Roll, Roll, Roll your Root: A Comprehensive Analysis of the First Ever DNSSEC Root KSK Rollover

Moritz Müller
University of Twente and SIDN Labs

Matthew Thomas
Verisign

Duane Wessels
Verisign

Wes Hardaker
USC/Information Sciences Institute

Taejoong Chung
Rochester Institute of Technology

Willem Toorop
NLnet Labs

Roland van Rijswijk-Deij
University of Twente and NLnet Labs

ABSTRACT

The DNS Security Extensions (DNSSEC) add authenticity and integrity to *the* naming system of the Internet. Resolvers that validate information in the DNS need to know the cryptographic public key used to sign the root zone of the DNS. Eight years after its introduction and one year after the originally scheduled date, this key was replaced by ICANN for the first time in October 2018. ICANN considered this event, called a *rollover*, *"an overwhelming success"* and during the rollover they detected *"no significant outages"*.

In this paper, we independently follow the process of the rollover starting from the events that led to its postponement in 2017 until the removal of the old key in 2019. We collected data from multiple vantage points in the DNS ecosystem for the entire duration of the rollover process. Using this data, we study key events of the rollover. These events include telemetry signals that led to the rollover being postponed, a near real-time view of the actual rollover in resolvers and a significant increase in queries to the root of the DNS once the old key was revoked. Our analysis contributes significantly to identifying the causes of challenges observed during the rollover. We show that while from an end-user perspective, the roll indeed passed without major problems, there are many opportunities for improvement and important lessons to be learned from events that occurred over the entire duration of the rollover. Based on these lessons, we propose improvements to the process for future rollovers.

ACM Reference Format:
Moritz Müller, Matthew Thomas, Duane Wessels, Wes Hardaker, Taejoong Chung, Willem Toorop, and Roland van Rijswijk-Deij. 2019. Roll, Roll, Roll your Root: A Comprehensive Analysis of the First Ever DNSSEC Root KSK Rollover. In *Internet Measurement Conference (IMC '19), October 21–23, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3355369.3355570

IMC '19, October 21–23, 2019, Amsterdam, Netherlands

ACM ISBN 978-1-4503-6948-0/19/10…$15.00
https://doi.org/10.1145/3355369.3355570

1 INTRODUCTION

The Domain Name System (DNS) is *the* naming system of the Internet. Since 2010, the root of the DNS is secured with the DNS Security Extensions (DNSSEC), adding a layer of authenticity and integrity. DNSSEC uses public-key cryptography to sign the content in the DNS and enables recursive resolvers[1] to validate that the information they receive is authentic. The sequence of cryptographic keys signing other cryptographic keys is called a *chain of trust*. The public key at the beginning of this chain of trust is called a *trust anchor*. Validators have a list of trust anchors, which they trust implicitly. The Root Key Signing Key (KSK) acts as the trust anchor for DNSSEC and this cryptographic key was added to the root zone in July 2010. Eight years later, and after a one year delay, the KSK was replaced for the very first time, following established policy that requires regular rollovers of the Root KSK [1]. This event, usually referred to as the Root KSK Rollover (hereafter "the rollover"), required years of preparation and was considered risky. Stakeholders expected, in the worst case, millions of Internet users (up to 13%) to become unable to resolve a domain name [2].

The Internet Corporation for Assigned Names and Numbers (ICANN), the organization responsible for coordinating and rolling the key, collected feedback from the community before the rollover. Two risks were most feared: (i) resolvers that would not update their local copy of the key [2] and (ii) resolvers that could not retrieve the key material from the root because it might exceed a packet size that cannot be safely handled by some networks (we explain these two risks in more detail in Section 2.2.1).

Leading up to the initially scheduled date of the rollover in October 2017, ICANN and its stakeholders carried out measurements to estimate the potential impact of both risks and considered the former acceptable. The actual impact of the former, however, was still hard to estimate. One of the reasons was the introduction of a new protocol that enabled resolvers to signal their configured key to the root server operators (RFC 8145 [3], we explain the protocol in more detail in Section 3.1). This protocol signaled that a significant number of resolvers only had the old key configured and this led to the decision to postpone the rollover [4]. Rescheduling the rollover gave researchers the opportunity to understand which resolvers sent this signal and estimations were that only a few users would be negatively affected by the rollover [5]. This gave ICANN the

[1]Today most, but not all, DNSSEC validation happens in recursive resolvers. For convenience we use the term "resolvers" in this paper, but the discussion applies equally well to validation that occurs elsewhere (e.g. in applications).

confidence to move forward with the rollover. The actual rollover was carried out on October 11th, 2018. In their March 2019 review of the rollover, ICANN concluded that *"there were no significant outages"* and that the rollover *"was an overwhelming success"* [6].

In this paper we provide a comprehensive analysis of the rollover, starting from the publication of the new key in July 2017 until the removal of the old key in March 2019. We use data that was actively and passively collected at key points in the DNS ecosystem over the entire duration of the rollover. We, as members of the DNS community, actively supported the rollover process with timely data analyses. This provides us with a unique perspective that covers multiple vantage points of the rollover. The main contributions of this paper are that we:

(i) Provide the first in-depth analysis of the root KSK rollover, a unique event with an impact on the global Internet;
(ii) Cover the event from multiple perspectives, that of root operators, of resolver operators, and end users;
(iii) Validate ICANN's conclusion that the event was a success and show that, while this conclusion generally holds for end users, there are observable challenges at all stages of the rollover;
(iv) Perform an in-depth analysis of the causes of the challenges seen at all stages of the rollover;
(v) Give recommendations for improving telemetry, processes for root key management and future rollovers.

In the remainder of the paper, we outline the basics of DNS and DNSSEC, as well as the stages of the root rollover and the risks involved (Section 2). Next, we introduce our measurement methods and data (Section 3). Then, we split the analysis of the rollover into three sections, *before*, *during* and *after* the rollover (Section 4). In Section 5 we discuss related work and in Section 6 we provide recommendations for better telemetry and rollover process improvements based on our analysis. We conclude the paper in Section 7.

2 BACKGROUND

This section explains the basics of DNS and DNSSEC, followed by a discussion of the Root KSK Rollover and its risks.

2.1 DNS and DNSSEC

The DNS uses resource records (RRs) to map domain names, such as `example.com`, to values. For example, an `A` record maps a domain name to an IPv4 address and an `NS` record maps a domain name to the authoritative name server for a domain. These records are stored in a *zone* and made available at the domain's authoritative name servers. End users usually employ recursive caching resolvers to query for records in the DNS. The DNS is a hierarchical naming system and at the top of the hierarchy sits the *root*. Assuming an empty cache, a recursive resolver that queries for the `A` record of `example.com` sends its first query to the authoritative name servers of the root, which refer the resolver further to the authoritative name servers of `.com` that finally refer it to the name servers of `example.com`. Each RR also has a Time-To-Live (TTL) field that defines how long a resolver may cache the RR. Until the TTL of the RR has expired, the resolver generally will not send another query for `example.com` but respond with the record from its cache.

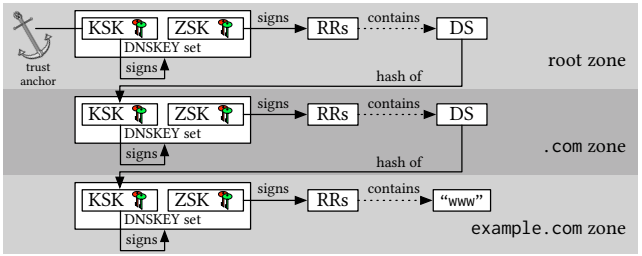

Figure 1: DNSSEC chain of trust, starting at the root.

DNSSEC allows a recursive resolver to validate that the response it receives from an authoritative name server has not been tampered with. Operators sign their records using public-key cryptography and publish the public key — in a `DNSKEY` RR — together with the signatures — in an `RRSIG` RR — in the zone file. Often, operators create two keys, a Zone-Signing-Key (ZSK) used to sign most RRs and a Key-Signing-Key (KSK) to sign only the `DNSKEY` RRset. This is also the case for the root zone of the DNS.

DNSSEC adds one central point of trust to the DNS at the root zone — a so called *trust anchor* (see Fig. 1). Validating recursive resolvers, or "validators," only need to trust the KSK of the root to validate signatures in the DNS. Because the root signs a hash (`DS`) of the `.com` KSK and publishes it in its zone, and because `.com` also signs and publishes a hash of the `example.com` KSK in its zone, a *chain of trust* between the different domains is created. Generally, DNSSEC validation leads to one of three results: the *secure* state, meaning the validator successfully verified the authenticity and integrity of the response, the *bogus* state, meaning the validator concluded the signatures in the response are invalid, or the *insecure* state, meaning the response was not signed or there is no chain of trust that allows validation. If a validator concludes a response is secure, it sets the Authenticated Data (`AD`) flag in its response to a client. If a response is bogus, the validator sends an error to the client with the `SERVFAIL` response code. If a response is insecure, the validator returns the response as-is, like a 'classical' DNS response.

2.2 The Root KSK Rollover

It is considered good operational practice that operators of zones signed with DNSSEC be able to periodically change, or "roll," the zone's cryptographic keys. A rollover might be necessary in case of a security breach, in case operators want to upgrade to a new algorithm, or because they follow a key management policy [7]. The root zone's ZSKs are rolled every calendar quarter [8]. When the root zone was first signed in 2010, it was generally accepted that the KSK would be rolled after a period of 5 years [1]. The parties involved in operating the root zone began discussing and planning a KSK rollover in 2013, but this work was put on hold when the NTIA announced its intention to transition oversight of the IANA functions to the Internet community [9]. Work on the rollover resumed in 2015, culminating in a 2016 Rollover Design Team report [2]. ICANN and Verisign, in their respective roles as the IANA Functions Operator and Root Zone Maintainer, used the design team report to develop a final set of operational plans [10].

