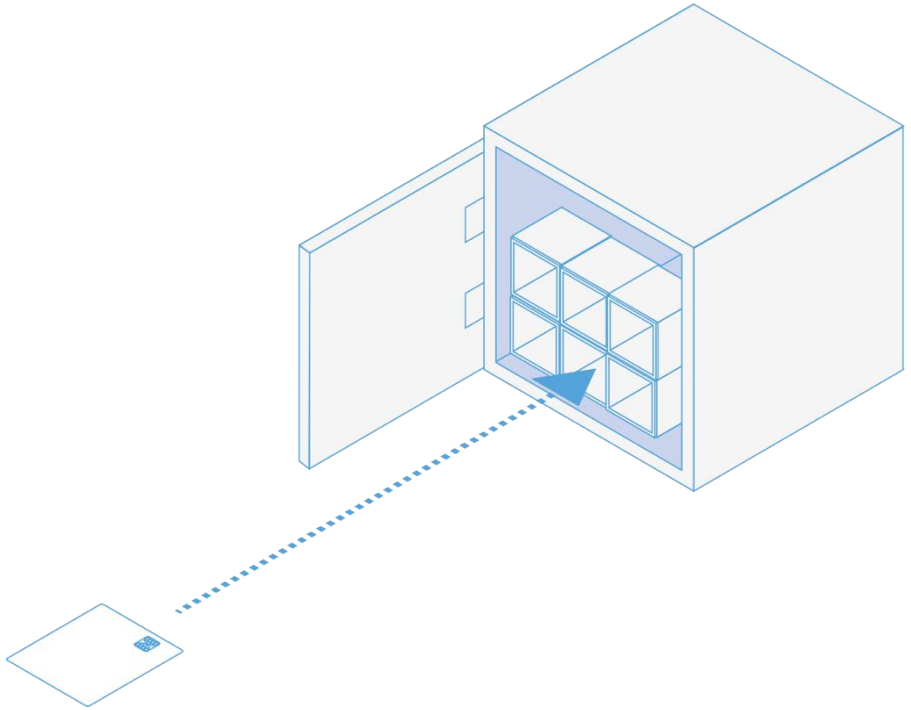
Root Signing Key Security

The HSM is stored inside a high-security safe, which can only be opened by a designated individual: the “safe security controller.” The integrity of the safe is monitored using seismic and temperature sensors, among others.



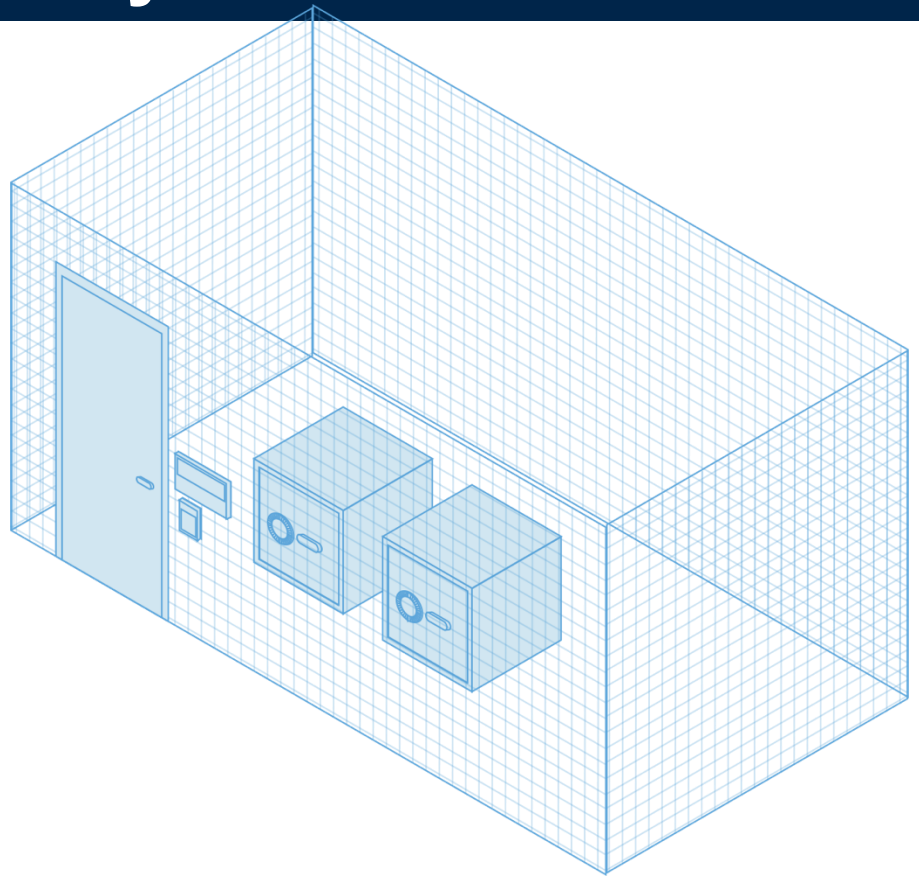
Root Signing Key Security

The smart card for each TCR is stored in a second credentials safe, which contains a series of security boxes. Each security box is accessed using a mechanical key that the TCR carries with them and keeps secure between ceremonies.



