The ISP Column
A monthly column on things Internet

May 2026
Geoff Huston

Rolling the Root Key

In cryptography, no cryptographic key is eternal. In the world of cryptography *breaking* a key is not a computationally impossible problem – it's just a computationally infeasible problem! What that means is that in practical terms it's infeasible to assemble a large enough pool of processing capability and have adequate time to try every possibility to derive the private key value. But if a key is used for an extended period of time, then a potential attacker has the same extended time to attempt to break the key. Limiting the time a key is used and limiting the time that you require the encrypted material to be considered a secure secret can help in ensuring the secrecy of the material within those bounds. The outcome is that cryptographic material, namely cryptographic keys, algorithms and key sizes, need to be regularly reevaluated and revised in light with the evolution of computational capacity and the duration of the intended secrecy of the material. A cryptographically acceptable algorithm and key should be incapable of being broken using available (and projected available) computational resources over the period of time that you wish to maintain integrity of the information that has been digitally signed or encrypted.

These days this requirement to factor in projected computational capacity is perhaps more challenging than ever before. If we are looking to protect the key used to encrypt information for a period of 20 years, which is apparently a common lifetime for secrets, then we need to include the consideration of whether *quantum computers* will form part of the arsenal of accessible computation capability in that period. If the answer is "yes", then immediately we are forced into the consideration of how to transition to so-called *post-quantum* cryptographic algorithms immediately. If, on the other hand, the intended lifetime of a secret is 5 years or less, then the likelihood of cryptographically relevant quantum computers being constructed and used in production environments is quite remote.

For DNSSEC and most of the DNS infrastructure the algorithms used by DNSSEC do not have to meet these tougher past-quantum cryptographic standards at present, as the intended lifetime of use of these keys is generally of the order of several months, or a year or two at most.

The single exception to this observation is the root key of the DNS, the Root Key Signing Key (KSK), where the first KSK was in service for 8 years and its successor has now been in service for just under 8 years. The key lifetime for the root KSK is because there is no convenient way to introduce a new KSK value and have it rapidly integrated into the trust anchor sets of all DNSSEC-validating DNS resolvers. The chosen solution is to use this KSK for extended periods and take more time in the step of introducing new keys into the root zone before their use. If such extended KSK key lifetimes are a common practice for the DNS Root Zone administrators at the IANA, then the prospect of shifting to

significantly larger post-quantum algorithms and keys is getting closer and closer!

What does this general consideration of the transient nature of cryptographic keys imply for DNSSEC? It means that all cryptographic keys used in the DNSSEC framework should be rolled regularly. What "regularly" means can be interpreted differently by different folk, but the basic consideration is that the chosen algorithm and key length should be resistant to attack over a period that minimally spans the lifetime of use of the key. The other consideration is that key rolls should be undertaken on a regular basis so that the operators of the DNS zone have regularly refreshed procedural familiarity with the key roll process.

Rolling a DNSSEC key every week is probably too short an interval, but leaving a cryptographic key in place for, say, twenty-five years is equally unwise. However, there is no single interval that applies to all DNSC zone administrators. A good operational practice is to introduce a new key into the zone's DNSKEY record, and then leave it in place for at least one DNS cache lifetime interval before switching over to the new key. It is also an option use a new key immediately, by choosing to publish multiple RRSIG records in the signed zone, which each Resource Records being signed by each key, but care should be taken when adding additional material to DNS responses that they do not bloat beyond a useful UDP packet fragmentation threshold of 1,210 bytes in size. It's also useful to bear in mind that the cache lifetimes defined in a zone are merely guidelines and recursive resolvers have been observed to apply their own lifetimes to cached DNS material, so a conservative approach should be taken to defining a suitable introduction period before deleting the old key and all records signed by that key.

RFC 6781 (https://www.rfc-editor.org/rfc/rfc6781.html), "DNSSEC Operational Practices, Version 2", from December 2012 and RFC7583 (https://www.rfc-editor.org/rfc/rfc7583.html) "DNSSEC Key Rollover Timing Considerations" from October 2015 both provide essential further reading on this topic.

