Ad-hoc trust associations with Trust Anchor Repositories

Stefan Roelofs
stefan.roelofs@os3.nl

July 2, 2009

UNIVERSITEIT VAN AMSTERDAM

Supervisor:
Yuri Demchenko

Abstract

This project report presents the results of the research done for the System & Network Engineering (SNE) research group at the University of Amsterdam on Trust Anchor Repositories (TAR). Three topics were investigated: comparison of the original global DNSSEC trust model and the TAR based island model; the future of TAR; and how DNSSEC and TAR can be used in Network Resource Provisioning (NRP).

TAR emerged as an interim solution to speed up DNSSEC deployment but is currently considered as a technology that can solve known issues with interdomain trust management in an open Internet environment. With TAR it is possible to implement DNSSEC in some zones without signing the root. This creates so-called islands of trust where the trust relations to those islands is maintained through TAR. The traditional DNSSEC implementation suggests out of band communication to verify the initial trust anchor, while the TAR based DNSSEC infrastructure can provide and requires periodic or on-demand lookup. There are in-band examples (DLV) as well as out-of-band examples (through HTTPS etc.). It is also important to mention that the availability of TAR directly impacts the availability of the DNS system.

Research on the future of TAR's shows that there is a different outlook by standardization organizations if TAR's should remain in effect after the DNS root zone has been signed. In particular, the report provides reference to and analyses the three models proposed by the NIST: global TAR, Community of Interest (COI) TAR, and Enterprise TAR. Global TAR's are used for supporting public DNSSEC deployment, COI TAR's for organizations to have secure relations (partnerships etc.) and Enterprise TAR's to collect trust anchors of internal signed namespaces. While policies in the hierarchal model were mutually agreed between a parent and a child zone, for TAR, there are multiple concepts on how they can be handled.

When a DNSSEC infrastructure will become a common practice in every domain, a COI TAR could be used for creating and managing trust associations in on-demand network resource provisioning that currently uses a shared secret model. The report proposes to move to Public Key Cryptography based on the DNSSEC ZSK public-private key pair of each domain. Every participating domain should publish its trust anchor in the TAR which should be trusted by all participating domains. During the reservation stage, the security context information of every passed domain is collected using pilot token type 3. During deployment stage, a Session Based Key (SBK) will be communicated backwards using pilot token type 4. The SBK is encrypted and the token is signed using DNSSEC domain keys.

The report discusses two possible implementation scenario's. The first scenario is build on the DNSSEC resolving capabilities being present in every domain in order to verify the neighbor domain keys. The second scenario suggests that only the destination host or domain will have DNSSEC resolving capabilities. Both scenarios use pilot token type 4 for communicating the encrypted SBK to each participating domain. In the second scenario however, the public key of the originating domain is also send with pilot token type 2, 3 and 4 because not all domains have resolving capabilities but need to check the token signature. For these domains there is no opportunity to verify the public key of the origi-

nating domain out-of-band. This means that in this scenario, the key could be replaced by a malicious key during transmission.

Acknowledgements

I would like to offer gratitude to the following people and organizations:

- The research group System & Network Engineering (SNE) for providing the means and opportunity to conduct this research project.
- Yuri Demchenko, my supervisor, for offering help, support and ideas.
- All those who gave their feedback on this document.

Contents

List of Figures

1 Introduction

This report presents the results of the research done for the System & Network Engineering (SNE) research group at the University of Amsterdam on Trust Anchor Repositories (TAR). TAR emerged as an interim solution to speed up DNSSEC deployment but is currently considered as a technology that can solve known issues with interdomain trust management in an open Internet environment.

Domain Name System Security Extension (DNSSEC) is designed to answer security vulnerabilities/concerns of the original DNS protocol such as cache poisoning, information leaks, software bugs and exploiting reverse DNS [1]. The first specification, released by the IETF in 1997 [2], had loads of comments for which a new revision called DNSSECbis was released in 2005 [3].
In order to deploy this specification over the whole DNS system, all zones need to be signed, starting at the root name servers. As of today, the root servers are not yet signed. Some owners of Top Level Domains (TLD) like .org, .se and RIPE-NCC for the reverse namespace (delegated by IANA) are early adopters of the technology [4].
Due to the delay in signing the DNS root, multiple initiatives [5] [6] [7] have been started (and some executed) to build (interim) trust anchor repositories. These repositories contain multiple trust anchors submitted by zone administrators. They create islands (or archipelagos) of trusted domains.

This project considers using the TAR based DNSSEC trust infrastructure for inter-domain trust management in on-demand Network Resource Provisioning.

1.1 Research Questions

This report will provide answers to the following research questions:

- What are the differences between the original DNSSEC global trust model and the island based model with Trust Anchor Repositories?
- What models are currently developed and what could or should be future developments?
- How can the Trust Anchor Repositories be of use in multi-domain on-demand network resource provisioning?

1.2 Approach

During the project, the following approach was used.

- Research on existing global and community oriented trust models such as DNSSEC, PKI, TAR and on-demand network resource provisioning.
- Interview RIPE-NCC technical staff members on DNSSEC and TAR.

- Investigation and comparison of the global trust models and TAR models.
- Investigation of future developments on TAR.
- Investigation use of TAR in network resource provisioning.
- Preparation of final report and presentation.
- Finishing report and processing of comments supervisor.

1.3 Report structure

This report provides an answer to the research questions stated earlier. The report is organized as follows. Chapter 2 provides a short introduction to DNSSEC and chapter 3 explains and compares the global trust hierarchies and trust anchor repository concepts. Practically implemented TAR's will also be discussed in this chapter. In chapter 4, future developments of TAR are discussed. Chapter 5 provides some of the essential concepts of Network Resource Provisioning (NRP) and chapter 6 provides two scenario's for TAR enhancements in NRP. Chapter 7 concludes the research questions and provides future work.

2 DNSSEC

Domain Name System Security Extension (DNSSEC) is designed in a response to security implications on the original DNS protocol. This section provides a short overview of these security implications and provides an introduction to DNSSEC.

2.1 DNS Threats

RFC3833 [8] provides a threat analysis of the Domain Name System. Threats are subdivided in different categories:

- Packet Interception: relative simple man-in-the-middle attack on DNS queries.
- ID Guessing and Query Prediction: executed through bugs in DNS Server software causing the transaction ID of the DNS query to be predictable [9] [10] allowing the DNS reply to be spoofed.
- Name Chaining: pollution of a caching DNS server with extra data in the DNS query that is not directly related to the original request.
- Betrayal By Trusted Server: making a malicious DNS server handle client replies through wrong DHCP offers or server hacks/bugs.
- Denial of Service: making the DNS server unavailable through the generation of too much server or network load.
- Authenticated Denial of Domain Names: removed entries from a DNS reply to deny existence of the data (actively discussed topic if this should be detectable).
- Wildcards: associated with the previous threat; the possible miss-use of wildcards in DNS records.

It is also possible for an attacker to combine any of these threats to maximize the effectiveness of an attack. A notable example is the Kaminsky attack [1]. The Kaminsky attack uses the ″Additional″ section of the DNS reply packet to advertise a malicious ″authoritative″ name server. Through this malicious name server false DNS replies can be generated.

2.2 DNSSEC

DNSSEC was designed by the Internet Engineering Task Force (IETF) in a response to the threats on the original DNS protocol. Although it does not prevent all the RFC3833 [8] listed threats (i.e. DOS attack), it solves most of the issues. DNSSEC provides origin authenticity, data integrity, and secure denial of existence using public-key cryptography. This is achieved using digital signatures: every DNS record is signed using a cryptographic algorithm. Resolvers can check these signatures proving the authenticity and integrity of the data [11]. Also, when there is no data to answer a query, authoritative servers can provide a response that proves no data exists.

The first specification of DNSSEC was released in 1997 in RFC2065 [12] and a revised edition in 1999. These specifications had loads of comments on their scalability. In order to solve this, a new specification called DNSSECbis (”bis” was borrowed from the ITU and is a French musical notation for ”repeated”) was released in 2005. It is described in RFC4033 through 4035.

DNSSEC relies on building a chain of trust. In order to delegate sub-zone signing, DNSSEC uses a technique similar to domain delegation. A domain holder can delegate the signing of a sub-domain by trusting the key that is used to sign the sub-zone. This creates a chain of trust starting at the root zone. The key in the root needs to be explicitly trusted by the resolver, this is called a trust anchor. Since the root servers are currently not signed, the concept of Trust Anchor Repositories (TAR) was invented. Trust Anchor Repositories hold the trust anchor of one or multiple zones. TAR's will be explained in detail in chapter 3.

