Project Overview for the DNSSEC KSK Management Tools RFP

Request for Proposal

12 September 2018

1 Introduction

1.1 About this Document

This document provides an overview of the Request for Proposal (RFP). It provides background and pertinent information regarding the requirements. The RFP itself is comprised of this as well as other documents that are hosted in the ICANN sourcing tool (SciQuest/Jaggaer). Indications of interest are to be received by emailing DNSSEC.KSK.Management.Tools-RFP@icann.org . Proposals should be electronically submitted by **23:59 UTC on 10 October 2018** using ICANN's sourcing tool, access to which may be requested via the same email address as above.

1.2 Overview of the Internet Corporation for Assigned Names and Numbers (ICANN)

The Internet Corporation for Assigned Names and Numbers' (ICANN) mission is to ensure the stable and secure operation of the Internet's unique identifier systems. To reach another person on the Internet, you have to type an address into your computer - a name or a number. That address has to be unique so computers know where to find each other. ICANN helps coordinate and support these unique identifiers across the world.

ICANN promotes competition and develops policy on the Internet's unique identifiers. ICANN has introduced over 1200 new generic top-level domains (gTLDs), each of which is operated by a Registry. In addition, as a function of ICANN's responsibility, it accredits domain name Registrars, who facilitate the registration of Internet domain names for individuals and organizations (i.e. Registrants). Currently, there are approximately 3,000 ICANN-accredited Registrars.

See www.icann.org for more information.

1.3 Overview of the Public Technical Identifiers (PTI), an affiliate of ICANN

Public Technical Identifiers (PTI) is responsible for the operational aspects of coordinating the Internet's unique identifiers and maintaining the trust of the community to provide these services in an unbiased, responsible and effective manner. Mainly, PTI is responsible for the operation of the Internet Assigned Numbers Authority (IANA) functions; Domain Names, Number Resources, and Protocol Parameter Assignments.

The IANA functions are the services by which the top-most level of all globally unique identifiers used on the Internet are allocated and managed. These functions, first performed by Dr. Jon Postel in the 1970s, are critical to the Internet today. The efficient, secure, and stable performance of the IANA functions ensures that the Internet's Domain Name System (DNS), Internet numbering, and protocol parameter assignments continue to be globally unique and can

support the continued operation and evolution of the global Internet. Since its inception in 1998, ICANN has provided the IANA functions to the Internet community and views this responsibility as core to our corporate mission of ensuring “the stable and secure operation of the Internet’s unique identifier systems”.

Public Technical Identifiers (PTI) was incorporated in August 2016 as an affiliate of ICANN, and, through contracts and subcontracts with ICANN, began performing the IANA functions on behalf of ICANN in October 2016.

For more information on PTI, please visit https://pti.icann.org and https://www.iana.org.

2 Scope

2.1 Project Objective

The Public Technical Identifiers (PTI), an affiliate and IANA Function Service operator for the Internet Corporation for Assigned Names and Numbers (“ICANN”), is soliciting proposals to identify a provider that will develop and maintain a new software that will replace the existing Domain Name System Security Extensions (DNSSEC) Key Signing Key (KSK) Management Tools. The selected provider, in coordination with PTI will be responsible for all aspects of development and implementation, including design, programming, testing and configuration. All deliverables must be created under formal guidelines with good documentation. The software will be open source and must incorporate industry best practices with documented test cases that will be available for the internet community.

The DNSSEC KSK Management Tools are intended to be a set of software utilities to manage the KSK life cycle and to process a Key Signing Request (KSR) and generate a Signed Key Response (SKR), as part of executing the KSK ceremonies.

PTI seeks a provider to develop and maintain this new software based on provided requirements and provide maintenance for changes and develop enhancements.
This software will help make the operation of the KSK Ceremonies to be more efficient, open and transparent.

2.2 Background

In June of 2010, ICANN held its first root zone DNSSEC KSK ceremony to secure the root zone of the DNS. Now, its affiliate, PTI, is responsible for the management of the KSK used in the deployment of DNSSEC in the root zone of the DNS. See https://www.iana.org/dnssec for more detail on DNSSEC.