However, an entirely different set of considerations apply to rolling the apex key, the Key-Signing Key (KSK) used by the root zone of the DNS. There is no upper-level context in which the KSK is published that can determine its acceptance. Instead, the new KSK must be introduced into the root zone and signed by the current KSK (*"old" signs "new"*). Then an extended period is required to allow DNSSEC-validating clients sufficient time to pick up and trust the incoming key. RFC 5011 (https://www.rfc-editor.org/rfc/rfc5011.html) "Trust Anchor Update" contains some guidance for this process. It states: "The add hold-down time is 30 days or the expiration time of the original TTL of the first trust point DNSKEY RRSet that contained the new key, whichever is greater. This ensures that at least two validated DNSKEY RRSets that contain the new key MUST be seen by the resolver prior to the key's acceptance."

The stated objective for the operational lifetime of the Root Zone KSK is nominally five years, but the practical experience has extended this period, with the first roll undertaken on the 11th October 2018. An analysis of this key roll can be found at https://www.potaroo.net/ispcol/2018-11/kskpm.html.

The next KSK key roll is underway as of May 2026. This is not a change of algorithm, but simply a change of the private key value. The incoming KSK (KSK-2024) was published on the IANA web site in July 2024 and added to the root zone's DNSKEY Resource Record in January 2025. The IANA plans to roll the KSK in October 2026, when it will be used for generating the root zone's DNSKEY Resource Record digital signature in place of the existing key, KSK-2017.

How are we going with this KSK roll? Have all DNSSEC-validating recursive resolvers learned of KSK-2024 by now and incorporated this key into their trust anchor set?

How can we measure this behaviour?

Trust Signal - RFC 8145 measurement

There are two techniques we can use to peer into the Internet's DNS infrastructure and find out which DNS resolvers have added KSK-2024 to their local trust anchor set.

The first is described in RFC 8145 (https://www.rfc-editor.org/rfc/rfc8145.html) "Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)". In this technique the resolver embeds the key tags of its trusted keys in queries that are sent towards the authoritative servers for the root zone. One way is to place the tag values of all trusted keys into a query name (i.e. a query name of "_ta-4f66-9728" indicates that the resolver trusts keys with tag values of 20326 and 38696). An alternate method is to embed the trust key tag values into an edns-key-tag option in queries.

This information is visible in the log of queries managed by the operators of root servers. An analysis of this RFC 8145 signal received by a root server operator is shown in Figure 1.

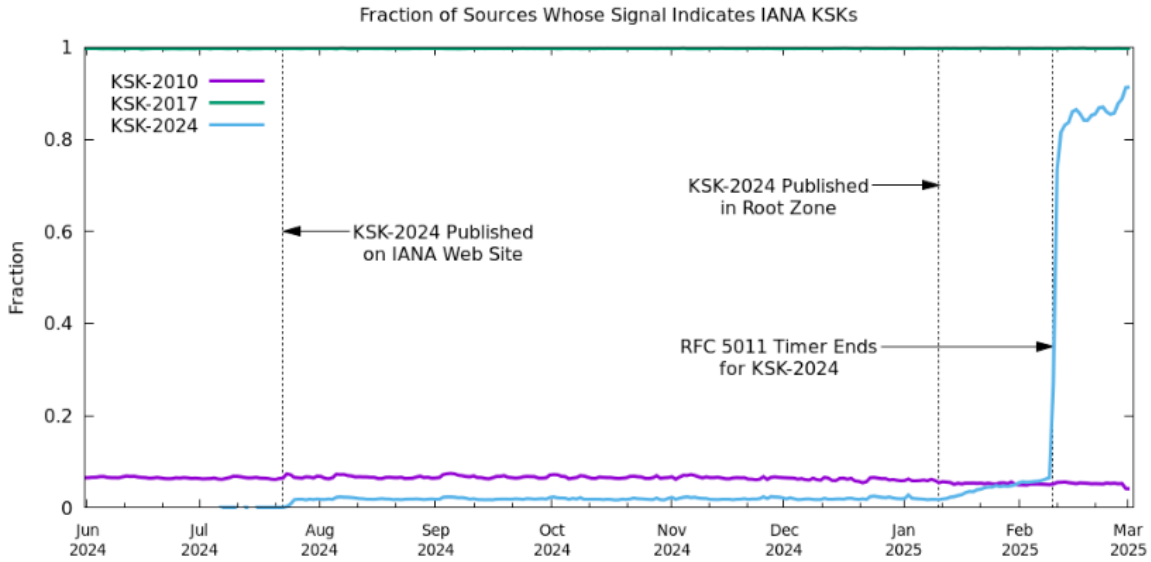

Figure 1: A sampling of resolvers and which KSK(s) they have configured in their trust anchor, as of March 3, 2025. (From https://blog.verisign.com/security/2024-2026-root-zone-ksk-rollover-initial-observations/)

