SAC063
SSAC Advisory on DNSSEC Key Rollover in the Root Zone

SSAC
ICANN Security and Stability Advisory Committee

An Advisory from the ICANN
Security and Stability Advisory Committee (SSAC)
07 November 2013

Preface

This is an Advisory to the ICANN Board from the Security and Stability Advisory Committee (SSAC) on DNSSEC Key Rollover in the Root Zone. The SSAC advises the ICANN community and Board on matters relating to the security, stability and integrity of the Internet's naming and address allocation systems. This includes operational matters (e.g., matters pertaining to the correct and reliable operation of the root name system), administrative matters (e.g., matters pertaining to address allocation and Internet number assignment), and registration matters (e.g., matters pertaining to registry and registrar services). SSAC engages in ongoing threat assessment and risk analysis of the Internet naming and address allocation services to assess where the principal threats to stability and security lie, and advises the ICANN community accordingly. The SSAC has no official authority to regulate, enforce, or adjudicate. Those functions belong to others, and the advice offered here should be evaluated on its merits.

A list of the contributors to this Advisory, references to SSAC members' biographies and statements of interest, and SSAC members' objections to the findings or recommendations in this Advisory are at the end of this Advisory.

Table of Contents

Executive Summary

There is consensus in the security and Domain Name System (DNS) communities that the root zone DNS Security Extensions (DNSSEC) system poses unique challenges for standard DNSSEC practices. While there is agreement that an eventual root zone Key-Signing Key (KSK) rollover is inevitable regardless of whether that rollover is caused by a key compromise or other factors, there is no solid consensus in the technical community regarding the frequency of routine, scheduled KSK rollovers.

In this Advisory the Security and Stability Advisory Committee (SSAC) addresses the following topics:

- Terminology and definitions relating to DNSSEC key rollover in the root zone;
- Key management in the root zone;
- Motivations for root zone KSK rollover;
- Risks associated with root zone KSK rollover;
- Available mechanisms for root zone KSK rollover;
- Quantifying the risk of failed trust anchor update; and
- DNS response size considerations.

The SSAC proposes the following five recommendations for consideration and discussion:

Recommendation 1: Internet Corporation for Assigned Names and Numbers (ICANN) staff, in coordination with the other Root Zone Management Partners (United States Department of Commerce, National Telecommunications and Information Administration (NTIA), and Verisign), should immediately undertake a significant, worldwide communications effort to publicize the root zone KSK rollover motivation and process as widely as possible.

Recommendation 2: ICANN staff should lead, coordinate, or otherwise encourage the creation of a collaborative, representative testbed for the purpose of analyzing behaviors of various validating resolver implementations, their versions, and their network environments (e.g., middle boxes) that may affect or be affected by a root KSK rollover, such that potential problem areas can be identified, communicated, and addressed.

Recommendation 3: ICANN staff should lead, coordinate, or otherwise encourage the creation of clear and objective metrics for acceptable levels of “breakage” resulting from a key rollover.

Recommendation 4: ICANN staff should lead, coordinate, or otherwise encourage the development of rollback procedures to be executed when a rollover has affected operational stability beyond a reasonable boundary.

Recommendation 5: ICANN staff should lead, coordinate, or otherwise encourage the collection of as much information as possible about the impact of a KSK rollover to provide input to planning for future rollovers.

1. Introduction

There is consensus in the security and Domain Name System (DNS) communities that the root zone DNS Security Extensions (DNSSEC) system poses unique challenges for standard DNSSEC practices. While there is agreement that an eventual root zone Key-Signing Key (KSK) rollover is inevitable regardless of whether that rollover is caused by a key compromise or other factors, there is no solid consensus in the technical community regarding the frequency of routine, scheduled KSK rollovers.

This Advisory explores the range of possible root zone KSK rollover scenarios and articulates many of the complications and complexities unique to the handling of root zone keys. As such, this Advisory is intended to facilitate discussion that will more fully examine the costs, risks, and benefits for the various root zone KSK rollover scenarios.

The intended audience of this Advisory is the Internet Corporation for Assigned Names and Numbers (ICANN) Board of Directors and others who have a basic familiarity with concepts related to the DNS in general and DNSSEC in particular and who are interested in understanding the issues related to transitioning an old root zone KSK to a new KSK (root key rollover). A basic level of understanding of public key cryptography will also be helpful.

The goals of this Advisory are to describe the DNSSEC root zone KSK rollover problem space at a high level, identify issues that constrain possible solutions, and where appropriate, make recommendations to the ICANN Board related to root zone KSK rollover. This document will provide context and background to allow readers to understand solutions being explored among the DNS technical community. It is a specific non-goal of this document to recommend specific solutions for root zone KSK rollover.

2. Definitions Relating to Root Zone KSK Rollover

This document makes use of various terms that are sometimes overloaded or whose meanings might otherwise be ambiguous. The definitions that follow provide clarity on how these terms are used within this document. The SSAC acknowledges that in some cases the definitions given here differ from conventional usage. These definitions are grouped alphabetically.

Chain of Custody

The Chain of Custody refers to a documented trail of control, disposition, and transfer of sensitive materials such as key data, equipment, credentials, access logs or audit material.

Domain Name System (DNS)

The DNS is the combination of protocols, hardware and software elements, operational practices, and information that together provide a mapping service that associates and translates hierarchically organized identifiers ("domain names") into one or more values. The most common illustration of the DNS is obtaining the Internet Protocol (IP) address(es) associated with a particular domain name, e.g., the IP version 4 (IPv4) address associated with the DNS name www.icann.org is 192.0.32.7 as of the publication of this document.

DNS Root Zone Key Rollover (also known as Root Key Rollover)

Since this Advisory focuses on issues and impacts of a root zone KSK rollover, it seems to be appropriate to point out potentially critical steps in that process. These are expected to be the following:

1. Creation of the new root zone key [NewKey];
2. Addition of the public part of NewKey to the root zone;
3. First use of the NewKey to sign root zone data;
4. Last use of previous [OldKey] to sign root zone data;
5. Removal of the public part of OldKey from root zone; and
6. Destruction of OldKey.

Although the root zone KSK rollover process will take place over a period of time, that period of time is not currently defined. Each of the above steps in the process has some potential for causing some disruptions that will be discussed in other parts of this advisory.

DNS Security Extensions (DNSSEC)

A set of extensions to the core DNS protocols and operational practices, documented in Request for Comments (RFCs) 4033,[1] 4034,[2] 4035,[3] and others, that specifies the use of strong cryptographic hashes or signatures over DNS data to allow consumers of that data to verify that it has not been altered in transit from the authoritative source to the validating resolver.

[1] See http://tools.ietf.org/html/rfc4033.

[2] See http://tools.ietf.org/html/rfc4034

[3] See http://tools.ietf.org/html/rfc4035.

Emergency Key Rollover

An Emergency Key Rollover takes place in response to an unforeseen critical event such as Key Compromise or Key Loss that requires immediate action.

Key

The term “key” signifies a piece of digital information used in a cryptographic algorithm to encrypt or decrypt other information. In the context of DNSSEC, the term “key” is used to refer to a public-private key pair, typically either a key signing key (KSK) or zone signing key (ZSK). The private part of the key pair is used for signing by encrypting a representation (hash) of DNS data. The public part of the key pair is published in the DNSKEY Resource Record and is typically used by validating resolvers in signature verification.

Key Ceremony

The Key Ceremony is an operational process by which cryptographic key material is generated or used. Key ceremonies typically include initial cryptographic key material generation, renewal of cryptographic key material where new keys are created and previously generated keys are revoked, and/or signing or resigning of all information that relies on the trust associated with the key pair through secure handling of the key, a documented chain of custody, etc. In the case of the Key Ceremony used in the creation and handling of the root key, the process is precisely scripted, recorded and audited for maximum transparency.

Key Compromise

Key Compromise refers to the unauthorized exposure of any portion of the private half of the public-private key pair. In most cases, a key compromise must be assumed to result in the private portion of the key being made public and thus being unusable for cryptographic purposes.

Key Loss

Key Loss occurs when either or both portions of the public-private key pair are made unavailable for use. Key Loss differs from Key Compromise in that loss does not necessarily imply that the private key has been made public, e.g., if a private key is destroyed by fire or if the passphrase used to “unlock” the private key is lost/forgotten. Since loss typically does not imply unauthorized exposure of the private key, Key Loss may have different timing considerations than Key Compromise.