DNSSEC is usually deployed using two keys, a Key Signing Key (KSK) and a Zone Signing Key (ZSK). The KSK is used to sign the ZSK and the ZSK is used to sign the zone. Although this could technically be the same key [15], in practice a different one is used. This deals with the different requirements to the lifetime and strength of each key. In the parent zone, a hash of the child's KSK called the DS record is stored to express the trust in this key. If the same key is used as KSK/ZSK in the child, this would require key rollover with the parent if the ZSK is changed. The ZSK is of a short length (resulting in a short lifetime) to limit its operational impact while signing all zone records [13]. The KSK on the other hand only signs the ZSK and can be of a greater length (resulting in a longer lifetime).

DNSSEC also has some cons which are discussed in RFC3833 [8], notably: complexity, increase in zone size, and increase in DNS response packet size. Next to this, DNSSEC does not provide confidentiality: it only proves that a response is genuine but does not keep it hidden. Competitors of DNSSEC like DNSCurve do provide confidentiality and some protection against DOS attacks [14].

3 Global Trust Hierarchy versus Island Based Trust concepts

Often, standards like DNSSEC are created with the intention that (some day) they will be deployed globally. Island based trust runs against this concept by partially deploying the standard. This section will compare global trust hierarchies with island based concepts.

3.1 Global Trust Hierarchy

A Global Trust Hierarchy is a hierarchy rooted at one or multiple, but limited, locations. These locations are called trust anchors. Trust anchors should be trusted by everyone who wants to use the (downstream) hierarchy. In this section, the (intended) global trust hierarchy of DNSSEC and the Public Key Infrastructure (PKI) will be discussed.

3.1.1 DNSSEC

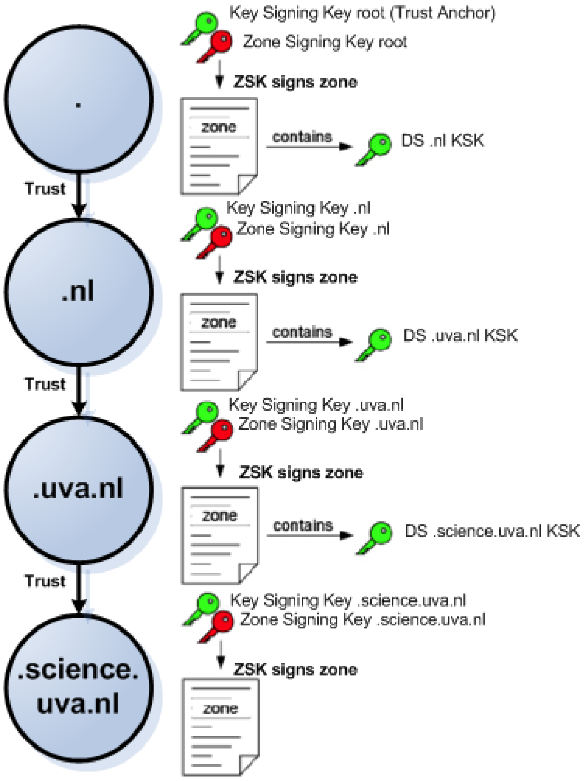

Figure 1: DNSSEC Chain of Trust
(picture partly derived from Surfnet)

When looking at the structure of DNS, it is a tree rooted at a single, empty label (””). Because of this single location, it was thought to use it in DNSSEC deployment as a secure entry point (SEP). DNS Resolvers who want to use DNSSEC use this entry point in order to start building a "chain of trust". This essentially means that, in order to validate data from a specific subzone, the resolver starts in the root zone and walks down the chain until it arrives at the appropriate zone. As explained in chapter 2 in the DNSSEC introduction, two keys are used to sign each zone, the KSK and the ZSK. These keys are stored in the DNSKEY resource record as specified in RFC4033 [3]. A parent stores the hash of each child's KSK in the DS record to express trust in that key. This DS record is also signed by the ZSK of the parent. Figure 1 shows an example chain of trust build from the root zone (.) down to the science.uva.nl domain. The KSK (or a hash of the KSK) in the DNSKEY resource record of the root zone is known by the resolver and is therefore the trust anchor in the (global) DNS hierarchy. The advantage of this hierarchy is that the resolver only needs to store and maintain the key used in the root zone rather than every single key of every zone in the DNS tree.

3.1.2 Public Key Infrastructures

When looking at Public Key Infrastructures for X.509 certificates as specified in RFC5280 [16], multiple tree's of Certificate Authorities can be imagined. These trees are rooted at a trusted Root Certificate Authority (CA). These trusted root CA's are the secure entry points to the Public Key Infrastructure. For users of X.509 certificates, their certificate validation software usually contains the public keys of multiple trusted root CA's. The trusted root CA's however are likely to have one or multiple subordinate CA's for security aspects or administrative purposes.

Like in DNSSEC, it is possible to build a "chain of trust" between an issued certificate and a trusted (root) CA. An X.509 certificate has an extension field called the Authority Information Access (AIA) [16]. In this AIA, an attribute "caIssuers" holds a method to obtain a copy of the Issuer's certificate. This will normally be a URL to a certificate (store). When obtaining the Issuer certificate of a subordinate CA, the resolver will find two certificates. One will be self-signed, and another certificate showing the Issuer has cross-certified with another, up in the hierarchy, CA. The "caIssuers" field of the cross-certificate can once again be checked to obtain a parent CA's certificate working up the hierarchy trust chain [17]. This is done until a trusted CA is found or the attribute is left empty meaning the chain stopped and the certificate cannot not be verified to a trusted CA. When comparing DNSSEC and PKI hierarchy, DNSSEC has no formal authority structure. There is no central CA in DNSSEC: just keys, signatures and a chain of trust [18].

3.1.3 Public Key Directory

Another example of an (intended) global PKI is the Public Key Directory (PKD) maintained by the International Civil Aviation Organization (ICAO) [20]. This directory is developed in order to check the digital signatures stored on ePassports (passport equipped with a chip) of countries that supply them to their citizens. Countries participating in the PKD must pay an administrational fee and for that can get access to other members public keys as well as publishing their own. However, not all countries are willing to participate in the project, mostly because of political reasons (e.g. Iran). Currently (June 2009) 14 countries are enrolled in the PKD [21].

3.2 Island Based Trust

Island based concepts are based on a hierarchy that is partitioned in multiple islands of trust. These islands are individually rooted at a single location. This location is the trust anchor of the island providing a secure entry point. In order to trust these anchors, a user of the infrastructure has to obtain and maintain multiple trusted keys. To circumvent this administration, repositories of public keys were invented which can be trusted as a whole. In DNSSEC, these repositories are called Trust Anchor Repository (TAR). DNSSEC TAR's were developed in order to speed up the deployment of DNSSEC while the root and numerous TLD's are not signed. This section will discuss a number of TAR implementations.

3.2.1 DLV TAR

DNSSEC Lookaside Validation (DLV) is based on the Trust Authority concept, launched by Samuel Weiler in 2004 in his paper "Deploying DNSSEC Without a Signed Root" [6]. This concept however, was not directly followed by a practical implementation. The concept splits the need for a repository in two designs: a trust authority for a single zone (e.g. a TLD) and an hierarchical trust authority (for an island of trust). It suggests to store the trust anchor of a zone in an other zone that can be verified using DNSSEC and a (at the validator) pre-configured trust anchor.
The practical implementation of Samuel Weiler's concept was DLV, developed by the Internet Systems Consortium (ISC), as an extension to DNSSECbis. The ISC is known for developing and maintaining BIND and DLV is therefore implemented in BIND 9.4.3-P2 and later. DLV is a method to store and publish trust anchors outside the DNS delegation chain. Lookaside Validation resolvers are able to validate DNSSEC signed data from zones whose parent or ancestors are not signed or do not publish DS records. RFC5074 [22] specifies the architecture of DLV:

> "DNSSEC Lookaside Validation allows a set of domains, called "DLV domains", to publish secure entry points for zones that are not their own children." With DNSSEC, validators may expect a zone to be secure when validators have one of two things: a preconfigured trust anchor for the zone or a validated Delegation Signer (DS) record for the zone in the zone's parent (which presumes a preconfigured trust anchor for the parent or another ancestor). DLV adds a third mechanism: a validated entry in a DLV domain (which presumes a preconfigured trust anchor for the DLV domain). Whenever a DLV domain contains a DLV RRset for a zone, a validator may expect the named zone to be signed. Absence of a DLV RRset for a zone does not necessarily mean that the zone should be expected to be insecure; if the validator has another reason to believe the zone should be secured, validation of that zone's data should still be attempted. [22]

The zone entry points are published using the DLV record type. The DLV record has the same semantics as the DS record and is specified in RFC4431 [19]. ISC set up their own DLV domain named ”dlv.isc.org”. Other TAR’s are also hosting DLV’s such as the SecSpider project [24] ”dlv.secspider.cs.ucla.edu” and the IKS-Jena TAR [27] ”dnssec.iks-jena.de”.