As the Verisign blog article notes about this data, all of the reporting resolvers have KSK-2017 configured as their trust anchor (green line at the top of Figure 1). We expect it to remain like that until after KSK-2017 is revoked, scheduled to occur in early 2027. At the time of the blog posting (March 2025) there were still quite a few resolvers (~0.5%) that trusted KSK-2010 (purple line in Figure 1), even though it was revoked in 2019. Perhaps due to its lingering presence in operating system software updates which have trust anchors contained in the software distribution. There was a small population of resolvers that began including KSK-2024 in their trust anchor configuration in July 2024, right after the key was published in the Internet Assigned Numbers Authority (IANA) trust anchor file on their website. The large jump occurred 30 days after the key's inclusion in the root zone's DNSKEY Resource Record. This behaviour is consistent with the timeline as specified by RFC 5011. The small rise in this initial 30-day period appears to be related to a small number of DNS resolvers who configured their trusted KSKs manually.

What the data in Figure 1 does not show is the population of users who are using these recursive resolvers. Obviously, some DNSSEC-validating recursive resolvers are used heavily, sometimes by millions of end users, while others, including the one on my local laptop, is used by just me! When the March 2025 figures show that up to 90% of reporting resolvers have added KSK-2024 to their trust

anchor set, that does not readily map to an estimate of the end user population that is served by these recursive resolvers, nor the population of end users who may be affected by an immediate roll of the KSK.

Also, this data does not show the population of DNSSEC-validating recursive resolvers who are not using this trust anchor signalling mechanism. This implies that it's challenging to place this data in the larger context of all DNSSEC-validating recursive resolvers, and what this implies for end users.

Key Sentinel - RFC 8509 Measurement

RFC 8509 "A Root Key Trust Anchor Sentinel for DNSSEC" proposes a different approach where the signal of a recursive resolver's trusted keys is folded back and sent to the querier rather than onward towards a root zone server. The approach involves a left-most label of a DNS query name, and requires the DNS resolver to recognise this label and generate a response based on the resolver's set of trusted keys, as shown in Table 1.

Label	Key is Trusted	Key is not Trusted
root-key-sentinel-is-ta-<key-tag>	return original answer	return SERVFAIL
root-key-sentinel-not-ta-<key-tag>	return SERVFAIL	return original answer

Table 1 – RFC 8509 Measurement responses

We use APNIC's ad-based measurement system to pose these queries to a collection of end users every day. What we are after here is not to identify individual validating recursive resolvers, but to quantify the population of end users who are located exclusively behind DNSSEC-validating resolvers where the resolver has so far failed to trust KSK-2024.

The test we are using has three queries, as shown in Table 2. A resolver that is configured to perform DNSSEC validation and supports this Root Key Trust Anchor Sentinel mechanism will provide DNS responses, also shown in Table 2.

DNS label	Response
https://root-key-sentinel-not-ta-20326...	SERVFAIL
https://root-key-sentinel-is-ta-20326...	Normal validated response
https://root-key-sentinel-is-ta-38696...	SERVFAIL **OR** normal validated response

Table 2 – RFC 8509 Measurement responses for KSK-2024

The first two tests are intended to determine if the resolver supports this Root Key Trust Anchor Sentinel mechanism by testing if it trusts KSK-2017 (key-tag 20326). If it does not it will return a validated response in both cases. If it does report responses as per Table 2 then we can classify the sample point as a "reporting user". The third query is intended to determine if the resolver has loaded KSK-2024 (key-tag 3896) into its local trust anchor set.

Results

The daily results of this measurement of the adoption of KSK-2024 as a Trust anchor are shown in Figure 2.

