Observing DNSSEC Key Lifecycles

ICANN Office of the Chief Technology Officer

Edward Lewis
OCTO-035
18 July 2022

ICANN

TABLE OF CONTENTS

This document is part of ICANN's Office of the Chief Technical Officer (OCTO) document series. Please see the OCTO publication page for a list of documents in the series. If you have questions or suggestions on any of these documents, please send them to octo@icann.org.

Executive Summary

The way in which an operator of a DNS zone, whether a top-level domain name (TLD) or not, manages the cryptographic keys used in Domain Name System Security Extensions (DNSSEC) can reveal much about the nature of the operations and the process of technology deployment. Visualizing the management of DNSSEC keys based on published records has recently been accomplished, this report covers the basics and discusses, without naming specific operators, some specific TLD use cases. The benefit gained from these visualizations for operators is the ability to examine their own history, compare their operations with other zones or TLDs, and visualize the impact of changes they have made. A benefit to protocol developers is the ability to see how a protocol is deployed, how it is used, visualizing the reality of what might have been only conceptually understood about a protocol or a component of a protocol.

1 Introduction

A prominent operational component of DNSSEC[1] deployment is the managing of the cryptographic keys that are used. DNSSEC is essentially the addition of digital signatures to DNS data. Digital signatures rely on cryptographic keys that work in pairs, one of which is publicly known (for validation) and one of which is kept private (for signature generation). With public keys and signatures published in DNS data, it is possible to observe and measure how a key is used by observing records published in the DNS.

It is common practice to replace existing keys with new ones. This is known as key rollover this can be scheduled or in response to a potential compromise of private key material. Key rollover events, or key supercessions, are significant events in the management of a DNSSEC operation, complicated by DNS caching. Key rollover events provide great insight into operational practices.

A generic life cycle of a cryptographic key consists of a few stages. First is the creation of the key, followed by distribution, active use, retirement, and then elimination. DNSSEC keys follow this general pattern, with some refinement and adjustments needed to work with the DNS, specifically in how data is cached throughout the system. The creation and elimination stages of DNSSEC keys are not visible in the DNS but the rest of the stages are.

A reason for operators to follow the lifecycle of their keys is to see the result of their own actions. Having access to a history of observations, it is possible for current operators to see what happened before they came on board (the DNS registry industry has matured to the point that turnover is a consideration). It is also helpful for operators to compare local decisions to what others are doing, especially for operators facing the same conditions.

Another reason to observe and visualize DNSSEC key life cycles is to see how DNSSEC is deployed and how it is configured, to determine best practices based operational experience, and to identify what, if any, unintended steps are taken. Lessons related to producing operations-friendly protocol modifications can be learned, answering questions including: What elements work? What elements are ignored?

[1] See https://datatracker.ietf.org/doc/rfc4033/, https://datatracker.ietf.org/doc/rfc4034/, and https://datatracker.ietf.org/doc/rfc4035/

2 Expected Key Life Cycles

For any DNSSEC key pair to be useful in a top-level domain registry setting, the public key must appear in a DNSKEY resource record. This resource record is how DNS distributes the public key of the key's pair. DNSSEC allows and relies upon multiple concurrent keys, so there is a DNSKEY resource record set (with *set* being the important word) that will be of special significance when analyzing key life cycles.

There are two cryptographic roles for DNSSEC keys. One role is to generate digital signatures to validate each of a zone's resource record sets. The other role is to be registered with the zone's parent zone registry to indicate whether and how DNSSEC is configured. The former role provides the security goal of authenticating the source of the data, the latter role allows for scaling the authentication of zone administrators through the hierarchy of the namespace.

Inherent in the use of cryptographic keys is the need to change keys from time to time. With cryptography based on secrecy and the recognition that over time secrets can be leaked, forgotten or lost altogether, keys will therefore need to be changed for long-lasting and reliable operations. Changes to keys are the very reason that there is a key life cycle to visualize.

A key called a Zone Signing Key(ZSK) performs the first role described earlier of authenticating data in the zone. (While this paper used the singular term “key”, in reality DNSSEC keys are public keys, consisting of a pair of keys, one public and one private.) Most zones use one key at a time in normal operations. ZSKs are usually changed on a relatively frequent basis. During such a change, multiple keys will be seen in the relatively short time it takes for the change to complete.

A Key Signing Key (KSK) performs the latter role of authenticating a zone within the DNS hierarchy. (A KSK is sometimes called a Secure Entry Point, abbreviated SEP.) KSKs are registered with the parent, so that a Delegation Signer (DS) resource record can be published by the parent, forming the “chain of trust.” The KSK signs just one set of zone data, the zone's DNSKEY resource record set.

It is possible to use the same key for both roles, in which case the key is called a Common Signing Key (CSK). Over the course of this study, there have been rare occasions when a CSK has been used in TLDs. One country code top-level domain (ccTLD) began DNSSEC operations with a CSK but quickly converted to the KSK and ZSK split role model. More recently one operator published a CSK as an unintended consequence of actions (the unintended nature was confirmed during a face-to-face meeting with the operator in 2019). Using a CSK in parts of the DNS tree lower than TLDs may be a preferred operational posture for other parts of the DNS namespace, i.e., below the TLD structure.

2.1 Life Cycles of a ZSK

The life cycle for a ZSK is different from the life cycle for a KSK. The expected life cycle of a ZSK key pair consists of five stages, three of which are detectable in the DNS:

- Created – The key is made but is not (yet) visible in the DNS, such as in an off-net Hardware Security Module (HSM)

- Pre-view – The key is in a DNSKEY resource record before being used to generate signatures
- Active use – The key is in a DNSKEY resource record and generating signatures
- Post-view – The key is in a DNSKEY resource record after being used to generate signatures
- Discarded – The key is not visible in the DNS and has been deleted

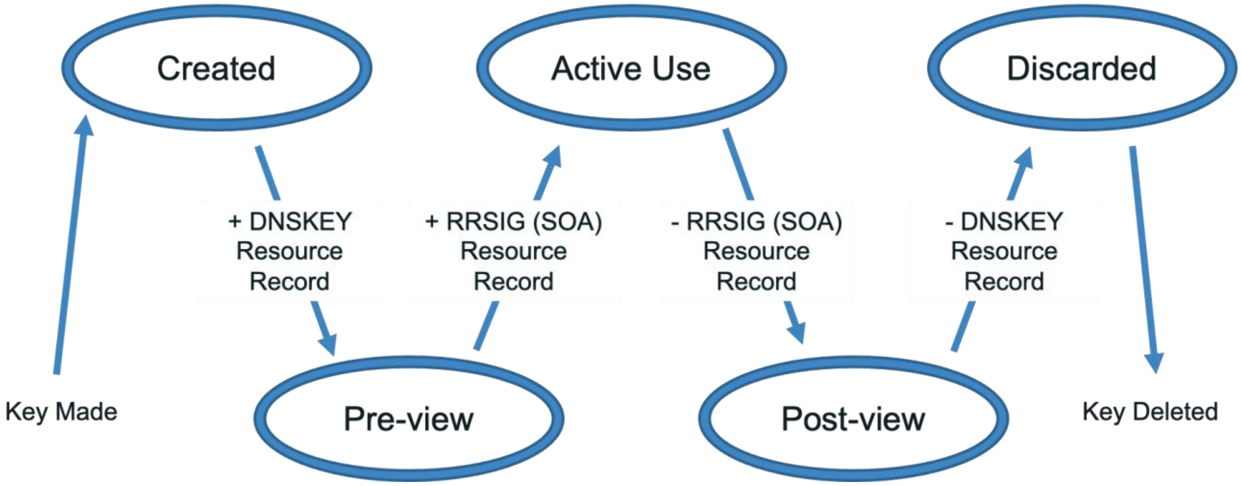


Figure 1 - Stages in a ZSK lifecycle

The need for the pre-published stage and the post-published stage (also known as retirement) is due to DNS caching. A key needs to be available while its signatures (in the form of RRSIG resource records) are published, or while those RRSIG records are held in caches, otherwise validation will fail. The length of time a key needs to be in either the pre-published or retirement stage is related to the time-to-live (TTL) settings of the zone, which for TLDs tend to range from hours to days. There will be overlap between the life cycles of a key and its successor such that at least one is always in its active state.

The most common deviation from this simple life cycle happens when the DNSSEC security algorithm is changed. This event is called an algorithm roll. When this happens, signatures from a key may appear first, due to protocol rules, before a key is published. When this happens, the signatures are not used; instead, DNS caches are prepared for the new key.

OCTO has observed that it is possible for keys to be pre-published and then withdrawn. The operator may have changed their mind about using a particular key. A key is not operationally significant when it appears in a DNSKEY resource record or is referred to by a DS resource record. The key only gains significance when there is an RRSIG resource record generated by it. But even when there is an RRSIG resource record, the key may be ignored if there are other keys in play.

2.2 Life Cycles of a KSK

The expected life cycle of a KSK key differs from the ZSK life cycle due to the interchange between the zone administrator and the administration that created the delegation for the zone, that is, the child and parent of the DNS zones. Across the TLD environment, the parent is the root zone. One of the life cycles a KSK might take is:

- Created – The key is made but not visible in the DNS, such as in an off-net HSM
- Pre-view – DNSKEY resource record before use in generating a DNSKEY signature
- Un-chained – DNSKEY resource record and generating a DNSKEY signature
- Chained (Active) –- DNSKEY resource record, signature, and a DS resource record at the parent
- De-chained – Same as the unchained in state but seen after having been chained
- Post-view – Same as Pre-view in state, but seen after having been Chained
- Revoked – The DNSKEY resource record for the key has a REVOKE bit set
- Discarded – The key is not visible in the DNS and is deleted altogether

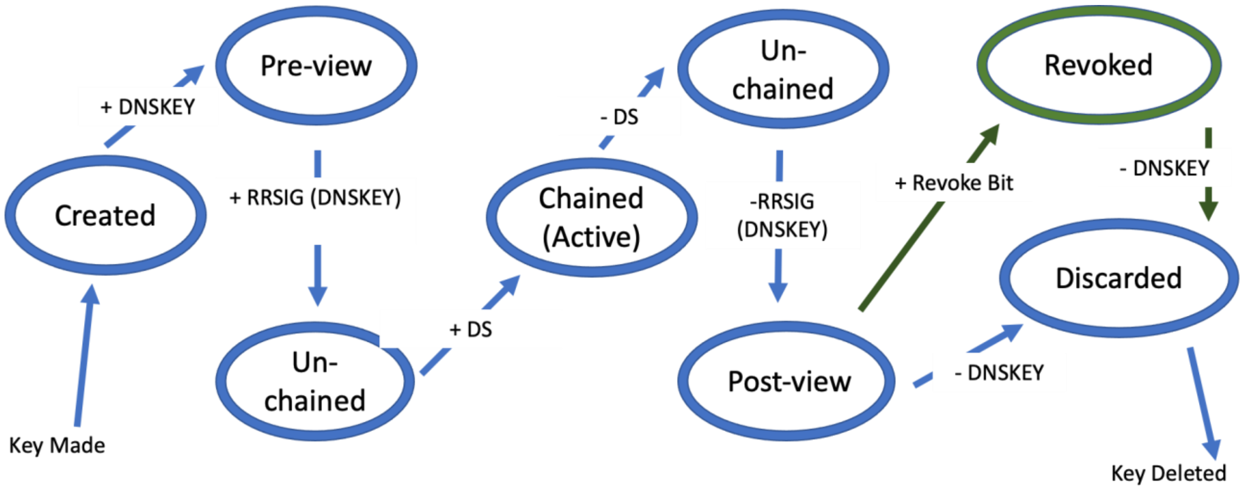


Figure 2 - One Set of States for KSK

The stage "Revoked" is rarely seen as it is a part of the Automated Updates for DNSSEC Trust Anchors process.[2] The events and the state bubble are drawn in green to distinguish it from the more common path.