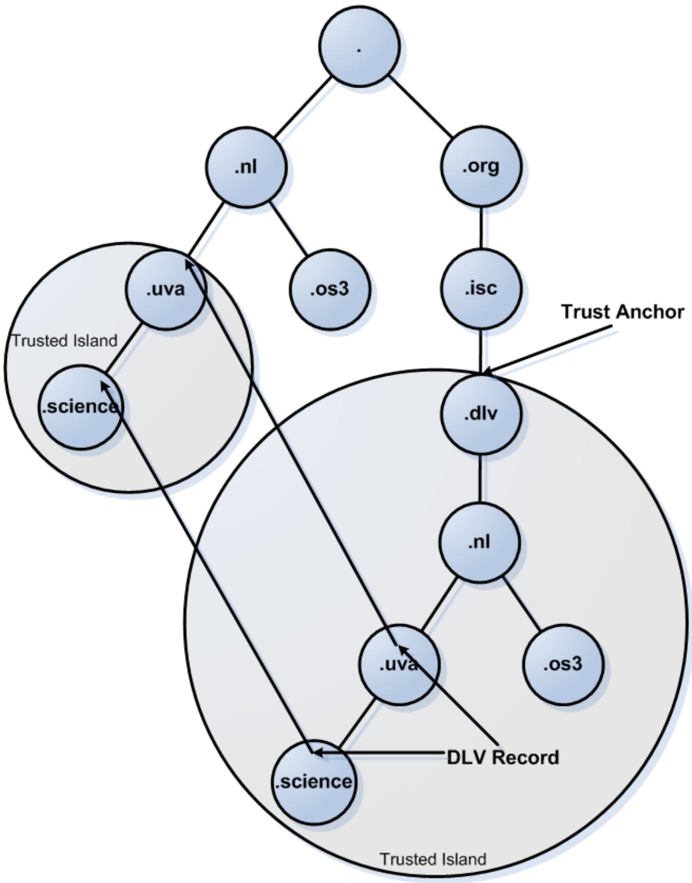

Figure 2: DNSSEC Lookaside Validation (DLV) Overview
DLV Records provide a secure entry point to DNSSEC signed domains.

Figure 2 depicts an example DLV situation for science.uva.nl using the ISC DLV TAR. In the figure, the .nl domain is not signed but the .uva and .science are. In the scenario, any resolver who wants to validate DNS data from the science or uva domain should trust the DLV anchor. Using DLV it is possible to ”mount” science.uva directly under dlv.isc.org. However, in order to limit resolver traffic and have no overlapping domains, the ISC repository maintains the full DNS name under dlv.isc.org.
When a resolver wants to validate science.uva.nl, it should first try to use a standard trust anchor. If this is not possible or results insecure, it should try using the DLV record. To find the DLV record for the queried domain, the validator starts by looking for a DLV corresponding to the QNAME. If there is no DLV record present, this will be verified using a (validated) NSEC record. If the name is also not the apex of the DLV domain, the validator

will remove the leading label from the QNAME and try again. This procedure is repeated until a DLV record is found or it is proven that no DLV record is present for the QNAME. All DLV records and NSEC records are validated using the normal DNSSEC procedure and trust delegations within the DLV domain.
Registration to the ISC DLV is done manually through a webform (however according to RIPE-NCC spokesman their signed zones are automatically and unasked listed by ISC). The registration requires a challenge sent via e-mail in order to proof access to the key. Updates (e.g. key-rollover) and deletions of trusted anchors are currently executed manually by the registrant but RFC5011 [28] (automatic key rollover detection and processing) will be implemented in the near future to automate this.

3.2.2 Manual TAR

While DLV provides an alternative lookup tree, it is also possible to let a resolver trust a repository containing keys of multiple trust islands. One way to acquire trust anchors is manually. IANA [7] has currently implemented this model in their (currently beta) "Interim Trusted Anchor Repository". The IANA ITAR only contains Top Level Domain (TLD) trust anchors (DS Records). The trusted anchors are manually submitted by zone administrators. Resolvers must download the trusted anchors in order to use them. The distribution of the anchors is done through HTTPS, RSYNC and FTP and a digest and PGP signature are available for verification. This means that an out-of-band mechanism is used, possibly impacting the network configuration on the resolver side. The policy of the IANA ITAR states that it will be removed as soon as the root zone is signed [23].
Updates and deletions of trust anchors at the IANA repository are done manually by the zone administrators.

3.2.3 Automatic TAR

Another way of acquiring trust anchors for a TAR is by automatic means. The SecSpider project from the University of California [24], operated since 2005, does so by crawling DNS zones. Next to building a repository, this information is used to monitor the DNSSEC rollout and to analyse deployment behavior [25]. SecSpider queries DNS zones for data an behavior and classifies them as secure or not. These queries are launched from 8 "Vantage points" (pollers) located throughout the US, Europe and Asia. Because of the distributed polling method, any attack on the automatic listing (e.g. spoofing) must be launched against all pollers [26]. SecSpider crawls zones that have been submitted by users (online), from a list of 2.5 million zones and walked through NSEC. A zone is classified as secure when all nameservers of the zone meet the following criteria [24]:

- A zone's nameservers must support EDNS0.
- A zone's nameservers must return RRSIG records with data when the DO bit is set.
- Returned RRSIGs must correspond to DNSKEYs that are also served by the zone.
- The zone must not have a CNAME for it's apex.

- The zone must return NSEC records for names that do not exist.

In order to determine an island of trust, SecSpider traces all trust delegations as far up the DNS hierarchy as possible. When zones are reached who's parent does not contain a DS record, the zone is considered the trust anchor for that island of trust and is listed.
SecSpider publishes the DNSKEY and DS record for all crawled and considered secure zones online. These are accessible for download through HTTP and can be verified using a GPG key. SecSpider has also deployed DLV at ".dlv.secspider.cs.ucla.edu" which contains DLV records for all crawled and classified secure DNS zones.
Because of SecSpider's daily crawling, updates and deletions of trust anchors are automatically processed.

Another implementer of the (semi) automatic TAR is IKS-Jena [27]. Sign-up on IKS-Jena is done manually through an open registration. Only domains directly under the TLD zone are allowed. Keys are automatically detected by scanning TLDs and the reverse DNS tree looking for DNSKEY RR's. Access to the trust anchors is provided through HTTP and DLV "dnssec.iks-jena.de".
Updates and deletions of trusted anchors at the IKS-Jena repository are processed automatically according to RFC5011 [28]. RFC5011 is explained in section 3.3.2.

3.2.4 Relation to Public Key Infrastructure

It is worth mentioning that a Trust Anchor Repository is about nothing more than providing trust relations to islands of trust. A TAR could be compared to the trusted CA store on a PKI resolver because they can both contain secure entry points to some point in a tree. However, in this report discussed TAR's are maintained by a trusted third party while the trusted CA store is under administration of the verification software manufacturer, a user or a network administration body. Also, regulations to be included in the trusted CA store by software manufactures are a lot different than the discussed TAR policies.

3.3 Comparing the concepts

This section provides an overview on the differences between the Global Trust Hierarchy and Island Based concepts.