Key elements of this management function take place during the KSK ceremonies. In those ceremonies, custom key management software is used to carry out operations such as KSK generation and Key Signing Request (KSR) processing.

This document assumes at least a basic understanding of DNSSEC and the design of the systems and processes involved in the use of DNSSEC in the Root Zone. The goal of this document is to provide sufficient information for a statement of work to an implementation to be agreed by PTI.

2.3 Scope of Work

The services requested in this RFP is to develop and maintain the DNSSEC KSK Management Tools.

The work is expected to be provided in three main areas:

1. Development of a software solution that meets ICANN/PTI requirements. The software development lifecycle (SDLC) for the Root Zone KSK Key management and signer software must implement relevant parts of NIST SP 800-64 for incorporating security and trustworthiness into the SDLC.

   The main tasks for software development are:

   - Create and configure a dedicated web server in the ICANN network and client that runs on that same dedicated server to exchange process for KSR and SKRs with an external party. These web services use TLS, including client-side authentication.
   - Validate and sign a KSR to generate a SKR.
   - Management of the KSK life cycle.
   - Interact with AEP Keyper Plus hardware security module (HSM).
   - Create configuration files that will be used to define the policies and parameters of the software (i.e. key lengths, algorithms, validity periods, etc.)
   - Maintain audit logs for all activities performed by each component of the software.
   - Monitor and react to specific signals sent from the HSM, these events are related to the KSK key life cycle, including but not limited to:
     - Key generation, backup, storage, recovery, archival, and destruction.
     - Exporting of public key components.
     - KSK signing and management events such as key activation, receipt and validation of public key material.
     - Successful or unsuccessful key management operations.
   - Calculate the SHA-256 and the PGP wordlist hashes of the input KSR and the output SKR.
   - Create and maintain the root trust anchor files.
   - The software shall be configurable to publish and sign the KSR with different KSK to facilitate the automated update of DNSSEC trust anchors in the root zone as outlined in RFC5011.
   - Future ceremonies may use different signing algorithms, the software should support this change.

2. Complete documentation and knowledge transfer of all functionalities of the software including the software development lifecycle (SDLC) and all necessary documentation and pertinent information including, but not limited to:

   - Process documentation: Documentation of all activities of the software development.
   - Product documentation: System and user documentation.
     - System documentation: Requirements, design, architecture, functional specification and testing (quality assurance).
     - User documentation: User, installation and reference manual.

   Documentation to support software audits as specified in the DNSSEC Practice Statement for the Root Zone KSK Operator (DPS) https://www.iana.org/dnssec/dps/ksk-operator/ksk-dps.txt. must include the core components of the System Architecture, Functional Specification and Implementation of the software.

   - System Architecture documentation. The DPS section 5.8.1. requires that:
     - There is a documented architectural design describing the security domains and functions maintained by the signer;
     - The architectural design demonstrates that the signer system prevents bypass of the security-enforcing functionality.

   A stand-alone System Architecture document should be produced that meets the requirements in the DPS. The document should emphasize the security elements in those requirements and describe clearly how they are met by the design.

   - Functional Specification documentation. The DPS section 5.8.1. requires that:
     - There is a functional specification completely representing the signer system and all operations associated with it.

   A stand-alone Functional Specification for implementations that meet the requirements of the Software Architecture should be produced. This Functional Specification must specify exactly how the various software components will be used in detail, including, but not limited to:
     - All available command-line arguments.
     - Formats of all supporting files (e.g. schemas for JSON-format configuration files).
     - Error codes and the way in which errors are reported.
     - Performance constraints (e.g. timeout values).
     - The formatting of critical output from all tools.
     - The Functional Specification should include and specify in detail all aspects of each implementation that could possibly be used to compare two different implementations with automation, e.g. as part of an automated regression test suite, including error conditions.

- Implementation. The DPS section 5.8.1. requires that:
  - There is a modular design description and a one-to-one correspondence with the modular decomposition of the implementation;
  - The implementation representation completely and accurately implements the security-enforcing functions.

  Each implementation must be assessed both for internal documentation and clarity, and also for functional correctness to meets the requirements of the Functional Specification.