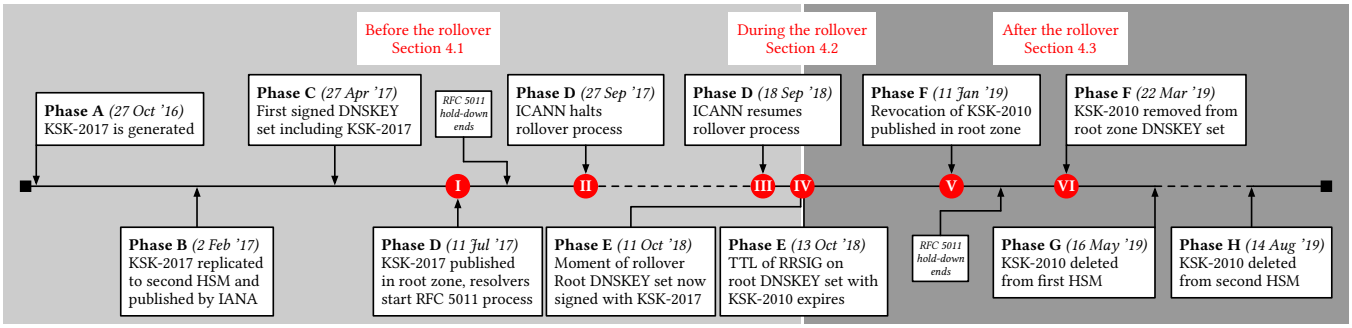

Figure 2: Time-line of the Root KSK rollover

These plans describe the process for replacing the old KSK, further referred to as *KSK-2010*, with a new KSK, now referred to as *KSK-2017*. Fig. 2 shows a timeline of each of the phases of the rollover as described in the operational plan. We have highlighted six key events in red labeled **I** – **VI**. These six events are the focus of this paper. In the rest of this section, we explain the risks as identified in the design team report and specific considerations that stem from the special role of the root's KSK as a *trust anchor*.

2.2.1 Risks during the Rollover. The design team report [2] identifies two major risks: validating resolvers that are unable to configure the new KSK as a trust anchor, and the increase in response size of the DNSKEY RRset at certain stages of the rollover process.

DNSKEY RRset Changes. Resolvers need a copy or a hash of the root KSK, and to configure it as a trust anchor. Some modern resolvers, e.g. BIND, ship with the current root KSK configured as a trust anchor. Thus, resolvers shipped with only *KSK-2010* need a mechanism to fetch *KSK-2017* before the rollover. If this does not occur, these resolvers fail validation as soon as they need to validate a signature signed with *KSK-2017*, when the root zone is published with its DNSKEY RRset signed by *KSK-2017* (**IV** in Fig. 2).

Resolvers that receive a DNSKEY RRset without a key that matches their trust anchor may start sending extra DNSKEY queries to the root. There are two reasons for this: First, some resolver implementations are designed to retry failures, including validation failures, at some or all of the available authoritative name servers. Second, resolvers typically cache such a failure for a short time only (so-called *negative caching*). Once the cached failure expires, the process starts anew. Negative caching times are typically much shorter than the TTL of the root DNSKEY RRset (currently 48 hours).

Clients relying on resolvers with an incorrectly configured trust anchor may receive responses with the SERVFAIL error code because the resolver failed to perform DNSSEC validation. ICANN's KSK rollover design team expected the number of resolvers that could not update their trust anchor to be low [2]. This degree of confidence was based on the RFC 5011 mechanism implemented by most resolvers and that we describe in the next section. In Section 4.2, we measure the actual impact of the rollover on resolvers and clients.

Response Size Changes. Due to the KSK/ZSK split, the size of most responses remains the same during the KSK rollover. Only the size of a DNSKEY response changes. Fig. 3 illustrates the sizes of various

Figure 3: DNSKEY response sizes during the rollover.

DNSKEY responses that occur throughout the rollover process, varying from 864 to 1,425 octets. The sizes shown in the figure include the question and standard EDNS0 data. Some root servers have deployed DNS cookies, which adds another 28 octets to the sizes shown. These response sizes can exceed the Maximum Transmission Unit (MTU) of some networks, which can cause fragmentation of UDP packets. Firewalls and other middle-boxes sometimes block fragmented packets [11, 12], which can hinder resolvers when trying to receive the DNSKEY record set and thus make it impossible for them to validate signatures. The measurements carried out by ICANN and the community leading up to the rollover indicated up to 6% of resolvers could be affected by this problem. These serve less than 1% of users and most do not perform DNSSEC validation [2]. Root servers may also receive an increased number of ICMP packets signaling the packet size exceeds the network's MTU. Clients relying on these resolvers could experience an increased response time or receive a DNS SERVFAIL response. We study the impact of increased response sizes during the revocation in Section 4.3, when the highest packet size during the rollover process occured.

2.2.2 Updating Trust Anchors. DNSSEC allows validators to automatically update their trust anchors through an in-band mechanism in the DNS, known as RFC 5011 [13], which works as follows. At the start of a rollover, the new key (*KSK-2017*, introduced at **I**) is added to the DNSKEY RRset, but the RRset is only signed with the then current trust anchor (*KSK-2010*). This signals to resolvers that support RFC 5011 that they should start the process of accepting the newly introduced key as a trust anchor. Acceptance is not effective immediately; instead, a *hold-down timer* starts, lasting 30 days. Only if the resolver has seen the new key consistently throughout the hold-down period will it accept the new key. This prevents malicious actors who have gained access to a trust anchor from instantly injecting a new trust anchor. Once the new trust anchor comes into

effect, the old one may be revoked. In RFC 5011 this is achieved by publishing a DNSKEY RRset in which the old key is marked with a revocation flag (at **V**). Again, after a 30-day hold-down the trust anchor is then removed by resolvers. Most resolver software (e.g. BIND, Unbound and Knot) supports RFC 5011 and among popular implementations, only PowerDNS lacks support. The widespread support of RFC 5011 gave the Rollover Design Team confidence that most resolvers would pick up the new key on time [2].

This KSK rollover was the first real test of RFC 5011. Since the publication of RFC 5011 in 2007, new technologies have been introduced that were not considered back then. This includes widespread use of virtual machines and containers, configuration management tools such as Puppet and Ansible, and DNS resolvers running on inexpensive, and hard-to-update home and small office routers.

Where RFC 5011 specifies an *in-band* approach, an *out-of-band* approach is discussed in RFC 7958 [14]. In this approach, resolvers and other applications can retrieve keys and/or hashes directly from the website of IANA as an XML document. Applications can use various approaches to validate correctness of this information, e.g., trusting protections provided by TLS or a digital (PGP) signature file, published separately. The Unbound resolver software uses this mechanism in situations when updates via RFC 5011 fail [15].

With both mechanisms, it is not possible for third parties to determine which resolvers have configured *KSK-2017*. To address this, new resolver software supports protocols that try to provide this insight. We use these protocols to measure the deployment of *KSK-2017* in Sections 4.1.1 and 4.3.1 and discuss their use in Section 6.

3 DATASETS AND METHODOLOGY

We use a broad set of passive and active measurements at different vantage points in the DNS hierarchy to cover the most critical phases of the rollover. We discuss these datasets and how we use them to analyse the rollover below. We also make the processed data sets and the accompanying scripts for each figure available [16].

3.1 Passive Measurements

The DNS root system has 13 root server identities, each of which is run by one operator [17]. At various stages of the rollover, we use passive datasets from select root servers or aggregate data for all of the root servers from a public repository. More specifically, we use the following datasets:

Root Queries. The Domain Name System Operations Analysis and Research Center (DNS-OARC) collects DNS traces from various name servers including the root system. This includes their well-known annual Day-in-the-Life (DITL) datasets [18]. Given the significance of the KSK rollover, DNS-OARC co-ordinated a DITL data collection from root operators spanning an 82-hour window around the dates of the actual rollover. We utilized this data, available to researchers and DNS-OARC members, to provide a holistic view of root query traffic during the rollover.

Our analysis, however, extends to well before and after the rollover. To support this, we make use of query datasets collected at three root servers, A, B and J. This non-public longitudinal data, spanning 2017–2019, was made available by Verisign (A/J Root) and the University of Southern California's Information Sciences Institute (B Root). These datasets are used throughout the analysis in Section 4 whenever we require detailed information about specific resolvers that exhibit anomalous behavior. Note, however, that other root servers might show different query patterns [19].

Query String	Which trust anchor(s)?
_ta-4a5c	Only *KSK-2010*
_ta-4a5c-4f66	Both *KSK-2010* and *KSK-2017*
_ta-3039	Has a non-IANA trust anchor
_ta-4a5c-4f66-8235	*KSK-2010* & *-2017* and a non-IANA trust anchor

Table 1: Root zone RFC 8145 trust anchor signals.

RSSAC Measurements. The ICANN Root Server System Advisory Committee (RSSAC) [20] advises ICANN about operational matters relating to the DNS root system. RSSAC defined a set of metrics that all root server operators are expected to publish on a daily basis [21]. The resulting data is published as YAML files, accessible through a public GitHub repository [22], with data going back to 2013. In this paper, we make use of the RSSAC002 data on traffic sizes to the root, as a proxy for DNSKEY queries in Section 4.3.2 and to estimate the impact of the increased DNSKEY RRset size in Section 4.3.3. The data is available for all root servers, except G Root.

Trust Anchor Signals. RFC 8145 [3] describes a protocol allowing DNSSEC validators to signal the keys in their trust anchor set. RFC 8145 signals are 16-bit "key tags," encoded as hexadecimal values in DNS queries. *KSK-2010* has key tag 19036, or 4a5c in hexadecimal. *KSK-2017* has keytag 20326, or 4f66 in hexadecimal. A validator that implements RFC 8145 periodically sends a query whose first label starts with the string "_ta-" followed by a hyphen-separated list of hexadecimal key tag values. It then appends the name of the zone to which the keys belong.[2] Table 1 shows root zone trust anchor signal strings and their meanings.

In this paper we use two RFC 8145 data sets: (i) all trust anchor signals received by A, B and J Root from up to 100,000 distinct IP addresses daily, and (ii) trust anchor signals provided to ICANN by most of the root server operators from up to 200,000 distinct IP addresses daily; ICANN provided us with a subset of this data covering February 1st to March 29th, 2018.

3.2 Active Measurements

Resolver State. By using only data collected at the root, we miss the perspective of the client. To add this perspective, we rely on public measurements [23], that make use of the RIPE Atlas measurement network [24]. An Atlas probe is a device from which we can actively send DNS queries through its recursive resolvers, pre-configured by the probe owner or learned through DHCP. This allows us to observe the transition from *KSK-2010* to *KSK-2017* (event **IV**) and the revocation of *KSK-2010* (event **V**) from the perspective of resolvers and measure whether they continue to validate DNSSEC signatures successfully. The public measurements we leverage consist of two queries sent every hour and check whether resolvers validate correctly. The first query asks for the A record of a domain with a valid signature, the second for a domain with

[2] In case of the root zone there is nothing to append. An example non-root zone trust anchor signal with appended zone is _ta-4b61.dlv.isc.org.