3.3.1 Governance

While the hierarchy model only requires parent - child communications, a TAR requires a form of governance. Governance is also a reason that Trust Anchor Repositories were developed in the first place, mainly concerning the signing process and key management of the root zone [6]. This discussion integrates with the question who should manage the DNS root zone and should therefore sign it. Two papers about "Signing the Root"' from Nominet [30] and ICANN [31] discuss governance. The discussion on who should sign the root is mainly a political discussion on the influence of the US government on the Internet. It is also enforced by messages that the US department of Homeland Security wants to have the ”Master key” of the root zone [32]. Several options on who should govern the root signing process are evaluated in the Nominet paper including [30]:

- The current manager of the root zone: the IANA
- The outsourced Root Zone Maintainer (RZM), Verisign Inc.
- A trusted third party

The Nominet paper suggest the following [30]:

> IANA should be responsible for creating and maintaining the Key Signing Keys (KSKs) used for the root zone. IANA and IANA alone should have the private portions of the keys and use those for the generation of Zone Signing Keys (ZSKs). IANA should send to the RZM the public portions of the KSKs and the public and private portions of the ZSKs for the RZM to use. The RZM should be responsible for publishing the public portions of both keys in the root zone and for using the ZSKs to create signatures following agreed algorithms that maintain the balance between security and manageability. We believe this maintains the appropriate balance of security and practicality of implementation, whilst reflecting the current separation of responsibilities between IANA and the RZM. [30]

Trust Anchor Repositories influence the governance discussion. Using TAR, everyone who is not satisfied with the proposed or eventually implemented governance structure is free to set up their own repository and associated policies.

3.3.2 Key management

Key management structures are different in a hierarchy and in island based concepts. The original idea on DNSSECbis is that the child has to contact its parent in order to (re-)publish its KSK using a DS record. This method is only valid in the hierarchy model and within the island. When looking at key management of the trust anchor(s) of the hierarchy and the island, the main difference is the number of trust anchors that should be published, updated or deleted. Also, the trust anchor of the hierarchy is directly published to the resolver while the trust anchors of the islands are collected in the repository and then published to the resolver.

RFC5011 [28] describes automated, authenticated, and authorized updating of DNSSEC ”trust anchors”. It provides protection against N-1 key compromises of N keys in the trust point key set. This protection is implemented using the existing trust anchors to authenticate new trust anchors for the same zone in the DNS hierarchy. In order to mitigate the compromise of an existing key or adding new malicious keys, a revoke bit is implemented for existing keys together with a hold down timer for new keys.

IKS-Jena has implemented RFC5011 and the ISC DLV is going to implement it [33]. Key management by the IANA repository is done manually because of its limited scope (TLD’s only). SecSpider crawls monitored zones everyday and does not implement the RFC5011

specification because their distributed poller approach should cover any security holes [26].
Where the global trust hierarchy concept is dependable on communication between a child and the parent, the TAR must state an admission policy. Initial publishing (enlistment in the repository) by SecSpider and IKS-Jena is conducted through an automated scan of the zone for the appropriate KSK. The results are presented to the user for verification. Although the distributed poller approach should also cover security holes at admission, manual verification at registration time provides an extra check with minimal operational impact.

3.3.3 Access

The global trust hierarchy suggests out of band communication to verify the initial trust anchor and after that, usage of RFC5011 [28] to update it. RFC5011 can be considered as an in-band mechanism, depending on the resolver implementation. When using a TAR, it must be accessed through extensions on the native protocol or by other means. DLV is an example of an extension on the native protocol and should therefore be implemented at the resolver side. However, communications are handled in-band through the DNSSEC protocol. Manual ways of access, through HTTPS, RSYNC or FTP present out-of-band communication. This could influence network configurations on the resolver side in order to be able to use the infrastructure. Of course, acquisition of the trust anchors should be performed in a secure way [29] preferably out-of-band.

3.3.4 Availability

The availability of the original hierarchy model is based on the deployment of multiple name servers. When deploying an alternate infrastructure to gain trust information from, this model should also guarantee this availability [30]. When heavily used, the TAR infrastructure should approach the current DNS root server deployment which might be to much for a(n) (non-profit) organization to manage/handle. The Nominet paper states that

> ”the robustness of DNS has been a major contributor to its success and DLV in its current form could undermine that.” [30]

3.3.5 Partitioning & complexity

Also actively discussed is the partitioning of the original global hierarchy by trust anchor repositories. Some fear that TAR removes the stimulation of zone administrators to completely sign the DNS tree [30]. Therefore, unsigned parts of the DNS hierarchy could continue to exist. These parts will remain susceptible to attacks posed on the original DNS infrastructure or even void what parent or child zones that are signed are trying to accomplish. Also unheard of in the hierarchy model are questions that will arise on who is responsible for the TAR’s and which one is the right to trust.

4 The Future of TAR

Outlook for the future of TAR is different among currently involved standardization bodies and operational organizations. Some, like IANA [7], suggest that trust anchor repositories in DNSSEC have no future and should be deleted when the root zone will be signed. The National Institute of Science and Technology (NIST) however, suggests that (some) TAR's do have a future and launched different TAR concepts: global (public DNS Tree), community of interest and enterprise TAR's [34]. This section will discuss the different concepts, their use and possible future developments.

4.1 Global TAR

The Global TAR type is thought of to support DNSSEC deployment on the Internet. The practical implemented concepts in chapter 3 can be classified as this type. Global TAR's can be used for multiple purposes or reasons:

- The parent zone remains unsigned
- There are DNSSEC pilot operations at the parent
- Availability of the parent to perform zone signing

Since key management and the signing process in DNSSEC are often done manually, the parent availability issue is a lot more feasible than in the traditional DNS protocol deployment.
The future of the global TAR is questionable since the ICANN announced that the DNS root zone will be signed by the end of 2009 [35]. In order to support the global trust hierarchy, IANA has clearly stated in their Interim Trusted Anchor Repository policy that:

> "the trust anchor repository is **temporary**. It will be decommissioned once the root zone is signed." [7]

However, the other TAR initiatives discussed such as the ISC DLV do not make any statement on the lifetime of their TAR. The future of global TAR's depends on if they will be actively maintained and used in the near future when the root is signed. Political discussion on the governance of the DNS root zone as discussed in section 3.3.1 can expand the lifetime of global TAR types.
The NIST suggest that if the Global TAR type will be disposed some day, this decision should be supported by one or both of the following pre-defined characteristics:

- A defined minimum threshold of trust anchor entries that should be in the TAR (not possible when using automatic key acquisition)
- A defined set of TLD's (e.g. large ones) that should be signed before the TAR is disposed.

4.2 Community of Interest TAR

A Community of Interest (COI) TAR should hold the trust anchors of a certain subset of zones (not necessarily islands of trust). When a partnership with other organizations is made, a TAR could be used to store the trust anchors of the internal DNS namespaces of the partners. This TAR is comparable to how organizations administer trusted CA's: a trusted CA of a partner can be added to a users validator software in order to establish trust in that domain. Usage scenario's of COI TAR's can be research networks, contractors, outsources and communities. COI TAR's could be created on demand, containing the trust anchors of all joined partners for a specific project.

4.3 Enterprise TAR

Some enterprises may have one or multiple DNS namespaces that should not be visible in the public DNS namespace. A method often used is split-view: the resolver has an internal and an external DNS view. In order to deploy DNSSEC, a TAR can be used to administer the trust anchors (secure entry point) of the separate namespaces.

4.4 Policies

The NIST suggest that a global TAR should acquire trust anchors either by registration of the zone administrator (like the IANA ITAR and ISC DLV) or by automatic means (like SecSpider and IKS-Jena). When manual registration is chosen, this can be handled in various ways including [34]:

- Open Registration: registration by zone administrator with some checks in order to prevent non-valid trust anchors.
- Strict Checking Registration: enforce a number of checks during registration mainly security driven to prevent malicious trust anchors.
- ″Same as Parent″' Registration: a varying policy throughout the repository; the policy level in place for parents must also be applied to descendants.

4.5 Future Requirements

RFC4986 [29] states a number of requirements a (future) TAR and resolver should adhere to:

- Scalability: (for global TAR's) must scale to root load comparable usage.
- Proprietary: royalty free usage of global deployed TAR's and ″TAR capable″ resolvers.
- Zone support: the TAR must be able to support all DNS zones including private ones.

- Detection of stale anchors: detection if trust anchors can no longer be updated.
- Key rollover: support for manual, planned and unplanned key rollover (also in a key compromise situation) while providing authenticity of the performed actions.
- Access time and high availability: the TAR must guarantee availability by distributing its database and must provide fast access.

5 Network Resource Provisioning Concepts

When building an optical lightpath there is a high chance of crossing different administration and security domains. While it is likely that in the near future a DNSSEC infrastructure will be mandatory in most domains, a TAR could be used to build inter-domain trust relations. This section provides an overview of NRP and the multi domain authorization infrastructure. Discussed techniques and models have been developed in the framework of the European Phosphorus project [36].