A simplified version of a KSK life cycle, one that is pretty common, combines the publication and removal of the DNSKEY and the RRSIG (DNSKEY) as shown in the next figure. The Pre-view and Post-view states are not strictly necessary for the KSK. The revoked state is also not shown.

[2] See https://datatracker.ietf.org/doc/rfc5011/

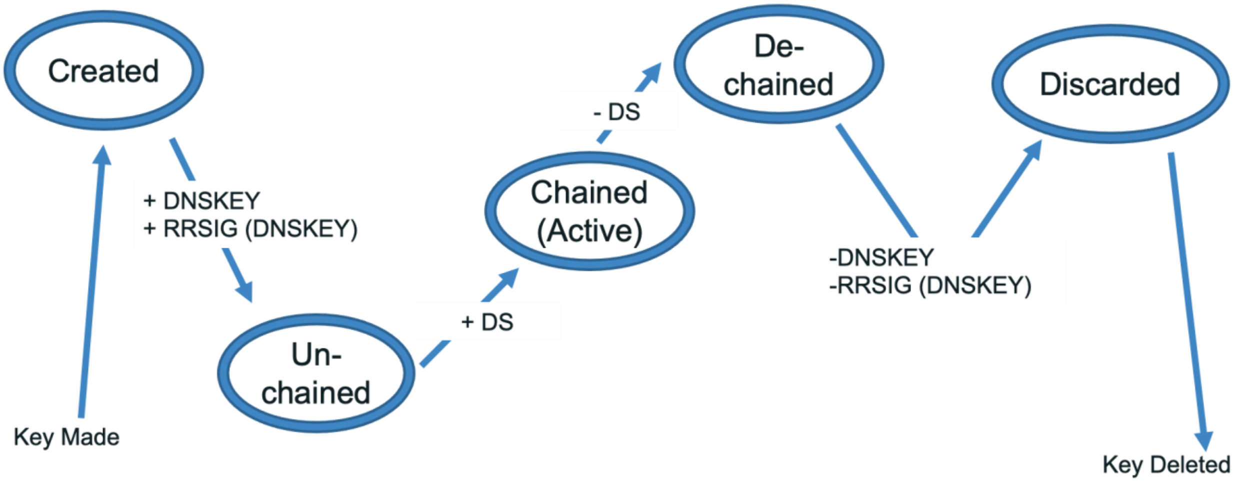


Figure 3 - Simplified Set of States for KSK

Across the observed key life cycles, some follow the pattern below, which features one more state that deserves highlighting.

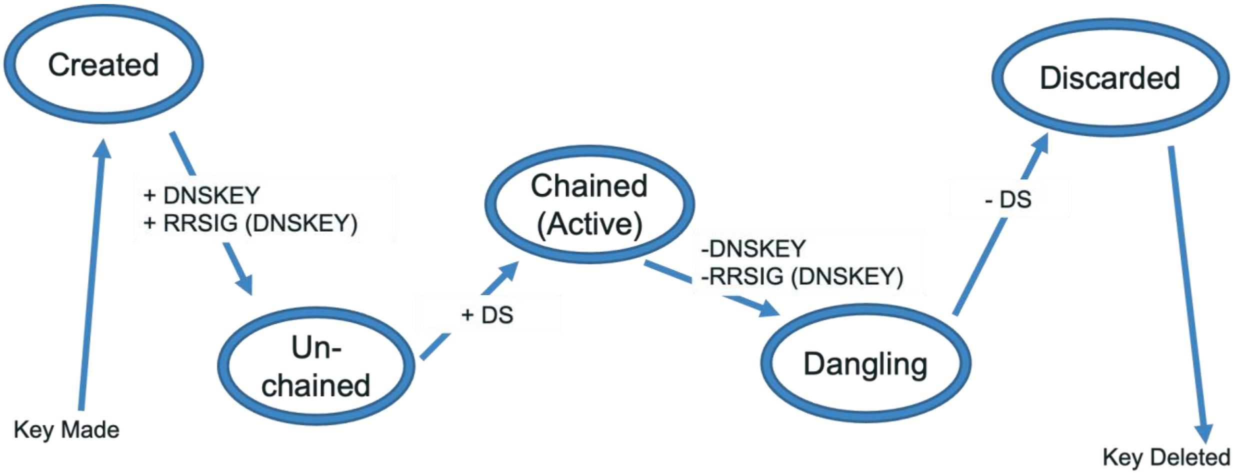


Figure 4 - KSK with Dangling State

The "Dangling" state occurs when a KSK is removed from the child zone before the DS record is removed from the parent. Why this happens is not specifically known, it could be because the operator delays the removal request to verify other steps, the removal request takes some time to process, or that the continued presence of the DS resource record was overlooked for some time. It could be argued that this is an operational error but there is little to no harm to the functioning of the protocol.

Chaining a KSK is a two-party process. In a parent-child zone situation, the child zone administrator publishes a DNSKEY resource record that is the subject of the chaining. The key is published in the DNSKEY resource record set of the child zone. The parent zone administrator publishes a DS resource record (or records) referring to that key in a DS resource record set owned by the child's name with the parent also producing and publishing a signature for the DS resource record set. When these records are in place, the key is chained from the parent to the child.

With chaining requiring an interaction between two independent operators, child operators have chosen different strategies to anticipate delays by the parent. The impact of this, on life cycles, is that there are times when a key may be seen in a DS resource record before or after being seen as active or chained. These situations are acceptable in the protocol when there are other chained keys in place, but might appear as errors when looking at a key's life cycle in isolation.

3 How a Key Life Cycle is Visualized

Based on the contents of the four categories of resource records, the state of any key is determined for a given time. By examining these states over time, the life cycle appears.

The four categories of records are:

- DNSKEY resource record – a key is published
- DS resource record – a key is recognized by the parent, a candidate to be "chained"
- RRSIG resource record covering DNSKEY – signing the key set
- RRSIG resource record covering start of authority (SOA) – signing zone data sets

ICANN's historical record[3] containing information on these records at the granularity of one day, with a day being defined by the UTC time standard. The data set has information for each day (with no gaps) and is built from multiple observations per day. This data set continues to grow. If any record is displayed on a day, it is shown as being present all day even if the record appears halfway through the day. On days when an operator makes a change, it might appear that the outgoing and incoming records are displayed together when in fact one replaced the other during a mid-UTC day.

For a ZSK, the key usually appears first in a DNSKEY resource record, establishing the pre-published state. An assumption is made, which holds true in the TLDs but may not be generally true, that the key used to sign the SOA resource record set is used throughout the zone. This is based on assuming an operator will choose to run as simply as possible, without dedicating different keys to different record sets, which is allowed by the protocol. In addition, each change to a zone requires a change of the zone's serial number resulting in a new signature, which means that the key used in the signature is the current key.

For a KSK, the difference is the use of the RRSIG resource records covering the DNSKEY resource record, to determine the active KSK(s). The presence of a DS resource record (in the parent zone) determines whether a KSK is chained or not.

3.1 A Key's Metadata

For visualizations a key is identified by the key tag and is a five-digit number. A key tag need not be unique but usually is at any given time. There has been one case seen since 2010 with two concurrent keys having the same key tag as well as the same DNSSEC security algorithm.

The DNSSEC security algorithm, which is a DNSSEC protocol identifier that indicates the cryptographic algorithm of the key as well as a hashing function used to generate the digital signature, is presented via background shading in the figures below. For keys based on the

[3] See https://observatory.research.icann.org/tld-apex-history/

RSA protocol, the length of the key (in bits) is included. For other protocols, the length of the key is always the same (according to the algorithm).

There is interest in seeing when and how the DNSSEC security algorithm is changed by an operator. Although the protocol was developed thinking multiple DNSSEC security algorithms would be used, reality shows that one is used at a time due to message size considerations, and again, simplicity. Due to this interest, the visualizations try to indicate a keys' Algorithm and, if applicable, key length.

3.2 Unexpected Key States and Operational Incidents

When observing the life cycle of a key, a key may enter an unexpected state, that is, not fitting into a usual pre-publish, active use, retirement sequence. For the sake of the observations, a color is used to draw visual attention to this in the expectation that there might be interest in further investigations. The choice of color is not intended to indicate an operational incident, because oftentimes, one key in an unexpected state does not contribute to an incident resulting in an operational abnormality.

Due to any introduced security extension naturally causing brittleness in an operating system, DNSSEC built in features to enhance, as much as possible, resiliency. Resilience means to retain functionality in the face of adverse conditions, such as a misconfigured key or signature. DNSSEC requires, for validation, one secure chain of trust to a trusted point. There may be many broken chains alongside one working chain, but one working chain is sufficient.

Whenever a key is observed as being in an unexpected state, look for other keys existing at the same time that could be used to form a secure chain, before concluding there was an operation incident. Historically, there usually is.

4 The Root Zone

This section describes some aspects of how the KSK and ZSK keys have been managed in the root zone, using that as a setting to walk through the visual charts.[4,5]

The global public DNS root zone began DNSSEC operations in the middle of 2010. The data set driving these charts begins later, the first keys shown were already in place. Given the passage of time, this gap in time coverage grows less significant, unless a study focuses only on the early days of DNSSEC.