Key Rollover

Key rollover is the process of replacing a DNSSEC key with, ideally, minimal disruption to systems relying on that key. This process normally includes adding a new key to the DNS while continuing to use the earlier key, subsequently beginning use of the new key while both keys are present, and eventually removing the earlier key from the DNS. The key rollover may be a scheduled (also known as pre-planned or routine) key change or it may be an unscheduled (e.g., as a result of a Key Compromise or Key Loss) key change.

Key-Signing Key (KSK)

The KSK is a key that signs the set of all keys for a given zone, including itself. When a validator chooses to use this key as a Trust Anchor (TA), its use for signing all of the zone's keys enables the validator to authenticate other data in the zone. A key that signs zone data is referred to as a Zone-Signing Key (ZSK). While a single key may simultaneously fill the role of both KSK and ZSK, it is common to employ two distinct keys to fill these roles so they might be handled differently. Local policy may require, for example, that a zone's ZSK be changed relatively frequently, while the KSK is used longer in order to provide a more stable secure entry point into the zone or to reduce the frequency of updating information in a zone's parent. Because their roles are different, KSKs and ZSKs might vary in other ways, such as the setting of the secure entry point (SEP) bit in their DNSKEY Resource Record (DNSKEY RR) or the lifetime of the signatures they produce. Designating a key as a KSK, ZSK, or both is purely an operational issue; DNSSEC validation of zone data only requires that trust be appropriately derived from a secure entry point into the zone. KSKs are discussed in more detail in RFC 3757.[4] The public part of the root zone KSK is typically configured as the root (top-level) trust anchor in DNSSEC validators. See also Zone-Signing Key (ZSK) below.

Root Key Rollover (see DNS Root Zone Key Rollover)

Scheduled Key Rollover

A Scheduled Key Rollover is a Key Rollover that takes place at a pre-determined published date and time usually as part of normal operations, such as refreshing stale keys to guard against brute force key guessing attacks. Examples of reasons for scheduled key rollovers would include protecting against (future) computational attack, discovery of potential cryptographic algorithm vulnerabilities, and changing cryptographic equipment used in signing.

Trust Anchor (TA)

A TA is a preconfigured public key that is associated with a specific zone. A validating resolver must be configured with one or more TAs to perform validation. A TA allows

[4] See http://www.ietf.org/rfc/rfc3757.txt.

the DNS server to validate DNSKEY resource records for the corresponding zone and establish a chain of trust to child zones if they exist. Due to the hierarchical nature of DNS, the root key is the top most TA in the entire DNSSEC validity chain. Additional TA definition details are provided in RFC 4033 and RFC 4986.

Validator

A Validator in the context of DNSSEC is software, hardware, or a combination that uses public keys (DNSKEY RRs), Resource Record signatures (RRSIGs), and other necessary information (e.g., cryptographic hashes, denial of existence proofs) to verify that data returned in a response to a DNS query has not been modified in transit. A Validator verifies digital signatures at each level of the DNS hierarchy, from the resource record requested to a configured TA, typically the root zone TA, thereby establishing a Chain of Trust. A Validator may be part of a Recursive Resolver residing at an Internet Service Provider (ISP) or enterprise or may stand alone in an end node. A Validator may also be referred to as a Validating Resolver, and the terms are used interchangeably in this advisory.

Zone Signing

Zone Signing is the process in DNSSEC of using one or more private keys to create digital signatures of the resource record sets within a zone. Note that in some systems, individual resource record sets can be signed on demand instead of signing the entire zone, however the term “Zone Signing” is typically used for both systems.

Zone-Signing Key (ZSK)

The ZSK is a key that signs data within a given zone. A ZSK must be authenticated by a KSK, which might be itself, or (more commonly) a distinct key designated for secure entry into the zone. See also Key-Signing Key (KSK) above.

3. Brief Overview of DNSSEC

DNSSEC is a collection of Internet Engineering Task Force (IETF) standards first introduced in the early 1990s to improve the security of the DNS by providing a mechanism by which DNS responses can be validated. DNSSEC incorporates public key cryptography into the DNS architecture to form a chain of trust originating at the root zone. When a resolver issues a DNS query for a resource record in a DNSSEC-signed zone, the response includes not only the requested data but also the signature(s) for the data, so the validity of the data can be determined. Successful validation indicates that the data in the response has not been modified or tampered with from the point in time when the data was signed until the data was validated.[5]

[5] It should be noted that DNSSEC does not protect against changes to zone data that occur prior to the data being signed or after validation occurs.

The digital signatures generated with zone signing are published in the DNS with RRSIG resource records.[6] The public key used to validate an RRSIG is stored in a DNSKEY resource record and is retrieved by a validating resolver during the validation process, so the validator can subsequently validate the signature and thus authenticate the data.

In order to DNSSEC-sign a zone, a signing strategy must be chosen. Common practice is to utilize two keys: a key that is used to sign the zone data, known as the ZSK, and a key used to act as a secure entry point into the zone and authenticate the ZSK, known as a KSK.[7] A single key may be used to accomplish the functions of KSK and ZSK, though a KSK/ZSK split adds versatility for maintaining a signed zone (see the definitions of KSK and ZSK for more information). In the case of a KSK/ZSK split, only one ZSK and one KSK are necessary to sign a zone with DNSSEC but additional keys are often required to carry out the key rollover process.

4. Key Management in the Root Zone

Because trust in a DNSSEC context is derived from parent zones and the root zone is the ultimate parent of all delegations within the DNS tree, the root zone is the topmost trust level of the entire DNS infrastructure. The implication of this level of trust is that root key compromise or loss would impact all delegations from the root, i.e., the top-level domains, their delegations, second-level domains, and so on. As such, the keys that are associated with the root zone must be protected in a manner that minimizes the risk of any loss or compromise. How the root zone keys are managed is described in detail in ICANN's "DNSSEC Practice Statement for the Root Zone KSK Operator"[8] (RZKO DPS) and includes the two key operational rolls, managing the KSK and managing the ZSK.

4.1 KSK Operational Role

ICANN, as the Internet Assigned Numbers Authority (IANA) Functions Operator, is the Root Zone KSK Operator performing the function of generating the Root Zone's KSK and signing the DNSKEY Resource Records published in the root zone (often referred to as the Root Keyset) using that KSK. The Root Zone KSK Operator is also responsible for securely generating and storing the private keys and distributing the public portion of the KSK (the Root Trust Anchor) to relying parties, typically the operators of DNSSEC-validating resolvers or software developers who maintain DNSSEC-validating resolvers.

As described in section 1.3.5 of the RZKO DPS, the Root Zone KSK (RZ KSK) Operator is responsible for:

[6] Zone signing also generates NextSECure (NSEC) or NextSECure3 (NSEC3) resource records that allow the non-existence of a domain name within the signed zone to be provable. Further details regarding the generation or use of NSEC/NSEC3 is outside of the scope of this advisory.

[7] See http://tools.ietf.org/html/rfc6781.

[8] See https://www.iana.org/dnssec/icann-dps.txt

- Generating and protecting the private component of the RZ KSK;
- Securely importing public key components from the RZ ZSK Operator;
- Authenticating and validating the public RZ ZSK keyset;
- Securely signing the RZ ZSK keyset;
- Securely transmitting the signed RZ ZSK key set to the RZ ZSK Operator;
- Securely exporting the RZ KSK public key components; and
- Issuing an Emergency Key Rollover within a reasonable time if any private key component associated with the zone is lost or suspected to be compromised.

Currently the RZ KSK is an RSA key pair,[9] with a modulus size of 2048 bits[10] and is scheduled to be replaced with a different key through a rollover process that includes a Key Ceremony. The rollover process is expected to be completed within 5 years of initial DNSSEC operation in the Root Zone.[11]

RZ KSK rollover is scheduled for "general cryptographic hygiene," ensuring that any ongoing brute force attacks against existing keys would be wasted effort. The public portion of the Root KSK is posted on ICANN's repository as the Root Trust Anchor. The publication formats and the methods to validate its integrity are in the process of being published within the IETF as an RFC.[12,13]

ICANN has established and maintains Emergency KSK rollover procedures to ensure readiness for key compromise situations. Upon the suspected or known compromise of a Root Zone KSK, ICANN KSK Operations Security personnel will assess the situation, develop an appropriate action plan, and implement the action plan with approval from the ICANN DNSSEC Policy Management Authority (PMA) and ICANN executive management.