5.1 Network Resource Provisioning

The Network Resource Provisioning (NRP) model [37] is designed to enable Grid users to allocate network resources as a virtualized resource just like computation and storage. In order to realize this, a service plane was developed for networks on top of the traditional control plane and data plane layer (see figure 3). The service plane is concerned with path allocation, optimization, monitoring, and restoration across two or more domains. It allows the adaptation of control plane interfaces like ASTN, GMPLS and JIT and is able to abstract their network view into the service plane. Using the control plane, domains are able to signal neighboring domains to create end-to-end paths. Every domain in the chain will have two agents implemented on the service plane: the AAA and Grid Network Service agent.

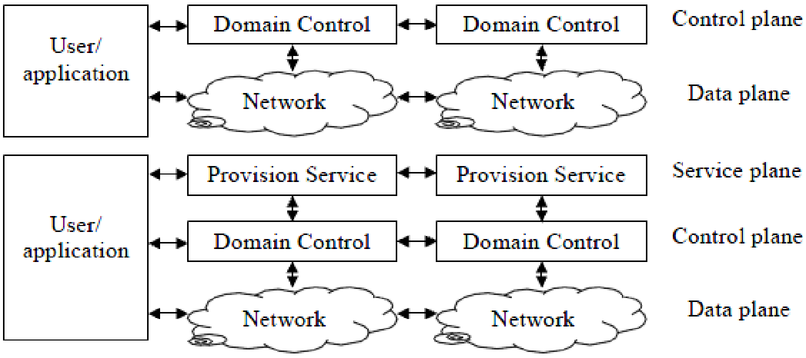

Figure 3: NRP Models
On top the traditional model, on the bottom the NRP model. [37]

The Grid Network Service agent is responsible for the topology view to construct optical end-to-end paths. To do so, it advertises networking capabilities to trusted neighbors. When a path is to be created, the Grid Network Service agent requires an authorization token. This authorization token is handled by the AAA agent of the service plane which obtains the authorization of multiple administrative domains. In NRP, a federation like model is used that can be joined by different parties. This allows the end-node to control the network resources of the federation.

The following quote [37] describes the inter-domain trust model used in traditional NRP:

> Since the path negotiation transits across multiple domains, the inter-domain trust model is a fundamental aspect to the overall provisioning strategy. There is a peer-to-peer relationship among the AAA servers representing an organization or domain. (..) Once the User is authenticated and authorized by the AAA agent of the source domain, this AAA agent represents the User during the setup process. Path establishment has been accomplished by means of transitive trust. (..) The requests are authorized because the requestor is known and trusted and the resource policy conditions are met. In our model, we used a token mechanism to ensure the authenticity of a request. [37]

5.2 Multi Domain Authorization Infrastructure for NRP

5.2.1 Network Resource Provisioning Model

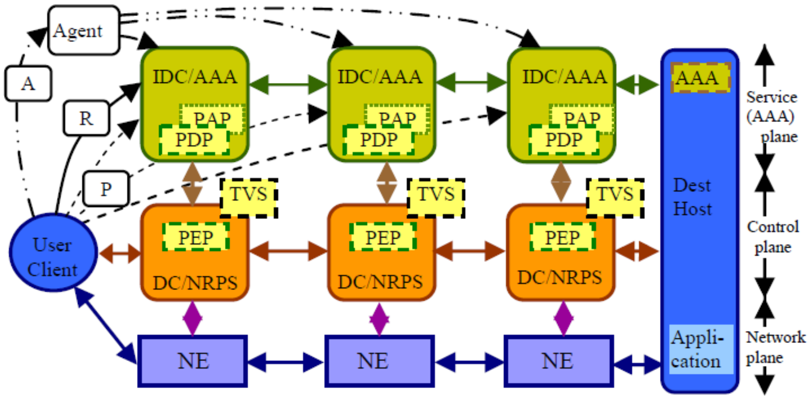

Figure 4: Network Resource Provisioning Authorization Model [38]

The Network Resource Provisioning (NRP) Authorization model [38] can be used as an authorization infrastructure in both on-demand NRP and the provisioning of Grid resources. The model is depicted in figure 4. The components of the NRP authorization model are [38]:

- The User client that wishes to contact the Destination Host delivering a service or application.
- Network plane Network Elements (NE): the network infrastructure.
- Control plane Network Resource Provisioning Systems (NRPS) acting as a Domain Controller (DC) for that domain.
- Service plane Inter-Domain Controller (IDC) and AAA service controlling access to the domain resources.

- Functional components of the authentication infrastructure: Policy Enforcement Point (PEP), Policy Decision Point (PDP), and Policy Authority Point (PAP).
- Token Validation Service (TVS): allows efficient authorization decision enforcement when accessing reserved resources and also supports token based service level signalling at the reservation stage.

The NRP authorization model describes three different sequences when provisioning a resource [38]:

- Relay (R) sequence (also called Chain or Provider sequence): the user contacts only the local network domain/provider providing the destination address, and each consecutive domain provides a path to the next domain reserving the necessary resources.
- Polling (P) sequence: the user client polls all resources or network domains, builds the path and makes the reservation.
- Agent (A) sequence: the user delegates network provisioning negotiation to an agent that will take care of all necessary negotiations to provide the required network path to the user.

5.2.2 Operational Stages

Establishing and using on-demand NRP goes through different stages [39]:

1. Reservation: resource lookup, composition (also composition of alternatives) and assigning a global reservation ID (GRI) to the reservation.
2. Deployment: reservation confirmation (binding of GRI to resources) and distributing components.
3. Access: providing authorization by using access tokens.
4. Decommissioning: session termination and accounting.

5.2.3 Provisioning Sessions Management with Tokens

In the proposed, and implemented in the Phosphorus project, Authorization infrastructure for NRP, tokens [40] are used for managing the overall provisioning session, its reservation and access stages. Tokens are divided in two classes: access tokens and pilot tokens [41]. Both tokens use the attribute SessionId (containing the GRI) and TokenId (containing a unique token number) for identification and authentication.
Access tokens are used by the end-host to perform authorization after the session is established. The tokens make it possible to bind traffic from an end-node application to end-to-end light paths. To achieve this, all IP packets from an end-node application are equipped with a token for authorization. A Token Based Switch (TBS) is used to process

these IP packets at wire speed. A PEP authorizes resource usage only if the provided token matches a cryptographic result. The key to this cryptographic result (the TokenKey) is provided by an authority during the path reservation stage.
Pilot tokens are used for session establishment during the reservation and deployment phase to collect and distribute provisioning information, in particular, security context and configuration information. They can be of the following 4 types [42], notably:

- Type 1: (PTT1) used for communicating the GRI during the reservation stage.
- Type 2: (PTT2) requester origin authenticating token. Contains a TokenValue that can be used as the authentication value for the token origin.
- Type 3: (PTT3) a type 2 extension that is used to store the security context information (including public key information) of multiple domains.
- Type 4: (PTT4) can be used at the deployment stage to set up a TVS infrastructure for the access stage.

During the resource reservation stage, the pilot token is sent through the domains participating in the path. To collect information of the domains participating, the token content changes:

1. The first token is created when authorization was positive, this is pilot token type 2 and is sent to the second domain.
2. When the second domain confirms the reservation, a pilot token type 3 is created that includes information from itself and any previous domains as a child element.
3. When the third domain confirms the reservation, pilot token type 3 now includes information of itself and all previous domains (domain 1, 2) as a child element.