Root Signing Key Security

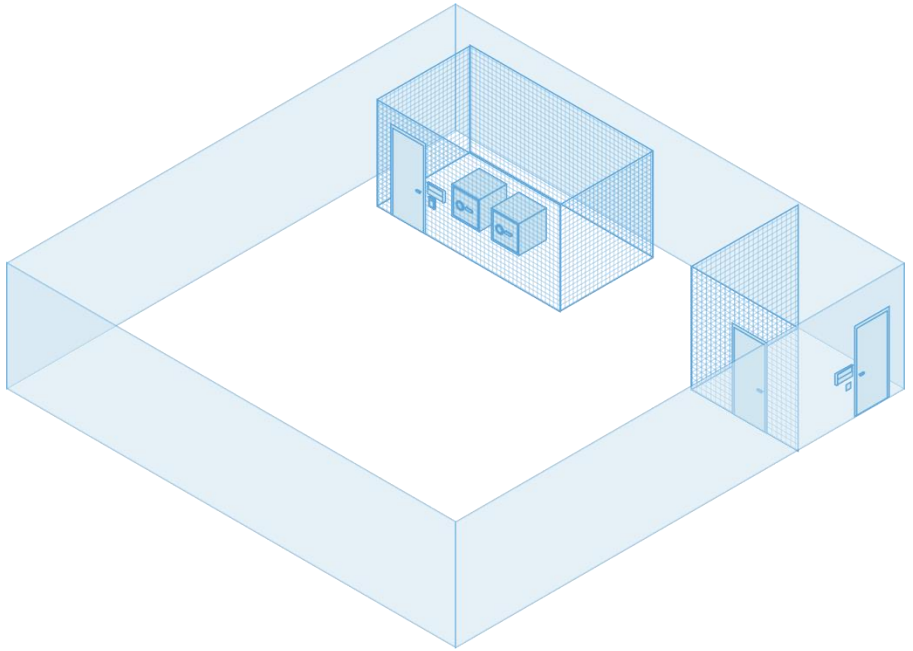
The two safes are housed within a secure, RF-shielded metal cage that can only be opened jointly by two designated individuals: the “Ceremony Administrator” and the “Internal Witness.” The room is monitored by intrusion and motion sensors, and access is controlled via biometric mechanisms.



Root Signing Key Security

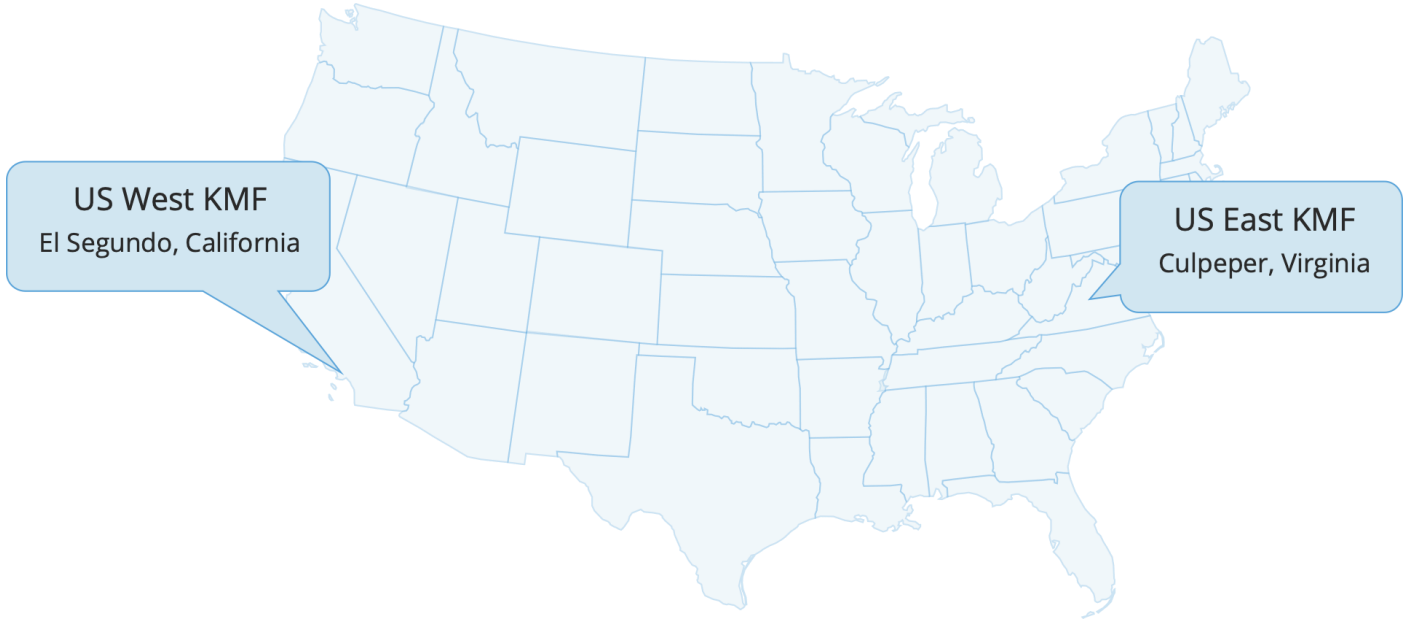
The secure room is located within a larger room where ceremonies involving TCRs and other individuals are conducted. These ceremonies are video-recorded, witnessed by the participants and other individuals, and audited by an external auditing firm.

Access to this room must be granted by a designated third party—the “Physical Access Control Administrator”—who is not present on-site, and is facilitated through biometric access controls.



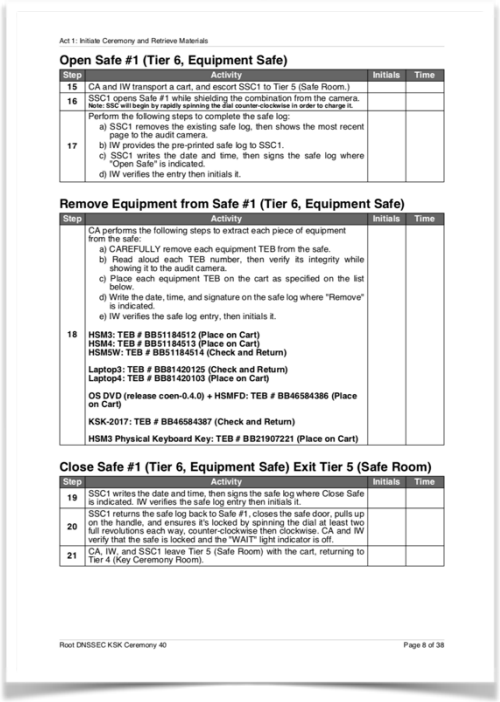
Root Signing Key Security

The ceremonial rooms, known as “key management facilities,” are located within two facilities guarded by third-party providers (Data Centers)—one on the East Coast and the other on the West Coast of the United States.