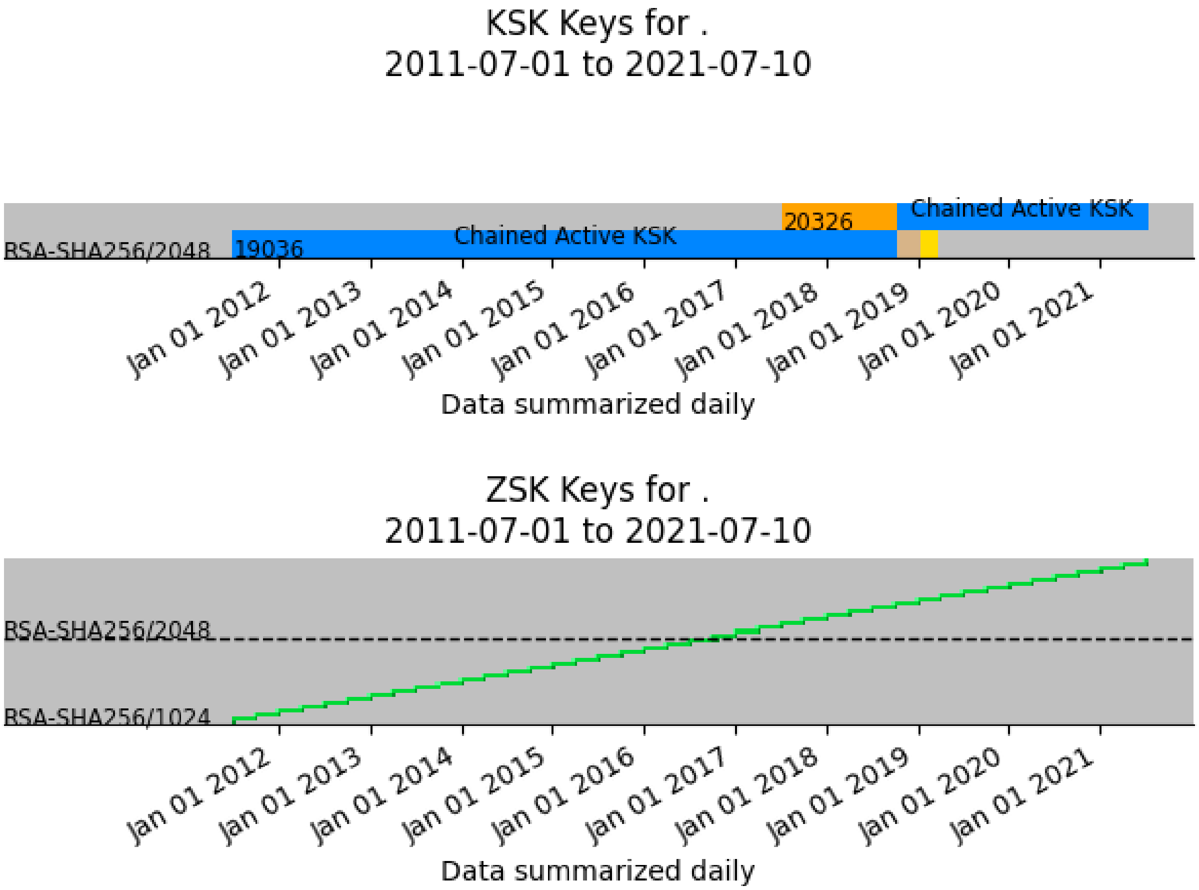
Different colors used to signify states of a key's life cycle. For the expected states of a KSK, shades of blue are used, from lighter to darker. For the expected states of a ZSK, shades of green are used, from lighter to darker. Other colors are used primarily to gain attention, not to signify an alarm, warnings, or error conditions.

[4] See https://www.iana.org/dnssec/procedures/ksk-operator/ksk-dps-20201104.html

[5] See https://www.iana.org/dnssec/procedures/zsk-operator/dps-zsk-operator-v2.1.pdf

4.1 10-Year History

These initial charts show the complete *recorded* history of the KSK and ZSK keys for the root zone. The KSK has been changed once in that time and the ZSK keys have been changed quarterly. The impact on the visuals is that there are 40 ZSK lifetimes to draw, which makes it hard to see at this time scale. To solve that, later visuals will narrow the timeline.


KSK Keys for .
2011-07-01 to 2021-07-10

ZSK Keys for .
2011-07-01 to 2021-07-10

Figure 5 – 10 Years of the Root Zone

The title "Keys for ." is using the protocol convention for referring to the root zone as ".".

The root zone's DNSSEC is special. The KSK for the root zone does not have a corresponding DS, so the ordinary life cycle of a KSK is not fully represented.

To read the KSK graph, start with the gray background color with the label RSA-SHA256/2048. This shows that the DNSSEC security algorithm used in the root zone has been RSA-SHA256 since the start of the data. The number 2048 is the bit length of the key used, which has also been unchanged.

With there being only two KSK keys to represent, there are just a few labels seen.

The labels 19036 and 20326 are the key tags of the two KSKs in the root zone's history. Despite the history beginning one year after the start of the root zone's DNSSEC, there were no operational key changes in the first year that were missed.

The orange rectangle in the KSK chart with "20326" in it represents the time period when the new KSK was present in the DNSKEY resource record set. The reason orange (for 20326) is used and not a light green is that the key did not sign the DNSKEY set (and was not in any DS resource record).

The tan rectangle which may be hard to see given the color choice, in the lower entry (for 19036) of the KSK chart, is the small rectangle between a deep blue rectangle and a yellow rectangle. This tan rectangle represents when the key was deactivated but remained in the set, up until the yellow rectangle representing the revocation of the key. During the revocation, the DNSKEY resource record for the key has the revoked bit set to one in accordance with RFC 5011, which notes the change in state of the key. These states will be re-examined in a later chart, focusing on the first root zone KSK rollover.

The lower chart in Figure 5 features the ZSK keys used. The ZSK has been rotated every quarter, so for 10 years, there are 40 keys represented. The most important feature to note here is the regular, staircase-like arrangement. This is a sign of a fully automated process. In any overview chart, like this one, the first item to notice is whether there is a regular cadence to changes (indicating automation), or perhaps a change in cadence. In anomalous cases, the pattern will reveal a level of chaos.

Looking very closely at the chart, perhaps invisible in the rendering of this document, there is one deviation in the ZSK staircase. One of the following charts will highlight this and explain the reason for the deviation.

4.2 One ZSK Lifetime

First, to help describe how the keys are managed for the root zone, a time span from 15 March 2021 through 17 July 2021 is chosen. This limits the view to one KSK key and three ZSK keys making the charts easier to read.

By design, each root zone ZSK is managed in 10-day increments, nine increments per quarter. (There may be one or two more days added to the last increment to match the calendar.) Each ZSK is pre-published for 10-12 days (depending on the calendar) before being made active on the first day of the quarter. Following the last day of the quarter, the key is kept in retirement for 10 days.

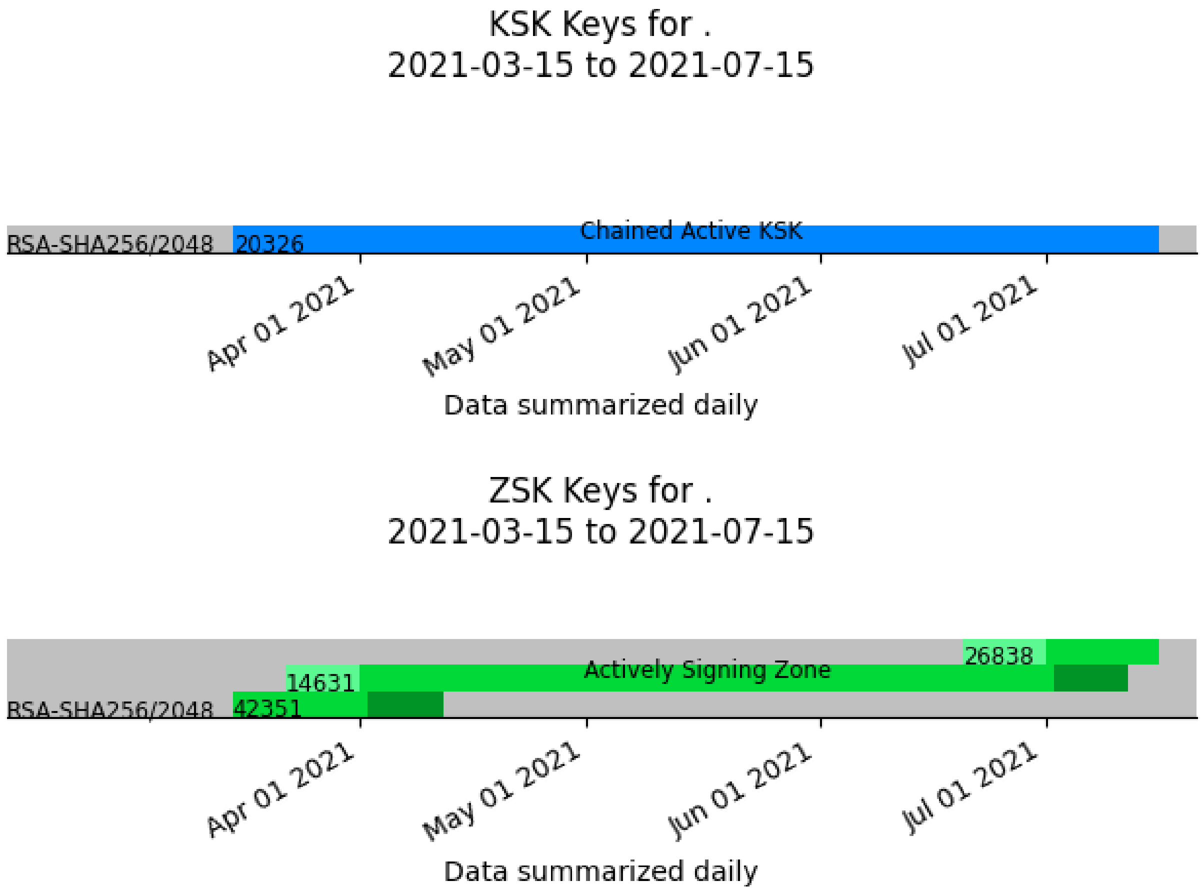


Figure 6 – Focus on one ZSK life cycle

During the dates shown, the KSK has not changed with key tag 20326 in place. The KSK key provides a signature for the DNSKEY resource record set and is assumed to be configured as a trust anchor in all validating DNS servers.

There are stages of life cycles for three different ZSK keys. Key tag 42351 ends its term as the active ZSK on 1 April 2021 and then is in the retirement phase from 1 April 2021 to 11 April 2021. Key Tag 26838 begins its pre-publish stage on 20 June 2021, becoming the active key on 1 July 2021.

Key tag 14631 is the key whose life cycle is wholly shown. It first appeared in a pre-publication stage (light green) on 21 March 2021 and waited 11 days until being made active on 1 April 2021 (medium green). On 1 July 2021, the key entered its 10-day retirement stage (dark green).

Changes to keys happen in the middle of the UTC day, so it appears there is an overlap in the active keys, but there is only one active at any one time.

4.3 The First Root Zone KSK Rollover

A significant event in the management of the root zone KSK began In 2017. The first change of the root zone KSK required a great coordination globally, as all DNSSEC validators needed to have the incoming key before the change, and there was no way to automatically check this by the root zone administrator. The rollover followed the automated updates of DNSSEC trust anchors process, with an important element being the readiness of DNS software to follow this as it had not been conducted on a large scale before. The process was planned to have occurred in about 9 months (counting only the publicly apparent stages) but was paused, mid-way for one year (from October 2017 until October 2018). The historical record shows this.