As part of the KSK emergency rollover procedures, ICANN maintains the capability of being able to generate and publish an interim TA within 48 hours. Since selection and use of TAs is a local policy determination per RFC 4033 and RFC 4986, DNSSEC Validator operators are expected to make use of the interim TA that will facilitate an automated rollover of an old KSK as described in the RZKO DPS private key compromise section. If an emergency rollover is required, some sort of manual intervention will be required for all DNSSEC Validators either by adding the interim TA or by adding the TA created in the subsequent Key Ceremony. Some number of Validators might be able to get the new TA via a software update while other Validators will require operators to manually insert new TA(s).

[9] RSA is an algorithm for public-key cryptography. See RSA (algorithm) at: http://en.wikipedia.org/wiki/RSA_%28algorithm%29.

[10] See section 6.1 of the RZKO DPS.

[11] See section 6.5 of the RZKO DPS.

[12] As of this writing still in Internet Draft form, see http://tools.ietf.org/html/draft-jabley-dnssec-trust-anchor-07 or its successors.

[13] See section 2.2 of the RZKO DPS.

If the private component of a TA is permanently lost, the latest point in time where this loss is detected will inevitably be at the Key Ceremony when it is supposed to be used. At this point in time, the Root Zone Maintainer/ZSK Operator has signatures for at least 33 days[14] of independent operations. If possible, a new KSK will be generated at this Key Ceremony or another ceremony scheduled within 48 hours. If ICANN is unable to accommodate the Key Ceremony, an interim KSK must be generated by ICANN and published as a TA within the stipulated 48 hours. In either case, the community is then given a minimum of 30 days notice to add the new TAs to the validating resolvers before the DNSKEY RRset has to be re-signed with the new TA. Failure to update a validating resolver will render that resolver unable to validate DNSSEC-signed RRsets.

The old TA will remain untouched in the root zone key set for one ten-day time slot.[15] In the next consecutive time slot, the old Trust Anchor will be marked as revoked, and after this time slot the lost key is permanently removed[16].

In either case, ICANN will inform the community of any emergency as soon as possible using the channels stipulated in the DPS.[17]

4.2 Zone-Signing Key (ZSK) Operational Role

Verisign, Inc. manages the root ZDK, acting in its role as Root Zone Maintainer. The ZSK is a 1024-bit RSA key that is rolled every three months. The ZSK derives trust solely from the root zone's KSK, which signs each new ZSK in a Key Ceremony run by ICANN in its role as IANA Functions Operator. Regular ZSK rollover has been part of the root zone management process from its inception and is a much different undertaking than rolling the root zone KSK. The root zone ZSK rollover process has few external dependencies outside of Verisign and ICANN staff. However, a number of Trusted Community Representatives (TCRs) are required at a Key Ceremony to oversee the use of the root zone KSK.

Verisign employs a dedicated group called Cryptographic Business Operations (CBO) to manage all-important key material within the company including the root zone ZSK. Verisign's CBO conducts its own Key Ceremony on a regular basis to generate new root zone ZSKs. In the event of a compromise or other event that requires an emergency ZSK rollover, Verisign's procedures allow for generating a new ZSK out of cycle. This out of cycle ZSK would have to be endorsed in a similar out-of-cycle Key Ceremony with the root zone's KSK.

Further details regarding the administration of the root zone ZSK are available in the DNSSEC Practice Statement for the Root Zone ZSK Operator.[18]

[14] See section 6.6 of the RZKO DPS.
[15] See section 6.6 of the RZKO DPS.
[16] See section 4.5.3.2 of the RZKO DPS.
[17] See section 2.1 of the RZKO DPS.
[18] See http://www.verisigninc.com/assets/dps-zsk-operator-1523.pdf.

5. Motivations for KSK Rollover

As mentioned previously, the motivations for a KSK rollover are tied, in part, to any scenario in which the trust in the secret part of the KSK has been reduced. These scenarios include:

- Key compromise;
- Loss of keying material;
- Reduced trust in the key management process;
- Reduced trust in signing algorithm or key size.
- Change in KSK or ZSK Key Management Entity
- Change of Hardware Security Module Vendors

In addition, a KSK rollover may be tied to ordinary operations.

5.1 KSK Compromise

A number of key compromise scenarios exist that have varying degrees of severity. Table 1 illustrates some of the possible compromise scenarios along with their expected severity levels and impact.

Scenario	Severity	Impact
Procedural Lapse	Low	Documentation or practice or personnel revision. Key rollover may be deferrable to next scheduled roll.
Key Management Facility Intrusion	Medium	Unless conclusive evidence demonstrates key was unmolested, emergency key rollover is required.
Compromise of Key Algorithm	High	Emergency key rollover with change of algorithm is required.

Table 1. Key Compromise Scenarios

A Procedural Lapse occurs when the policies and/or processes documented within the DPS are not followed. In some cases, a Procedural Lapse can be relatively benign, signifying a bug in the documented policies or processes that necessitate a revision of the DPS. In other cases, where the policies or processes of the DPS are deemed to be correct, failure to follow the DPS may result in reduced trust in the generated KSK.

A Key Management Facility Intrusion, which includes everything from unauthorized entry into the secured Key Management Facility to exposure of the actual KSK private key, is generally more serious than Procedural Lapse. In the best case, where Key Management Facility Intrusion is detected, monitored, and at no time is the KSK private key demonstrably at risk, it is likely the intrusion is a Procedural Lapse that can be remedied by updating policies or processes. In the worst case, in which the KSK private key is exposed, the KSK will need to be regenerated before trust can be restored.

Finally, Compromise of Key Algorithm denotes a fundamental failure in the ability to trust the algorithm used to generate the KSK or the algorithm that uses the KSK to sign zone data. In the best case, the KSK will need to be rolled with a new algorithm that is already supported in deployed resolvers. In the worst case, in addition to the KSK being rolled, resolvers will need to be updated to support a new signing algorithm.

While most of the key compromise scenarios in Table 1 are extremely unlikely, prudent risk management practice suggests that their probabilities be treated as non-zero and that a corresponding rollover of the root KSK is a possibility.

5.2 KSK Loss

KSK loss differs from a key compromise in that rather than having the private portion of the KSK be (potentially) accessible to a non-authorized party, the key is not accessible by authorized parties. The scenarios in which a KSK loss can occur would include (among others):

- Damage to physical facilities that renders the key permanently inaccessible;
- Loss of the passphrase or other mechanism used to "unlock" the private key for use;
- Intentional destruction of key to prevent exposure; and/or
- Not having the required number of people to access the key.

Given a KSK that is stored in a physical facility, there is always the possibility that a natural or man-made disaster can occur that would make the KSK unavailable. Examples could include earthquakes, fires, floods, and bombings. In these sorts of situations, a new KSK would likely need to be generated (along with the new key management facilities).

The second scenario, in which the mechanism used to protect the private key is lost is not applicable in the case of the root KSK as the private key is protected within a Hardware Security Module (HSM) and the code used to "unlock" the private key for use is published for transparency reasons.

The third scenario, in which the KSK is intentionally destroyed to prevent exposure of the private portion of the public-private key pair, is a function of anti-tampering mechanisms frequently used in HSMs. The theory behind these anti-tampering mechanisms is that it is better to destroy a key than to allow the surreptitious use of the key.

However, one of the multiple ways in which HSMs attempt to sense tampering is by detecting a strong jolt to the device, e.g., when an attacker tries to forcefully remove the HSM's casing. Unfortunately, it is impossible for an HSM to distinguish between this sort of attack and an accident such as dropping the HSM when it is being moved or an event such as an earthquake. In any of these cases, the destruction of the private key results in the KSK being made unusable and thus, a need for a key rollover.[19]

The final scenario is a procedural failure in which an insufficient number of people are available to implement a Key Ceremony. This situation could occur temporarily due to weather or other natural conditions or more permanently in the case of kidnapping or terrorist attack.

In all of these scenarios, if there is a ZSK/KSK split, the KSK regeneration must be completed before the ZSK expires.