Root Signing Key Security

Each ceremony is organized using a complete script that identifies every individual step that must be performed.



Act 1: Initiate Ceremony and Retrieve Materials

Open Safe #1 (Tier 6, Equipment Safe)

Step	Activity	Initials	Time
15	CA and IW transport a cart, and escort SSC1 to Tier 5 (Safe Room.)		
16	SSC1 opens Safe #1 while shielding the combination from the camera. **Note: SSC will begin by rapidly spinning the dial counter-clockwise in order to charge it.**		
17	Perform the following steps to complete the safe log: a) SSC1 removes the existing safe log, then shows the most recent page to the audit camera. b) IW provides the pre-printed safe log to SSC1. c) SSC1 writes the date and time, then signs the safe log where "Open Safe" is indicated. d) IW verifies the entry then initials it.		

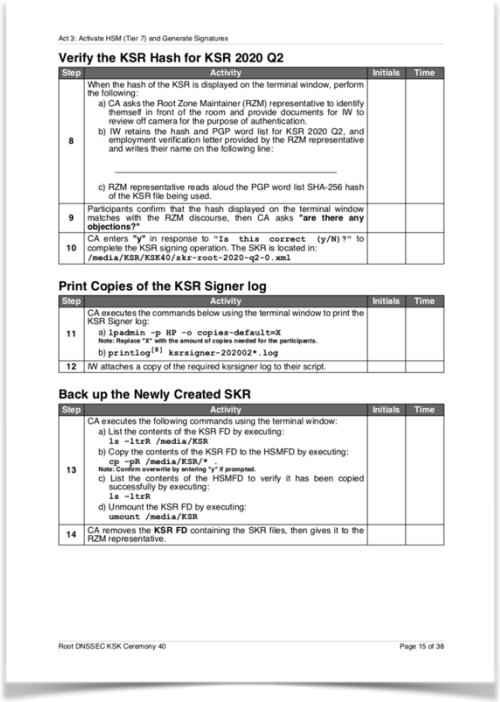
Remove Equipment from Safe #1 (Tier 6, Equipment Safe)

Step	Activity	Initials	Time
18	CA performs the following steps to extract each piece of equipment from the safe: a) CAREFULLY remove each equipment TEB from the safe. b) Read aloud each TEB number, then verify its integrity while showing it to the audit camera. c) Place each equipment TEB on the cart as specified on the list below. d) Write the date, time, and signature on the safe log where "Remove" is indicated. e) IW verifies the safe log entry, then initials it. **HSM3: TEB # BB51184512 (Place on Cart)** **HSM4: TEB # BB51184513 (Place on Cart)** **HSM5W: TEB # BB51184514 (Check and Return)** **Laptop3: TEB # BB81420125 (Check and Return)** **Laptop4: TEB # BB81420103 (Place on Cart)** **OS DVD (release coen-0.4.0) + HSMFD: TEB # BB46584386 (Place on Cart)** **KSK-2017: TEB # BB46584387 (Check and Return)** **HSM3 Physical Keyboard Key: TEB # BB21907221 (Place on Cart)**		

Close Safe #1 (Tier 6, Equipment Safe) Exit Tier 5 (Safe Room)

Step	Activity	Initials	Time
19	SSC1 writes the date and time, then signs the safe log where Close Safe is indicated. IW verifies the safe log entry then initials it.		
20	SSC1 returns the safe log back to Safe #1, closes the safe door, pulls up on the handle, and ensures it's locked by spinning the dial at least two full revolutions each way, counter-clockwise then clockwise. CA and IW verify that the safe is locked and the "WAIT" light indicator is off.		
21	CA, IW, and SSC1 leave Tier 5 (Safe Room) with the cart, returning to Tier 4 (Key Ceremony Room).		

Root DNSSEC KSK Ceremony 40 Page 8 of 38



Act 3: Activate HSM (Tier 7) and Generate Signatures

Verify the KSR Hash for KSR 2020 Q2

Step	Activity	Initials	Time
8	When the hash of the KSR is displayed on the terminal window, perform the following: a) CA asks the Root Zone Maintainer (RZM) representative to identify themself in front of the room and provide documents for IW to review off camera for the purpose of authentication. b) IW retains the hash and PGP word list for KSR 2020 Q2, and employment verification letter provided by the RZM representative and writes their name on the following line: _____ c) RZM representative reads aloud the PGP word list SHA-256 hash of the KSR file being used.		
9	Participants confirm that the hash displayed on the terminal window matches with the RZM discourse, then CA asks **"are there any objections?"**		
10	CA enters **"y"** in response to `"Is this correct (y/N)?"` to complete the KSR signing operation. The SKR is located in: `/media/KSR/KSK40/skr-root-2020-q2-0.xml`		

Print Copies of the KSR Signer log

Step	Activity	Initials	Time
11	CA executes the commands below using the terminal window to print the KSR Signer log: a) `lpadmin -p HP -o copies-default=X` **Note: Replace "X" with the amount of copies needed for the participants.** b) `printlog[8] ksrsigner-202002*.log`		
12	IW attaches a copy of the required ksrsigner log to their script.		

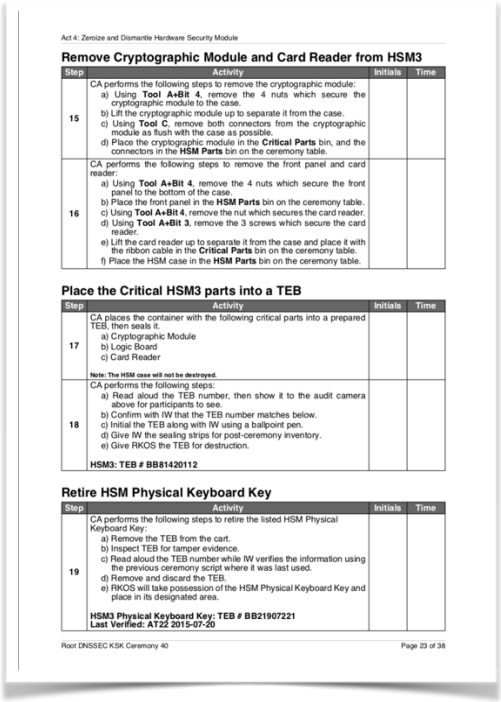
Back up the Newly Created SKR

Step	Activity	Initials	Time

13	CA executes the following commands using the terminal window:
	a) List the contents of the KSR FD by executing:
	`ls -ltrR /media/KSR`
	b) Copy the contents of the KSR FD to the HSMFD by executing:
	`cp -pR /media/KSR/* .`
	Note: Confirm overwrite by entering "y" if prompted.
	c) List the contents of the HSMFD to verify it has been copied successfully by executing:
	`ls -ltrR`
	d) Unmount the KSR FD by executing:
	`umount /media/KSR`		