DNS response code		State
Valid Signature	Bogus Signature	
NOERROR	NOERROR	insecure
NOERROR	SERVFAIL	secure
SERVFAIL	other	bogus

Table 2: Combination of response codes indicating the state of the measured resolver.

a bogus signature. The response codes of both measurements can be combined (see Table 2) to establish if a resolver (i) does not validate DNSSEC signatures (state *insecure*), (ii) validates signatures correctly (state *secure*) or (iii) fails to validate (state *bogus*). Secure resolvers changing state to *insecure* or *bogus* at any stage of the rollover may be indicative of that resolver experiencing problems. In addition to the public measurements, we schedule our own measurement which queries each resolver for the DNSKEY RRset of the root, to measure uptake of *KSK-2017* during the rollover.

Using 10,004 RIPE Atlas probes (all probes available at the time of our measurement) and their recursive resolvers gives 18,277 vantage points (VPs), located in 3,647 autonomous systems (ASs). To find how many resolvers these VPs cover, we send hourly queries for a domain under our control, using the probe ID and a random string as a sub-label to avoid caching. Our authoritative name server responds with the IP address of the resolver that served the query. Using this method, we observe 35,719 upstream IPs located in 3,141 ASs over the period in which we conducted the measurement.

Root Sentinel. As discussed, RFC 8145 allows resolvers to signal which trust anchors it uses to upstream authoritative name servers. What was lacking, however, is a way for resolver users and other third parties to actively ask resolvers which trust anchors they use. This led to the introduction of RFC 8509, the so-called "Root Sentinel" [25]. Given that the specification was only finalized in December 2018, it could not reliably be used to monitor the root KSK rollover (although we do observe early implementations). We do, however, include Root Sentinel measurements to study adoption of this new form of telemetry and to observe the revocation of *KSK-2010* in 2019 from the perspective of resolvers.

The Root Sentinel is an *active* measurement mechanism. A client can send two special queries to resolvers to ask what trust anchors they currently have to validate DNSSEC responses. The first query type allows a client to ask if a DNSKEY with a certain key tag *is* a trust anchor, the second type allows a client to ask the inverse (whether a specific DNSKEY *is not* a trust anchor). The resolver returns a valid response to the first type if the specified key is a trust anchor, and a SERVFAIL error if it is not. For the second query type, the opposite behavior applies. Table 3 shows what the queries look like. Note, while RFC 8145 uses hexadecimally encoded key tags, RFC 8509 uses decimal key tags. Thus, to query for the presence of *KSK-2010* and *KSK-2017*, ...`-is-ta-19036` and ...`-is-ta-20326` are used.

Our goal is to examine (i) how many resolvers support Root Sentinel queries, and for those that do, (ii) if they correctly have the new key (*KSK-2017*) and remove the old key (*KSK-2010*) when it is revoked (event **V**). To do so, we set up a domain under our control. The name server for this domain is configured to return

Query String	Is <KEY-TAG> a trust anchor?	
	Yes	No
root-key-sentinel-is-ta-<KEY-TAG>	Valid response	SERVFAIL
root-key-sentinel-not-ta-<KEY-TAG>	SERVFAIL	Valid response

Table 3: RFC 8509 Root Sentinel queries

a DNSSEC-signed A record for Root Sentinel queries. We then use RIPE Atlas to issue four Root Sentinel queries (i.e., each of the two Root Sentinel queries for the old and new key) under our test domain. For this measurement, we extended our coverage of the global resolver population by including additional measurements using the Luminati proxy network [26]. This gives us more visibility in residential networks. Luminati is a paid HTTP/S proxy service enabling clients to route traffic via the Hola Unblocker Network. Luminati currently provides over 187 million potential exit nodes. When receiving an HTTP request, exit nodes send a DNS request to *their resolver* and then issue the HTTP/S request. This allows us to measure resolver behavior. For more details on using Luminati for network and DNS measurements, we refer to Chung et al. [27, 28].

3.3 Ethical Considerations

The measurement data collected at the root of the DNS consists of aggregate data (RSSAC002), telemetry signals (RFC 8145), DNSKEY queries and aggregates of popular queries for telemetry sources identified as showing non-standard behavior. Only in rare cases do we identify specific resolver operators (not end users) so we can contact them in order to gain an understanding of unexpected resolver behavior (cf. Section 4.3.2).

Most of our active measurements leverage well-established public measurement platforms, such as RIPE Atlas, where strict guidelines exist. The exception to this are our Luminati measurements. To use the Luminati service, we first note that we paid the operators of Luminati for access, and strictly follow their License Agreement [29]. The owners of exit nodes agreed to route Luminati traffic through their hosts. Furthermore, we took great care to ensure that all traffic only flowed toward domains under the authors' control, which serve empty web pages. Given that we are only interested in information about the RFC 8509 behavior of DNS resolvers, we discard any end user IP addresses from our logs.

4 ANALYSIS

The next sections discuss the most relevant events of the rollover (**I** – **VI** in Fig. 2), starting before the rollover (**I** – **III**) in Section 4.1, followed by the rollover itself (**IV**) in Section 4.2 and ending after the rollover (**V** – **VI**) in Section 4.3.

4.1 Before the Roll

4.1.1 Early RFC 8145 Data. RFC 8145, published April 2017, was quickly adopted by open source resolver implementers. BIND supports it from mid-2016 with the functionality enabled by default, Unbound since April 2017, enabling it by default in October 2017, and Knot since November 2017, again enabled by default.

We began looking for evidence of RFC 8145 signals in A/J Root data from May 2017. By September 2017 we see trust anchor signals from approximately 1,300 unique source IPs per day. Fig. 4 shows these early trust anchor signals. The *KSK-2010* line shows what

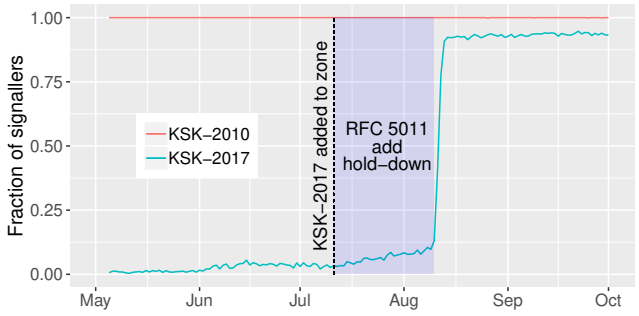

Figure 4: Early RFC 8145 trust anchor signals (2017).

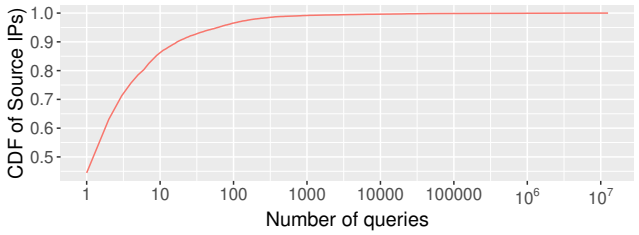

Figure 5: CDF of addresses vs. queries in B Root data sending only *KSK-2010* signals.

fraction of RFC 8145 sources sends signals for the old trust anchor, and the *KSK-2017* line shows signals for the new trust anchor. Note that these signals are independent; in other words: a single source may send signals for both *KSK-2010* and *KSK-2017*.

As Fig. 4 shows, initially almost all sources had only *KSK-2010*. There is some slight increase in uptake of *KSK-2017* starting in June, before *KSK-2017* was published in the root zone. This increase can be explained by installations that received the new trust anchor as part of a software update, or from those where an administrator manually added it. ISC, for example, added the new key to BIND's code repository on the same day it was made operational and published by IANA (February 2nd, 2017).

When *KSK-2017* is published in the root zone on July 11th, 2017, validators that implement RFC 5011 begin the process of accepting the new key. After seeing the key published (and correctly signed) for 30 continuous days (the RFC 5011 *Add Hold-Down Time*), a validator adds the new key to its trust anchor set. Thus, from August 10th, we observe a rapid rise in signalers reporting *KSK-2017* over the two days after the hold-down period ends. Because the TTL of the DNSKEY record set is 48 hours, the shift is not immediate.

After the 30-day hold-down ends, some 8% of signalers still do not report having *KSK-2017*. Operators watching this data hoped this population would continue to shrink. However, it remained at this level through the end of September. This is the primary reason why, on September 27th 2017, ICANN made the difficult decision to postpone the rollover [4]. As late as August 2019, around 1% of signalers still report only having *KSK-2010*.

4.1.2 Unusual KSK-2010 *RFC 8145 signalers.* During continued monitoring of the RFC 8145 signals, ICANN began observing two

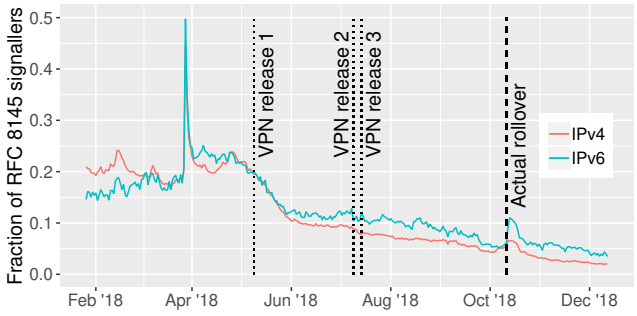

Figure 6: Addresses signaling only *KSK-2010*.

	Description	Count
A	Unique sources in ICANN data	1,206,840
B	Sources from **A** signaling *KSK-2010*	508,533
C	Sources from **B** sending only one signal	310,839
D	Unique Sources in ICANN data to B Root	309,140
E	Sources from **D** signaling *KSK-2010*	113,467
F	Sources from **E** signaling just once	16,403
G	Sources from **F** sending 1-9 queries	6,702

Table 4: Narrowing the observed data.