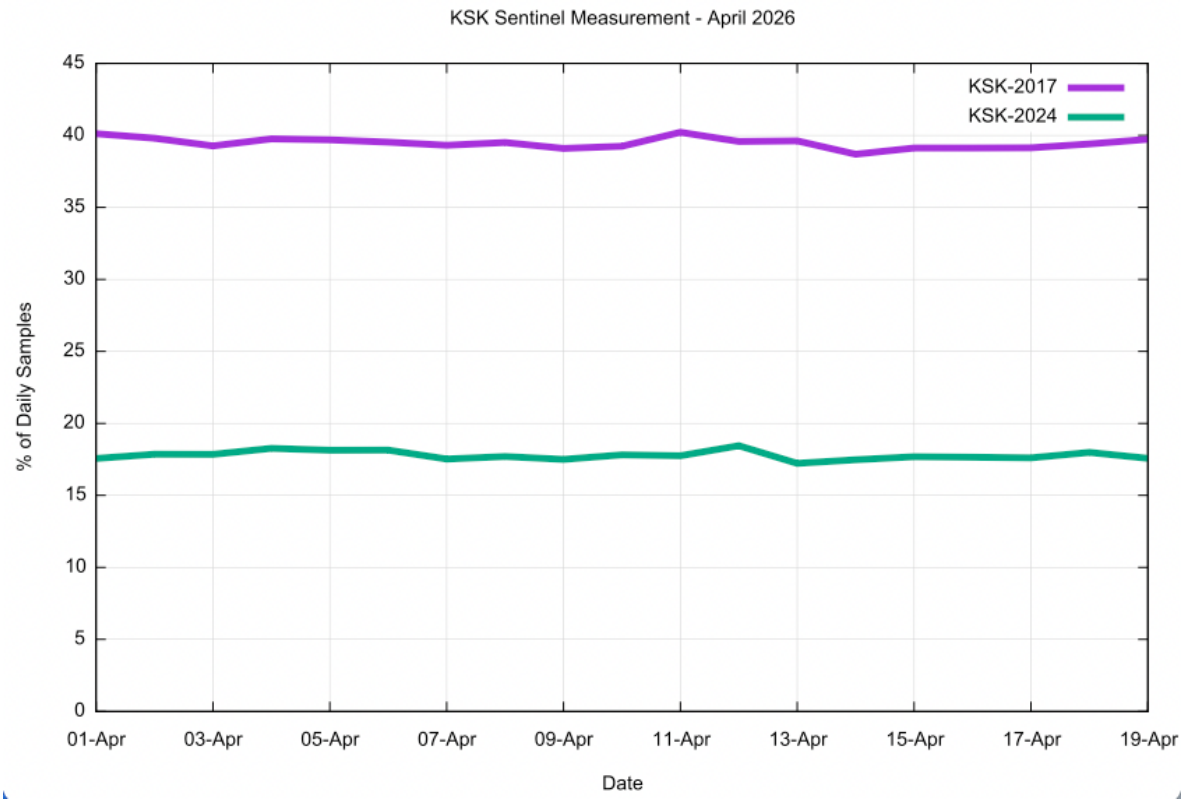

Figure 2: Daily Measurement of Adoption of KSK-2024 as a Trust Anchor using RFC 8509 Sentinel during April 2026

The time span of this measurement does not include the introduction of KSK-2024 into the root zone DNSKEY resource record in January 2025.

The Y axis of this measurement is unlike that of Figure 1. This is not a count of recursive resolvers, nor a ratio of reporting DNS resolvers, but a ratio of reporting users who are located behind DNS resolvers that are observed to perform DNSSEC validation. The test for DNSSEC validation has been described previously (https://www.potaroo.net/ispcol/2023-10/measure-dnssec.html). The daily counts of the various DNSSEC capabilities of tested end users is shown in Figure 3.