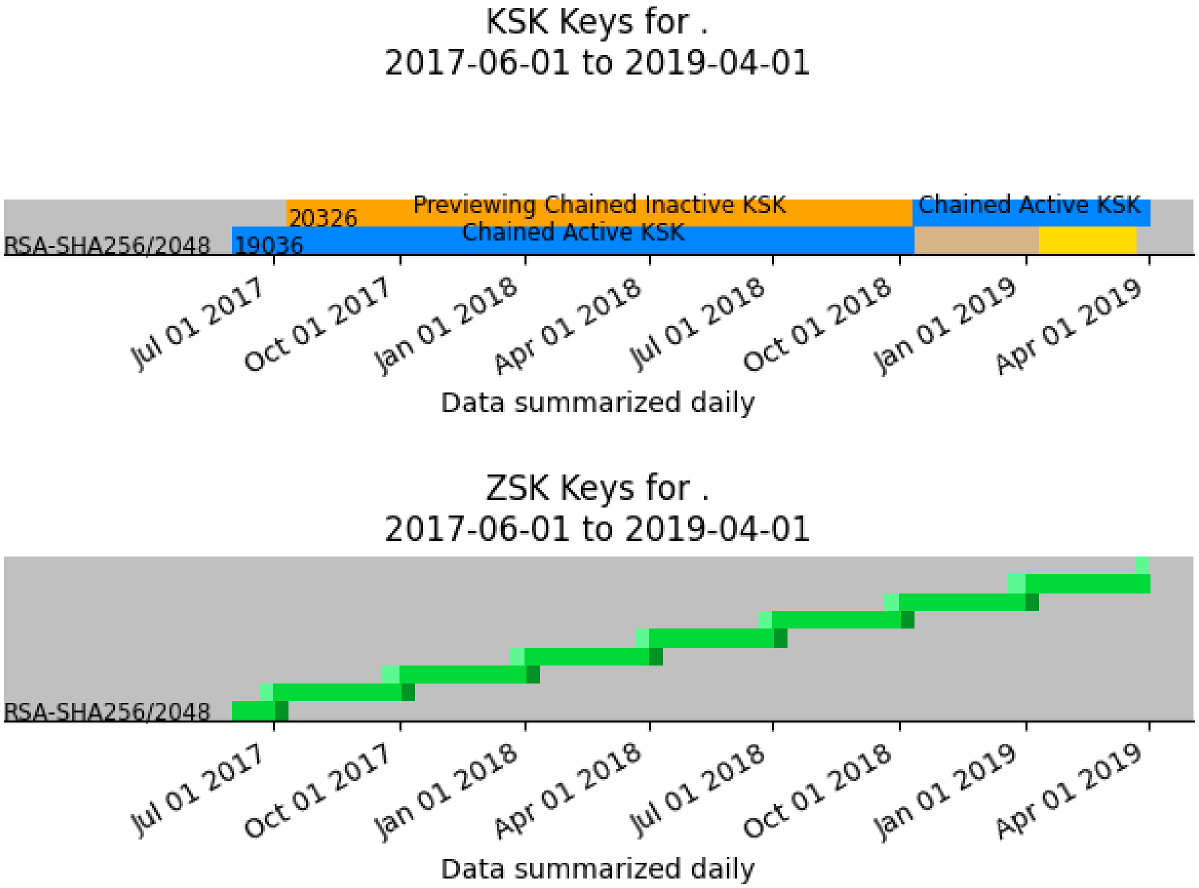


Figure 7 – Root Zone First KSK Rollover

The initial KSK key, with key tag 19036, had been in place since 2010 and is labeled as "Chained Active KSK". The chain here is the embedded trust anchors in all DNS validators attached to the Internet as opposed to a DS resource record. The tan rectangle is the retirement of that key, meaning it no longer was used to generate signatures, this state occurred from 11 October 2018 until 11 January 2019. From 11 January 2019 until 20 March 2019 it was published as a revoked DNSKEY resource record.

The replacement KSK key, with key tag 20326, was presented in the DNSKEY resource record set on 11 July 2017 (orange rectangle). On 11 October 2018, the second KSK began signing the DNSKEY set while the first KSK ceased its period of activity. From that date on the second key has been in the "Chained Active KSK" state.

The tan rectangle for key tag 19036 and the orange rectangle for key tag 20326 represent a time when the key is considered to be chained (trusted in validators) and is in a DNSKEY resource record but there is no RRSIG resource record created by the key. Although the two states look the same, the color is different because one comes before the key's period of activity and the other after the key's period of activity.

Over this time period, the ZSK operations were held steady. As has been noted parenthetically, the operations of the ZSK were reserved for certain days and the KSK for the other days, to prevent any required fall back from being complicated.

The yellow rectangle for key 19036 indicates the period over which it was published but with its revoke bit set.

4.4 The Lengthening of the Root Zone ZSK

The early root zone ZSK keys had a DNSSEC security algorithm of RSA-SHA256 and a key length of 1024 bits. The key length had begun to be seen as inadequate for cryptographic reasons, so a switch was made to keys of 2048 bits. But there was some operational concern that the larger key length (resulting in larger signature lengths) may cause network problems, so a fallback plan was created in which the older ZSK would resume its activity. The chart covering this event shows how the lengthening happened.

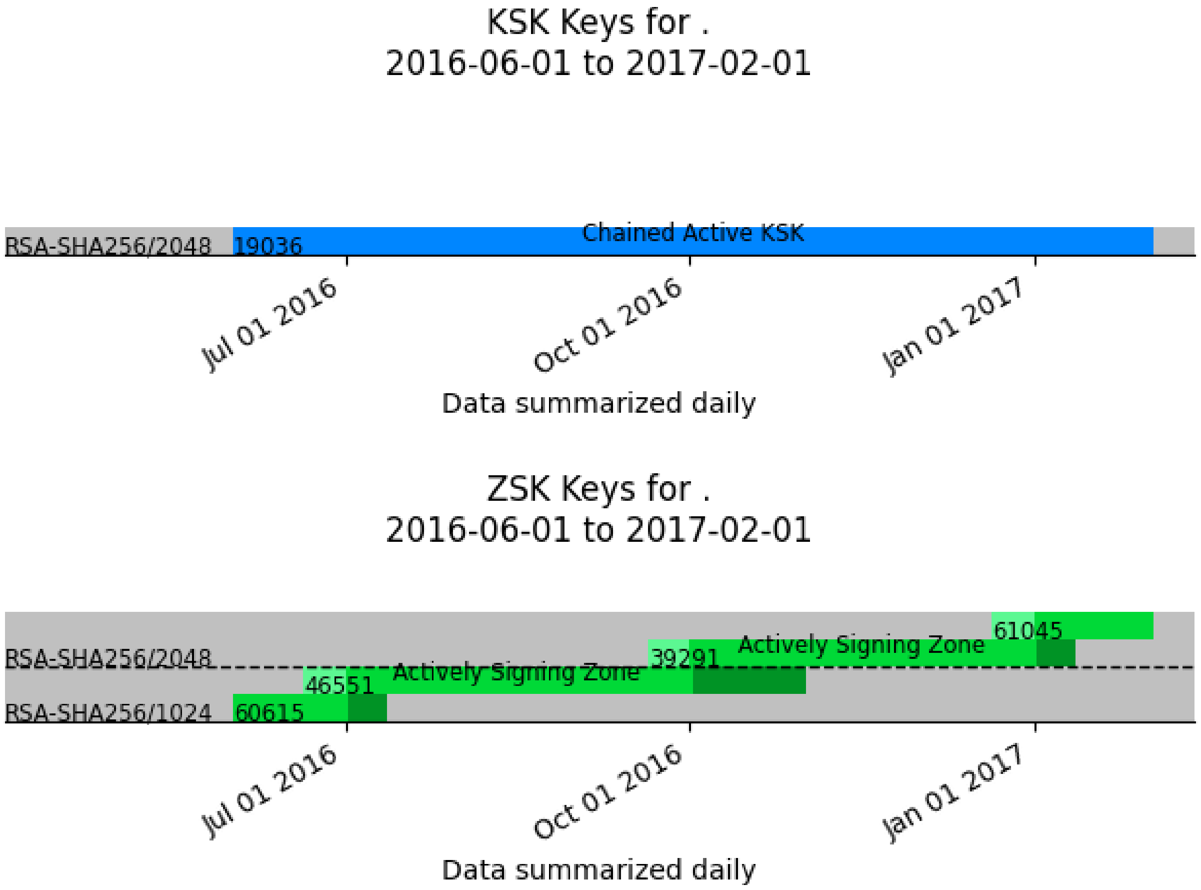


Figure 8 – Root Zone ZSK Lengthening

For this time period, the KSK chart is inconsequential. In the ZSK chart, key 46551 had an elongated retirement period. This countered the possible need to fall back to 46551 if the ZSK tagged as 39291's longer size proved to be a problem.

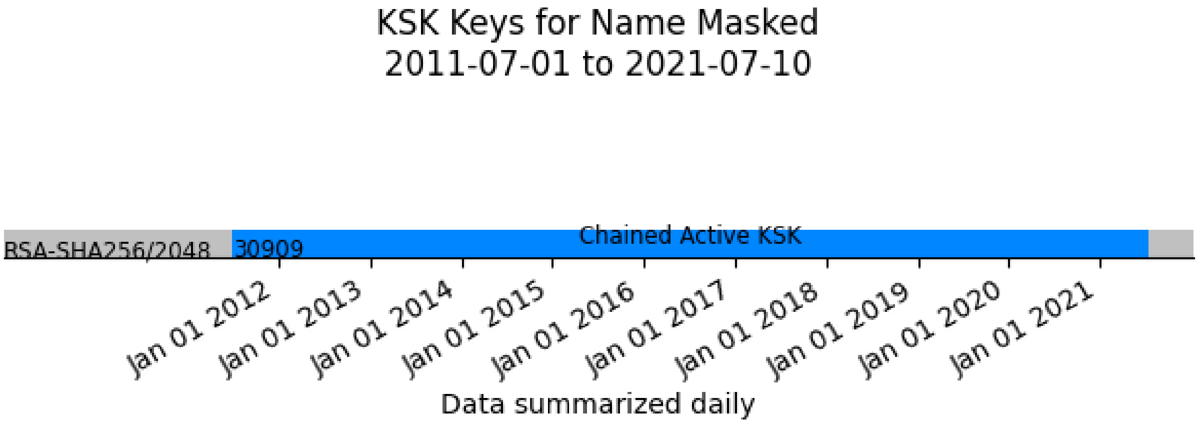
Note in the chart the dashed line. While colors are used to note changes to the DNSSEC security algorithm, dashed lines help highlight when key lengths change. A key length change is only relevant to the RSA-based DNSSEC security algorithms defined to date.