14	CA removes the **KSR FD** containing the SKR files, then gives it to the RZM representative.		

Root DNSSEC KSK Ceremony 40 Page 15 of 38



Act 4: Zeroize and Dismantle Hardware Security Module

Remove Cryptographic Module and Card Reader from HSM3

Step	Activity	Initials	Time
15	CA performs the following steps to remove the cryptographic module: a) Using **Tool A+Bit 4**, remove the 4 nuts which secure the cryptographic module to the case. b) Lift the cryptographic module up to separate it from the case. c) Using **Tool C**, remove both connectors from the cryptographic module as flush with the case as possible. d) Place the cryptographic module in the **Critical Parts** bin, and the connectors in the **HSM Parts** bin on the ceremony table.		
16	CA performs the following steps to remove the front panel and card reader: a) Using **Tool A+Bit 4**, remove the 4 nuts which secure the front panel to the bottom of the case. b) Place the front panel in the **HSM Parts** bin on the ceremony table. c) Using **Tool A+Bit 4**, remove the nut which secures the card reader. d) Using **Tool A+Bit 3**, remove the 3 screws which secure the card reader. e) Lift the card reader up to separate it from the case and place it with the ribbon cable in the **Critical Parts** bin on the ceremony table. f) Place the HSM case in the **HSM Parts** bin on the ceremony table.		

Place the Critical HSM3 parts into a TEB

Step	Activity	Initials	Time
17	CA places the container with the following critical parts into a prepared TEB, then seals it. a) Cryptographic Module b) Logic Board c) Card Reader **Note: The HSM case will not be destroyed.**		
18	CA performs the following steps: a) Read aloud the TEB number, then show it to the audit camera above for participants to see. b) Confirm with IW that the TEB number matches below. c) Initial the TEB along with IW using a ballpoint pen. d) Give IW the sealing strips for post-ceremony inventory. e) Give RKOS the TEB for destruction. **HSM3: TEB # BB81420112**		

Retire HSM Physical Keyboard Key

Step	Activity	Initials	Time
19	CA performs the following steps to retire the listed HSM Physical Keyboard Key: a) Remove the TEB from the cart. b) Inspect TEB for tamper evidence. c) Read aloud the TEB number while IW verifies the information using the previous ceremony script where it was last used. d) Remove and discard the TEB. e) RKOS will take possession of the HSM Physical Keyboard Key and place in its designated area. **HSM3 Physical Keyboard Key: TEB # BB21907221** **Last Verified: AT22 2015-07-20**		

Root DNSSEC KSK Ceremony 40 Page 23 of 38

DNSSEC

DNS Security Extensions (DNSSEC) constitute a hierarchical public-key cryptographic system that mirrors the hierarchical delegation of DNS itself. At the apex of this hierarchy resides the Root Zone Signing Key (KSK), the sole anchor of trust for the DNSSEC system.

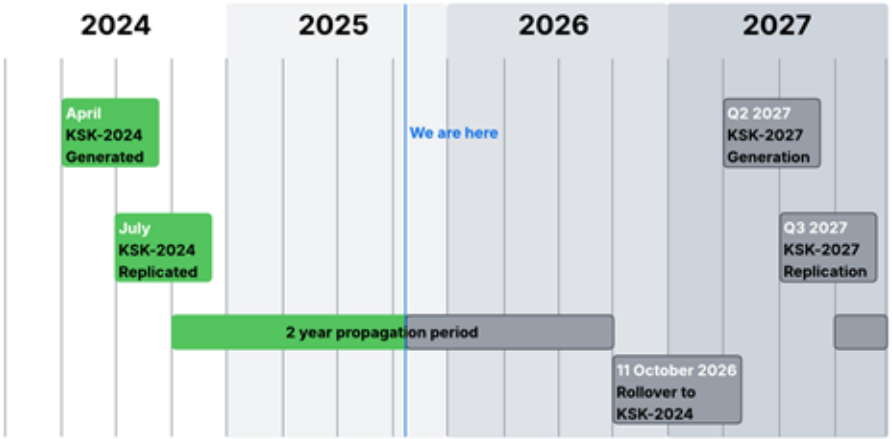
We manage this key through a highly transparent process, which includes public-key signing ceremonies and an open design model.

Understanding the KSK Rollover

Why Does the KSK Change?

The KSK rollover:

- Introduces a new Root Zone KSK (KSK-2024)
- Updates the DNS root trust anchor
- Gradually transitions systems to trust the new key



Key status

This table provides information on the keys generated for Root Zone KSK operations. Software implementers should rely on the XML trust anchors file for normative parameters on keys.

INFORMAL NAME	STATUS	DETAILS
KSK-2024	Pre-Publication	Generated 2024-04-26 (attestation) with key tag 38696 and label Kmyv6jo. Expected to supersede KSK-2017.
KSK-2017	Active	Generated 2016-10-27 (attestation) with key tag 20326 and label Klajeyz. Signing since 2018-10-11.
KSK-2023	Abandoned	Generated 2023-04-27 (attestation) with key tag 46211 and label Kmrfl3b. Will not be used, superseded by KSK-2024.
KSK-2010	Retired	Generated 2010-06-16 (attestation) with key tag 19036 and label Kjqmt7v. Signing between 2010-07-15 and 2018-10-11.

Why Does the KSK Change?