3. Development of a test framework. An automated test framework must be specified and implemented, consistent with modern software development practices, to ensure that maintenance of any implementation can be tested in a reliable and consistent manner.

   For example, in the case where a code change has been made in an implementation to remedy a bug report, it ought to be sufficient for an external auditor familiar with the tools and programming languages used in an implementation to complete a visual code review to confirm that no unexpected functionality has been introduced, and then provide output of the standard test framework to confirm that the essential functionality of the software has been preserved through the change.

   The Test Framework must be published with sufficient documentation that interested third parties can use it to test particular implementations without PTI or ICANN staff involvement.

3 High Level Selection Criteria

The decision to select a provider as an outcome of this RFP will be based on, but not limited to, the following selection criteria:

1. Capability and experience developing DNS and DNSSEC software.
2. Experience with developing software for HSMs.
3. Ability to document the proposed implementation approach including software architecture, design and solution.
4. Documentation on their test methodology.
5. Responsiveness and flexibility in working with clients to meet specific requirements and agreement terms.
6. Quality and process excellence.
7. Value added services.
8. Financial value / pricing

4 Business Requirements

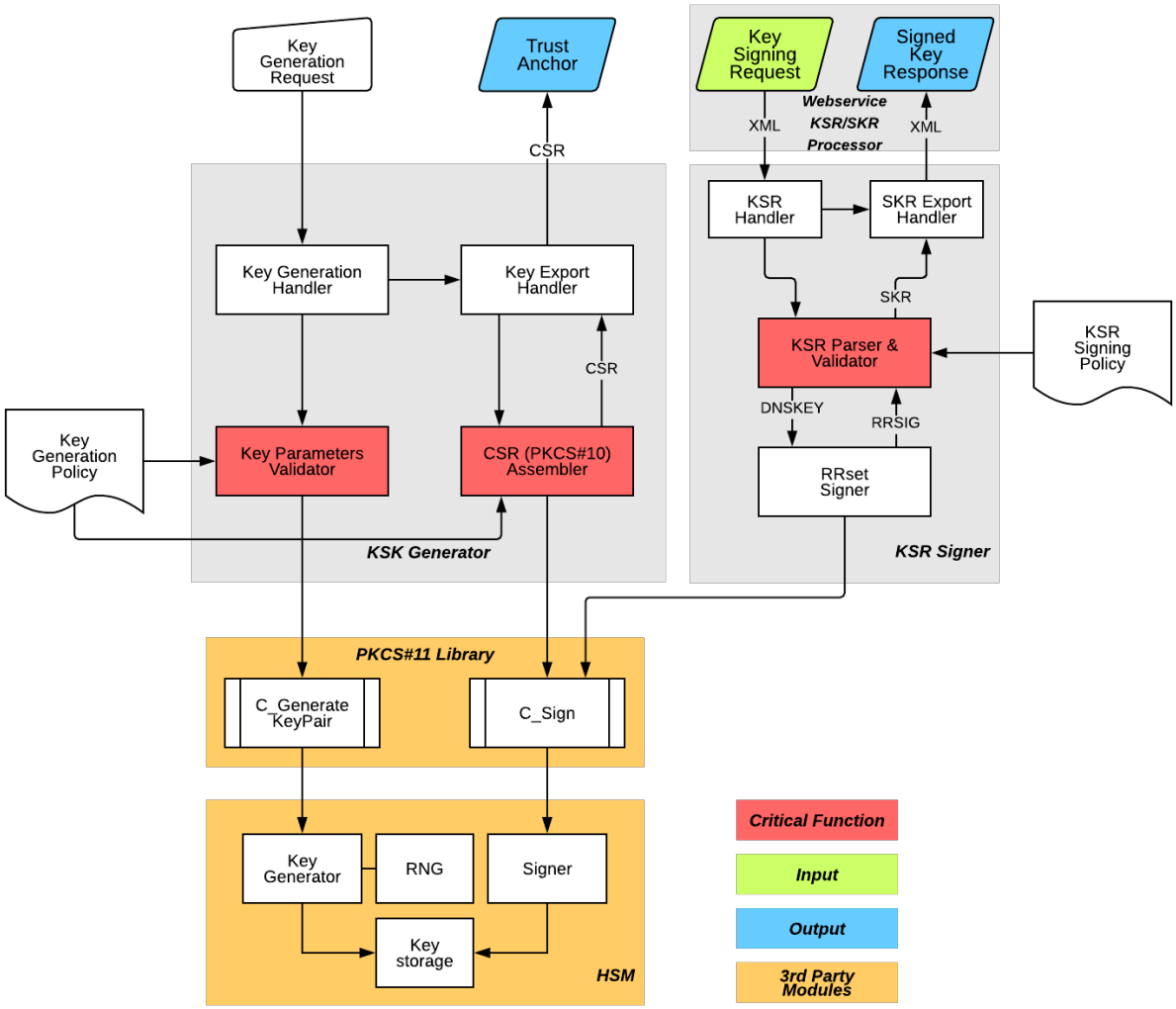
Below is a high level summary of the business requirements:

1. Source code must be supplied for all components.
2. Copyright of all components will be retained by ICANN.
3. The software is to be fully and solely owned by ICANN.
4. Software will be published by ICANN as open source under a license of ICANN's choosing.
5. The software should be properly documented with comments in the source code and stand-alone companion documentation using a tool such as Doxygen sufficient to allow future maintenance of the software and to support third-party software audits.
6. The preferred format for the documentation is Markdown.
7. The key management software must interact with hardware security modules (HSMs) to perform all cryptographic operations and must be broadly compatible with vendor-supplied PKCS#11 libraries in general and with those used by ICANN's HSM vendors. (Currently, compatibility with the AEP Keyper HSM and with OpenDNSSEC SoftHSM for testing is required).
8. Use of third-party packages should be avoided to reduce software complexity and reduce the footprint of code that needs to be covered in an audit.
9. Selection of programming languages is not prescribed, but the preferred languages for the deliverable software is Python or Java. The code shall be well-documented and modular in design such that it is easily understood by third parties for audit purposes.
10. The software should have minimal dependencies on particular CPU architectures and operating systems, allowing the software to be ported easily to different target environments.
11. Comprehensive unit and integration tests that provide full test coverage must be included. Instructions for executing tests shall be provided.
12. Development environment configuration and build instructions shall be included.
13. The software must implement all essential functionalities of the existing software available at https://github.com/iana-org/dnssec-keytools, plus new requirements: KSK algorithm rollover, test framework, and a trust anchor XML generator.
14. The software should be capable of verbose debugging output in all relevant functions to facilitate troubleshooting and audit logging.
15. Policies and parameters (such as key lengths, algorithms, validity periods, etc.) used in key signing operations should be definable in configuration files that are passed at runtime, rather than hard-coded.
16. All components must interact with the user via the Posix command line interface.
17. All configuration files and command-line parameters should maintain equivalent functionality to those in the existing software, so that the operations carried out in KSK Ceremonies can use either the existing software or the new software without major changes in procedures or ceremony scripts.
18. Revision of the software based on the external review is required.
19. The development should be follow the reproducible binary standards available at https://reproducible-builds.org.
20. The current OS where these tools will be executed is available at https://github.com/iana-org/coen.

21. Review, provide recommendation for improvement and update the tools (written on scripting language) that support the KSK ceremonies, for example: printing, audit logs capture and hash calculation scripts.
22. Provide a plan for knowledge transfer.

5 Functional Requirement

The functional diagram below illustrates the KSK Generator and KSR Signer components in a high level summary:



KSK Generator

The KSK Generator generates a KSK inside an HSM. The public key is signed by the private key and exported as a PKCS#10 Certificate Signing Request (CSR). Fingerprints of the public key and the CSR in various formats are returned to the user for review during the KSK Ceremony and will be printed on paper.

Key Generation Handler

The process of generating a new key involves the validation of the Key Generation Policy, successful processing of key generation requests in the HSM to express that policy, the successful generation of a signature over the public part of the generated key pair with the private part, the export of the resulting proof of possession of private key as a PKCS#10 CSR and the output of the keytag of the new KSK and fingerprints in various formats of both the public part of the generated key pair and the CSR.