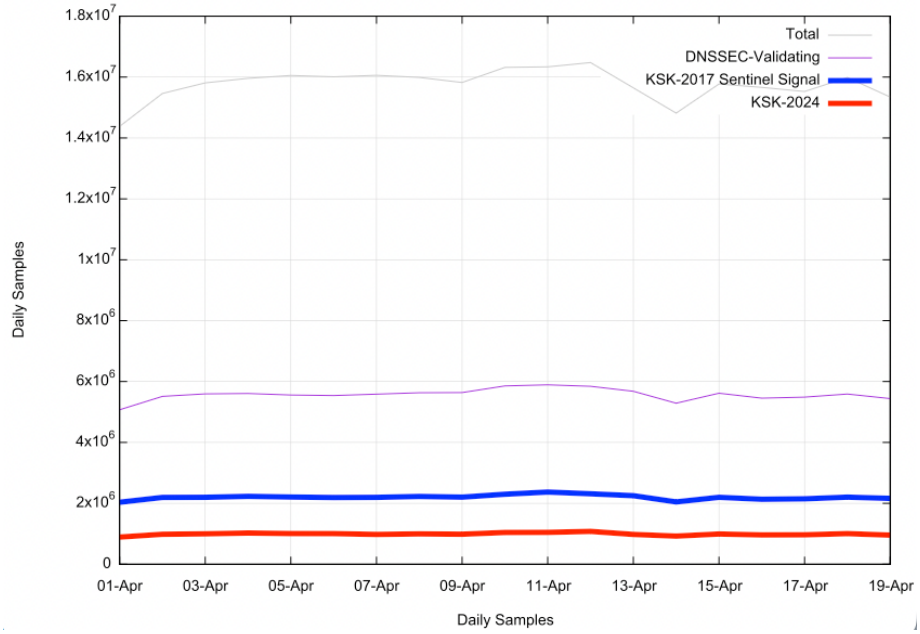

Figure 3: Daily Measurement Count of DNSSEC Validation and KSK Sentinel Signals

Discussion

This is certainly not the result that we had anticipated. The data in Figure 1 using the RFC 8145 signal indicated that reporting resolvers had followed the procedure specified in RFC 5011 and had adopted KSK-2024 in March 2025. But the RFC 8509 sentinel measurement in Figure 2 shows a completely different story.

In April 2026, some 13 months after the 30-day introduction timer had finished for KSK-2024, the data shows that under 20% of users who were behind DNSSEC validating resolvers had added KSK-2024 to its local trust anchor set.

Not all recursive resolvers that show DNSSEC validation behaviour also show that they support the Key Sentinel mechanism. We are selecting only those samples where the results of the two tests, *is-ta-20326* and *not-ta-20326* show a completed DNS resolution in the first case and an unsuccessful DNS resolution attempt in the second case.

There are some challenges in this measurement, which are worth noting. The advertisement framework presents two URLs to the tested used. Here's an example:

```
https://root-key-sentinel-not-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-
i00000000-10.ap.dotnxdomain.net/1x1.png?udd44ca59-s1777599902-i00000000.ap.kn
https://root-key-sentinel-is-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-
i00000000-30.ap.dotnxdomain.net/1x1.png?udd44ca59-s1777599902-i00000000.ap.ko
```

A DNSSEC-validating DNS recursive resolver that supports RFC 8509 will return different responses to the resolver's attempt to resolve these names. For the *is-ta-20326* query we receive a normal NOERROR DNSSEC-signed response, as follows:

```
$ dig +dnssec A root-key-sentinel-is-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-
i00000000-30.ap.dotnxdomain.net @localhost

; <<>> DiG 9.20.21 <<>> +dnssec A root-key-sentinel-is-ta-20326.0ds-udd44ca59-c13-a04C5-
s1777599902-i00000000-30.ap.dotnxdomain.net @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57993
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;root-key-sentinel-is-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-i00000000-
30.ap.dotnxdomain.net. IN A

;; ANSWER SECTION:
root-key-sentinel-is-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-i00000000-
30.ap.dotnxdomain.net. 60 IN A 172.104.187.92
root-key-sentinel-is-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-i00000000-
30.ap.dotnxdomain.net. 60 IN RRSIG A 13 5 60 20260529015205 20260501005205 13681 0ds-
udd44ca59-c13-a04c5-s1777599902-i00000000-30.ap.dotnxdomain.net.
kZtSMNqUAIq10zi+nP+pJRFGIkicH0HF8wFa9G+BNR2kn4dImK7OCdR7 LBFk4JzC58yv1Bz2xnEVnfnIVjh5SA==
```

For the second query we see:

```
$ dig +dnssec A root-key-sentinel-not-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-
i00000000-30.ap.dotnxdomain.net @localhost

; <<>> DiG 9.20.21 <<>> +dnssec A root-key-sentinel-not-ta-20326.0ds-udd44ca59-c13-a04C5-
s1777599902-i00000000-30.ap.dotnxdomain.net @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 44304
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
```

```
;root-key-sentinel-not-ta-20326.0ds-udd44ca59-c13-a04C5-s1777599902-i00000000-
30.ap.dotnxdomain.net. IN A
```

It is not apparent that these DNS results will make it back to the users who are running the tests, as we cannot directly interrogate the user's application. Normally, we would look at our Web Server's access logs, and if the DNS resolution succeeded than we would expect a fetch for that named web object to be logged.

But this URL format presents us with a problem. We use a uniquely generated label within the DNS name to ensure that the user's efforts to resolve the name cannot trigger a response from a resolver's cache. Normally the unique name element is the left-most label, allowing us to use a wildcard domain name certificate to support the TLS handshake as part of the HTTPS object retrieval. However, in this case the unique label component is not the left-most label so we cannot a conventional wildcard certificate.