Figure 5 displays the changing structure of the pilot token type during the reservation stage.
The Token Validation Service (TVS) is responsible for checking if the service or resource presenting the token has the permission to access or use a resource. During this check the TVS looks if the token has a reference to a previously reserved resource. The TVS supports a method called Validate & Relay which verifies the pilot tokens authenticity and generates a new token with local domain attributes to be sent to the next domain. The authenticity of the pilot token is verified by calculating and matching the TokenValue field. The TokenKey and the TokenValue are calculated using the HMAC-SHA1 algorithm [41]:

- TokenKey = HMAC(GRI, tb_secret)
- TokenValue = HMAC(GRI, TokenKey) - for access tokens
- TokenValue = HMAC(concat(DomainId, GRI, TokenId), TokenKey) - for pilot token type 2 and 3

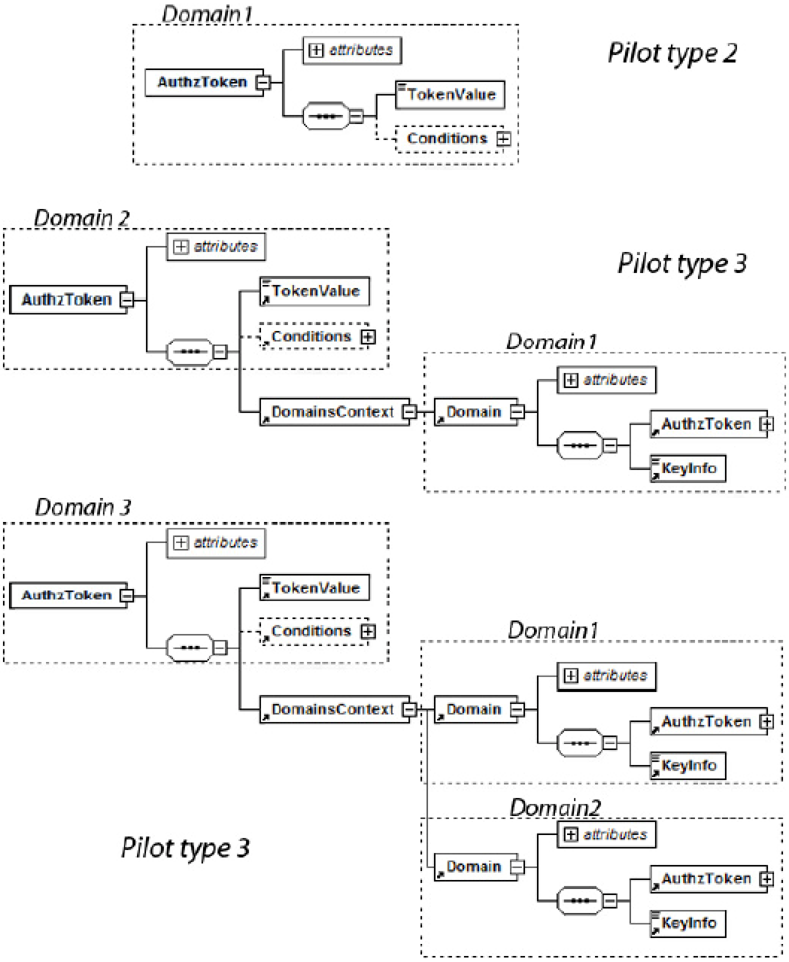

Figure 5: Pilot Token Type change during reservation [42]

The tb_secret (token builder shared secret) is currently implemented in the testbed software as a hard coded 3DES encrypted secret. In an operational situation, the shared secret needs to be replaced by a key management model. DNSSEC can be used to as a key management system since its presence is assumed to be mandatory in the near future.

During the reservation stage, the NRP model relies on a suggested pre-established service provisioning agreement or relations (that may or may not include formal security trust relations) at least between neighbor domains. Then in case of reservation (path) confirmation, a dynamic security/trust association is established that will be used at the later deployment and access stages.

6 Using TAR to support inter-domain trust associations in on-demand NRP

This section explores the possibility of incorporating a TAR in Network Resource Provisioning (NRP) during session establishment. A session based key is proposed which, using public key cryptography of DNSSEC, will be exchanged to all participating NRP domains. First, enhancements made to the original model are discussed after which two implementation scenarios are unfolded. Also, software that can be used for the implementation is introduced.

6.1 Proposed modifications and extensions

6.1.1 Moving to PKI based token validation

The Public Key Cryptography proposed in this report is based on the ZSK public/private keypair of DNSSEC. The ZSK is used because of its minimal operational impact and the (assumable) better online availability of its private key pair (e.g. through an HSM).
During the reservation stage, a signature is used as TokenValue. This signature will be generated using the domain's ZSK private key on the SHA1 hash of the concatenation of DomainId, GRI and TokenId to verify authenticity:

$$\text{TokenValue} = \text{SIG}(\text{SHA1}(\text{concat}(\text{DomainId}, \text{GRI}, \text{TokenId})))$$

This TokenValue is inserted in PTT 2 or 3 in order to provide token authenticity during reservation. While cryptographic calculations remain time intensive, a session based key (SBK) will be exchanged during the deployment stage. This key will be known by all traversed domains for the session lifetime and is used during the access stage. The SBK replaces the current token builder secret (tb_secret) and will be generated at the destination host. Using the SBK, the TokenKey will be generated in the following way:

$$\text{TokenKey} = \text{HMAC}(\text{GRI}, \text{SBK})$$

This leaves the generation of the TokenValue for access tokens intact as described in section 5.2.3.
The SBK is communicated backwards through the participating domains using PTT4 . The key will be encrypted using the DNSSEC ZSK public keys of the traversed domains to ensure confidentiality. Authenticity will be achieved using the TokenValue signature (same as PTT 2/3).

6.1.2 Community of Interest TAR

The proposed deployment model requires a Community of Interest (COI) TAR as described in section 4.2 of this report. The COI TAR provides a way to store the trust anchors of (internal) namespaces of participating domains. All NRP participating domains should publish the digest of their DNSSEC (KSK) public key in this TAR.
On submission, the KSK of the domain could be automatically detected like IKS-Jena and

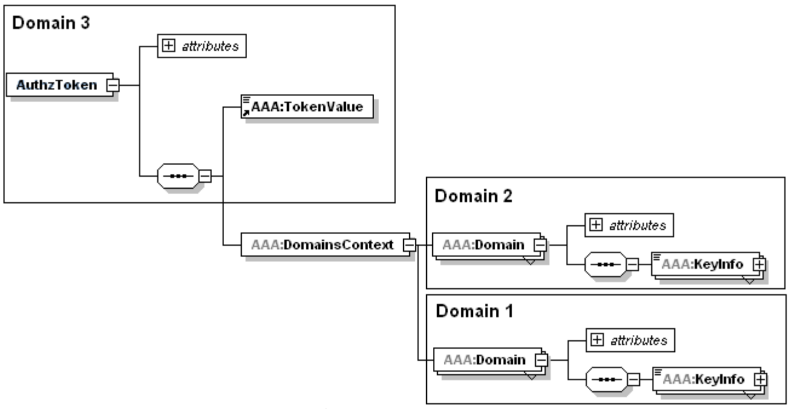

Figure 6: Pilot Token Type 4 Data Model
Structure when generated in destination domain.

SecSpider (see section 3.2.3). However, the actual submission must require a manual check in order to prevent security issues (e.g. spoofing). Key rollover should be automatically detected and processed by implementing RFC5011 [28].
The scope of the resolvers participating in the COI TAR should not extend beyond it, meaning that the ZSK public keys of domains are not to be verified through other trust anchors (e.g. from the root). Only domains (actively) participating in the COI TAR are to be queried since they are explicitly trusted by the community for path building.
Access to the TAR, as discussed in section 3.2, could be performed using DLV or downloaded (manually/scripted) through HTTPS etc. Since DLV provides the possibility to lookup current information without delay, this lookup method is preferred. In order to make use of DLV, every domain should trust (configure) the DLV trust anchor in their resolver. Using DLV, it should be possible to acquire the hash of the KSK public key of the domain. With this hash, it is possible to validate the KSK in the domain following the ZSK DNSKEY record (public key) and acquire it.
The DLV could be hosted somewhere in the common namespace of the project (e.g. dlv.ist-phosphorus.eu). By putting the domain name (e.g. uva, i2cat) of the desired domain in front of the DLV namespace, the DLV record can be acquired.
The TAR should have a general policy that may reflect some of the following statements, dependable on research/project goals:

- Limited admission.
- Signed parent is no issue (because of DLV traversal).
- Governance by a common accepted administration body.
- Guaranteed availability of the TAR by spreading and duplication of the DLV nameservers.

6.1.3 Pilot Token Type 4

Pilot Token Type 4 will be used during the deployment stage to carry the Session Based Key (SBK). The datamodel of PTT4 is displayed in figure 6. It is largely consistent with PTT3 in containing multiple ”Domain” child elements under the ”DomainContext” element. In the ”KeyInfo” field of every domain, the encrypted SBK is stored in hexagonal form. The TokenValue of PTT4 is re-calculated at every traversing domain providing origin authenticity.

6.2 Basic NRP TAR oriented scenarios: Scenario 1

Scenario 1 describes a possible implementation of TAR in NRP where all domains have DNSSEC resolving capabilities.