The Key Generation Handler is responsible for receiving the Key Generation Request, coordinating those various functions with the Key Parameters Validator, the HSM via the PKCS#11 library and the Key Export Handler, and returning appropriate success or failure indications to the user with other information about the generated KSK as described above.

The Key Generation Handler may be implemented as part of the KSK Generator tool, or as library functions or separate executables invoked by the KSK Generator tool.

Key Parameters Validator

The Key Parameters Validator ensures that the Key Generation Policy is present, well-formed, internally-consistent and is syntactically correct. The Key Parameters Validator may be implemented as part of the KSK Generator tool, or as library functions or separate executables invoked by the KSK Generator tool.

CSR Assembler

The CSR Assembler constructs a Certificate Signing Request (CSR) using the generated KSK. The CSR attributes are specified in the Key Generation Policy. The result provides proof of possession of the private key, since it represents a signature by the private part over the public part of the generated KSK key-pair.

The CSR Assembler may be implemented as part of the KSK Generator tool, or as library functions or separate executables invoked by the KSK Generator tool.

Key Export Handler

The Key Export Handler exports the CSR constructed by the CSR Assembler as a file in PKCS#10 format.

The Key Export Handler may be implemented as part of the KSK Generator tool, or as library functions or separate executables invoked by the KSK Generator tool.

Trust Anchor

The Trust Anchor is the CSR constructed by the Key Export Handler [RFC7958].

Key Generation Policy

The Key Generation Policy provides the parameters necessary for KSK generation. The policy must ensure that when a new KSK is generated the key id (a.k.a. keytag) and keylabel are not equal to any previously generated key.

KSR Signer

The KSR Signer will validate a Key Signing Request (KSR), generate signatures to be used in the root zone as RRSIGs over the apex DNSKEY RRSet as requested in the KSR and output the result as a Signed Key Response (SKR). The KSR Signer will generate signatures using one or more KSKs available on attached HSMs according to the KSR Signing Policy.

Key Signing Request (KSR)

A KSR is supplied by the Root Zone Maintainer for processing as an XML document. The KSR specifies a set of time intervals together with the DNSKEY RDATA relating to the ZSKs that will be used during each of them. Every key bundle is signed using every included ZSK as proof of possession of the corresponding private key.

The SKR that was generated by the processing of the immediately preceding KSR (usually at the previous KSK Ceremony) is also supplied to the KSR Signer, in order that various consistency checks can be carried out.

KSR Handler

The processing of a KSR involves successful completion of the KSR Parser & Validator component, the construction of DNSKEY RRSets that include the relevant ZSKs and KSKs to be published during each time interval, the generation of signatures over those RRSets by the RRSet Signer and the construction of a corresponding SKR to be exported using the SKR Export Handler.

The KSR Handler may be implemented as part of the KSR Signer tool, or as library functions or separate executables invoked by the KSR Signer tool.

KSR Parser & Validator

The KSR Parser confirms that the KSR being processed is well-formed, is syntactically-correct, internally-consistent and does not violate the constraints specified in the KSR Signing Policy (for example, a KSR should not be allowed to request a signature with a validity period in the past, or too far in the future)

The KSR Parser also verifies that the KSR is compatible with the SKR produced during the previous KSK Ceremony, for the previously-submitted KSR.

RRSet Signer

The RRSet Signer generates signatures over a supplied DNSKEY RRSet using one or more KSKs specified in the KSR Signing Policy. It uses the PKCS#11 interface to the local HSM to carry out signing functions.

The RRSet Signer may be implemented as part of the KSR Signer tool, or as library functions or separate executables invoked by the KSR Signer tool.

SKR Export Handler

The SKR Export Handler constructs an SKR from the KSR being processed and the signed RRSets constructed by the RRSet Signer. The result is stored in a file as well-formed XML consistent with and validated against the SKR schema.

The SKR Export Handler may be implemented as part of the KSR Signer tool, or as library functions or separate executables invoked by the KSR Signer tool.