Query-Name	Count
`_ta-4a5c`	15,447
`.`	9,182
`VPN-PROVIDER.com`	3,156
`VPN-PROVIDER-ALTERNATE.com`	415
`_sip._udp.OTHER-DOMAIN.com`	86

Table 5: Top query names from anomalous sources.

unusual artifacts: (i) a large fraction of resolvers failed to pick up and trust *KSK-2017*, as measured by resolvers sending only RFC 8145 *KSK-2010* signals and seen in Fig. 6, and (ii) many of the data points came from IP addresses sending only small numbers of queries, as seen in Fig. 5. Note that the fraction of resolvers not trusting *KSK-2017* actually got worse, not better, between the end of Fig. 4 and the beginning of Fig. 6. These artifacts led to the question *"Why do so many new addresses appear that send RFC 8145 signals indicating they only trust* KSK-2010*?"*

To answer this question, we compare the RFC 8145 signal data from ICANN to all DNS queries arriving at B Root over a four week period from March 1st–29th, 2018. We focus this analysis on B Root, because unlike the data from ICANN which only contains RFC 8145 signals, for B Root we have full access to all queries received. We narrow the data to those addresses that behave unexpectedly: they send *a single signal* for *KSK-2010* to B Root, and send *only 1–9 other queries* to B Root in the period covered. The narrowing down of the full list of IP addresses ICANN observed to just these anomalously behaving addresses is shown in Table 4.

To test if there is any commonality in other query names sent by these sources, we extract and correlate the top query names sent by these addresses (shown in Table 5). Beyond the RFC 8145 signals

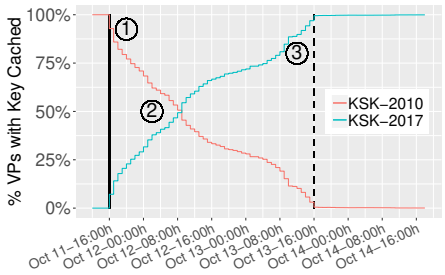

Figure 7: Key transition for all VPs.

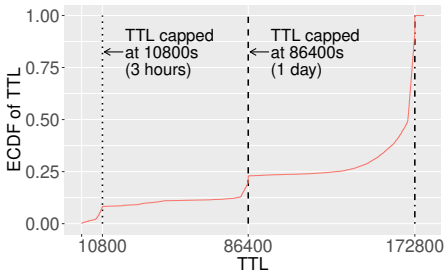

Figure 8: Reported DNSKEY TTL.

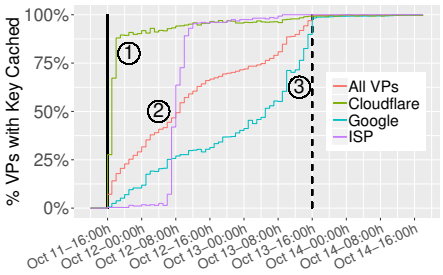

Figure 9: *KSK-2017* on large resolvers.

("_ta-4a5c") and queries for root-zone data (".") (period)), the next highest two requested names are a Virtual Private Network (VPN) provider's primary and secondary domain (anonymized in Table 5). This commonality in top queries strongly indicates the discovery of a likely cause of *KSK-2010* signals from sources sending otherwise low-volume traffic. Searching the VPN provider's software, taken from their Android release, revealed an embedded "root.key" file containing just *KSK-2010* and not *KSK-2017*. The embedded libraries found in the software also revealed a library name matching the Unbound project [30], a popular DNSSEC-validating resolver.

We contacted the VPN provider on April 17th, 2018. They confirmed our findings and indicated that multiple products were affected. Subsequently, they released updated versions of their product to address the issue, as marked in Fig. 6. The desktop software update had the most dramatic impact, significantly decreasing the number of *KSK-2010* signals seen at the root. The first mobile update with the new key set also showed a small dip in *KSK-2010* signals, though the second mobile update exhibited a less visible impact.

Key Takeaway Before the Roll. A single application can significantly influence trust anchor signaling, and the fact that it was an end-user application is largely responsible for the high number of signals. Given that DNSSEC validation in end-user applications will become more common in the future, this needs to be considered for future rollovers.

4.2 During the Roll

As *KSK-2010* signals returned to the 8% range by mid-2018, ICANN revised its plans for the rollover [31]. After community feedback on these plans, ICANN proceeded with the rollover [32]. On October 11th, 2018, at 16:00h UTC the KSK is rolled (event **IV**). From then on, root servers return a DNSKEY RRset signed with *KSK-2017*. In this section we show how resolvers picked up the new RRset. We then examine what happens to resolvers that do not have *KSK-2017* as a trust anchor, and how operators solve the problems this causes.

4.2.1 The Key Transition. To measure the transition from the old to the new RRset, we use RIPE Atlas probes (see Section 3.2) to send DNSKEY queries and then analyzed the results. Fig. 7 shows when resolvers drop the old RRset from their cache and query the root for the new one. [3] Right after the new key is published, resolvers begin showing cached signatures from *KSK-2017*. Within the first hour 7% of the resolvers have the new RRset. Sixteen hours later over 50% of resolvers have the new RRset. At 48 hours after the roll, the old RRset should have been removed from the caches of all resolvers; 99.5% of our vantage points return *KSK-2017* signatures at that point. After 11 more days, the last "lagging" vantage points pick up the new RRset (not shown in Fig. 7).

Because the root DNSKEY RRset has a TTL of 48 hours, we expected half of vantage points to have the new RRset after 24 hours. As Fig. 7 shows, however, this point is already reached after just 16 hours. In Fig. 8 we plot the TTLs for the root DNSKEY RRset as reported by each vantage point when it receives the new RRset for the first time. More than 20% of vantage points report a TTL that is lower than 1 day, and around 10% even report a TTL lower than three hours. This indicates that some resolvers cut the TTL to a value lower than 48 hours, also explaining why the new RRset was picked up earlier than expected.[4] What this also means is that had a failure occurred during the rollover, we would likely have seen this sooner than intuitively expected, which is important to consider for future rollovers.

Another thing that stands out in Fig. 7, are sudden "jumps" in the adoption of *KSK-2017* (marked ①–③). We correlate these jumps with adoption at resolvers often used by RIPE Atlas probes in Fig. 9. The jumps respectively correspond to adoption of the new RRset by Cloudflare (①), a German ISP (②) and Google (③). Operators of the Cloudflare resolvers publicly commented that someone used their web interface to purge the DNSKEY RRset of the root from the cache right after the rollover [34]. This explains why the resolvers fetched the new RRset soon after the roll. This spurred us to check if other operators *purposely flushed their caches* before or after the rollover to either keep the old status for as long as possible, or force the new situation as soon as possible. To find evidence, we looked for vantage points that report a TTL close to 48 hours just before or after the rollover. We find three resolvers that fetched the keyset just before the roll (effectively locking in the old situation for almost 48 hours). A large European ISP privately confirmed they did this to avoid problems right after the rollover, allowing them to monitor the news from other operators after the roll [35].

4.2.2 Impact on Validating Resolvers. Now that we know how resolvers picked up the new RRset, we check if they experience any problems once they have the new RRset. For resolvers that do experience problems, we expect them to either fail validating signatures (become *bogus*) or turn off validation altogether (become

[3] We published updates of this figure on social media and on the website of NLnet Labs to give the community insight into the progress of the roll.

[4] E.g., Unbound caches RRsets for a maximum of 24 hours by default [33].

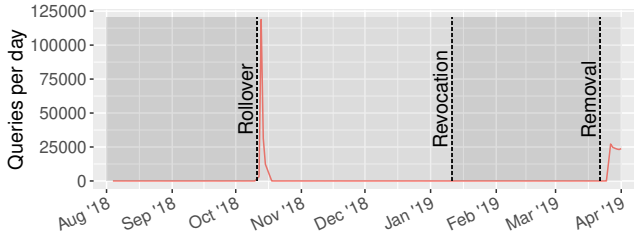

Figure 10: DNSKEY queries from ISP "EIR" to A/J Root.

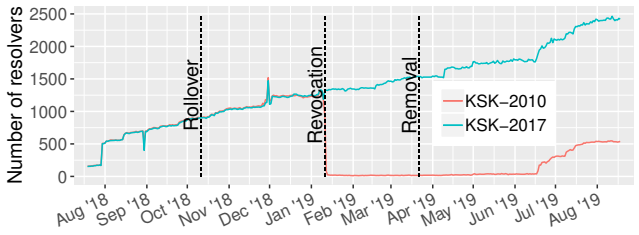

Figure 11: Root Sentinel observations with RIPE Atlas

insecure). We use RIPE Atlas measurements (see Section 3.2) to identify resolvers that were continuously *secure* 88 hours before the roll but turned *bogus* or *insecure* at any point within 56 hours after the roll.

We summarize resolver behavior observed through RIPE Atlas in Table 6. Row **A** shows the total number of resolvers observed during the rollover. Of these, 1,717 (**B**+**C**) always validate signatures correctly before the roll but 970 (2.7%) turn *bogus* and 747 (2.1%) *insecure* some time after. We check how often problematic resolvers query for the DNSKEY of the root, using DNS-OARC DITL data collected during the rollover (see Section 3.1). If a resolver changes state *and* sends more DNSKEY-queries, we conclude that this change is caused by problems with the rollover. We see DNSKEY queries from 519 sources at the root (**D**). Of these, 509 (**E**) send more DNSKEY queries after than before the roll. For 359 resolvers, the increase in DNSKEY queries exceeds 1.5 times (**F**). The majority, 342 resolvers (**G**), return to their normal DNSKEY query pattern within an hour. We assume operators intervened and fixed these resolvers. For 138 resolvers (**H**) we keep observing unusually high numbers of DNSKEY queries for over an hour. They only return to their normal behavior after a median of more than 39 hours. Only three resolvers (**I**) continue sending unusually high numbers of queries throughout the entire measurement period. The fact that more than 60% of the resolvers get fixed within one hour is a strong sign that resolvers in our data set are used actively and that operators noticed issues during the rollover relatively quickly. We discuss resolvers that send excessive numbers of DNSKEY queries in more detail in Section 4.3.2.

4.2.3 The User's Perspective. From the analysis above, we cannot gauge the actual impact on end users. During our measurements, 175 RIPE Atlas probes (1% of all vantage points) relied exclusively on one of the *bogus* resolvers (set **B** in Table 6), thus were not able to receive any valid response at some point after the rollover. More than 70% of these probes, however, suffered problems only an hour

	Upstream Resolvers	Count
A	Unique sources in RIPE Atlas data	35,719
B	↳ from **A** always secure before and bogus after	970
C	↳ from **A** always secure before and insecure after	747
D	↳ from **B** and **C** sending DNSKEY queries	519
E	↳ from **D** reach maximum DNSKEY queries after	509
F	↳ from **E** w. 1.5× DNSKEY queries after	359
G	↳ from **F** fixed within 1h	218
H	↳ from **F** fixed after 1h	138
I	↳ from **F** that did not get fixed	3

Table 6: Data of RIPE Atlas measurements.

or less. 166 probes could rely on at least one other resolver to serve their queries and were not affected by the failing resolver.