Why this matters:

- Cryptographic keys should be updated periodically
- Planned maintenance is safer than emergency replacement
- Regular updates strengthen long-term DNS security and resilience

Technical Aspects of the KSK Rollover

Root Zone KSK Rollover Technical Preparedness

This session focuses on the operational and technical aspects of the rollover, including:

- How validating resolvers use DNSSEC trust anchors
- What happens during the KSK transition process
- RFC 5011 automated trust anchor updates
- Monitoring, testing, and operational readiness
- Common failure scenarios and troubleshooting considerations
- Actions operators can take to help ensure a smooth transition

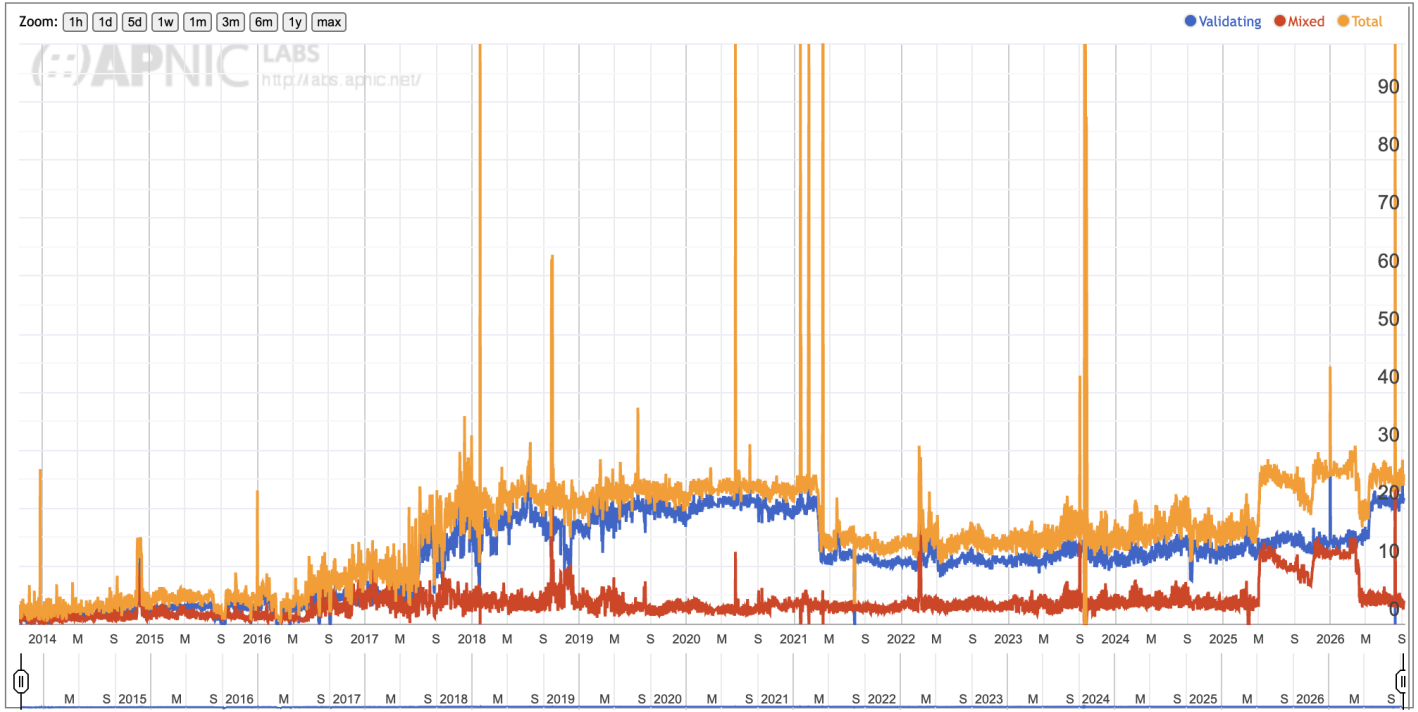
Who Should Pay Attention?

This session is most relevant for:

- DNSSEC-validating resolver operators
- ISPs and network operators
- Enterprise DNS administrators
- DNS software developers
- System integrators and infrastructure teams

Who Should Pay Attention?

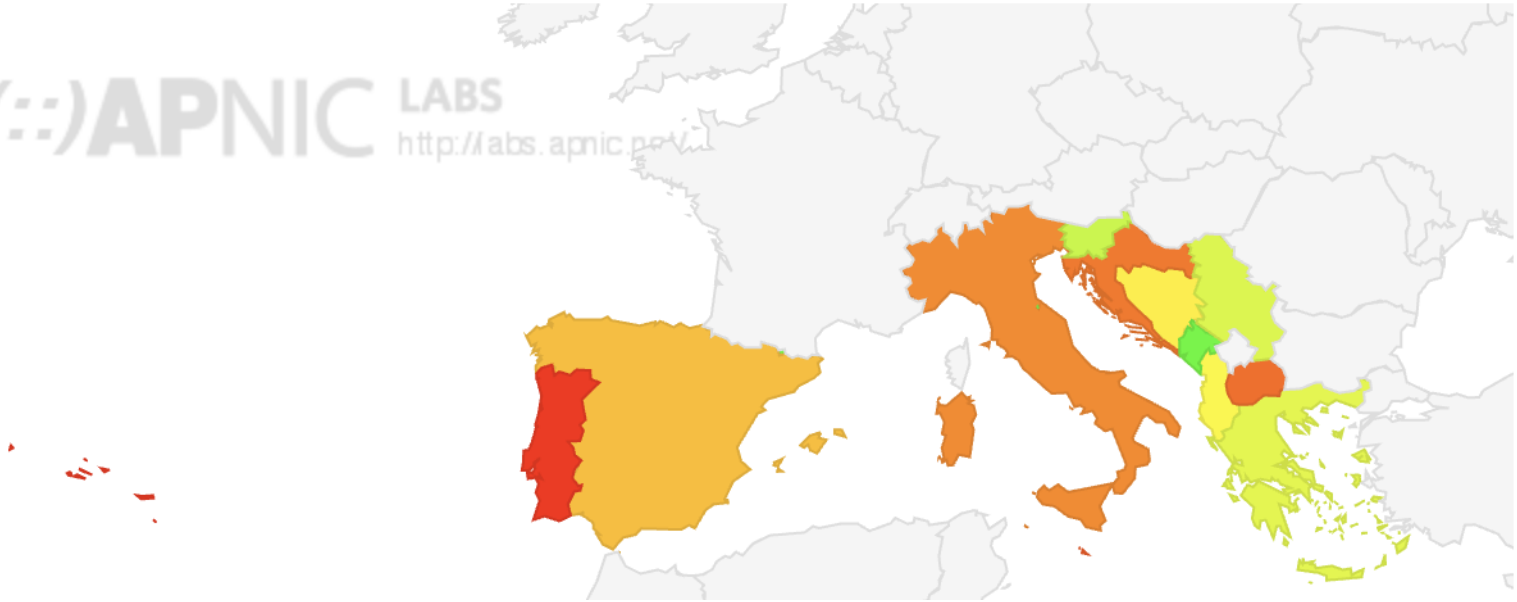
Use of DNSSEC Validation for Croatia (HR)