5.3 Reduced Trust in Key Management

The processes developed by the Root Management Partners to manage the root keys precisely detailed to ensure the outcome of any aspect of root key management is verifiable and trustable. However, as a result, these processes tend to be relatively complex, involving a large number of steps, each of which must be performed correctly before the next step can take place. If any one of these steps is mis-performed, the entire Key Ceremony and the instance of key management process for which the Key Ceremony was being executed can be called into question. In the most likely case, a mis-performance of the Key Ceremony will result in a minor delay as the Key Ceremony is restarted, presumably correctly. However, in the event that the mis-performance is not noted either due to accident or malicious intent, the level of trust associated with the root key may be reduced.

A more serious concern however is related to the execution of the key management functions themselves. The risk associated with touching any part of the infrastructure associated with key management, no matter how small, is higher than not touching that infrastructure. For example, as previously noted the HSMs in use for root zone key management use anti-tampering technology that will destroy the contents of the HSM if the device receives a sufficiently strong physical shock. During the Key Ceremony necessary for any key management function including rolling the root key, the HSM must be physically removed from a safe, placed onto a cart, rolled to a table, lifted from the cart and placed onto the table for execution of the key management function. When the key management function is completed, the HSM must be returned to the safe. At any point in this process, mishandling of the HSM can result in the contents of the HSM, including the private key of the KSK, being wiped. This situation may be exacerbated by complacency in which frequent routine behaviors tend to be short circuited, albeit at increased risk of errors being introduced. As such, the risk of key loss when executing this key management function is higher than if the key management function is not performed.

[19] To protect against hardware failures such as this, current root KSK operations duplicate key material across four HSMs and backed up on smartcards that can only be restored onto replacement hardware with the participation of 5 out of 7 trusted community representatives holding smartcards containing portions of the key used to protect the KSK backup.

Every time the root key is rolled, the key management infrastructure must be exercised. Assuming there is a small but non-zero probability of key loss or compromise, each key roll provides an independent opportunity for failure of the key rollover process, with the cumulative effect of multiple key rolls being that there is a higher probability of key loss/compromise than if no key rolls are done.

For example, if we assume key rollovers are independent events and the probability of a successful key rollover is 99.9 percent, the probability of two consecutive successful key rollovers would be 99.801 percent, of ten consecutive successful key rollovers: 99.004 percent, of 100: 90.5 percent, of 1000: 36.77 percent.[20]

This key loss/compromise implies that it would not be possible to recover via an automated RFC 5011-style[21] key rollover, thereby forcing the worst case equivalent of a key bootstrap process in which every validator in the world would need to have their root TA manually reconfigured.

Of course, the probability of key loss/compromise per rollover is likely far smaller than 0.1 percent (the actual probability is difficult to determine due to the limited history of key rollovers, the various potential failure modes and their mitigations, and other factors), however the significant impact of that loss must be factored into the risk assessment associated with root key rollover.

Finally, failure to roll the key also carries with it some risk since best practice for "general cryptographic hygiene" is to replace keying material after some length of time to minimize the risk of successful brute force attacks.

5.4 Reduced Trust in Signing Algorithm or Key Size

From the moment keying material is generated, it is vulnerable to compromise. As time goes by, the window during which it *could be* compromised increases until the key is securely destroyed. This leads to the observation that as keying material ages, the faith one places in it should decrease until it eventual reaches a point where it must be assumed that the key has been compromised and thus should no longer be used.

Unfortunately the rate at which this aging occurs cannot be accurately predicted. While estimates about key longevity based on the cryptographic techniques and strength of the underlying algorithms do exist, they necessarily make linear assumptions about technology, that is, that technological breakthroughs can't be predicted, and as a result, are of limited value. In this advisory the SSAC is not able to recommend any specific timeline for suspected key compromise probabilities.

[20] Note that this is true despite the fact that each independent event would be successful 99.9 percent of the time – it is the difference between asking, "what is the probability of flipping a coin 10 times and always getting heads? (0.001)" and asking, "given I've gotten 9 heads, what is the probability of getting a tenth head? (0.5)".

[21] See http://tools.ietf.org/html/rfc5011.

5.4.1 Improvements in Factoring/Computational Capacity

As discussed previously, the root key is comprised of two parts, a public part and a private part. Under the current algorithms being used for the root key, these two components are derived from the product of two (extremely large) prime numbers. The security provided by the root key thus relies entirely upon the difficulty in factoring this product and determining the value of the two primes.[22]

In theory, a suitably determined attacker can simply multiply candidate numbers together until the two constituent primes are found, however due to the size of the numbers involved this will require an infeasibly large number of tries (and so will take an extremely long time).

In practice, a number of factors are working to the attacker's advantage. These factors include:

- Increase in computing power ("Moore's law");
- The ability to distribute the work across multiple computers, e.g., through the use of botnets;
- Dedicated and/or optimized hardware, e.g., the use of ASICs, FPGAs, or GPUs;
- Technological advances (e.g., quantum computing); and
- Improvements in mathematics (such as the number field sieve).

Collectively, these factors means that as time goes by longer keys will need to be used to provide the same level of protection as provided by shorter keys used in previous years and eventually the current key length will be become inadequate.

The amount of effort an attacker is willing to expend to "break" (factor) a key is related to the value of the information that the key protects. For example, it would not make sense for an attacker to spend $1,000,000 to steal something worth $10. However it would be more worthwhile for an attacker to spend $1,000 to gain access to information worth $1,000,000. As the value of the information protected by the DNSSEC root key is potentially extremely large, it is prudent to assume that an attacker would be willing to expend significant money or effort to factor it. In addition, given the criticality of the DNS root, it is appropriate to assume that the attacker may be very well resourced (such as a nation state) and has significant technical expertise.

How long various key lengths should be considered "secure" is beyond the scope of this advisory,[23] but eventually the current key lengths, regardless of what they might be, will not be sufficient to provide adequate protections. As a result, before the risk of root key

[22] As a simplified example, if handed 27680466418840896028326181, it would be difficult for most to determine that that number is the result of multiplying 376765654387 by 73468656435463.

[23] See Recommendation for Key Management, Special Publication 800-57 Part 1 Rev. 3, NIST, 05/2011 for more information.

compromise becomes too large, key rollover to a longer key length should be performed.

In addition, advances in cryptanalysis and factoring may suggest the current keys and/or algorithms may not provide the required level of assurance in the future and so necessitate earlier rollover.

5.6 Change in KSK or ZSK Key Management Entity

A change in the Root Zone Maintainer, which currently administers the ZSK, would have little impact on the KSK and the larger Internet community. At worst, an out-of-cycle key ceremony by the IANA Functions Operator would be required to sign the first ZSK generated by the new operator.

A change in the IANA Functions Operator, which currently administers the KSK, may be more complicated because of the external dependencies inherent in the root zone KSK, which is configured in many DNSSEC Validators as a trust anchor. Such a change does not necessarily require a KSK rollover. If the new operator is able to use the existing HSM infrastructure, no special actions would be necessary. Alternatively, if the new operator uses the same type of HSM, it would theoretically be possible for the current operator to export the current KSK in encrypted form only readable by the same manufacturer's HSM. This encrypted key material could be moved securely, with a clear chain of custody, to the new operator and imported into the new operator's HSMs in a key ceremony. (Indeed, a similar process was used by ICANN to transport the current KSK's private portion from the U.S. East Coast to West Coast to a second Key Management Facility.)

If such a smooth transfer is not possible, either because the new operator did not use the same HSM for whatever reason or other unforeseeable technical, procedural or political circumstances, then a KSK rollover would certainly be required: the new KSK operator would need to generate a new KSK.

5.7 Change of Hardware Security Module Vendors

It is likely that at some stage of KSK root operations the HSM vendor may need to change, e.g., if that vendor were to cease business operations.[24] If this were to occur, there would be a need to introduce a new HSM vendor. Although an upgrade or transition path is usually provided by manufacturers of HSM equipment, the vagaries of the market cannot guarantee an orderly transition. Furthermore, it may be impractical to transfer key material from one HSM to another.[25] As a result, the generation of a new KSK on the

[24] The vendor of the HSM currently holding the KSK has supported the product for well over a decade through a series of ownership changes (now ULTRA Electronics a UK defense department contractor). This bodes well for the nature of KSK operations and other HSM customer applications.

[25] The use of the PKCS11 standard in KSK operations software does provide for the support of multiple HSM vendors. However the mechanisms used to transfer the overarching keys used to securely export and import keys between units varies greatly from vendor to vendor making this step impractical or insecure.

incoming HSM and scheduling a KSK rollover[26] would be necessary.