5 gTLDs From the Largest Pre-2012 Operators

Prior to 2012, the largest commercial gTLDs were operated primarily by three organizations. This is a look at their operations, focusing on each organization's largest TLD. The reason for focusing on these three is that, early in the history of domain name registries, they had the most experience with the DNS.

To deflect the temptation to use charts to grade operator performance and to continue to focus on protocol deployment, the operators names are masked in the subsequent charts.

5.1 Operator α (alpha)



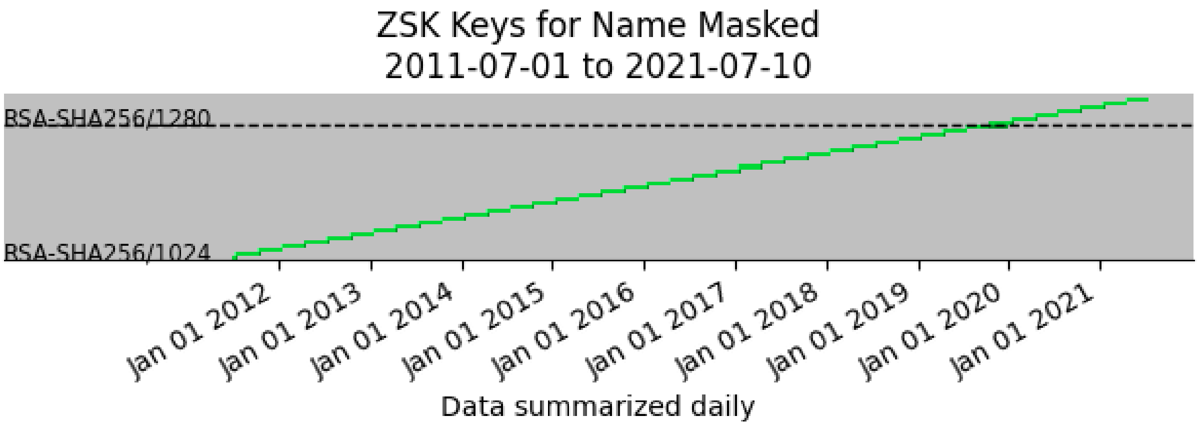


Figure 9 – Large, Pre-2012 gTLD Operator α

Figure 9 shows an operator has never changed the KSK but has followed a strict operational schedule in rolling the ZSK. A closer look at the ZSK chart shows that they did change the length of the ZSK in late 2019, from 1024 bits to 1280 bits.

5.2 Operator β (beta)

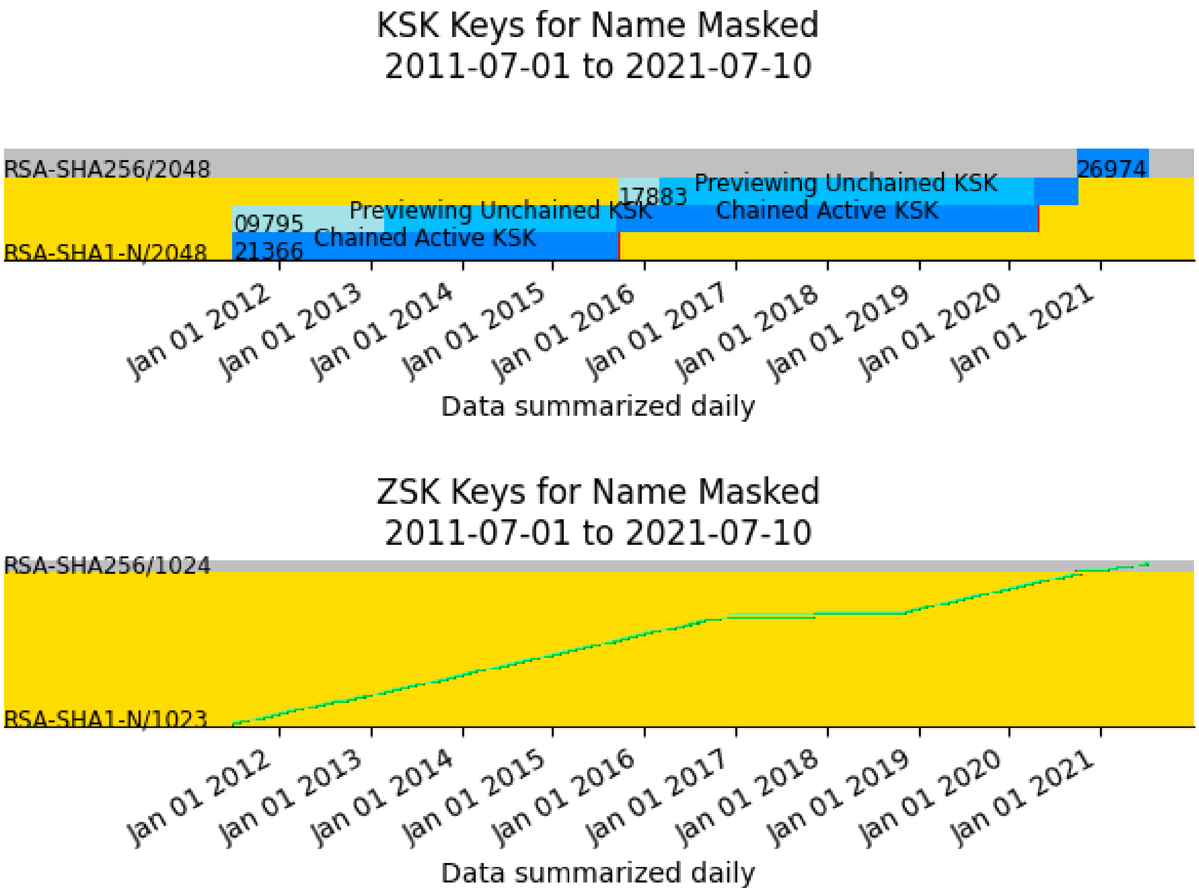


Figure 10 – Large, Pre-2012 gTLD Operator β

In Figure 10, the operator β recently changed from using RSA-SHA1-N as its DNSSEC security algorithm. Switching from one DNSSEC security algorithm to another is commonly called "algorithm rollover.". An operator may find it is necessary to change from one DNSSEC security algorithm to another due to a weakness or perceived weakness in the current DNSSEC security algorithm. An algorithm roll is a complicated procedure for an operator, the first few TLDs to perform algorithms were considered brave pioneers. As time has passed and more experience has been accrued, more TLDs are now performing algorithm rolls due to weaknesses in the older DNSSEC security algorithms.

In the name of the DNSSEC security algorithm, RSA-SHA1-N. The '-N' suffix denotes that the zone is able (and does) use NSEC3 as a negative answer approach. This is an artifact of the very early days of the DNSSEC protocol design.

This operator rolled their KSK on a super-annual basis, the data suggests five years (although the data does not capture the first KSK key's entire life cycle). The ZSK was changed on a monthly basis until 2017, then on an annual basis for two years, before resuming a monthly process.

Looking closely at the end of the early KSK lifecycles, there is a small dark red rectangle. This indicates a state where the KSK is removed from the system but there is still a DS resource record referring to it. This situation is not operationally impacting as long as there is another possible chain and is representative of two independent organizations (one the parent, the other the child, in DNS delegation terms) interacting. The DNSSEC protocol did account for the independence of organizations.

5.3 Operator γ (Gamma)

Operator γ began DNSSEC with an active key and a backup key for each of the KSK and ZSK, rolling the KSK annually and ZSK quarterly. But in late 2017, apparently plans were changed.

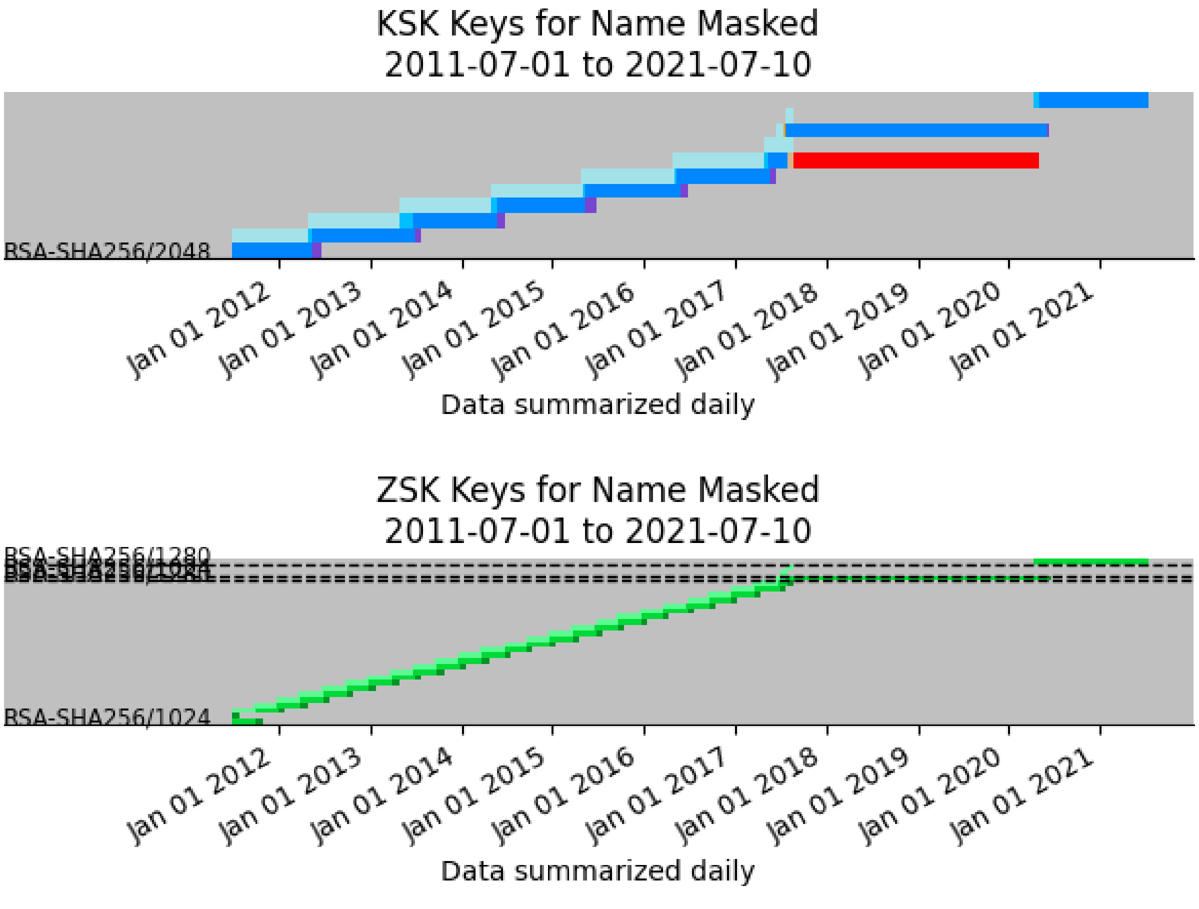


Figure 11 – Large, Pre-2012 gTLD Operator γ

From late 2017, no cadence indicating automation is seen. The red rectangle in the KSK chart represents a period of time when a KSK was retired and removed from the zone, but the DS resource record referring to it was still in place. Still, with a valid validation chain in place, this was not operationally impacting. In the ZSK chart, the jumble of DNSSEC security algorithm labels and dashed lines represents times when keys of different lengths were put into the zone. But the key in active use only changed once, in 2020.