Source: APNIC https://stats.labs.apnic.net/dnssec/HR?o=cXEw1v1p1x0l1

Who Should Pay Attention?

Region Map for Southern Europe (039)



Source: APNIC https://stats.labs.apnic.net/dnssec/HR?o=cXEw1v1p1x0l1

What Happens During the Rollover?

High-level rollover process

- New KSK is generated
- New KSK is published in the root zone
- Resolvers learn the new trust anchor
- Root zone signatures transition to the new KSK

Trust Anchor Management

Trust anchors are critical for validation

Resolvers using RFC 5011 can:

- Automatically learn new trust anchors
- Update trust anchors without manual intervention

Resolvers not using RFC 5011 may require:

- Manual trust anchor updates
- Configuration validation
- Additional operational testing

Risk: Resolvers without the new trust anchor may return SERVFAIL responses

Recommended Operator Actions

Operators and developers should:

- Verify RFC 5011 support
- Test resolver behavior
- Monitor rollover updates
- Update trust anchors where needed

ICANN also provides:

- Operational testing resources
- Trust anchor publication files
- Technical guidance and outreach

What Operators Should Monitor

Expected behavior and operational monitoring

Prepared resolvers should:

- Transition automatically
- Continue validating normally
- Experience no service disruption

Operators should monitor for:

- SERVFAIL increases
- Validation failures
- Resolver log anomalies
- Unexpected traffic patterns

Potential Operational Issues

Possible operational problems

Issues may occur if:

- Trust anchors are outdated
- RFC 5011 is disabled or malfunctioning
- Software mishandles multiple KSKs
- Policies or firewalls block updates

Possible symptoms:

- SERVFAIL responses
- Intermittent DNS failures
- User connectivity complaints

What Could Happen if Systems are Unprepared?

Potential issues:

- Validation failures
- SERVFAIL responses
- DNS lookup failures
- User-facing connectivity problems

To protect stability:

- The rollout may pause
- Phases may be extended
- Changes may temporarily back out if necessary

Goal: A seamless transition for Internet users

Monitoring and Stability Protections

Protecting DNS stability

Operational monitoring includes:

- Resolver telemetry
- Validation failure monitoring
- RFC 8145 trust anchor signaling
- Root server system observations

If significant issues appear:

- The rollout may pause
- Timelines may extend
- Changes may temporarily back out

Operator Preparation Checklist

Preparation checklist

- ✓ Verify RFC 5011 support
- ✓ Confirm trust anchor configuration
- ✓ Test validating resolvers
- ✓ Review monitoring and alerting
- ✓ Update operational documentation
- ✓ Follow ICANN and IANA updates

Checking my Resolver Readiness

Let’s check if your resolver does do DNSSEC

Run the following command for all your Resolvers, where you replace <ip-address> with the IP address of each resolver.

dig @<ip-address> dnssec-failed.org +dnssec

If the answer contains status: SERVFAIL
Your resolver does do DNSSEC validation and you should follow on with this guide.

If you got an answer with an IP address, your resolver does not perform DNSSEC validation and you don’t need to do anything about the Root-Key rollover. But you should of course consider switching on DNSSEC validation.

DNS as a Service

Protective DNS is used by many organizations. Obviously you can't do anything about the Root KSK Rollover as you don't run DNS Resolvers your own, but you can ask your service provider what they are doing.

Commercial Software

If you use a commercial software the new key should already be included in one of your latest updates. You should install the latest updates and if you for some reason can't do that, you should plan for an update some time before 11 October 2026. Talk to your vendor about the Root KSK Rollover and ask in what update the new key is/will be included.

New Trust Anchors, where to find them

You can use *curl* or *wget* to obtain the official trusted anchors file in XML format directly from the IANA data repository:

```
curl -O https://data.iana.org/root-anchors/root-anchors.xml
```

New Trust Anchors, where to find them

You can use a command similar to the following to extract all the keys in the generally accepted format for the BIND Resolver configuration for example:

```
awk '/<KeyDigest/,/<\/KeyDigest>/ {print}' root-anchors.xml | awk '
/<\/KeyDigest>/ { if(tag!="") print "    . static-ds " tag " " alg " " type " \"" dig
"\";"; tag=""; alg=""; type=""; dig="" }
/<KeyTag>/ {gsub(/<\/?KeyTag>/,""); gsub(/[ \t]/,""); tag=$0}
/<Algorithm>/ {gsub(/<\/?Algorithm>/,""); gsub(/[ \t]/,""); alg=$0}
/<DigestType>/ {gsub(/<\/?DigestType>/,""); gsub(/[ \t]/,""); type=$0}
/<Digest>/ {gsub(/<\/?Digest>/,""); gsub(/[ \t]/,""); dig=$0}
'
```

You should get something similar to this:

```
. static-ds 19036 8 2 "49AAC11D7B6F6446702E54A1607371607A1A41855200FD2CE1CDDE32F24E8FB5";
. static-ds 20326 8 2 "E06D44B80B8F1D39A95C0B0D7C65D08458E880409BBC683457104237C7F8EC8D";
. static-ds 38696 8 2 "683D2D0ACB8C9B712A1948B27F741219298D0A450D612C483AF444A4C0FB2B16";
```

Open Source Software: BIND

There are several methods of configuring DNSSEC validation in BIND.