> DNS wildcards can encompass up to 255 levels of delegation, which wildcards used in the Subject name of Domain Name certificates can only encompass a single (left-most) label.

In this case we can turn to a log of packet captures, and use the observation that the SNI field is in the clear in a TLS handshake (if no encryption key is provided) and the inspection of the SNI informs us of the intended URL being fetched using TLS, even if the TLS handshake fails.

What the results displayed in Figure 2 are saying is that slightly less than one half of the users who appear to be using DNSSEC-validating resolvers that recognise the RFC 8509 sentinel label for KSK-2017 are reporting that they also trust KSK-2024 via the same mechanism. In other words, of the users who were behind resolvers who appeared to support the RFC 8509 sentinel mechanism (by the virtue of a failure to provide a resolution to the query https://root-key-sentinel-not-ta-20326...) only slightly fewer than one half of these users completed a DNS resolution of the query https://root-key-sentinel-is-ta-38696... This seems to be an improbable result.

Let's review exactly what we are measuring here.

The first condition is that the user is using one or more DNS recursive resolvers, **all** of which perform DNSSEC validation. We test for this by passing the user a URL for a dual-stack hosted web object where the DNS name for this object is DNSSEC-signed, but the signature cannot be validated. In this case the DNSSEC-validating resolver will return a SERVFAIL error code to the local stub resolver, which will try the next configured recursive resolver. If all of these resolvers that are called to resolve this DNS name are performing DNSSEC validation, then the DNS name cannot be resolved and the web object will not be retrieved.

We then pass the user two further tests, both using validly DNSSEC-signed DNS names, with the pair of *is-ta* and *not-ta* for the KSK-2017 key tag value (2036). If all the recursive resolvers that process the user's query support the RFC8509 sentinel mechanism, then we should see the *is-ta* object successfully resolved, and we then would expect to see an attempt to perform a web fetch for this URL. Similarly, we should see all the recursive resolvers return SERVFAIL for the *not-ta* object and no web fetch will be attempted.

The final test is for *is-ta* with the key tag value 38696, which should be resolved successfully if this key has been integrated into the resolver's trust anchor set.

Why are these results not showing a much clearer picture of DNSSEC-validating resolvers adding KSK-2024 into their trusted key set?

Part of the issue here is that SERVFAIL is a challenging response code. It does not say that the queried name does, or does not, exist. SERVFAIL responses are not normally cached (except for some resolvers that cache the response in the short term of 1 -2 seconds). SERVFAIL is an implicit invitation for the resolver to attempt the same query to a different recursive resolver. A recursive resolver will conclude its query (and return a SERVFAIL response code) when all possible authoritative servers have been queried, or when the resolver has exceeded its internal time or query limit to attempt to resolve the name. A stub resolver will conclude its query when all configured recursive resolvers have returned SERVFAIL, or when the stub resolver has exceeded its time or query limit to attempt to resolve the name.

Can we refine the test to see if the signal can be improved? On the 20th April another URL was added to the test set, `root-key-sentinel-not-ta-38696`. Now the test for the resolver trusting KSK-2024 is the successful resolution of the `is-ta-3896` domain name and the failure to resolve the `not-ta-3896`. The change in results from this additional test is shown in Figure 4.