Other work [36] shows users move to public DNS providers in case of issues with the resolver of their ISP. Therefore, we analyzed if vantage points change to the public resolvers of Google, Cloudflare or OpenDNS. We found only two vantage points. One of these used the resolver of the Irish ISP EIR. This ISP experienced a well-publicized DNS outage [37] during the rollover, and the DNS community speculated this outage was caused by EIR's resolvers failing validation. Using the RIPE Atlas measurements, we identify the IP addresses of EIR's resolvers. Then, we count how many DNSKEY queries these resolvers send to A/J Root per day (see Fig. 10). Starting from October 12th, queries increase, reaching a peak one day after the roll and returning to normal after 3 days. Keeping in mind that RIPE Atlas probes actively switched resolvers at the same time, this is a strong sign that the outage of EIR was indeed caused by validation errors. Note, Fig. 10 shows the number of DNSKEY queries from EIR rising again after removal of *KSK-2010*. We discuss this increase Section 4.3.2.

Key Takeaways During the Roll. We observed few resolvers with serious problems. Where such problems occurred, they were solved promptly by operators. Less than 0.01% of the resolvers we monitored during the rollover experienced problems that lasted beyond our observation window.

4.3 After the Roll

We now discuss what happened after the rollover, from the point when all resolvers should have a DNSKEY RRset signed by *KSK-2017*, to the removal of *KSK-2010* from the root zone.

4.3.1 Revocation of KSK-2010. As discussed in Section 3.2, the Root Sentinel standard (RFC 8509) was published too late to be useful for the actual rollover. We can, however, study revocation of *KSK-2010* with resolvers that adopted this protocol. Using all RIPE Atlas probes, we send out Root Sentinel queries from August 2018 to August 2019. Fig. 11 shows the Root Sentinel signals observed over this period. As the figure shows, overall, the number of resolvers supporting Root Sentinel queries steadily increases to 2,419 resolvers in 720 ASs by the middle of August 2019. This is encouraging given the early stage of deployment of the protocol. After the revocation of the old key (event **V**), the number of resolvers with *KSK-2010* drops to almost zero while the number of resolvers with *KSK-2017* keeps increasing. Interestingly, some 20 resolvers continue to signal having *KSK-2010* in their trust anchor store. This implies either a manually configured trust anchor, or a failure in

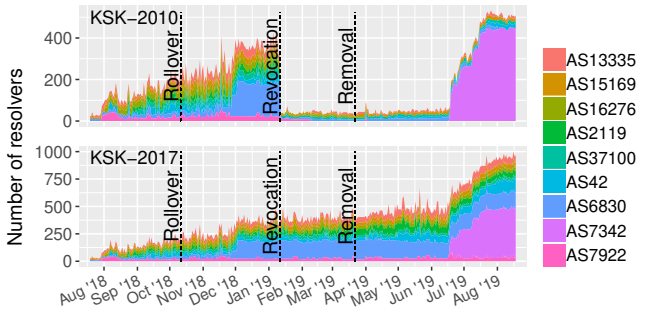

Figure 12: Top 9 ASs supporting Root Sentinel queries observed through RIPE Atlas

their RFC 5011 processing. Then, from the middle of June 2019, *KSK-2010* starts making a surprising comeback. We explain why further down in Section 4.3.4.

As RIPE Atlas provides a limited view, we also used Luminati to measure a total of 52,378 resolvers serving 589,928 exit nodes — from 210 countries and 7,867 ASs — over a period of 14 days from March 28th 2019. From these, we select resolvers on which we were able to test all four combinations of Root Sentinel queries (cf. Table 3). This leaves 21,563 resolvers, to which 385,520 exit nodes sent queries at least once. We further split these into resolvers that support Root Sentinel queries and ones that do not.[5] We finally determine which trust anchor(s) resolvers that support the Root Sentinel signal as present in their trust store. The vast majority — 21,056 (97.63%) resolvers from 5,311 ASs — do not support RFC 8509. These resolvers cover 330,891 (85.8%) exit nodes. Only 468 (2.2%) resolvers from 164 ASs support Root Sentinel queries and have only *KSK-2017*; these resolvers cover 33,266 (8.6%) exit nodes indicating that a few large ASs support RFC 8509 queries, including Telenor (Norway), Bezeq (Israel) and Meo (South Africa). We also note that 39 resolvers (0.19%) still signal they have *KSK-2010* configured.

Finally, we compare our observations through RIPE Atlas and Luminati. Fig. 12 shows the top 9 ASs with resolvers supporting RFC 8509 in our RIPE Atlas measurements. Comparing this to Luminati, we find that 43 resolvers from AS2119 (Telenor), 10 from AS16276 (OVH), 10 from AS6830 (Liberty Global), and 2 from AS7922 (Comcast), are observed in the same state through both RIPE Atlas and Luminati. Fig. 12 also shows a surprising increase of *KSK-2010* from June 2019, we explain why in Section 4.3.4.

4.3.2 Increase in DNSKEY *Queries.* As mentioned at the end of Section 4.2, we observed an increase in DNSKEY queries from certain resolvers at various stages of the roll. We analyse this phenomenon in more detail here, especially because of the sharp increase in queries after the revocation of *KSK-2010* to the extent that at some point a worrying amount — up to 10% — of traffic to the root consisted of DNSKEY queries.

We start by analyzing the total amount of DNSKEY queries to the root. DNSSEC validators must regularly verify their locally configured trust anchor(s) against the zone's published DNSKEY

[5]Note: a resolver that supports RFC 8509 correctly will return a valid response to *only one* of the two queries with the same key tag.

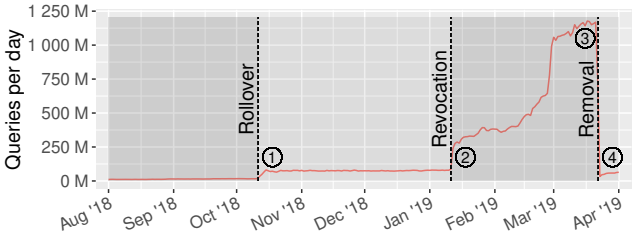

Figure 13: DNSKEY queries to A/J Root after the rollover.

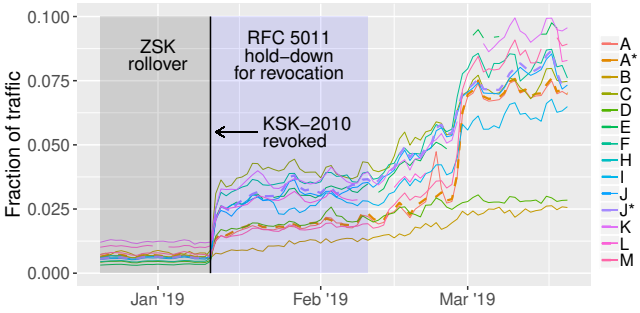

Figure 14: DNSKEY query increases for all root servers.

records. In other words: validators periodically issue DNSKEY queries for the root zone. Due to the retry behavior of implementations, a validator with an out-of-date trust anchor is likely to send more than the normal amount of DNSKEY queries. This behavior was already observed in 2009 — before the root zone was signed — during a KSK rollover for an `in-addr.arpa` zone operated by RIPE. The group investigating that incident called it "rollover and die" [38].

Just after the root KSK rollover on October 11th, 2018, root name servers observed an increase in DNSKEY queries. Fig. 13 shows the query rate for A/J Root. The increase was gradual, ramping up over the course of two days as the DNSKEY RRset timed out from resolver caches. Pre-rollover the rate was around 15 million queries per day. Post-rollover it increased five-fold, to 75 million (①). An even more dramatic increase occurred when *KSK-2010* was revoked (Event **V** in Fig. 2). Immediately after the revocation, A/J Root see a sudden spike in DNSKEY queries (②), jumping from 75 million to over 200 million queries per day within 24 hours. The DNSKEY query rate continued to climb over the following weeks and months, exceeding one billion per day in March 2019 (③). At this point, DNSKEY queries comprised 7% of the total traffic received at A/J Root. The final phase of the rollover sees *KSK-2010* removed from the root zone on March 22nd, 2019. To everyone's surprise, the DNSKEY query rate dropped dramatically immediately after *KSK-2010* was removed. As Fig. 13 shows (④), the rate dropped and slowly crept back up to post-rollover levels as seen in October, November, and December 2018.

Fig. 13 only shows data for A/J Root. To confirm similar increases at other root servers, we use the RSSAC002 data (see Section 3.1). The RSSAC002 data does not have a dataset specifically identifying DNSKEY queries, however we can infer the presence of such queries by examining the response size dataset. Fig. 14 shows the percent of responses between 1232–1472 bytes as solid lines. The dashed

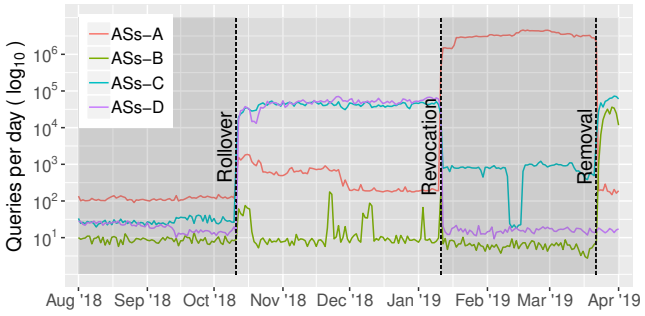

Figure 15: AS DNSKEY query patterns to A/J Root.

lines — marked A* and J* — are actual A/J Root traffic and show a strong correlation. Not all root servers saw the same increase in queries, but we currently lack sufficient information to explain this.

Deeper inspection of the A/J Root traffic shows vastly differing DNSKEY query patterns on a per AS basis. Fig. 15 shows the average of multiple ASs whose DNSKEY queries exhibit distinct patterns at different times throughout the rollover. Some ASs expressed a systemic trend of increased DNSKEY queries post-rollover and even higher rates post-revocation (ASs-A). Other ASs only exhibited an increase in DNSKEY queries after the removal of *KSK-2010* (ASs-B). Likewise, some ASs show increased rates post-rollover until revocation (ASs-D) and again after removal (ASs-C). To better profile these resolvers, we issued `version.bind` queries to IP addresses expressing the various behaviors. While the response rate was low (4.3% of ±18K resolvers), the majority returned older versions of BIND (45% BIND 9.9.x, 34% BIND 9.8.x, and 13% BIND 9.10.x).