5.8 Operational and Procedural Exercise

While a scheduled key rollover is necessary for root KSK "hygiene," the tradeoff as to how often a scheduled key rollover should occur to minimize any potential attack window must be carefully balanced against:

- Key rollover operational costs and risks;
- Likelihood of users disabling validation to avoid having to deal with rollover; and
- Inherent security risks associated with KSK loss/compromise during generation of new keys.

While the timeframes for scheduled key rollovers can be debated, due to the non-zero probability of a potential root key compromise, it is clear that at some point the probability of a need for a key rollover will reach certainty. As in any emergency situation, it is always useful to clearly understand the operational implications and have definitive processes and procedures in place. Perhaps equally important is testing the processes and procedures to ensure they operate as intended and do not have unacceptable side effects.

Both a scheduled and an emergency key rollover will, in all likelihood, have very similar predefined sets of operational procedures and processes. However, the emergency key rollover will not have the luxury of advance notice to prepare DNSSEC Validators of imminent changes in Trust Anchors. To make matters worse, in the case of an emergency key rollover caused by key compromise, the mechanisms defined in RFC 5011 for automated key rollover will likely not be available for some Validators, e.g., operators may not know about the interim TA or choose not to use it. The emergency key rollover procedures for the root zone impact the entire DNSSEC hierarchy and require an immediate coordinated and efficient process.

For an emergency root KSK rollover, expedient communication across the hierarchy of Validators is critical and, unfortunately, extremely difficult given Validators are operated independently of the root zone. All Validator operators must be able to initiate their own emergency processes to modify their TAs as soon as the root KSK has been renewed and associated timing parameters have been met. All Key Management operational procedures for an emergency root KSK rollover must be clearly understood by Validator operators using the root TA, to discern in advance what operational and procedural processes their Validators will need to follow and to understand any potential impacts to those Validators.

[26] Support for KSK rollover as per figure 2 of "DNSSEC Root Zone High Level Technical Architecture" (http://www.root-dnssec.org/wp-content/uploads/2010/06/draft-icann-dnssec-arch-v1dot4.pdf) document is built into the current software used to manage the KSK.

6. Risks Associated with Key Rollover

It is generally accepted that the DNS is a critical Internet system. The root zone is a critical component of the DNS, given its unique importance as the starting point for all resolution and as a starting point for trust in DNSSEC. A root zone KSK rollover would thus mean changing a critical component of a critical system. This section enumerates some of the risks associated with such a high impact change.

The first and largest risk is that some portion of DNSSEC Validators using the root zone KSK as a TA will not, for whatever reason, properly install the new root zone KSK TA during a rollover. The result for affected Validators is failed DNSSEC validations for all DNS records except those for which a Validator has a more specific TA configured. However, it is reasonable to assume that most, if not all, validating resolvers exclusively use the root zone's KSK as a TA. For such a Validator, an out-of-date and invalid root zone KSK remaining configured as a trust anchor means failed validation for any DNS response it attempts to authenticate.

An analysis of the potential impact of this risk is discussed in Appendix A of this report. Referencing that discussion, it is estimated that as of this writing, 8.3 percent of all Internet clients use resolvers that perform DNSSEC validation using the root KSK as a TA. This represents the population of users that might be affected by errors at the root level. However, only about 87 percent of those clients are using validators that are expected to properly update their TA with some confidence, leaving the fate of 1.1 percent (i.e., 13 percent of the 8.3 percent of users using validation) of users in question with a root KSK rollover.

There is also some risk of increased traffic to the root or other authoritative servers, particularly from validating resolvers that failed to update to the root TA after a rollover. The basis for this concern is the 2009 incident documented as "Roll Over and Die?"[27] in which an outdated TA for a number of resolver implementations resulted in a large increase in traffic to authoritative DNS servers. Appendix A explains some small-scale testing of this scenario, showing that newer versions of some validator implementations still cause increased traffic to authoritative servers when an invalid TA is used. However, the observed increase is of a much smaller magnitude than that observed in 2009 and the root and TLD zones are not impacted at all.

As discussed previously, a number of procedural and technical risks associated with the rollover process itself exist. Although the rollover process is expected to be similar to the process used for initially signing the root zone, any differences between the initial signing process and the rollover process have not as yet been operationally exercised and, thus, may introduce more risks.

[27] See: http://www.potaroo.net/ispcol/2010-02/rollover.pdf.

7. Available Mechanisms for Key Rollover

By design, DNSSEC validators must have at least one TA that provides the basis for doing DNSSEC validation of DNS responses. In most if not all cases, Validators on the Internet will have a TA that is based upon the root zone KSK. As a result, changing the root zone KSK, i.e., performing a KSK rollover, requires the operators of validating resolvers to change their root zone TA in a trustworthy manner. The IETF has published the TA rollover requirements in RFC 4986[28] and a protocol to facilitate rollover, specified in RFC 5011. Resolvers that implement RFC 5011 have an automated method to update their TA given at least one non-compromised TA. For validators that do not implement RFC 5011 or in the case when no trustable TAs exist (e.g., a worst case scenario where the root KSK has been compromised or lost) some other form of TA update is required.

7.1 RFC 5011 Rollover

RFC 5011 describes a key rollover protocol whereby an existing root zone KSK can be used to authenticate new root zone KSKs. This would enable multiple valid KSKs where one acts as the active key while the other acts as a standby key. The standby key would not actively participate in signing but it would be accepted as a TA if a validator sees a signature from it.

Creating a standby root zone KSK is straightforward. Assuming a valid KSK_1 exists, a new key pair is created to produce KSK_2, which is added to the DNSKEY RRSet. The existing KSK_1 is then used to sign the new DNSKEY RRSet.

Scheduled key rollovers assume both KSK_1 and KSK_2 are valid TAs. A new key pair would be generated to produce KSK_3, which is added to the DNSKEY RRSet. Both KSK_1 and KSK_2 are used to sign the new DNSKEY RRSet. KSK_1 would get revoked, the KSK_2 would become the new active key and KSK_3 would become the new standby TA.

An emergency key rollover would be required if either KSK_1 or KSK_2 were compromised. In the case of a KSK_1 compromise, the procedure would be the same as for a scheduled key rollover. In the case of a KSK_2 compromise, a new key pair would again be generated to produce KSK_3, which is added to the DNSKEY RRSet and signed, by both KSK_1 and KSK_2. However, now KSK_2 would get revoked, KSK_1 would still remain the active key and KSK_3 would now become the stand-by key.

The RFC 5011 protocol has provisions for ensuring that revocation can only be realized through trusted mechanisms. Additionally, there are timing requirements imposed to ensure that there are sufficient sanity checks to prevent scenarios where both the attacker and the valid root zone KSK are able to sign data and be accepted as valid.

[28] See http://tools.ietf.org/html/rfc4986.

7.2. Non-RFC 5011 Rollover

There are a number of situations where use of the RFC 5011 rollover is not possible. Examples include installation of a validator for the first time, failure in the automated mechanism, or when all root KSKs have been compromised. In such cases, a non-RFC 5011 rollover will be required. The actual mechanics of doing such a rollover will be different for various validators and the details of those mechanics are beyond the scope of this advisory, however all non-RFC 5011 rollover approaches will have some common elements, including "bootstrap-ability" and mechanisms by which the Validator operator can ensure proper root zone TAs can be installed.

When a resolver is moving from not using DNSSEC to using DNSSEC and doing validation, the then-current root zone TAs will need to be installed. This is termed "bootstrapping" the validator. Given RFC 5011's need for a valid, non-compromised TA to install a new TA, RFC 5011 is obviously not applicable to bootstrapping.

In addition to bootstrapping, a Validator operator needs to be able to ensure the root zone TAs in use in the Validator have not been tampered with. A number of methods for publicizing and distributing an updated root zone TA have been suggested. Although there is no Internet standard specification describing these methods, there are discussions underway and some draft specifications[29] that may result in one or more specifications in the future.