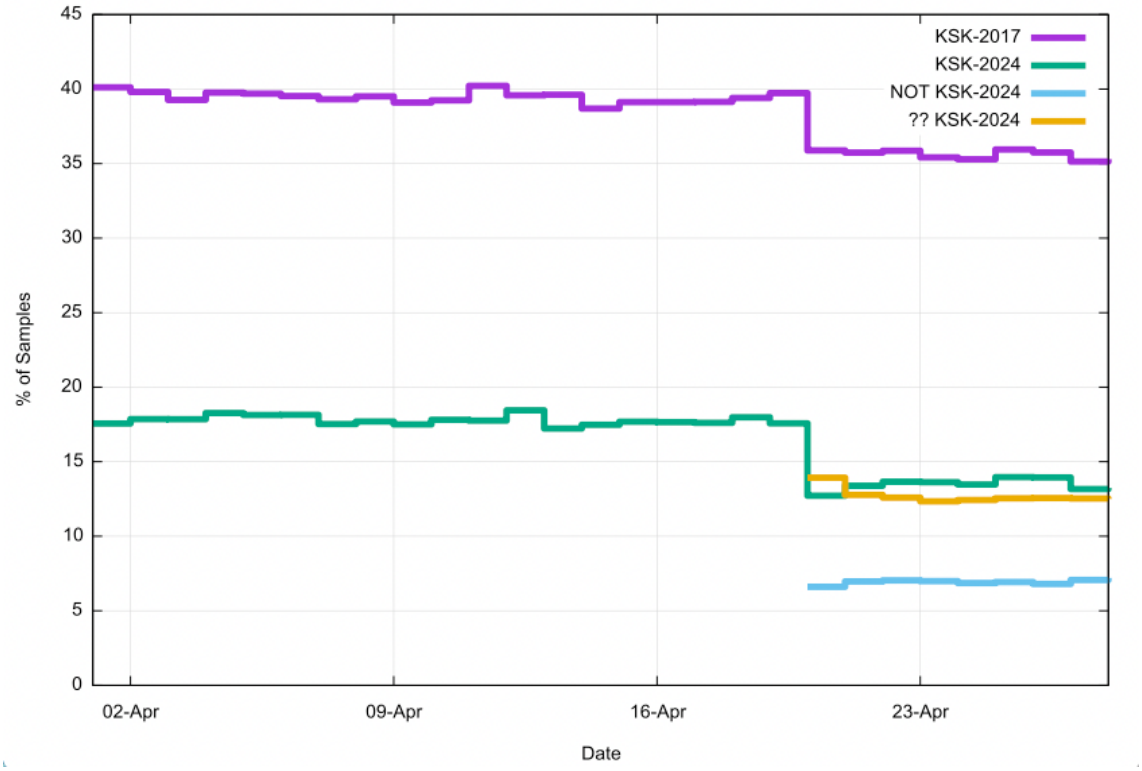

Figure 4: Daily Measurement of Adoption of KSK-2024 as a Trust Anchor using RFC 8509 Sentinel during April 2026

While the test for KSK-2017 has not changed, the proportion of samples reporting that they do not trust KSK-2017 has dropped. It is likely that the addition of a URL into the measurement script has caused a higher dropout rate in the transition from DNS resolution to web object fetch within the tested users' browsers.

The test for KSK-2024 is now a test for a successful resolution of `is-ta-3896` and a failure to resolve `not-ta-3896`. With this change the proportion of reporting samples reporting trust in KSK-2024 has fallen from 17% of DNSSEC-validating samples to 12%. There is now a visible opposite signal, reporting that KSK-2024 is not trusted, which accounts for 7% of DNSSEC-validating samples. There is significant noise in this signal, and the case called "?? KSK-2024" is that where the sample has a clear signal that KSK-2017 is trusted, but both DNS names "`is-ta-3896`" and "`not-ta-3896`" successfully resolve, indicating that at least one of the user's resolvers does not support the RFC 8509 Sentinel mechanism, which contradicts the supposedly clear signal of trust in KSK-2017.

The sample counts for this measurement for April 2026 are shown in Figure 5.