Explaining the increase in DNSKEY queries. To find the cause of the increased query rates, we studied traffic coming from individual, high-volume sources. Outreach efforts at a global DNS scale are challenging, but we were able to contact multiple operators willing to help diagnose the DNSKEY query increase. One operator (a large French cloud hoster), stated their servers were running BIND 9.8.2 on CentOS 6.7 and the logs contained large numbers of validation errors. Another set of sources identified as sending excessive DNSKEY queries to the root, came from 8 addresses in a single subnet at a large midwestern university. Their staff quickly identified a DNS lab exercise that had been left running inside virtual machines (VMs). After shutting down the VMs, we confirmed that the excess DNSKEY traffic had stopped. From the university's class instructions, we hypothesized that the DNSKEY query spikes were the result of ISC's BIND software running in a specific state: (i) the DNSSEC `managed keys` did not contain *KSK-2017*, but did contain *KSK-2010*; (ii) the `dnssec-enable` flag was set to `false`; and (iii) the `dnssec-validation` flag was unset, leaving it in its default state of `yes`.

To verify this hypothesis, we performed experiments to test for bugs related to BIND's behavior in the absence of a valid trust anchor. We set up a *BIND 9.11.5-P4* resolver (the oldest supported release at the time), configuring it as per the university's class instructions. We also ensured that BIND's managed keys file contained only *KSK-2010*. Then, we ran 20 experiments in which we started a fresh copy of BIND configured as specified above. In each

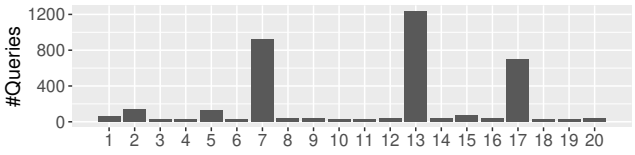

Figure 16: DNSKEY queries for root during experiments.

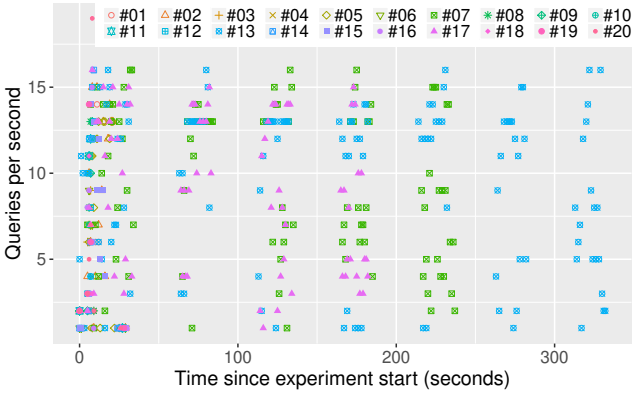

Figure 17: Time-normalized graph of experiments.

run, we sent ten sets of queries to BIND for test domains in seven TLDs at 30-second intervals, recording DNSKEY queries sent by the resolver, along with timestamps. Fig. 16 shows the results. Each experiment start time was normalized to zero and overlayed in Fig. 17, showing highly variable query patterns in each run (note experiments 7, 13 and 17).

Both plots show wide variations in behavior of the resolver under test. At times it behaves as expected, sending only a few DNSKEY queries after initializing. At other times, the resolver seems stuck in a state where every incoming request causes the resolver to send out a flurry of DNSKEY queries.

From the analysis of events **V** and **VI**, and the corresponding DNSKEY loads seen at the root (Fig. 13 and Fig. 14) we conclude there are likely two different bugs causing the increase in queries. One bug is likely the cause of the increase in DNSKEY queries shortly after the rollover (event **IV**) and after *KSK-2010* is removed (event **VI**). Another bug is likely the cause of the extreme query loads seen in Fig. 14, when *KSK-2010* was present but with the revoke bit set. We have reached out to the developers of BIND to confirm our hypotheses, but have not received any feedback as of September 13th, 2019. What remains unclear is why operators have not noticed this broken resolver behavior, as we expect these resolvers to return SERVFAIL errors to every query. We speculate only one resolver in a group is failing, with an alternate succeeding on behalf of their clients. This behavior is a well-known fact from other work [39].

To facilitate reproducibility, we published experiment configurations and scripts in a public GitHub repository [40].

4.3.3 Increased Response Size. Another potential risk during the rollover, identified in the 2016 Rollover Design Team report [2], was the increase in size of the DNSKEY RRset (see Section 2.2.1). When

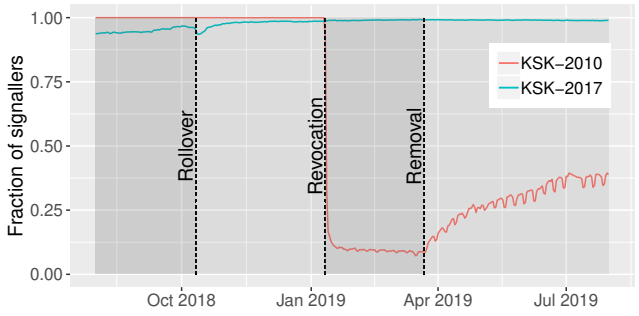

Figure 18: RFC 8145 signals August 2018 to August 2019.

KSK-2010 was revoked, this size reached its maximum value of 1,425 bytes. We analyzed if this increase hindered resolvers fetching the record set and, as a result, caused validation errors. While there are other moments during the rollover at which the response size is significantly higher than usual, we focus on the revocation event since that is when the maximum size was reached.

The first sign we expected to see if resolvers experience problems is an increase in fallback to TCP. We studied the RSSAC002 data concerning traffic types, and found no evidence of such an increase during revocation. Note, however, this data does not contain information on individual query types such as DNSKEY. If resolvers are also unable to fall back to TCP, then they may become unable to fetch the DNSKEY RRset altogether. We use the measurements from RIPE Atlas to detect whether any vantage points were unable to retrieve the DNSKEY RRset from the root after the increase in size. Resolvers are marked as unable to retrieve the DNSKEY RRset if they cannot fetch the RRset within 5 seconds.

Out of 17,925 vantage points, 1,975 (11%) are able to fetch the DNSKEY RRset before revocation, but fail to fetch it at least once 48 hours after the revocation. Only 67 of these (0.4%) never manage to fetch the key set after the revocation. Even though the IPv6 minimum MTU is 1,280 bytes, vantage points that contact resolvers via IPv6 did not fail more often than those using IPv4. We also found no resolvers that turned *bogus* after the revocation. This leads us to conclude that the increased response size during revocation only caused problems for a few resolvers and did not impact validators. This was also expected by the KSK rollover design team [2].

4.3.4 The return of KSK-2010. We end this section with a surprising comeback. As mentioned in Section 4.3.1, the number of resolvers that signal support for *KSK-2010* is on the rise again since its removal from the root zone DNSKEY RRset. This increase is also visible in the RFC 8145 signals sent to root servers. Fig. 18 shows that by the end of July 2019 almost 39% of signalers again report having *KSK-2010* in their trust anchor set. This, of course, raises the question why a retired trust anchor is making this comeback. While it is impossible to attribute the observed rise to a single source, we have convincing evidence of the most likely cause: DNS resolver software shipping with built-in or pre-configured trust anchors.

First, we note that the current long-term supported version of Ubuntu (18.04 LTS) ships with Unbound version 1.6.7, which supports RFC 8145. In addition, Ubuntu also includes a pre-configured trust anchor package that includes both *KSK-2010* and *KSK-2017*, and enables DNSSEC validation by default. We verified that, upon startup, Unbound loads both trust anchors, marks *KSK-2010* as "missing", but as the trust anchor is still configured, Unbound signals its presence in its RFC 8145 telemetry. Any installation of Ubuntu 18.04 LTS with Unbound that was running for at least 30 days[6] when *KSK-2010* was published as revoked will have cleaned up the old trust anchor. However, any installation (or re-installation) after February 20, 2019 could not complete RFC 5011 revocation and retained *KSK-2010* as a trust anchor. We also verified the behavior of another popular open source DNS resolver implementation on the same OS. Ubuntu 18.04 LTS ships with BIND version 9.11.3, which includes both *KSK-2010* and *KSK-2017* as built-in trust anchors. By default, the Ubuntu package for BIND is configured to perform DNSSEC validation using the built-in trust anchors. Upon startup, however, if BIND does not find a configured trust anchor in the DNSKEY RRset returned by the root servers, it will not signal this trust anchor in its RFC 8145 telemetry. This does not mean, however that the trust anchor is removed. We verified that BIND retains *KSK-2010* in its trust anchor file on disk, so if the key were ever to return in the root DNSKEY RRset we expect BIND to accept it as a valid trust anchor again.

Second, as mentioned previously, Fig. 12 shows an increase in *KSK-2010* beginning in the middle of June 2019 from a single network, AS7342. As it happens, this is the origin AS for Verisign's public DNS service.[7] The rise in *KSK-2010* signalers corresponds to an upgrade of the software used on the public DNS resolver. The newly deployed version supports the Root Sentinel (RFC 8509) and is packaged with a configuration that includes both *KSK-2010* and *KSK-2017* as trust anchors.

The two examples above explain most of the return of *KSK-2010* in Fig. 12 and at least some of the return in Fig. 18. They are illustrative of software still shipping with *KSK-2010* as trust anchor. This does not mean that these are the only examples, though, there are likely other packages with similar behavior. One question we have not discussed yet is whether the comeback of *KSK-2010* can be considered problematic. We discuss this in more detail in Section 6.

Key Takeaways After the Roll. The biggest problem during the whole process, arguably, occurred after the roll with the significant increase in DNSKEY queries. This problem was not foreseen in the design report [2], underlining the importance of independent studies of such major events on the Internet and confirming the need for meaningful telemetry. Additionally, it is clear trust anchor management is complex and that shipping trust anchors with software has long-lasting effects. We come back to this in Section 6.

5 RELATED WORK

As we discussed in the introduction, the root DNSSEC KSK rollover is a first-of-its-kind event. Thus, our discussion of related work will focus on earlier studies that have looked at the operation of the DNS root server system and the impact of DNSSEC on the performance of DNS resolvers. Huston [41] independently confirms our finding that the Irish ISP EIR suffered outages but does not provide a more thorough analysis.