Signed Key Response (SKR)

The SKR is the product of the SKR Export Handler. It is delivered as a single file suitable for delivery to the Root Zone Maintainer.

KSR Signing Policy

The KSR Signing Policy provides the parameters necessary for KSR processing.

Webservice KSR/SKR Processor

The KSR/SKR Processor receives a KSR from an authorized client (operated by the Root Zone Maintainer) over a secure network channel (HTTP over TLS with client authentication), provides some validation of the KSR (such as integrity and validation of the previous SKR) to provide a useful, real-time response to the client, and generates notification when a KSR has been submitted.

More details can be found in the DPS section 5 and 6 at https://www.iana.org/dnssec/dps/ksk-operator/ksk-dps.txt.

6 Project Timeline

The following dates have been established as milestones for this RFP. ICANN reserves the right to modify or change this timeline at any time as necessary.

Activity	Estimated Dates
RFP published	12 September 2018
Participants to indicate interest in submitting RFP proposal	**26 September 2018 by 23:59 UTC**
Participants submit any questions to ICANN	**26 September 2018 by 23:59 UTC**
ICANN responds to participant questions	3 October 2018
Participant proposals due by	10 October 2018 by 23:59 UTC
Evaluation of responses	11 October through 16 November 2018
Final evaluations, contracting and award	19 November through 14 December 2018

7 Terms and Conditions

General Terms and Conditions

1. Submission of a proposal shall constitute Respondent's acknowledgment and acceptance of all the specifications, requirements and terms and conditions in this RFP.

2. All costs of preparing and submitting its proposal, responding to or providing any other assistance to ICANN in connection with this RFP will be borne by the Respondent.

3. All submitted proposals including any supporting materials or documentation will become the property of ICANN. If Respondent's proposal contains any proprietary information that should not be disclosed or used by ICANN other than for the purposes of evaluating the proposal, that information should be marked with appropriate confidentiality markings.

Discrepancies, Omissions and Additional Information

1. Respondent is responsible for examining this RFP and all addenda. Failure to do so will be at the sole risk of Respondent. Should Respondent find discrepancies, omissions, unclear or ambiguous intent or meaning, or should any question arise concerning this RFP, Respondent must notify ICANN of such findings immediately in writing via e-mail no later than ten (10) days prior to the deadline for bid submissions. Should such matters remain unresolved by ICANN, in writing, prior to Respondent's preparation of its proposal, such matters must be addressed in Respondent's proposal.

2. ICANN is not responsible for oral statements made by its employees, agents, or representatives concerning this RFP. If Respondent requires additional information, Respondent must request that the issuer of this RFP furnish such information in writing.

3. A Respondent's proposal is presumed to represent its best efforts to respond to the RFP. Any significant inconsistency, if unexplained, raises a fundamental issue of the Respondent's understanding of the nature and scope of the work required and of its ability to perform the contract as proposed and may be cause for rejection of the proposal. The burden of proof as to cost credibility rests with the Respondent.

4. If necessary, supplemental information to this RFP will be provided to all prospective Respondents receiving this RFP. All supplemental information issued by ICANN will form part of this RFP. ICANN is not responsible for any failure by prospective Respondents to receive supplemental information.

Assessment and Award

1. ICANN reserves the right, without penalty and at its discretion, to accept or reject any proposal, withdraw this RFP, make no award, to waive or permit the correction of any informality or irregularity and to disregard any non-conforming or conditional proposal.

2. ICANN may request a Respondent to provide further information or documentation to support Respondent's proposal and its ability to provide the products and/or services contemplated by this RFP.

3. ICANN is not obliged to accept the lowest priced proposal. Price is only one of the determining factors for the successful award.

4. ICANN will assess proposals based on compliant responses to the requirements set out in this RFP, responses to questions related to those requirements, any further issued clarifications (if any) and consideration of any other issues or evidence relevant to the Respondent's ability to successfully provide and implement the products and/or services contemplated by this RFP and in the best interests of ICANN.

5. ICANN reserves the right to enter into contractual negotiations and if necessary, modify any terms and conditions of a final contract with the Respondent whose proposal offers the best value to ICANN.