Some possible approaches for root zone TA distribution would include:

- Via current software distribution mechanisms: A number of software packages that provide a DNSSEC Validator currently also include the root zone TA. When newer root zone TAs are available, these software packages can also include the newer TAs.
- Automated (or semi-automated) software distribution updates: Many current software providers (especially operating system providers) have automated or semi-automated software distribution methods that could also provide root zone TAs that include updates resulting from root zone KSK rollover.
- ISP provided TA updates: Many Internet users, particularly home users, depend upon their ISP to provide them with various security related services, e.g., anti-virus software. In addition to existing services, an ISP could provide their customers with current and updated TAs.
- Well publicized publication of root zone TAs: ICANN is currently publishing root zone TAs in various formats on a web site https://www.iana.org/dnssec/. Other web sites (especially DNSSEC related sites) could provide references to this site.
- Publish information via traditional media: The root zone TAs and information

[29] See, for example, "DNSSEC Trust Anchor Publication for the Root Zone" http://tools.ietf.org/html/draft-jabley-dnssec-trust-anchor-07 (as of this writing).

related to the state of root zone TAs (such as a planned rollover) could be published in traditional print publications. The particulars of which publications (including geographic and language factors), the frequency of publication and the specific information to be provided would need to be determined but this would provide a fully "out of band" method to provide root zone TA information.

Many of these approaches are either already automated (e.g., via Microsoft's Windows Update system) or can be automated with scripts by competent administrators (e.g., software that would periodically fetch, verify, and install the TAs made available at https://www.iana.org/dnssec/).

7.3 Resolver Rollover Requirements

In many if not all cases, DNS resolvers, whether they are performing validation or not, have minimal requirements for persistent local storage, typically requiring a minimum of (at least) read-only configuration data. When validation is being used, it is necessary to have a configured TA before the first query requiring information that must be validated. Typically, this implies the TA must be stored in persistent storage. In the case of RFC 5011 being used for root KSK rollover, the resolver must be able to write to that local storage to update the TA when the root key is rolled. This requirement may be problematic for systems with limited persistent storage (e.g., embedded systems) or for environments where servers with access to the Internet have limited or no permission to write to local storage for security reasons.

7.4 Response size issues

DNSSEC is often already using messages larger then the classic default 512 byte User Datagram Protocol (UDP) limit. During a (KSK) key rollover these messages will be larger. This might have consequences for how UDP fragmentation is handled and causes some concern, especially when IPv6 is used a transport protocol. A detailed discussion of this issue can be found in Appendix B: Response Size Considerations.

8. Recommendations

The SSAC proposes the following five recommendations for consideration and discussion:

Recommendation 1: ICANN staff, in coordination with the other Root Zone Management Partners (United States Dept. of Commerce, National Telecommunications and Information Administration (NTIA), and Verisign), should immediately undertake a significant, worldwide communications effort to publicize the root zone KSK rollover motivation and process as widely as possible.

For the initial rollover, specific contact should be made with known major providers of DNSSEC validation, both software and operational providers. As part of this effort, some way in which users who believe they have been negatively impacted by rollovers can notify the root management partners of the circumstances and impacts should be made available and clearly described.

Recommendation 2: ICANN staff should lead, coordinate, or otherwise encourage the creation of a collaborative, representative testbed for the purpose of analyzing behaviors of various validating resolver implementations, their versions, and their network environments (e.g., middle boxes) that may affect or be affected by a root KSK rollover, such that potential problem areas can be identified, communicated, and addressed.

This testbed should be made available to any person or entity, especially software and operational providers, who have a demonstrated need to participate in the analysis of potential issues during a root KSK rollover.

The testbed should be designed to collect as much information as possible about the impact of a KSK rollover and any mitigation strategies that are developed. The information should be made available to all participants, DNS experts, and others who have a demonstrated commitment to analyze the information for the purpose of improving future rollover events.

In addition to testing the root KSK rollover process, the testing of an algorithm rollover should be included and encouraged within the testbed.

Recommendation 3: ICANN staff should lead, coordinate, or otherwise encourage the creation of clear and objective metrics for acceptable levels of "breakage" resulting from a key rollover.

It is expected that there will be some issues during at least the first KSK rollover, and probably the next few. It will not be possible to anticipate all the problems that may occur but an agreed understanding of when the rollover has affected operational stability beyond a reasonable boundary is essential so the decision to rollback the rollover can be made quickly and efficiently.

Recommendation 4: ICANN staff should lead, coordinate, or otherwise encourage the development of rollback procedures to be executed when a rollover has affected operational stability beyond a reasonable boundary.

As part of the rollback procedures, a clear chain of command and/or set of people should be identified as responsible parties to identify when to abort a rollover and to rollback to a previous KSK. These procedures should be in place for both when a new trust anchor is published and when an old trust anchor is removed.

Recommendation 5: ICANN staff should lead, coordinate, or otherwise encourage the collection of as much information as possible about the impact of a KSK rollover to provide input to planning for future rollovers.

Particular emphasis should be on whether manual or automated methods were used to get the new trust anchor in place. As part of this effort, a baseline of DNS traffic, particularly at the root should be collected in order to facilitate before and after comparisons.

DNS experts should be invited to participate in the analysis of the information.

The analysis and the supporting information should be published and made available to the ICANN community.

9. Acknowledgements, Statements of Interests, and Objections, and Withdrawals

In the interest of transparency, these sections provide the reader with information about four aspects of our process. The Acknowledgments section lists the members who contributed to this particular document. The Statements of Interest section points to the biographies of all Committee members and any conflicts of interest—-real, apparent, or potential—-that may bear on the material in this document. The Objections section provides a place for individual members to disagree with the content of this document or the process for preparing it. The Withdrawals section is a listing of individual members who have recused themselves from discussion of the topic. Except for members listed in the Objections and Withdrawals sections, this document has the consensus approval of all members of the Committee.

9.1 Acknowledgments

The committee wishes to thank the following SSAC members for their time, contributions, and review in producing this Report.

SSAC Members:

Greg Aaron
Alain Aina
Jaap Akkerhuis
Roy Arends
KC Claffy
David Conrad
Patrik Fältström
Jim Galvin
Merike Kaeo
Warren Kumari
Matt Larson (Work Party Co-Leader)
Russ Mundy (Work Party Co-Leader)

Staff:

Joe Abley
Casey Deccio (Research Fellow)
Richard Lamb
Barbara Roseman
Julie Hedlund (Editor)

9.2 Statements of Interest

SSAC member biographical information and Statements of Interest are available at: http://www.icann.org/en/groups/ssac/biographies-08oct13-en.htm.

9.3 Objections and Withdrawals

There were no objections or withdrawals.

Appendix A – Quantifying the Risk of Failed Trust Anchor Update

Measuring Domain Name System (DNS) Security Extensions (DNSSEC) validating resolvers and their clients is a complex task, though it has been the subject of recent research. It has been observed in various measurement studies that less than 1 percent of resolvers are performing validation [1], validating resolvers are responsible for 12 percent queries [1], and nearly 8.3 percent of all clients use resolvers that perform DNSSEC validation [2].[1] All these numbers have different implications in terms of impact in the case of a failed root Trust Anchor (TA) update.

While any validator might have some possibility of failure, it is expected that validators associated with high-profile DNS services would be less likely to miss a TA update than smaller, perhaps less connected DNS services. For example, it is likely that Google's Public DNS Services and Comcast DNS Services, both prominent DNSSEC validation deployments, would properly install a new root KSK in their validating resolvers when the new one was introduced. It is estimated that nearly 75 percent of DNSSEC validating queries were attributed to Comcast DNS servers, prior to the enabling of DNSSEC validation on Google's Public DNS servers [1], and more recently, 47 percent of clients using validating resolvers use Google's Public DNS [2]. Assuming that the number of DNS queries made by resolvers is proportional to the number of clients using those resolvers and that current validation trends remain consistent with those prior to the introduction of Google's Public DNS, then roughly 13 percent[2] of clients are using validating resolvers other than Google or Comcast. Of course, this is only an estimate, but it represents the client population for which a reliable TA update is less certain and perhaps provides a probable upper bound of the potentially impacted user base.

Quantifying the number of validators that might miss a root KSK rollover among those used by the remaining 13 percent of clients is a difficult problem and an accurate measurement is likely unrealistic. However, there are several considerations that might provide insight into the potential for missed TA updates at the root zone by validating resolvers. Among those are DNSSEC algorithm awareness and automated TA update support in validating resolver implementations. The analysis in this report considers the capabilities and histories of two prominent, open-source, validating resolver implementations, recognizing that these are only two of many that are currently deployed.