[6] The RFC 5011 *Remove Hold-down Time.*
[7] https://www.verisign.com/en_US/security-services/public-dns/index.xhtml

The earliest work to study DNS traffic to root servers by Danzig et al. [42] dates back to 1992, five years after DNS was adopted as the Internet's naming system [43]. This study illustrates that software bugs that cause excessive traffic are a problem of all ages, as they find multiple bugs in algorithms meant to improve DNS resilience. In 2001, Brownlee et al. [44] study almost two weeks of traffic to F Root. Again, they find a surprising amount of problematic traffic to the root, with 14% of queries consisting of malformed address (A) queries. In 2003, Wessels et al. [45] studied 24 hours of F Root traffic and concluded an astonishing 98% of queries were malformed or unnecessary. Since 2006, DNS-OARC collects so-called Day-in-the-Life (DITL) datasets [18], which typically includes traffic to most root servers. In 2008, Castro et al. [19] analyzed three years of DITL data to characterise root server traffic and also found that 98% of queries were unnecessary.

Apart from studying traffic at the root, past work also looked at operational changes to the root system. A particularly impactful event is the change of the IP address of a root server. Since resolvers have to be configured *a priori* with the IP addresses of root servers to bootstrap DNS resolution, such events have a major impact. Many root servers have undergone such changes, and Lentz et al. [46] study one such change for D Root in an academic paper in 2013. This study concludes that such address changes take a long time to propagate to the global resolver population, with the old address still seeing significant amounts of traffic months after the change. The authors suggest that such IP address changes may actually be beneficial, as they serve as some form of a "garbage collection" for old implementations. A similar notion could be said to apply to rollovers of the root KSK. In 2015, Wessels et al. [47] show how the aftereffects of an address change linger, finding that the old IP address for J Root still receives on average 400 queries per second from some 130,000 sources *thirteen years* after the address change.

The effects of the root KSK rollover on resolvers studied in this paper are part of the impact of DNSSEC on resolvers. Earlier work studies other aspects of the impact of DNSSEC, including the performance impact of DNSSEC validation [48–51] and the risks, in terms of availability and security, of packet fragmentation of large DNSSEC responses [11, 52]. Even though [11] conclude that up to 10% of resolvers could have problems handling larger DNSSEC responses, we did not observe failures when the DNSKEY response size increased. Other popular DNSSEC signed zones have served records larger than 1,425 bytes and validating resolvers probably took measures to handle large responses already. Finally, the way DNSSEC is organized as a Public Key Infrastructure is highly relevant for the root KSK rollover studied in this paper. Yang et al. provide a detailed overview of why the DNSSEC PKI is organized the way it is today [53].

6 DISCUSSION AND RECOMMENDATIONS

Improving Telemetry. A key challenge faced during the KSK rollover was sparse and distorted telemetry from resolvers. Ideally, those responsible for the rollover would want to know both the exact state of resolvers (in terms of DNSSEC validation) and how important these resolvers are (in terms of the number of clients relying on them). This provides actionable intelligence that allows prioritisation of "important" resolvers (serving millions of users).

	RFC 8145	**RFC 8509**
Signaling	Automatic	Requires query
Which TAs are revealed	All configured	Only those queried
Supports non-root TAs	Yes	No
Collection method	Passive	Active
Vulnerable to manipulation	Yes	Only to on-path attackers

Table 7: Supported features of existing telemetry.

Clearly, during the root KSK rollover discussed in this paper such comprehensive telemetry was not available. While RFC 8145 saw significant deployment before the rollover, it was difficult to interpret its signals. This was mostly due to four reasons: first, RFC 8145 only allows for passive observations by — in this case root — DNS operators. Thus, in case of problems, it is impossible to query resolvers for further state information. Second, there is no telemetry on the query volume a resolver processes, making it hard to judge how relevant or risky a resolver with problems is. Third, RFC 8145 may propagate through upstream systems (NATs, DNS forwarders, caches and other middle-boxes), leading to distorted signals and hiding systems with actual problems. Fourth, although we have not seen any evidence of tampering, an attacker could artificially inflate the number of resolvers that have not acquired the new key by spoofing RFC 8145 telemetry signals. Such an attack could adversely influence the decision-making process around whether or not to proceed with a planned rollover. Despite the limitations of RFC 8145, however, *without it* ICANN and the DNS community would have been completely blind and some problems were actually solved due to RFC 8145 telemetry.

The Root Sentinel (RFC 8509) addresses the first limitation of RFC 8145. It uses active measurements from the client perspective to establish the DNSSEC trust anchors configured on a resolver. While standardized too late to be of use during the current rollover, our analysis shows RFC 8509 is seeing rapid deployment and provides useful signals as of September 13th, 2019. Nevertheless, RFC 8509 also suffers from the second and third limitations discussed for RFC 8145 albeit with different signal distortion (e.g. assuming a Root Sentinel query is sent to resolvers at a large ISP while it is actually handled by a local forwarder). Table 7 summarizes the supported features of the existing telemetry protocols.

Based on our analysis of the current rollover, we recommend exploring incremental improvements to both RFC 8145 and RFC 8509. The quality of such signaling would be greatly improved if it were possible to identify true signal sources, identify cases where signals are forwarded, and estimate the number of users being serviced. We recognize that there are serious concerns around such detailed signaling. Weighing the tradeoffs requires further thought and debate in the community.

Another issue compounding the difficulties of interpreting resolver validation problems is the ambiguity of the SERVFAIL error code validators send upon failure. Effectively only by combining results from different measurements (cf. Table 2) can we be reasonably confident that a resolver has issues with DNSSEC validation. We therefore strongly support a draft under review in the IETF that proposes to send extended error codes for DNSSEC failures [54].

Introducing a Standby Key. There is an ongoing debate in the DNS community about introducing a KSK standby key in the root zone by default [55]. Effectively, because the rollover was postponed by a year, this has already been tested for a single standby key, without leading to issues with, e.g., response sizes. We therefore think it safe to introduce such a standby key as multiple community members have suggested. An immediate benefit of this is that resolvers are much more likely to pick up the new key if it is pre-published for a longer period. Given the rollover policy of the root [1], such a standby key could even be published years in advance.

Trust Anchor Distribution. The 2018 KSK rollover was the first time a large population of DNSSEC validators needed to update their trust anchor. At the start of the process, the design team expected RFC 5011 to be the main means through which validators keep their trust anchors up to date [2]. Our observations suggest that where RFC 5011 was used, it generally worked as intended. In the few instances where problems did occur, this was either due to validators lacking permission to persist state to disk, or loss of state due to, e.g. container or virtual machine teardown and reinitialisation. The latter issue has the potential to become a bigger problem moving forward, as the proliferation of container technologies was not envisioned when RFC 5011 was authored 11 years ago. Lastly, we are also beginning to see DNSSEC validation in end user applications (e.g. the VPN client from Section 4.1.2), often with hard-coded trust anchors (a search on GitHub yields thousands of examples of this). This raises the question if *in-band* updates through RFC 5011 remain the main means for trust anchor management going forward.

As noted earlier, some resolver implementations distribute trust anchors in their software packages (thus these get refreshed with software updates). While this works to some extent, it does not scale to encompass applications performing validation. Additionally, we observed that there may be significant delays when retiring trust anchors, as evidenced by the surprising comeback of *KSK-2010*.

Based on these results, we advocate that the preferred method to distribute trust anchors should be with operating systems *out-of-band*. Some distributions (e.g. Debian Linux) have already started doing so. Applications can then rely on the OS and we strongly urge against hard-coding of trust anchors. In addition to this, OS distributors should tightly manage these trust anchors when they are replaced. In Section 4.3.4, we ended with the question if the retention of the retired *KSK-2010* was problematic. On the face of it, the answer to this question is "No", since the key was retired according to a schedule, and all copies of the key have now been destroyed. Consider, however, two scenarios, one in which a key is revoked because it has been compromised, and one in which the algorithm for the key has been compromised. It is evident that a speedy retraction of such a key as a trust anchor is imperative, and it is also evident that the current practice we observed does not suffice. Given the inertia of solving this issue Internet-wide, we would recommend an additional security practice: if a key needs to be revoked, then the root `DNSKEY` RRset should include the revocation signal until there is a reasonable certainty that systems have been updated to remove the trust anchor. This practice guarantees that software that correctly implements RFC 5011 will not use the compromised key as a trust anchor.

7 CONCLUSIONS

In this paper we provide a comprehensive analysis of the very first DNSSEC Root KSK Rollover. We show the rollover did not pass without problems: hundreds of actively used resolvers failed to validate signatures at some point during the rollover. Nevertheless, this is only a minute share of the total resolver population and most problems were fixed quickly. Additionally, thousands of resolvers exhibit anomalous behavior during the rollover process, though it remains unclear if this caused problems for end users. The significant traffic increase to root servers, seen after the revocation of *KSK-2010* requires attention from the DNS community with future rollovers in mind. We demonstrated that at least some of these queries can likely be attributed to bugs in resolver software.

We also demonstrate that telemetry, used to measure deployment of new keys, was significantly distorted by a single application (a VPN client). We analyzed a complementary protocol, which while potentially a valuable addition, still has drawbacks. Based on our experiences, we provide recommendations for incremental improvements to both protocols. In addition to this, we observe that trust anchor distribution — which the rollover design team expected to happen mostly *in-band* — requires attention for future rollovers, and provide recommendations for alternatives.

While, of course, our work focused heavily on anomalies, our analysis supports ICANN's conclusion that the rollover was indeed an overall success. As with earlier changes to the root system, some systems will fail and this study shows that the Root KSK rollover was no different. These failures, however, were limited to a very small set of resolvers and got fixed fast, limiting the impact. This gives us confidence that this first ever rollover certainly should not be the last.

Finally, taking a step back from the specifics of the DNS, there are valuable lessons to be learned from this event that apply much more broadly to Internet protocols. Firstly, the experience with this event shows that telemetry is a key factor in the understanding of, and decision-making for, major changes to the Internet. The event is also demonstrative of the well-known inertia of the installed base of networking software across the Internet that hampers the deployment of such telemetry enhancements, and underlines what others in the network research community have argued about making measurability an explicit concern when designing protocols [56]. Second, there are lessons to be drawn about trust anchor management. The more different places in which trust anchors are stored (i.e. in different applications and services), the harder it becomes to predictably manage them. We posit that trust anchors should preferably be managed centrally, in the OS. While not a perfect solution, it limits the risk of hard-coded or mismanaged trust anchors. This is a lesson that equally applies to other Public Key Infrastructures.