1. Please check your BIND configuration for the following statement “dnssec-validation auto;”. BIND will update the trust-anchor automatically. In the latest versions of BIND this configuration is the default. If “dnssec-validation” is not configured at all, it means you are on “auto”.

2. If your configuration says “dnssec-validation yes;” please consider changing to “auto”. If you can’t change to “auto” you will need to update the key file manually. Please see below for the correct content.

3. If your bind configuration contains a section called “trust-anchors”
   a. you should have a list of keys, which starts with “. initial-ds” or “. initial-key”.
   b. If you see a definition containing “. static-ds” or “. static-key” you need to either change the “static” to “initial” or add the new key manually to your configuration.

Open Source Software: BIND

4. **Run “rndc managed-keys status”**
   It should display two keys, one with keyid 20326 (current KSK from 2017 to be replaced) and one with keyid 38696 (new replacement KSK 2025).

5. **Run “rndc secroots”**
   It writes a file called “named.secroots” (see below for folders to check). It should contain two keys as
   ./RSASHA256/20326
   ./RSASHA256/38696

6. If you do not use views, your current initial set of trust-anchors should be in a file named “bind.keys” (usually in /etc/bind) and the dynamic set of trust-anchors “managed-keys.bind” (usually in /var/cache/bind), usually found where your bind configuration lives. If you use views, you should have one or more “.mkeys” files. Check the files, they should contain two keys.

Open Source Software: BIND

Here is an example of the file bind.keys contents that is created by a current version of bind when you first install bind on a machine:

```
trust-anchors {
        # This key (20326) was published in the root zone in 2017, and
        # is scheduled to be phased out starting in 2025. It will remain
        # in the root zone until some time after its successor key has
        # been activated. It will remain this file until it is removed
        # from the root zone.
        . initial-key 257 3 8 \
              "AwEAAaz/tAm8yTn4Mfeh5eyI96WSVexTBAvkMgJzkKTOiW1vkIbzxeF3
               +/4RgWOq7HrxRixHlFlExOLAJr5emLvN7SWXgnLh4+B5xQlNVz8Og8kv
               ArMtNROxVQuCaSnIDdD5LKyWbRd2n9WGe2R8PzgCmr3EgVLrjyBxWezF
               0jLHwVN8efS3rCj/EWgvIWgb9tarpVUDK/b58Da+sqqls3eNbuv7pr+e
               oZG+SrDK6nWeL3c6H5Apxz7LjVc1uTIdsIXxuOLYA4/ilBmSVIzuDWfd
               RUfhHdY6+cn8HFRm+2hM8AnXGXws9555KrUB5qihylGa8subX2Nn6UwN
               R1AkUTV74bU=";

        # This key (38696) will be pre-published in the root zone in 2025
        # and is scheduled to begin signing in late 2026. At that time,
        # servers which were already using the old key (20326) should roll
        # seamlessly to this new one via RFC 5011 rollover.
        . initial-ds 38696 8 2 \
              "683D2D0ACB8C9B712A1948B27F741219298D0A450D612C483AF444A
               4C0FB2B16";
};
```

Open Source Software: UNBOUND

There are several methods of configuring DNSSEC validation in unbound.

1. **Please check your unbound configuration for the following statement "auto-trust-anchor-file:" Unbound will update the trust-anchor automatically. Check the file for correct contents.**
2. **If your configuration uses "trust-anchor-file:" or "trusted-keys-file:" statements please consider changing to auto-trust-anchor-file. If you can't change to auto-trust-anchor-file you will need to update the key file manually. Please see below for the correct content.**
3. **If your unbound configuration contains "trust-anchor:" the keys are defined explicit. Check that both keys (keyid 20326 and 38696) are in the list. Preferably switch to "auto-trust-anchor-file:".**

Open Source Software: UNBOUND

Depending on the configuration statement you use, please check the file configured to have a content similar to the content below.

```
; autotrust trust anchor file
;;id: . 1
;;last_queried: 1773685071 ;;Mon Mar 16 18:17:51 2026
;;last_success: 1773685071 ;;Mon Mar 16 18:17:51 2026
;;next_probe_time: 1773728077 ;;Tue Mar 17 06:14:37 2026
;;query_failed: 0
;;query_interval: 43200
;;retry_time: 8640
.     3600  IN    DNSKEY     257 3 8
AwEAAaz/tAm8yTn4Mfeh5eyI96WSVexTBAvkMgJzkKTOiW1vkIbzxeF3+/4RgWOq7HrxRixHlFlExO
LAJr5emLvN7SWXgnLh4+B5xQlNVz8Og8kvArMtNROxVQuCaSnIDdD5LKyWbRd2n9WGe2R8PzgCmr3E
gVLrjyBxWezF0jLHwVN8efS3rCj/EWgvIWgb9tarpVUDK/b58Da+sqqls3eNbuv7pr+eoZG+SrDK6n
WeL3c6H5Apxz7LjVc1uTIdsIXxuOLYA4/ilBmSVIzuDWfdRUfhHdY6+cn8HFRm+2hM8AnXGXws9555
KrUB5qihylGa8subX2Nn6UwNR1AkUTV74bU= ;{id = 20326 (ksk), size = 2048b}
;;state=2 [  VALID  ] ;;count=0 ;;lastchange=1773667482 ;;Mon Mar 16 13:24:42
2026
.     3600  IN    DNSKEY     257 3 8
AwEAAa96jeuknZlaeSrvyAJj6ZHv28hhOKkx3rLGXVaC6rXTsDc449/cidltpkyGwCJNnOAlFNKF2j
BosZBU5eeHspaQWOmOElZsjICMQMC3aeHbGiShvZsx4wMYSjH8e7Vrhbu6irwCzVBApESjbUdpWWmE
nhathWu1jo+siFUiRAAxm9qyJNg/wOZqqzL/dL/q8PkcRU5oUKEpUge71M3ej2/7CPqpdVwuMoTvoB
+ZOT4YeGyxMvHmbrxlFzGOHOijtzN+u1TQNatX2XBuzZNQ1K+s2CXkPIZo7s6JgZyvaBevYtxPvYLw
4z9mR7K2vaF18UYH9Z9GNUUeayffKC73PYc= ;{id = 38696 (ksk), size = 2048b}
;;state=2 [  VALID  ] ;;count=0 ;;lastchange=1773667482 ;;Mon Mar 16 13:24:42
2026
```