The jumbled text in the upper left of the ZSK chart represents a mixture of different key lengths seen in mid-2018. DNSSEC does not distinguish between keys of different lengths but of the same DNSSEC security algorithm, so there may be a mixture of lengths at any one time. In this chart there is no convenient way to clearly denote this.

5.4 Summary of These gTLD Examples

Each of the operators shown began DNSSEC during 2009, a few years before the data history begins. So, during the span of these figures, more than 10 years of experience has been gained. Operators, even large-scale ones, show signs of reconsidering their approach to DNSSEC. There has been staff turnover (not documented here, but known in the industry), and technology no doubt refreshed. These factors were not considered in the protocol development of DNSSEC, but operators have had to adjust and are doing so as processes age.

One topic to follow into the future is the impact of mergers and acquisitions on key life cycle history. Two of the operators featured in this section have recently changed hands although the back-end technology platforms have not reflected the changes (yet).

6 Examples of Three ccTLDs

There are many different scenarios that could be studied, some known from external events and some by skimming all of the visualizations. Three examples are shown here, chosen because they highlight some lessons related to the deployment of DNSSEC.

6.1 One That Initially Deployed Automated Updates

The operator of this ccTLD is one of two operators that, early on, performed KSK changes according to RFC 5011. Based on the charts seen as well as conversations had with the operator in more recent times, the hypothesis is that initially the zone was administered by a pioneering and enthusiastic engineer who was long gone by the time the current operations staff came to be.

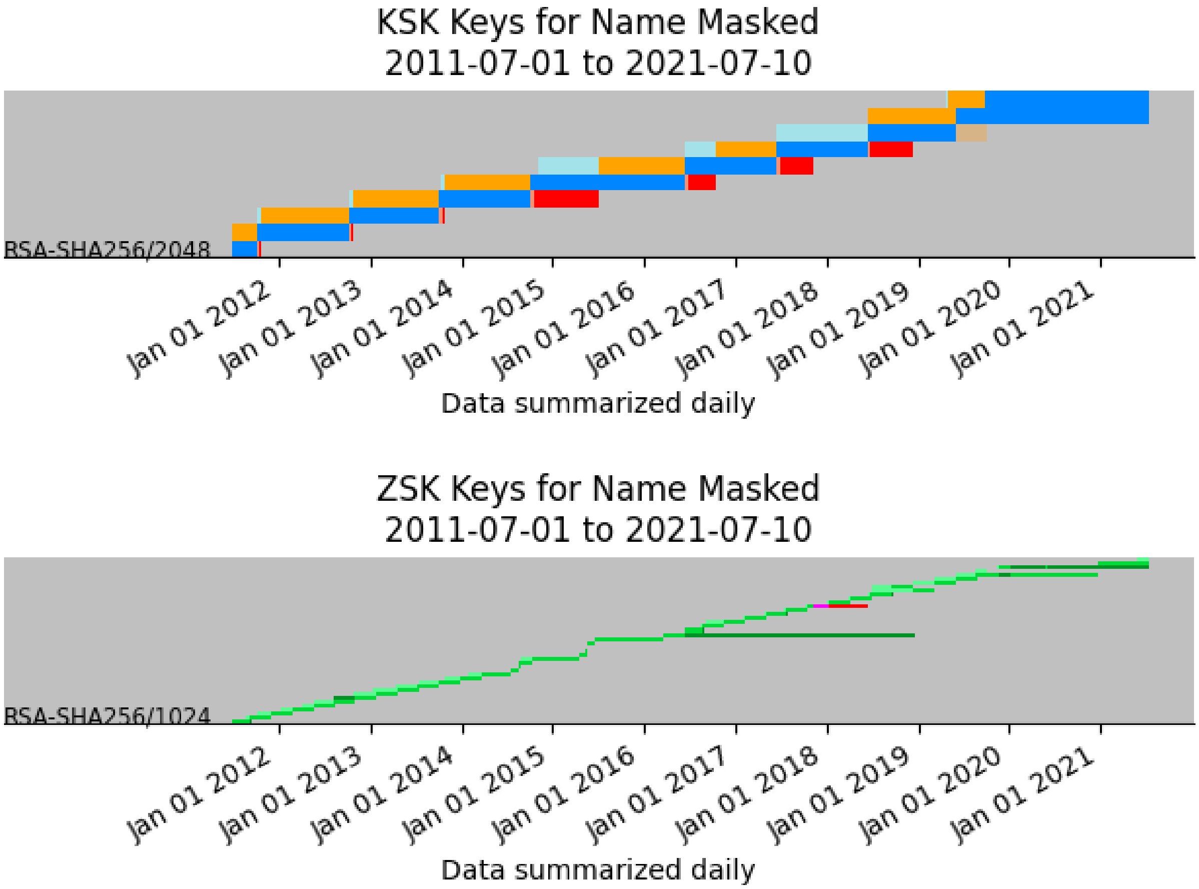


Figure 12 – A ccTLD with a rich history

Figure 12 shows that there were patches of time where the operations changed. From the start of the data through 2014, there was a steady cadence of operations. Then from 2014 until 2019 the key changes followed, loosely, another cadence, and since then a period where there have been few changes to the keys. A superficial glance at the patterns suggests that there was some, but not full, automation in place, as it appears that some keys seem to have been in place longer than expected, perhaps even forgotten.

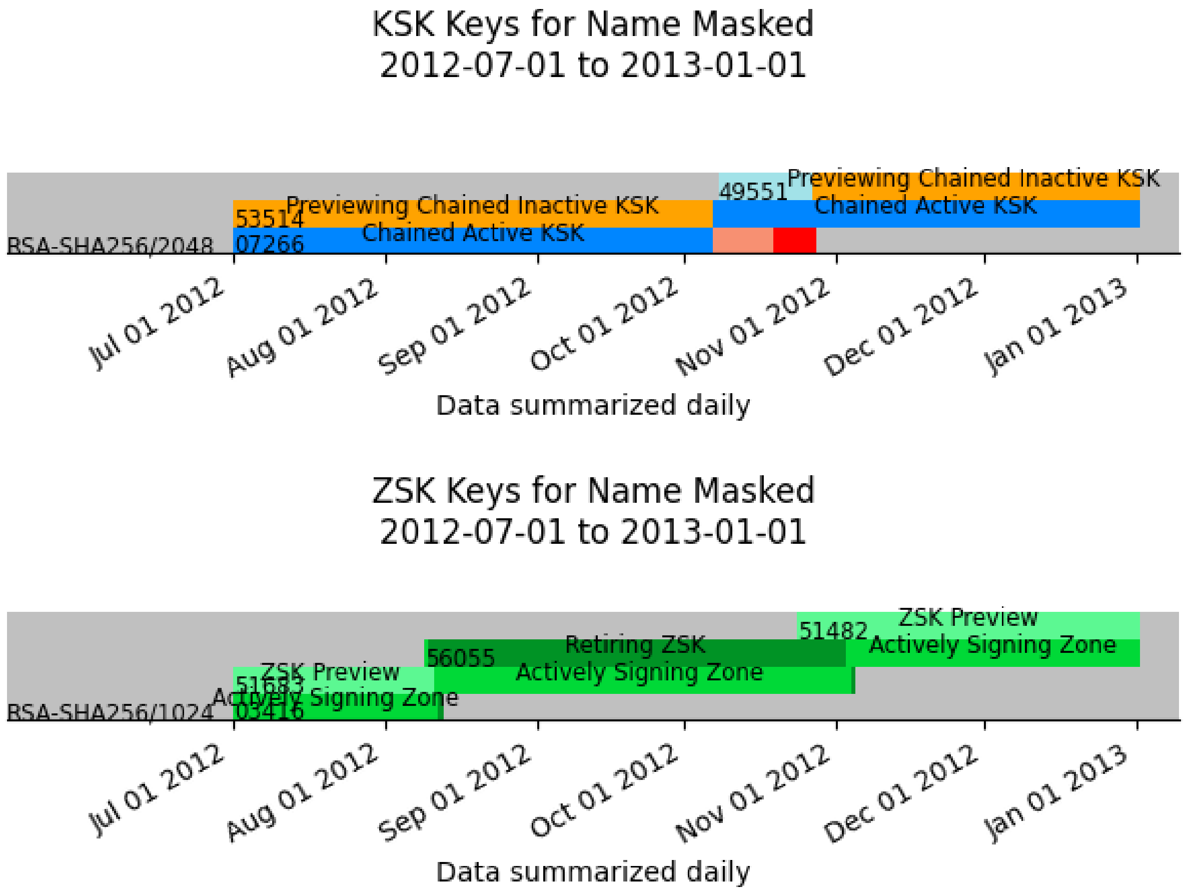


Figure 13 – Late in 2012

Figure 13 focuses on the latter half of 2012. While there is an even cadence going on from 2011 to 2015, there is one exception in the ZSK chart. Looking at more detailed charts (not shown), key tag 56055 is seen as retiring before it becomes active. The day in which the key was first in a DNSKEY resource record it generated a signature for the SOA resource record, but just on that one day. With key tag 03416 active, this would not interrupt operations but does reveal some manual work happening.