ACKNOWLEDGEMENTS

The authors would like to thank the following organisations (in alphabetical order): Amazon, DNS-OARC, ICANN, NIC.at, OVH, Purdue University, RIPE and SURFnet. Furthermore, we would like to thank Anna Sperotto, Evan Hunt, our shepherd Matthew Luckie, Ondřej Surý, and the anonymous IMC reviewers. This research was supported in part by NSF grants CNS-1850465, CNS-1901090 and EC H2020 Project CONCORDIA GA 830927.

REFERENCES

[1] IANA. DNSSEC Practice Statement for the Root Zone KSK Operator. https://www.iana.org/dnssec/dps/ksk-operator/ksk-dps.txt, 2016.

[2] KSK Rollover Design Team. Root Zone KSK Rollover Plan. https://www.iana.org/reports/2016/root-ksk-rollover-design-20160307.pdf, 04 2016.

[3] D. Wessels, W. Kumari, and P. Hoffman. Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC). RFC 8145 (Proposed Standard), April 2017. Updated by RFC 8553.

[4] ICANN. KSK Rollover Postponed. https://www.icann.org/news/announcement-2017-09-27-en, 2017.

[5] ICANN Board. Board Approval of KSK Roll. https://www.icann.org/resources/press-material/release-2018-09-18-en, 2018.

[6] ICANN. Review of the 2018 DNSSEC KSK Rollover. https://www.icann.org/en/system/files/files/review-2018-dnssec-ksk-rollover-04mar19-en.pdf, 03 2019.

[7] Ramaswamy Chandramouli and Scott Rose. Secure Domain Name System (DNS) Deployment Guide. *NIST Special Publication*, 800, September 2006.

[8] Verisign DNSSEC PMA. DNSSEC Practice Statement for the Root Zone ZSK Operator. https://www.iana.org/dnssec/dps/zsk-operator/dps-zsk-operator-v2.0.pdf, 2017.

[9] NTIA. NTIA Announces Intent to Transition Key Internet Domain Name Functions. https://www.ntia.doc.gov/press-release/2014/ntia-announces-intent-transition-key-internet-domain-name-functions, 2014.

[10] ICANN. Operational Plans for the Root KSK Rollover. https://www.icann.org/resources/pages/ksk-rollover-operational-plans, 2016–2018.

[11] Gijs Van Den Broek, Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. DNSSEC Meets Real World: Dealing with Unreachability Caused by Fragmentation. *IEEE Communications Magazine*, 52(4):154–160, 6 2014.

[12] Christian Kreibich, Nicholas Weaver, Boris Nechaev, and Vern Paxson. Netalyzr: Illuminating the Edge Network. In *Proceedings of ACM IMC 2010*, pages 246–259. ACM, 2010.

[13] M. StJohns. Automated Updates of DNS Security (DNSSEC) Trust Anchors. RFC 5011 (Internet Standard), September 2007.

[14] J. Abley, J. Schlyter, G. Bailey, and P. Hoffman. DNSSEC Trust Anchor Publication for the Root Zone. RFC 7958 (Informational), August 2016.

[15] NLnet Labs. Man-Page: Unbound Anchor. https://www.nlnetlabs.nl/documentation/unbound/unbound-anchor/.

[16] Moritz Müller, Matthew Thomas, Duane Wessels, Wes Hardaker, Taejoong Chung, Willem Toorop, and Roland van Rijswijk-Deij. Roll Roll Roll Your Root: Accompanying Data Sets. https://github.com/SIDN/RollRollRollYourRoot.

[17] Internet Assigned Numbers Authority (IANA). Root Servers. https://www.iana.org/domains/root/servers.

[18] DNS Operations and Analysis Center (DNS-OARC). Day-in-the-Life Datasets. https://www.dns-oarc.net/oarc/data/ditl.

[19] Sebastian Castro, Duane Wessels, Marina Fomenkov, and Kimberly Claffy. A Day at the Root of the Internet. *ACM SIGCOMM Computer Communication Review*, 38(5):41–46, 2008.

[20] ICANN. Root Server System Advisory Committee. https://www.icann.org/groups/rssac.

[21] RSSAC Caucus. RSSAC002 version 3 – RSSAC Advisory on Measurements of the Root Server System, Jun 2016.

[22] RSSAC. RSSAC002 Datasets. https://github.com/rssac-caucus/RSSAC002-data.

[23] Roland van Rijswijk-Deij, Taejoong Chung, David Choffnes, Alan Mislove, and Willem Toorop. The Root Canary: Monitoring and Measuring the DNSSEC Root Key Rollover. In *Proceedings of the 2017 SIGCOMM Posters and Demos, Part of ACM SIGCOMM 2017*, Los Angeles, CA, USA, 2017. ACM Press.

[24] RIPE NCC Staff. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal (IPJ)*, 18(3), Sep 2015.

[25] G. Huston, J. Damas, and W. Kumari. A Root Key Trust Anchor Sentinel for DNSSEC. RFC 8509 (Proposed Standard), December 2018.

[26] Luminati IO. Residential IP and Proxy Service for Businesses. https://luminati.io/, May 2018.

[27] Taejoong Chung, David Choffnes, and Alan Mislove. Tunneling for Transparency: A Large-Scale Analysis of End-to-End Violations in the Internet. In *Proceedings of ACM IMC 2016*, 2016.

[28] Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. In *Proceedings of USENIX Security 2017*, 2017.

[29] Luminati. Luminati End User License Agreement. https://luminati.io/license.

[30] NLnet Labs. Unbound DNS Resolver. https://www.unbound.net/.

[31] ICANN. 2018 KSK Rollover Operational Implementation Plan. https://www.icann.org/en/system/files/files/2018-ksk-roll-operational-implementation-plan.pdf, 04 2018.

[32] ICANN, Office of the CTO. Staff Report of Public Comment Proceeding. https://www.icann.org/en/system/files/files/report-comments-ksk-rollover-restart-23apr18-en.pdf, 04 2018.

[33] NLnet Labs. Man-Page: unbound.conf. https://nlnetlabs.nl/documentation/unbound/unbound.conf/.

[34] Ólafur Guðmundsson. DNSKEY cache purge. Comment at the mic during DNS-OARC 29 meeting in Amsterdam, https://www.youtube.com/watch?v=yT51FwPG0jE&t=6782, Oct 2018.

[35] Not Disclosed. European ISP flushing DNSKEY from cache before the rollover. Private correspondence, Oct 2018.

[36] Wouter B De Vries, Roland Van Rijswijk-Deij, Pieter-Tjerk de Boer, and Aiko Pras. Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–8. IEEE, 2018.

[37] Stephen Murphy. 'Significant percentage' of Eir customers affected by broadband outage. https://www.rte.ie/news/2018/1013/1002966-eir-outage/, October 2018.

[38] Geoff Houston. Roll Over and Die? http://www.potaroo.net/ispcol/2010-02/rollover.html, February 2010.

[39] Geoff Houston. Measuring the Root Zone KSK Trust. https://blog.apnic.net/2018/04/11/measuring-the-root-zone-ksk-trust/, April 2018.

[40] Wes Hardaker. Configurations and Scripts to Test BIND Behavior in the Absence of a Valid Trust Anchor. https://github.com/hardaker/isc-bind-dnskey-bug-test.

[41] Geoff Huston. APNIC Blog: Analyzing the KSK Roll. https://labs.apnic.net/?p=1181, 10 2018.

[42] Peter B Danzig, Katia Obraczka, and Anant Kumar. An Analysis of Wide-Area Name Server Traffic. In *Proceedings of ACM SIGCOMM 1992*, pages 281–292, Baltimore, MD, USA, 1992. ACM Press.

[43] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Internet Standard), November 1987.

[44] N. Brownlee, K.C. Claffy, and E. Nemeth. DNS Measurements at a Root Server. In *Proceedings of IEEE GLOBECOM 2001*, volume 3, pages 1672–1676, San Antonio, TX, USA, 2001. IEEE Computer Society.

[45] Duane Wessels and Marina Fomenkov. Wow, That's a lot of packets. In *Proceedings of the Passive and Active Network Measurement Workshop (PAM 2003)*, San Diego, CA, Apr 2003. PAM.

[46] M Lentz, D Levin, J Castonguay, N Spring, and B Bhattacharjee. D-mystifying the D-root Address Change. In *Proceedings of ACM SIGCOMM 2013*, pages 57–62, Barcelona, Spain, 2013. ACM Press.

[47] Duane Wessels, Jason Castonguay, and Piet Barber. Thirteen Years of "Old J-Root". In *DNS-OARC 24*, Montréal, Canada, 2015.

[48] Bernhard Ager, Holger Dreger, and Anja Feldmann. Predicting the DNSSEC Overhead Using DNS Traces. In *Proceedings of the 40th annual IEEE Conference on Information Sciences and Systems, CISS 2006*, pages 1484–1489, Princeton, NJ, USA, 2007. IEEE Comput. Soc.

[49] Wouter C A Wijngaards and Benno J. Overeinder. Securing DNS: Extending DNS Servers with a DNSSEC Validator. *IEEE Security and Privacy*, 7(5):36–43, 2009.

[50] Daniel Migault, Cédric Girard, and Maryline Laurent. A Performance View on DNSSEC Migration. In *Proceedings of the 6th International Conference on Network and Service Management (CNSM 2010)*, pages 469–474, Niagara Falls, Canada, 2010. IFIP.

[51] R. Van Rijswijk-Deij, K. Hageman, A. Sperotto, and A. Pras. The Performance Impact of Elliptic Curve Cryptography on DNSSEC Validation. *IEEE/ACM Transactions on Networking*, PP(99), 2016.

[52] Amir Herzberg and Haya Shulman. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *2013 IEEE Conference on Communications and Network Security, CNS 2013*, pages 224–232, 2013.

[53] Hao Yang, Eric Osterweil, Dan Massey, Songwu Lu, and Lixia Zhang. Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC. *IEEE Transactions on Dependable and Secure Computing*, 8(5):656–669, 2011.

[54] Warren "Ace" Kumari, Evan Hunt, Roy Arends, Wes Hardaker, and David C Lawrence. Extended DNS Errors. Internet-Draft draft-ietf-dnsop-extended-error-05, Internet Engineering Task Force, March 2019. Work in Progress.

[55] Various Authors. KSK Rollover Mailing List Archive, March 2019. https://mm.icann.org/pipermail/kskA-rollover/2019-March/thread.html.

[56] Mark Allman, Robert Beverly, and Brian Trammell. Principles for Measurability in Protocol Design. *ACM SIGCOMM Computer Communication Review*, 47(2):2–12, 2017.