Open Source Software: UNBOUND

Finally run

```
dig @<ip-address> trustanchor.unbound CH TXT +noall +nocomments +answer
```

Against your resolver. The answer should contain one line with

```
trustanchor.unbound. 0 CH TXT ". 20326 38696"
```

Open Source Software: KNOT

Since version 4.0 Knot Resolver has switched on DNSSEC validation by default. No trust-anchor file is needed, everything is compiled into the executable.
Check your configuration for any appearance of “trust_anchors”. Please check with the Knot documentation at https://knot-resolver.readthedocs.io/en/stable/config-dnssec.html how to proceed.

There is an undocumented way of checking (given to us by the Knot developers), this may or may not work in your version of the knot-resolver.

```
$ echo 'trust_anchors.summary()' | sudo socat - /run/knot-resolver/control/0
```

```
.                    86400    DNSKEY   257 3 8
AwEAAaz/tAm8yTn4Mfeh5eyI96WSVexTBAvkMgJzkKTOiW1vkIbzxeF3+/4RgWOq7HrxRixHlFlExO
LAJr5emLvN7SWXgnLh4+B5xQlNVz8Og8kvArMtNROxVQuCaSnIDdD5LKyWbRd2n9WGe2R8PzgCmr3E
gVLrjyBxWezF0jLHwVN8efS3rCj/EWgvIWgb9tarpVUDK/b58Da+sqqls3eNbuv7pr+eoZG+SrDK6n
WeL3c6H5Apxz7LjVc1uTIdsIXxuOLYA4/ilBmSVIzuDWfdRUfhHdY6+cn8HFRm+2hM8AnXGXws9555
KrUB5qihylGa8subX2Nn6UwNR1AkUTV74bU= ; Valid: ; KeyTag:20326
.                        86400    DNSKEY   257 3 8
AwEAAa96jeuknZlaeSrvyAJj6ZHv28hhOKkx3rLGXVaC6rXTsDc449/cidltpkyGwCJNnOAlFNKF2j
BosZBU5eeHspaQWOmOElZsjICMQMC3aeHbGiShvZsx4wMYSjH8e7Vrhbu6irwCzVBApESjbUdpWWmE
nhathWu1jo+siFUiRAAxm9qyJNg/wOZqqzL/dL/q8PkcRU5oUKEpUge71M3ej2/7CPqpdVwuMoTvoB
+ZOT4YeGyxMvHmbrxlFzGOHOijtzN+u1TQNatX2XBuzZNQ1K+s2CXkPIZo7s6JgZyvaBevYtxPvYLw
4z9mR7K2vaF18UYH9Z9GNUUeayffKC73PYc= ; Valid: ; KeyTag:38696
```

Open Source Software: PowerDNS

Please check your configuration for "`trustanchorfile`". It gives you the location of the file to check for the trust-anchors. Please make sure the file contains both keys, similar to

```
. IN DNSKEY 257 3 8
AwEAAaz/tAm8yTn4Mfeh5eyI96WSVexTBAvkMgJzkKTOiW1vkIbzxeF3+/4RgWOq7HrxRixH
lFlExOLAJr5emLvN7SWXgnLh4+B5xQlNVz8Og8kvArMtNROxVQuCaSnIDdD5LKyWbRd2n9WG
e2R8PzgCmr3EgVLrjyBxWezF0jLHwVN8efS3rCj/EWgvIWgb9tarpVUDK/b58Da+sqqls3eN
buv7pr+eoZG+SrDK6nWeL3c6H5Apxz7LjVc1uTIdsIXxuOLYA4/ilBmSVIzuDWfdRUfhHdY6
+cn8HFRm+2hM8AnXGXws9555KrUB5qihylGa8subX2Nn6UwNR1AkUTV74bU= ; keytag
20326
. IN DNSKEY 257 3 8
AwEAAa96jeuknZlaeSrvyAJj6ZHv28hhOKkx3rLGXVaC6rXTsDc449/cidltpkyGwCJNnOAl
FNKF2jBosZBU5eeHspaQWOmOElZsjICMQMC3aeHbGiShvZsx4wMYSjH8e7Vrhbu6irwCzVBA
pESjbUdpWWmEnhathWu1jo+siFUiRAAxm9qyJNg/wOZqqzL/dL/q8PkcRU5oUKEpUge71M3e
j2/7CPqpdVwuMoTvoB+ZOT4YeGyxMvHmbrxlFzGOHOijtzN+u1TQNatX2XBuzZNQ1K+s2CXk
PIZo7s6JgZyvaBevYtxPvYLw4z9mR7K2vaF18UYH9Z9GNUUeayffKC73PYc= ; keytag
38696
```

Open Source Software: PowerDNS

Additionally you can run

```
rec_control get-tas
```

The output should look like this

```
Configured Trust Anchors:
.
  20326 8 2
     e06d44b80b8f1d39a95c0b0d7c65d08458e880409bbc683457104237c7f8ec8d
  38696 8 2
     683d2d0acb8c9b712a1948b27f741219298d0a450d612c483af444a4c0fb2b16
```

When you restart your resolver, please check for a message saying something similar to

```
you need to update package with trust anchor
```

In that case you should strongly consider changing to an updated version of Knot Resolver. The new key is shipped with Knot since summer 2024.

ICANN Webpage and Social Media Links


icann.org


@icann


facebook.com/icannorg


youtube.com/icannnews


flickr.com/icann


linkedin.com/company/icann


instagram.com/icannorg