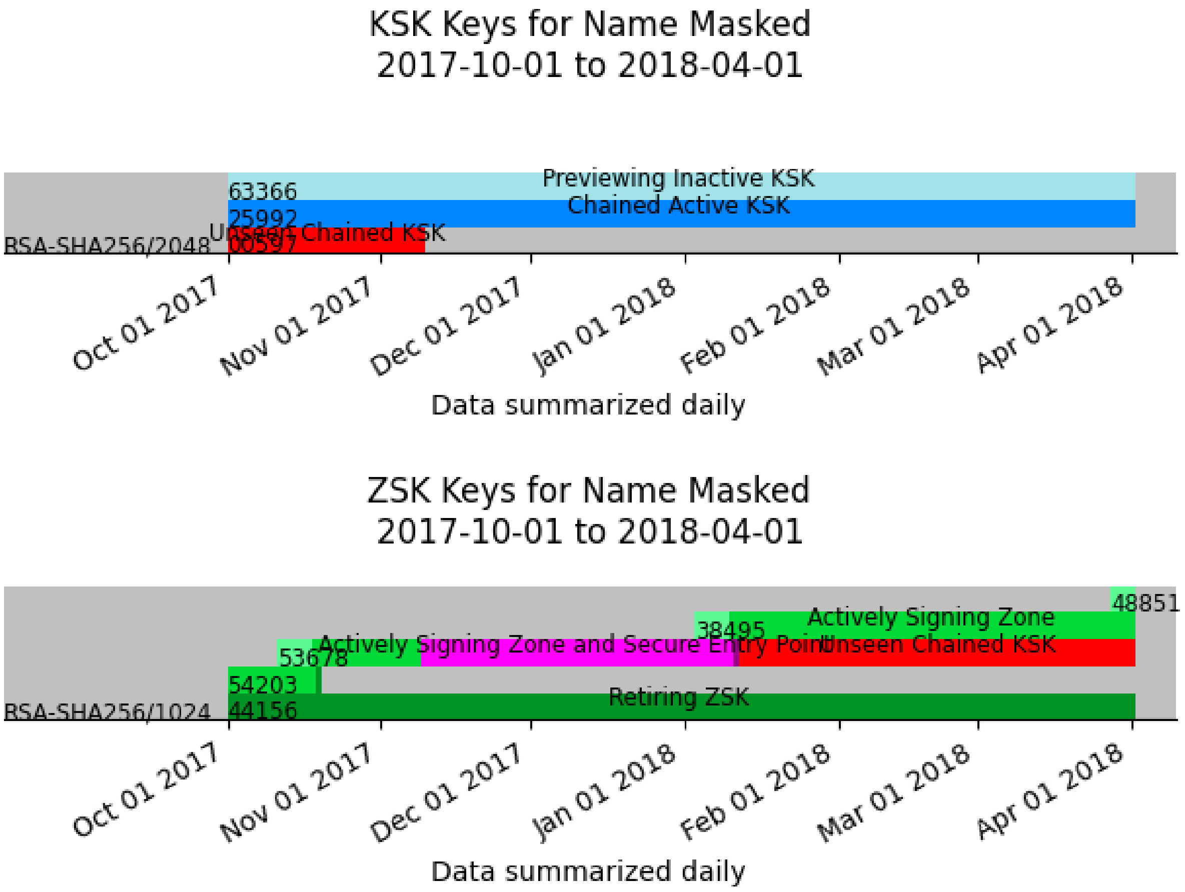


Figure 14 – Late 2017 thru early 2018

In Figure 14, detail is shown of a time when a ZSK was included in a DS resource record, generating RRSIG resource records for the zone and for the DNSKEY resource record set. This makes the key a Common Signing Key (CSK) for the period of time its rectangle is pink. (The key is called a ZSK based on the SEP bit being "off" and the key being used to sign records other than the DNSKEY resource record set. Diving deeper, the SEP bit is not meaningful to the DNSSEC validation protocol, it is a hint to key management software. In this instance, the ZSK role is determined by signing the SOA resource record set and others.) Despite this, caused perhaps by an accidental configuration, the zone maintained a working, active ZSK and KSK. Unless there were other problems, there would be no noticeable operational impact. But, oftentimes, when there is one mishap, there are others as well.

The detailing of key management in this case study is done to remind us that operations teams and organizations change. Documentation is important, as well as only deploying what makes sense and has a clear tie to requirements, or the hand off of operations from one to another may be troubled. What is also seen here is the resiliency of DNSSEC, and why that is important, when keys are changed.

6.2 One That Rolled DNSSEC Security Algorithms, Twice

This ccTLD exhibits a keen awareness of registry engineering, deploying advanced services. This assessment is based on overall interactions, in industry fora, and not just these charts. Looking at the cadence of key changes, consistency is very apparent, a hallmark of a highly automated process. The main reason for highlighting this operator is that they have changed their DNSSEC security algorithm twice while maintaining their usual cadence.

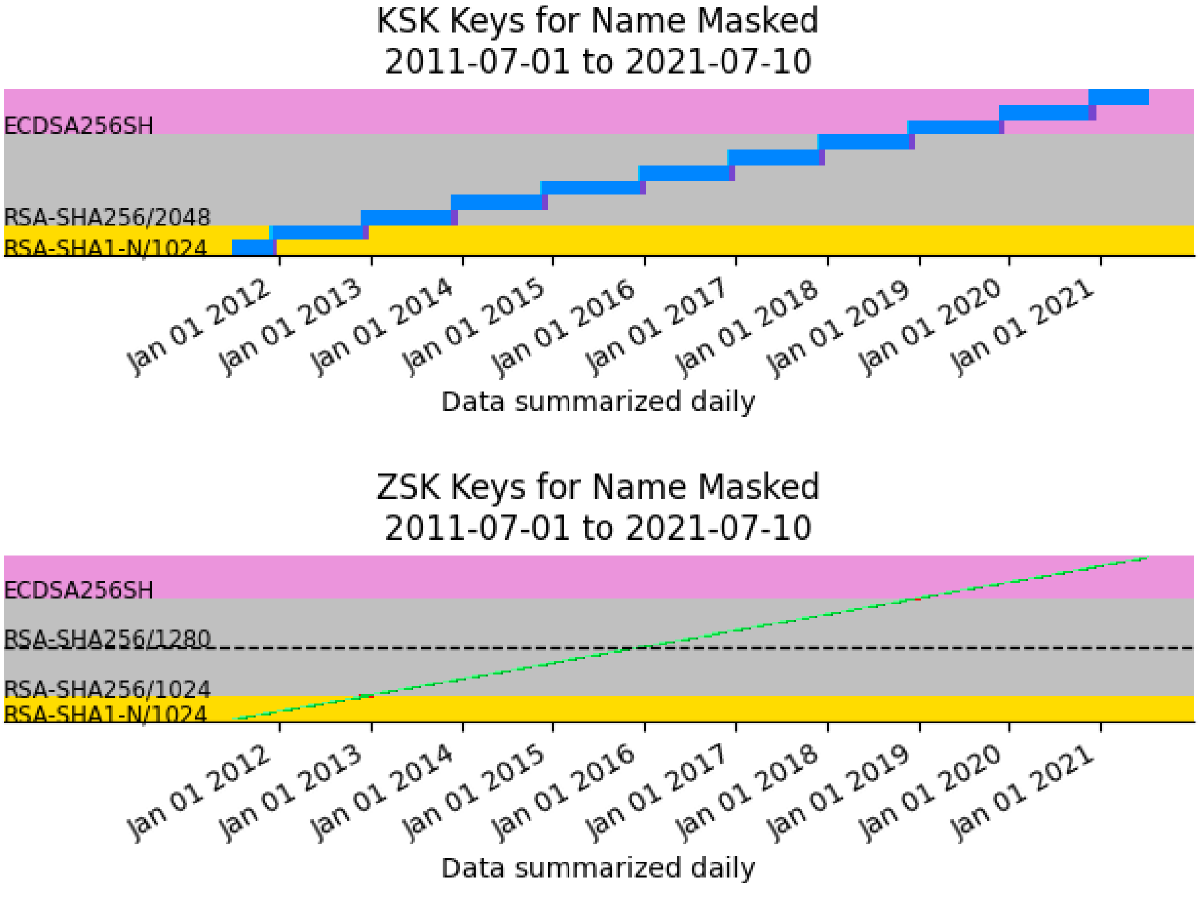


Figure 15 – Twice-changed DNSSEC security algorithms

In Figure 15, The pink background represents the use of ECDSA256SH keys, the gray (seen earlier in the document) RSA-SHA256, and the yellow (also seen previously) RSA-SHA1-N. The registry also lengthened the RSA key in use (without changing the DNSSEC security algorithm). The registry has maintained the clock-work-like process despite these changes.

The reason this is surprising is the expectation that changing the DNSSEC security algorithm is difficult. Due to protocol rules, when an algorithm is used, it must be used everywhere. If a public key of a new algorithm is published in the zone, it will cause validators to expect new signatures throughout the zone. The theory is that adding all the new signatures at once would be a burden, and with caching in place, nearly impossible to avoid confusion. Trying to use two algorithms at once (instead of an immediate change) might make packets too big.

Squinting closely at the full, 10-year history, there are two small red rectangles, hinting there is something interesting to see leading to the question, “How did the registry change the algorithms?” By focusing the timeline to the points where the algorithms change, about 2012–13 and again about 2018–19, one can see the steps. In both cases, the steps were the same, although the dates of changes were slightly different.

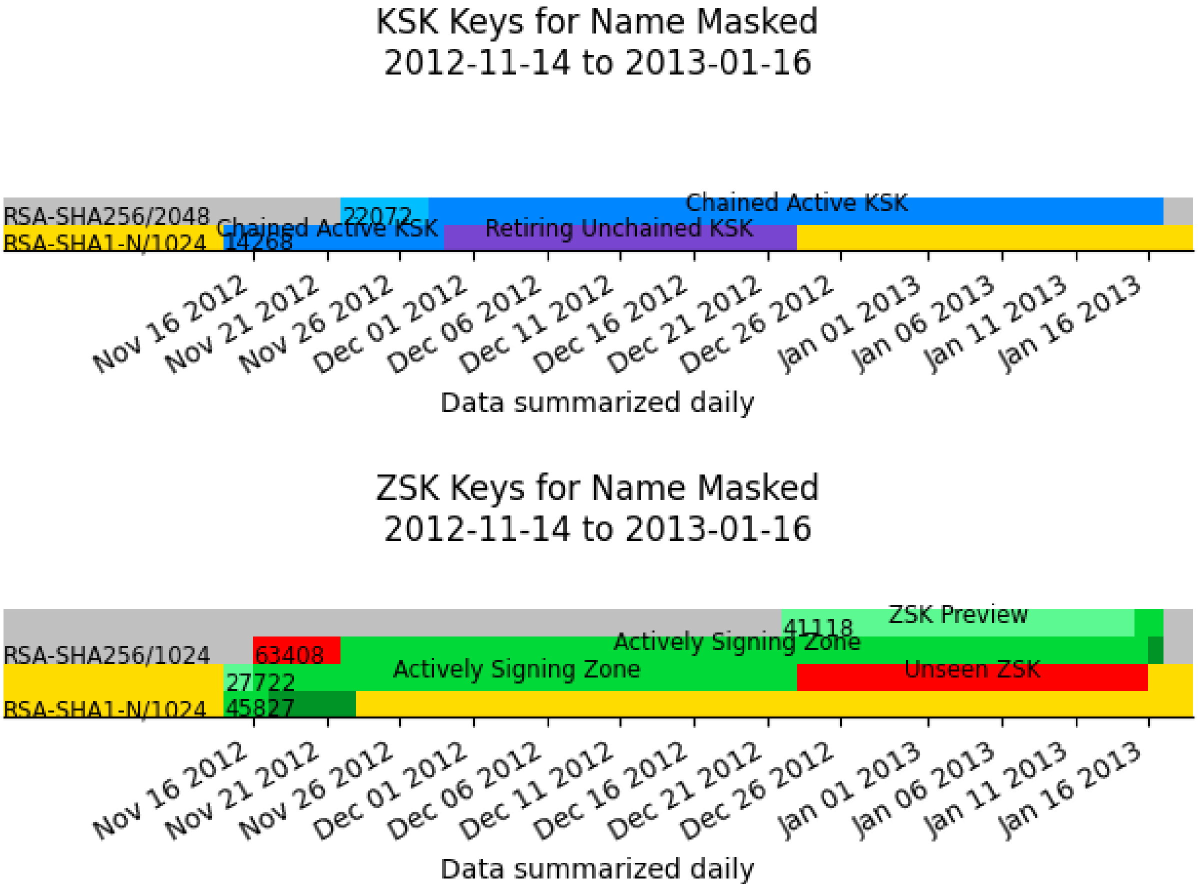


Figure 16 – From RSA-SHA1-forNSEC3 to RSA-SHA256

In Figure 16, one can see that, to begin the algorithm change, a new key of the new DNSSEC security algorithm generated signatures for a few days before the key was published and instantly active. This is the leftmost red rectangle, in this case, not an error but a moment of interest. For a few weeks, the zone was signed by two keys of two algorithms before the outgoing key was removed from the DNSKEY resource record set. The signatures of the outgoing key were still published (the rightmost red rectangle, the one barely visible on a 10-year time scale) to avoid validation failures in caches still holding copies of the outgoing key.