While it is not expected that this analysis will be comprehensive enough to apply to all implementations, it can be used to gain some insight into the possibility of failure, and the principles can likely be applied elsewhere. The two implementations analyzed are the

[1] It is difficult to compare the numbers from the DNSSEC validation measurements [1-2], in part because the methodologies differed, but also because [1] occurred prior to Google's deployment of DNSSEC validation on their public DNS service, and [2] is from after.

[2] If 47 percent of clients behind validating resolvers use Google DNS and 75 percent of the remaining 53 percent (roughly 40 percent) use Comcast, then the identity of resolvers for 13 percent of clients using DNSSEC validation is unknown.

Berkeley Internet Name Domain (BIND) by Internet Systems Consortium (ISC) and *unbound* by NLnet labs. *BIND* version 9.6.2[3] (released March 2010) and later have included support for the RSASHA256 DNSSEC algorithm. Likewise, *unbound* has included support for RSASHA256 since version 1.4.0[4] (released November 2009). Thus, resolvers using these or more recent releases of this software are capable of using the root zone's KSK as a TA, it also utilizing signing algorithm RSASHA256.

Although an automated key rollover mechanism has been defined (see the section on RFC 5011 Rollover), client implementations of this mechanism are not universal — that is, not all validating resolvers have integrated RFC 5011 support. *Unbound* added RFC 5011 support in release 1.4.0 — the same release in which it became capable of using the root zone KSK as a TA by supporting algorithm RSASHA256. As such, there is no apparent *unbound* release that supports validation using the root zone's KSK as a TA but does not support automated TA updates using RFC 5011.

BIND included RFC 5011 support in releases of the 9.7 and later branches but did not backport support to the 9.6 branch, including 9.6-ESV (Extended Support Version), which continues to be active at the time of this writing and has expected end-of-life in January 2014.[5] Thus, *BIND* 9.6 releases beginning with 9.6.2 and including 9.6-ESV (still active and supported) are capable of using the root zone KSK as a TA but do not have any built-in mechanism for automatically updating it in the case of a rollover. Such resolvers are susceptible to breakage in the case of a root KSK rollover if they are using the root zone KSK as a TA and no manual intervention is performed. Success in this case depends on the knowledge and competence of the administrators in initially configuring (or not) DNSSEC and their involvement in the DNS community, including pertinent mailing list subscriptions for awareness of and action taken for the impending rollover.

The default resolver behavior as obtained directly from the vendor or distributed with an operating system can also increase the risk of failure, depending on the configuration associated with the software version or distributed with the operating system. *BIND* version 9.6-ESV, which has the greatest identified potential risk, includes no stock configuration file (named.conf) for the validating resolver software. As such, the user would need to create her own configuration file and explicitly include a configuration directive to enable DNSSEC validation with the root zone KSK as a TA ("*trusted-keys*" in named.conf syntax). Presumably, the installed TA would be up-to-date at the point of deployment, yet it still must be manually maintained, as mentioned previously. User awareness of this responsibility may vary, just as it does with deployment of DNSSEC on an authoritative server.

Releases of *unbound* since 1.4.0 and *BIND* releases throughout branches 9.7, 9.8, and 9.9 require explicit configuration in the configuration to enable DNSSEC validation with the

[3] See ftp://ftp.isc.org/isc/bind9/9.6-ESV-R9-P1/CHANGES

[4] See http://unbound.net/download.html

[5] While ISC declares an "end-of-life" date for *BIND* 9.6-ESV, it is unlikely that use will cease at that point, but probably taper off, perhaps over years beyond January 2014.

root zone KSK as a TA. Each is also capable of maintaining the root zone TA with RFC 5011 (assuming the underlying infrastructure upon which the name server allows it), so the configuration apparently presents less risk. However, both *BIND* and *unbound* provide two ways to enable DNSSEC validation, one of which includes automatic trust anchor support (i.e., "*managed-keys*" in *BIND* and "*auto-trust-anchor*" in *unbound*) and one that does not (i.e., "*trusted-keys*" in *BIND* and "*trust-anchor-file*", "*trusted-keys-file*", or "*trust-anchor*" in *unbound*). *BIND* and *unbound* resolvers that are capable of both using the root zone KSK as a TA and obtaining an updated TA using RFC 5011 must be properly configured to do so. The alternative introduces risk similar to that identified with *BIND* 9.6-ESV with regard to missing a TA update in conjunction with a root KSK rollover.

The following table summarizes the features of the *BIND* and *unbound* resolver implementations, as they relate to potential risk associated with a root KSK rollover:

Validating Resolver Implementation	RSASHA256 (8) Support	Validation with Root TA Enabled by Default	RFC 5011 Support (Enabled by Default?)	Risk of Missed TA Update
unbound < 1.4.0	No	No	No	None
unbound >= 1.4.0	Yes	No	Yes (Y/N*)	Low
BIND < 9.6.2	No	No	No	None
BIND 9.6.2 – 9.6-ESV (latest)	Yes	No	No	Medium
BIND 9.7.x	Yes	No	Yes (Y/N*)	Low
BIND 9.8.x	Yes	No	Yes (Y/N*)	Low
BIND 9.9.x	Yes	No	Yes (Y/N*)	Low

* - As explained earlier, automatic trust anchor support depends on the configuration options used to enable DNSSEC validation on the resolver.

While the default behaviors of validating resolver implementations *BIND* and *unbound* have been discussed, there is some variance in behavior based on the default configuration distributed with an operating system. For example, the *BIND* (version 9.8.4) and *unbound* (version 1.4.17) packages distributed with Debian 7.0 both include stock configuration files that enable DNSSEC validation with auto trust anchor maintenance. The same is true for the *BIND* (version 9.8.2) package distributed with Red Hat Enterprise Linux (RHEL) 6 and CentOS 6.

The risk of increased traffic to the root or other authoritative servers should also be considered. Increased traffic is attributed to excessive queries from validating resolvers that failed to update to the root TA after a rollover and might negatively impact these authoritative servers. Such an increase was observed following the distribution of an outdated trust anchors in 2009 as documented in “Roll Over and Die?”. It is estimated that only 13% of clients behind validating resolvers and 1.1 percent (i.e., 13 percent of the 8.3 percent using validation) of all Internet clients are of potential concern, based on the aforementioned measurement studies. The behaviors of *BIND* (version 9.8.4) and *unbound* (version 1.4.17) were examined in a small test environment in the wake of a staged failure of a TA update. It was observed that *unbound* implements a “bad” cache, as permitted in RFC 4035 and recommended in RFC 6840. When an *unbound* resolver encounters a broken chain of trust due to an outdated TA, it caches that response for one minute. In the absence of a “bad” cache entry, a single query is issued by the resolver for the desired RRset to the authoritative server closest to the queried name — not the servers authoritative for its ancestors. *BIND* appears to have no “bad” cache with the same setup, but otherwise follows the *unbound* behavior. This is notably different than the behavior observed in versions of *BIND* prior to the March 2010 release of *BIND* 9.6-ESV.

While neither validating resolver in this experiment reacted as aggressively as was observed in the “Roll Over” analysis, the short-lived (*unbound*) and non-existent (*BIND*) negative caches could result in a significant increase in queries to second-level domain servers and below. However, if observation of these two resolvers is representative, then the impact on the root and top-level domain servers is minimal.

References

[1] Yingdi Yu, Duane Wessels, Matt Larson, and Lixia Zhang. “Check-Repeat: A New Method of Measuring DNSSEC Validating Resolvers.” In *The 5th IEEE International Traffic Monitoring and Analysis Workshop (TMA 2013). Apr 2013. http://irl.cs.ucla.edu/~yingdi/web/pub/yingdi_tma13.pdf.*

[2] Geoff Huston. “DNS, DNSSEC and Google's Public DNS Service.” Online at: http://www.circleid.com/posts/20130717_dns_dnssec_and_googles_public_dns_service/.

[3] George Michaelson, Patrik Wallström, Roy Arends, Geoff Huston. “Roll Over and Die?” Online at: http://www.potaroo.net/ispcol/2010-02/rollover.pdf.

Appendix B – DNS Response Size Considerations

In periods of KSK rollover, Domain Name System (DNS) key (DNSKEY) responses from root servers will contain three DNSKEY Resource Records (RRs) and two Resource Record Signatures (RRSIG) RRs — one made by the current Key Signing Key (KSK) and one made by the new KSK. The size of the keys determines the size of the signatures, and together they determine the size of the response.