6.2.1 Reservation Stage

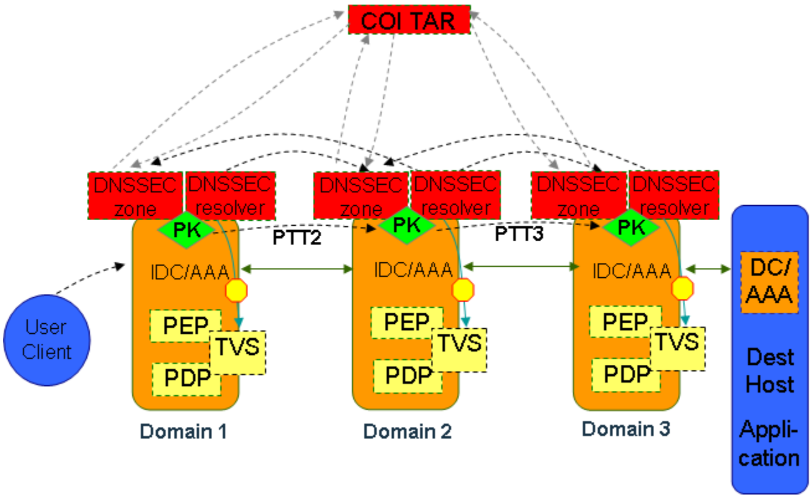

Figure 7: CRP Scenario 1 Reservation Stage
Arrows; Gray: trust establishment, Black: Pilot Tokens and DNSSEC queries.

During the reservation stage, PTT2 and 3 are used to collect the public keys of traversed domains. Figure 7 displays the acquisition of the public keys (PK). In the figure, the gray dotted lines represent every domain has published the hash of its KSK in the TAR letting the TAR trust every participating domain. The black dotted lines represent the pilot token sent and DNSSEC queries. The ZSK DNSKEY record is extracted from the local DNSSEC signed zone. This information is inserted in binary form in PTT3 in the ”KeyInfo” field under the applicable ”Domain” child element in the ”DomainContext” element (see figure 5 for datamodel).
PTT2/3’s TokenValue is signed as described in section 6.1.1. In order to verify the signature, every domain will acquire the ZSK public key of its neighbors through the DNSSEC

resolver. The neighboring namespace(s) can be recovered in three ways:

1. The DNS namespace is composed from the ”domainId” field found in every ”Domain” element. Currently the domainId is presented in the URI format (e.g. http://tesbed.ist-phosphorus.eu/viola/ which could be translated to viola.ist-phosphorus.eu).
2. An extra attribute is added to the PTT3 to carry the DNS namespace.
3. Neighboring domainId's are configured through the gaaapi-NRP configuration file.

While the namespace recovery of domain N-1 is possible through all options, N+1 can only be recovered through option 3 during reservation (there is no backward communication yet). When the Public Key's of domain N-1 and N+1 are recovered, they are used to verify the signature of the TokenValue used in PTT2, PTT3 and PTT4.

6.2.2 Deployment Stage

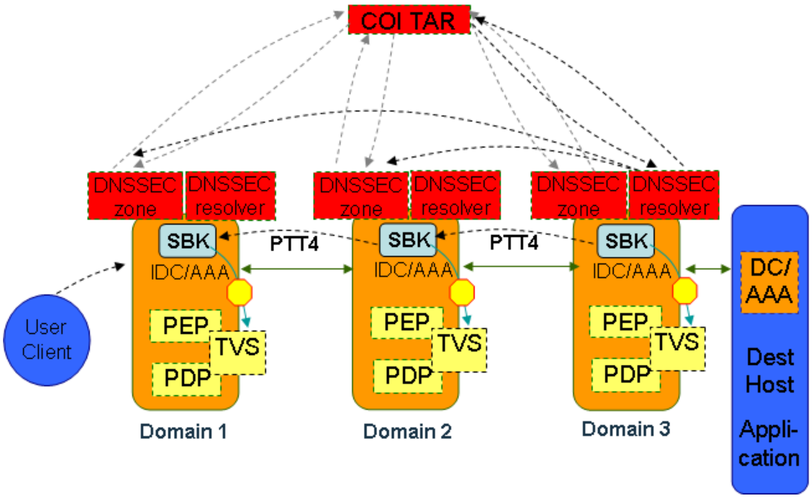

Figure 8: CRP Scenario 1 Deployment Stage
Arrows; Gray: trust establishment, Black: Pilot Tokens and DNSSEC queries.

When PPT3 reaches the destination domain, the public keys are verified using the TAR. Again, the DNS namespaces must be matched. Using the DNS namespace, the trust anchor of the domain can be acquired from the TAR (see figure 8). While this is the DLV record (hash) of the KSK, using DNSSEC, the ZSK DNSKEY record must be acquired from every domain. The value of this record must match the ”KeyInfo” field of the looked up domain in order to verify positive.

Using the public key of every domain, the SBK can be encrypted for each domain. While this covers confidentiality, authenticity is covered by deployment of the SBK using PTT4.

Authenticity could also be accomplished by signing the token using the private key of the destination host. In the deployment stage however, key lookup should be local (in TVS cache) to be non time consuming. For this purpose we cached the public key of neighboring domains during reservation.
In the scenario, PTT4 will provide token origin authentication in a backward scenario (just like token type 3 in a forward scenario) by inserting the signature on the hash of concatenation of DomainId, GRI and TokenId in the TokenValue. When token type 4 is received, every domain verifies the TokenValue signature using the cached public key of the (neighboring) origin domain. Using their own ZSK private key, every domain is able to decrypt the SBK. The SBK can now be used by the TVS to build the TokenKey as explained in section 5.2.3.

6.3 Basic NRP TAR oriented scenarios: Scenario 2

Scenario 2 describes an implementation scenario that does not require any DNSSEC key resolving capabilities in other domains than the destination domain or host.

6.3.1 Reservation Stage

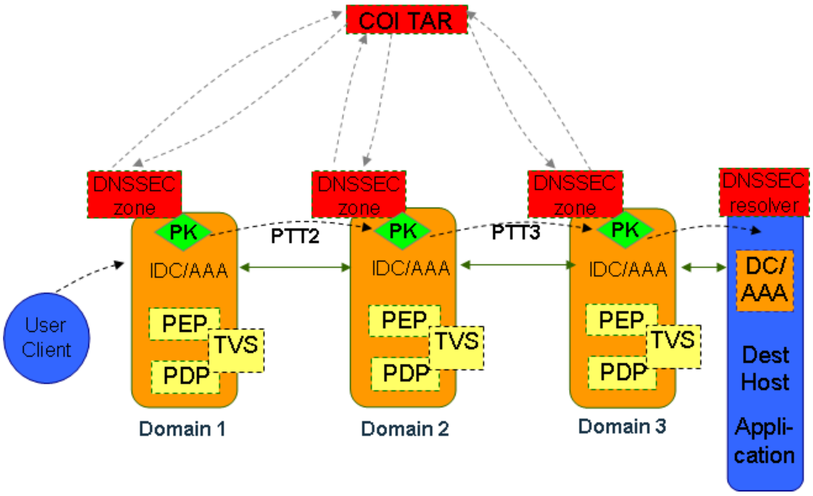

Figure 9: CRP Scenario 1 Reservation Stage
Arrows; Gray: trust establishment, Black: Pilot Tokens and DNSSEC queries.

Figure 9 depicts an alternative reservation stage. PTT2 and 3 are used during reservation to communicate every domains public key. Authentication is performed by inserting the signature of the TokenValue in PTT2/3 using the ZSK private key. Traversed domains can verify this signature by the PTT2/3 contained public key.

6.3.2 Deployment Stage

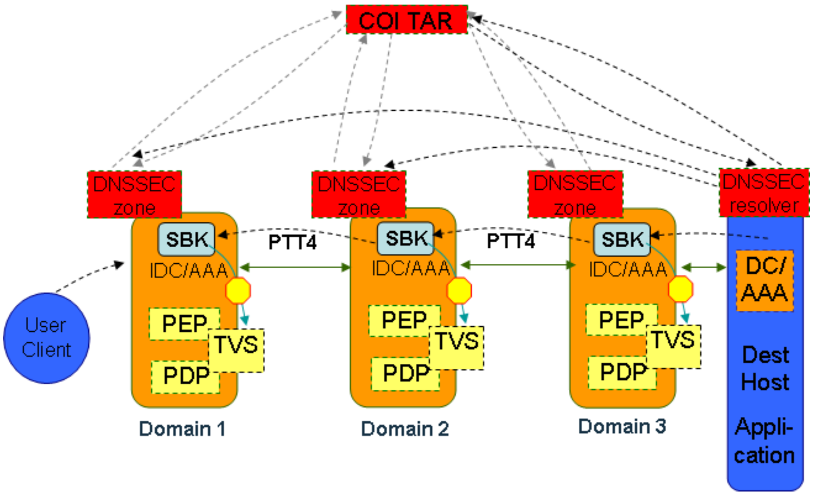

Figure 10: CRP Scenario 2 Deployment Stage
Arrows; Gray: trust establishment, Black: Pilot Tokens and DNSSEC queries.