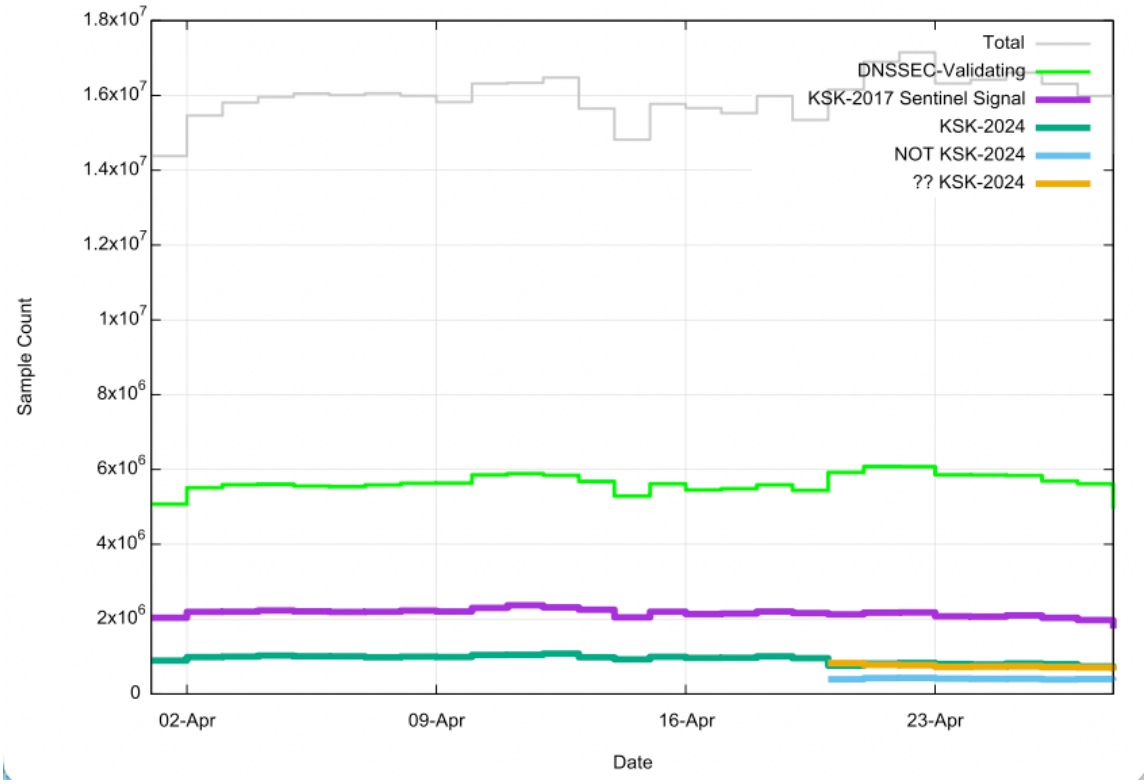

Figure 5: Daily Measurement of Adoption of KSK-2024 as a Trust Anchor using RFC 8509 Sentinel during April 2026

Conclusions

The DNS system is designed with resilience in mind, and the name resolution function is designed with a clear preference to perform a diligent search for a clear response (NOERROR or NXDOMAIN) over a potentially faster but indeterminate response (SERVFAIL). Setting up a measurement framework that relies on a SERVFAIL signal, which is used by the RFC 8509 Sentinel Mechanism, is a case of inviting a large quantity of measurement uncertainty into any measurement that uses this approach.

Secondly, the DNS has no equivalent of a trace for queries and responses. When a user makes a query to their locally configured recursive resolvers they have no a priori knowledge where and how their query may be forwarded within the DNS "cloud", and when a response is provided there is no trace of where the query was forwarded, where response was generated, and which resolvers handled the response in delivering this response back to the original querier. The internal behaviour of the DNS is quite simply opaque, and it seems to me that it is deliberately so. This makes isolating the behaviours of individual DNS resolvers quite challenging.

The next issue is the somewhat unclear measurement objective in this case. Are we looking for individual DNS recursive resolvers who perform DNSSEC validation, but are failing to follow the procedure described in RFC 5011 to invest trust in a new KSK that has been validly signed by the current KSK? Or are we trying to answer a completely different question, namely, to estimate the negative impact of the planned October 2026 KSK roll and provide an estimate of the user population who may be impacted by this roll?

Both measurement techniques, RFC 8145, Trust Anchor Signalling and RFC 8509, Trust Anchor Sentinel, have their problems and measurement uncertainties.

In the case of RFC 8145 only Root Server operators can collect this signal, and attribution to an individual DNSSEC-validating recursive resolver is potentially distorted when the DNS forwarding path between the resolver and a root nameserver contains forwarders and other recursive resolvers. The measurement also cannot determine the number of resolvers who do not support RFC 8145 signalling, so if the question is to determine the extent of DNSSEC-validating resolvers that are not trusting KSK-2024, then

this data does not provide a clear answer. And of course, the user impact question is not answerable in this framework, as this measurement does not expose the population of users that depend on each resolver's behaviour.

In the case of RFC 8509 TA sentinel there is the challenge of setting up a large-scale sampling measurement, and then there is the issue of an extremely noisy signal because of the behaviour of DNS resolvers to try and find a "better" response when they receive a SERVFAIL response code. We have no idea of the number of recursive resolvers that support this RFC 8509 mechanism. The result is that we are moderately confident when we assert that some 35% of the Internet's user base appear to be located behind DNS recursive resolvers that perform DNSSEC validation, and we can assume that most implementations of DNSSEC validation in recursive resolvers will "learn" the introduction of a new KSK as per RFC 5011. However, the data we have gathered from our measurements of this RFC 8509 behaviour give us no clarity in making this assumption. We just don't know.

We will keep this measurement running through October 2026, but frankly I don't expect any tangible improvement in the quality of the results!

Necessarily, we are going to have to trust that implementations of DNSSEC-validating recursive resolvers operate with an accurate implementation of the relevant RFCs relating to introducing a new KSK when we roll the KSK in October 2026.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston AM, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net