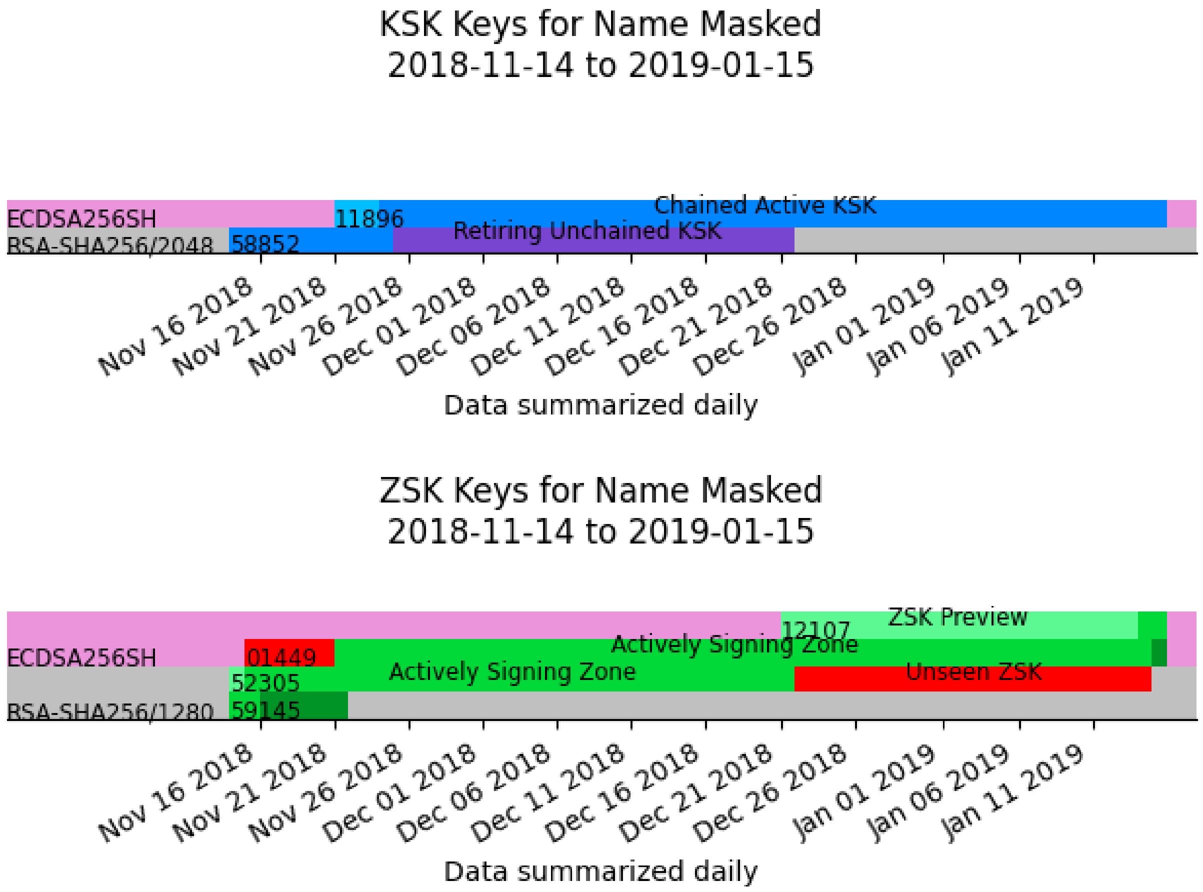


Figure 17 – From RSA-SHA256 to ECDSA256SH

In Figure 17, one can see the same sequence of steps happening. As the process worked once, it is unsurprising it would be followed again. The only deviation is that the process seems to run one day later (starting and ending) in 2012 when compared to 2018. A first thought might be that the calendars for November 2012 and November 2018 were different, but, no, 1 November 2012 and 1 November 2018 were both Thursdays.

6.3 One That Seemed to Have Followed the Fired and Forgot Tradition

It has been said that DNS used to be "fire and forget," once a zone was published it never changed. DNSSEC eliminated this by requiring new signatures and key changes. The latter, key changes, is not strictly required by DNSSEC although cryptographers generally recommend it. To the point of not requiring key changes, one ccTLD has operated DNSSEC for 10 years, never changing its keys for the lifetime of the data set.

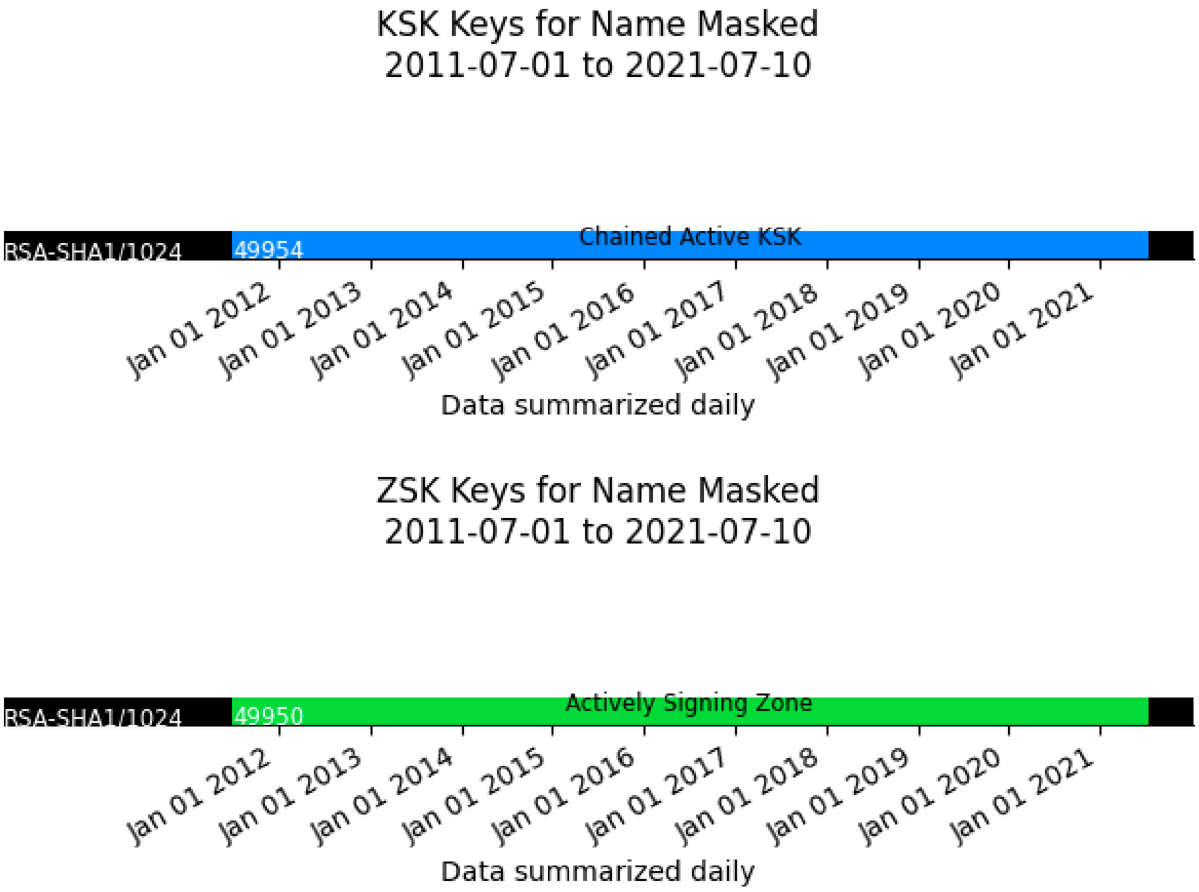


Figure 18 – Fire and Forget Key Management

Figure 18 shows a ccTLD that has featured one KSK (labeled 49954) and one ZSK (49950) for the duration of the observations. Whether this needs to be changed has not been demonstrated, given a lack of any reports of operationally impacting or security events.

7 Wrap-Up

Observing DNSSEC key lifecycles gives a good view into the state of operations, which permits evaluation of the protocol, the processes used to operate it, and to some extent (although not highlighted in this document) the tools available. This document featured just a few case studies, providing an orientation to the visuals and giving a hint as to the information that can be gained. Not shown are many other interesting case studies, from a pool of around over 1,500. Note that there may now (as one reads this) more than 1,500 or less than 1,500 active TLDs, the number fluctuates over time.

A common theme across the case studies is that the industry of domain name registries is maturing. From just a few operators before 2010, many have entered and have grown. In many other areas, the role of a registry has gained prominence, from a research project to a critical element in some geographic regions. With these changes, approaches change and this can be seen in the life cycles of DNSSEC keys.

When looking at the different cases and seeing "unexpected" events, it is tempting to think that operators are not following the appropriate guidelines. In some cases this may be the case, but further research has suggested that, oftentimes, there are reasons why operators deviate from what a protocol engineer might expect. What this suggests is that the strategy to encourage deployment of new technologies does not need to stress the education of operators, rather that the development of new technologies needs to incorporate operational reality early in the development process. This observation is consistent with other studies of developer operations (sometimes called “devOps”).

The development of this document has occurred at a time when it is inconvenient to speak directly to the operators involved, due to sporadic local and global travel restrictions in place. It is a given that observing symptoms of operations from afar is possible but diagnosing the drivers from a distance is often inaccurate. What is not part of the record here is whether or not validation errors occurred and whether they had an impact on the changes seen. This is just one question that arose in a review of this document. Once travel is possible again, a set of follow-up interviews will be very enlightening.