A large DNS response might lead to fragmentation of User Datagram Protocol (UDP) messages. Filtering of UDP fragments might cause responses to be dropped. In the first part of this section, the DNS response during a KSK rollover, given a 1024 bit RSASHA256 Zone Signing Key (ZSK) and two 2048 bit RSASHA256 KSKs, will be calculated. The second part of the section will detail the impact of this response size during a transmission, separately for IPv4 and IPv6.

Maximum DNS Response size during a KSK rollover

Section 3.2[1] shows that there will be a period when there is a root zone with a ZSK and two KSKs and signatures by both KSKs. Section 6.1[2] shows that the current KSK DNSKEY contains a RSA public key with a modulus of 2048 bits. Section 6.1[3] shows that the current ZSK DNSKEY contains a RSA public key with a modulus of 1024 bits.

During this period, a response for a DNS request for the DNSKEY RRset for the root zone that indicates that the resolver is able to receive DNSSEC Resource Records (via the DNSSEC-OK bit), will contain the ZSK, the two KSKs and the two signatures over the DNSKEY (a signature by each KSK) RRset.

To calculate the exact response size, consider that the DNS message contains a set of fixed size elements and various variable parts. The variable parts are solely determined by the size of the RSA modulus. We'll first calculate the fixed part of the response:

1. A DNS response consists of a 12-byte DNS header (DNSH = 12) and 11 byte OPT record carrying EDNS0 (EDNS = 11).[4]
2. The question section contains the QNAME = the root label (1 byte), QTYPE = "DNSKEY" (2 bytes), QCLASS = "IN" (2 bytes). The combined size of the Question Section is 5 bytes (Qsection = 5).
3. All Resource records have the following format: Name (1 byte root label), Type (2 bytes), Class (2 bytes), TTL (4 bytes), RDLEN (2 bytes), RDATA (Variable length). Hence, a Resource Record with a root label has a fixed size of 11 bytes plus a variable size.

[1] See http://www.root-dnssec.org/wp-content/uploads/2010/05/draft-icann-dnssec-keymgmt-01.txt.
[2] See https://www.iana.org/dnssec/icann-dps.txt.
[3]See http://www.root-dnssec.org/wp-content/uploads/2010/06/vrsn-dps-00.txt.
[4] Given a response that carries DNSSEC RRtypes must support the DNSSEC OK bit in the EDNS0 header, the OPT record isn't actually optional.

There are 5 resource records in the response, three DNSKEYS and two RRSIGS.

4. The DNSKEY Resource Records have the following fields: Flags (2 bytes), Protocol (1 byte), Algorithm (1 byte) and Public Key (Variable). The Public Key field consists of the Exponent and the Modulus. The exponent used for root keys is 65537, which costs 3 bytes, plus a byte to indicate the length of the exponent in bytes (4 bytes total). This exponent is generally accepted as safe and unlikely to change, so for the purpose of this exercise, we consider this fixed. Hence, a DNSKEY Resource Record with a root label has 19 bytes fixed size, plus the size of the Modulus. (DNSKEYfixed = 19 bytes)

   The RRSIG Resource Records have the following fields: Type Covered (2 bytes), Algorithm (1 byte), Labels (1 byte), Original TTL (4 bytes), Signature Expiration (4 bytes), Signature Inception (4 bytes), Key Tag (2 bytes), Signer's Name (root label, 1 byte) and Signature (Variable). Hence, a RRSIG Resource Record with a root label has a fixed size of 30 bytes plus a variable size. (RRSIGfixed = 30 bytes)

We can now determine the fixed size of a DNS response message:

FixedSize = DNSH + EDNS + Qsection + 3 * DNSKEY + 2 * RRSIG = 145 bytes

We'll now calculate the variable part of the response:

1. The variable part of the DNS response message is determined by the different RSASHA256 module sizes. Currently the ZSK modulus is 1024 bits (Mzsk = 128 bytes), and the KSK modulus is 2048 bits (Mksk = 256 bytes).
2. The size of the modulus is equal to the size of the signature field in the RRSIG, hence a signature generated by the KSK is 2048 bits (Sksk = Mksk = 256 bytes). Note that the DNSKEY set is not covered by a signature generated by the ZSK.

We can now determine the variable size of DNS response message, with two KSKs using 2048 bit keys and one ZSK using 1024 bit keys:

VariableSize = Mzsk + 4 * Mksk = 1152 bytes.

Together, the FixedSize and the VariableSize result in a response size of 1297 bytes. (For later reference and ease of calculation, the formula to calculate a response size during KSK rollover = 145 + 4 * Mksk + Mzsk).

Interaction of Response Size and Fragmentation

A standard DNS message over UDP is limited to 512 bytes of payload; as specified in RFC 1035[5] section 4.2.1, longer messages are truncated and will have TC bit set in the

[5] See http://tools.ietf.org/html/rfc1035.

header. As clarified in RFC 5966,[6] when a client receives a response, it takes the TC flag as an indication that it should retry over TCP instead.

EDNS0, defined in RFC 2671[7] permits an increase of the DNS over UDP payload size by allowing both the client and server side to indicate what their perceived maximum UDP payload size is. RFC 2671 indicates that choosing a 1280 bytes UDP Payload Size on an Ethernet connected client is reasonable. The maximum payload size used is the smallest of the size indicated by both the client and the server. At the moment of writing, the root-servers advertise a UDP payload size of 4096, which can be considered an absolute maximum. However, for the purpose of this exercise the effect of 1280 UDP message size for IP will be considered, finding an acceptable response size limit, the effect of using 1280 bytes UDP payload size will be considered.

There is some concern that a response size of 1297 bytes will be problematic for various reasons. Internet Protocol version 6 (IPv6) guarantees a minimal size of 1280 bytes for UDP, but due to tunneling and various other reasons, this is often also the maximum path MTU size found. This will naturally lead to fragmentation if the Extension Mechanisms for DNS (EDNS) Maximum payload size is set higher than 1280. UDP fragments are often filtered, including the Internet Control Message Protocol (ICMP) response going back to the server. This leads to a black hole effect. Over IP version 4 (IPv4), the Maximum Transmission Unit (MTU) is often around the 1480 limit, but is not guaranteed.

One data point to assess the potential concern of this issue at the root zone is that of the 317 TLDs delegated at the time of writing, 28 (nearly 9 percent) have a DNSKEY response that is greater than 1280. The TLDs include some significant names, such as US (the largest, with a payload of 1883), ASIA, INFO, ORG, INFO, DK, FR, and BE. With this frequency of high payloads, it can be expected that the effects of low PMTU would have been felt by validating resolvers when validating names under these TLDs. To date, there are no known reports that that is the case. Of course, lack of this evidence does not prove that there is no potential problem, but it does suggest that perhaps the concern is minimal or else it would have been experienced at the Top Level Domains (TLDs).

A possible mitigation strategy would be for the root servers to change their EDNS0 Maximum Payload size from 4096 to 1280, which will cause the response having the truncate (TC) bit set. It is reasonable to expect a significant increase in Transmission Control Protocol (TCP) traffic due to this change, which might raise other concerns.

Interaction of Response Size and IPv6 Fragmentation

With IPv4, the path MTU between the root server and requester does not matter, because the network handles fragmentation and reassembly transparently. With IPv6 only end-

[6] See http://tools.ietf.org/html/rfc5966.

[7] See http://tools.ietf.org/html/rfc2671.

points may fragment and reassemble. Given a DNS response size of 1297, an IPv6 UDP packet containing that answer will be 1297 + 40 IPv6 header + 8 UDP header = 1345 bytes. If the Path MTU from the root server to the requester is 1345 or more, such an answer will be received without problems. A recent study[8] by Maikel de Boer and Jeffrey Bosma suggests that when the root servers don't fall back to IPv4 and the requestor also doesn't fall back, 5 percent of the requesters will not be able to get a response. Various proposals[9] suggest the use of Path MTU Discovery (PMTUD) in the name servers and/or a minimum Extension Mechanisms for DNS (EDNS0) size of 1232[10] will mitigate the problem. An obvious strategy is of course to properly configure the resolvers and validators to allow fall back to TCP.

[8] http://www.nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf.

[9] https://www.nlnetlabs.nl/blog/2013/06/04/pmtud4dns/.

[10] https://ripe65.ripe.net/presentations/167-20120926_-_RIPE65_-_Amsterdam_-_DNSSEC_reco_draft.pdf.