Figure 10 depicts the deployment stage of scenario 2. The destination host is responsible for verifying the PTT3 collected public keys (as described in scenario 1). Deployment of the SBK in scenario 2 is done authenticated using PTT4's TokenValue. In this scenario, the public key of every traversed domain is saved under a new field in the PTT4 called "OriginKey". This new field should be placed under the Domain child element in the DomainsContext element.
What follows is that every traversed domain can check the authentication of PTT4 using the TokenValue by the origin public key. After this it can also decrypt the SBK contained in the KeyInfo field under the applicable "Domain" child element in the "DomainsContext" using its own private key.
Scenario 2 however, is susceptible to a man-in-the-middle attack replacing the Public Key in PTT2/3 or OriginKey in PTT4 with a malicious public key and signing the TokenValue with the malicious private key pair.

6.4 Software

Currently, the only implementation of the DLV TAR is in the BIND software of ISC [5]. When using manual trust anchors as "feed", multiple tools are available for a DNSSEC resolver implementation [43]. One of these tools is ldns of NLnetlabs [44] which comes with an extensive library for verifying DNSSEC material, TSIG, and DNS operations. With the example program ldns-keyfetcher it is possible to get all DNSKEY records of a given zone. This is implemented by first finding all authoritative nameserver of a zone by tracing it from the root down (this is configurable using a hints list). All authoritative nameservers are then queried (using TCP to minimize spoofing) for the DNSKEY RRset of the zone apex. If the results are all the same, all DNSKEY RRsets are printed. Below is an example output of ldns-keyfetcher that shows two ZSK's and one KSK public key of the RIPE website zone.

```
root@stefan-desktop:/home/stefan/Desktop# ldns-keyfetcher ripe.net
-r named.root
ripe.net. 3600 IN DNSKEY 256 3 5 AwEAAbzkOONqZz4CK4aQCXTluszwC
GYUrwINCVCn5SL5qQNLMN/DRuDmP+OGQSo241kegEzDSGW6HMQeqhZAtMTYmuuMGi
fMSnGAru6gh9W+wVu38qI6emM/NnhwFuCmQbOPXgJSs+YWSGxqRS8pRbGQgZ4ZQvo
alEKlf4UZrNMoHFbgZuYSz+XtlVJXNEBG3XAPBhxmYG0KkQ== ;{id = 52245
(zsk), size = 1200b}
ripe.net. 3600 IN DNSKEY 256 3 5 AwEAAcQFvKfZN3774/s6qY1jPBDX
PZ83n4AKk8VVQPcSjvlLvEbLnYbWjbiiHxoSCN3TwhcUky+1SheuFS9NKhNCIq85
276CXDXZv7yd595mLJ5jwXNBSUw5SJdyZAXY+xlWAmK2wjHNIVUk2MbJyowjhMoZ
2TKMK2HzNTl/uPMJxcPOFfDeT/Yikt7VUz7u920WoBiSribB2w== ;{id = 5058
6 (zsk), size = 1200b}
ripe.net. 3600 IN DNSKEY 257 3 5 AwEAAaFW0PI6SQlvxNyBGBUyHz6f
xquW34RLerkYVDb0Gh3Y2+N/rf7bMdlRWTiBbkkXtgMZnI+ug3FKxsZIiobXcHDm
P9pBJzQp01sfJgum+PqRQDucIG64MM/KitVas6SHnszuOlV/d3wtHhx4dXoDGf5Y
FZU33vmGBHwVy/En4KmX8jc/7RNm9YS02ZJ4d3qBMZB1uE52re7V0ZhUBWkm5HfF
eBw74HmsA2GUSBTZVpfxWZwqVLRseCbWm6OZr7snDSNJfo/TDC/zloHd7b/mbn9i
xNs/wk074iIFe6T7+GsGJMZEE0vpH10B+2Mko63mD4FioQIegmXCxwT5wGdGU20=
 ;{id = 59569 (ksk), size = 2048b}
```

7 Conclusion & future research

This project report presented the results on Trust Anchor Repositories (TAR). Three topics were investigated: a comparison of the original global DNSSEC trust model and the TAR based island model; the future of TAR; and how DNSSEC and TAR can be used in Network Resource Provisioning (NRP).

TAR emerged as an interim solution to speed up DNSSEC deployment but is currently considered as a technology that can solve known issues with interdomain trust management in an open Internet environment. With TAR it is possible to implement DNSSEC in some zones without signing the root. This creates so-called islands of trust where the trust relations to those islands is maintained through TAR. The comparison has shown that governance reasons on the management of the DNS system (especially the root) have contributed to the development of TAR. With TAR it is possible for parties to issue own policies, depending on political goals. The traditional DNSSEC implementation suggests out of band communication to verify the initial trust anchor, while the TAR based DNSSEC infrastructure can provide and requires periodic or on-demand lookup. There are in-band examples (DLV) as well as out-of-band examples (through HTTPS etc.). It is also important to mention that the availability of TAR directly impacts the availability of the DNS system.

Research on the future of TAR's shows that there is a different outlook by standardization organizations if TAR's should remain in effect after the DNS root zone has been signed. In particular, the report provides reference to and analyses the three models proposed by the NIST: global TAR, Community of Interest (COI) TAR, and Enterprise TAR. Global TAR's are used for supporting public DNSSEC deployment, COI TAR's for organizations to have secure relations (partnerships etc.) and Enterprise TAR's to collect trust anchors of internal signed namespaces. While policies in the hierarchal model were mutually agreed between a parent and a child zone, for TAR, there are multiple concepts on how they can be handled.

When a DNSSEC infrastructure will become a common practice in every domain, a COI TAR could be used for creating and managing trust associations in on-demand network resource provisioning that currently uses a shared secret model. A move to Public Key Cryptography was proposed, based on the DNSSEC ZSK public-private key pair of each domain. Every participating domain should publish its trust anchor in the TAR which should be trusted by all participating domains. During the reservation stage, the security context information of every passed domain is collected using pilot token type 3. During deployment stage, a Session Based Key (SBK) will be communicated backwards using pilot token type 4. The SBK is encrypted and the token is signed using DNSSEC domain keys.

There are two possible implementation scenario's. The first scenario is build on the DNSSEC resolving capabilities being present in every domain in order to verify the neighbor domain keys. The second scenario suggests that only the destination host or domain will have DNSSEC resolving capabilities. Both scenarios use pilot token type 4 for communicating the encrypted SBK to each participating domain. In the second scenario however, the public key of the originating domain is also send with pilot token type 2, 3 and 4 because not all domains have resolving capabilities but need to check the token signature. For these domains there is no opportunity to verify the public key of the originating domain out-of-band. This means that in this scenario, the key could be replaced by a malicious key during transmission.

7.1 Future research

Future research will mainly concentrate on developing the communication protocol and API to allow the NRP AAA (Authentication, Authorisation and Accounting) system to interact with TAR. A DNSSEC (TAR capable) resolver must be build and integrated with the current software. Section 6.4 provides some suggestions/recommendations regarding DNSSEC and TAR software that could be used. Future research should also consider the need to establish a COI TAR that involves all participating domains and/or organizations.

References

[1] Shane Kerr
IETF Response to the Kaminsky DNS Vulnerability, 2008
`http://www.isoc.org/tools/blogs/ietfjournal/?p=275`

[2] D. Eastlake, C. Kaufman
Domain Name System Security Extensions, 1997
`http://tools.ietf.org/html/rfc2065`

[3] R. Austein, M. Larson, D. Massey, S. Rose
DNS Security Introduction and Requirements, 2005
`http://tools.ietf.org/html/rfc4033`

[4] Xelerance
Worldwide DNSSec Deployment
`http://www.xelerance.com/dnssec/`

[5] Internet System Consortium
ISC Launches DLV registry to kick off worldwide DNSSEC deployment, 2006
`https://www.isc.org/node/62`

[6] Samuel Weiler
Deploying DNSSEC Without a Signed Root, 2004
`http://www.watson.org/~weiler/INI1999-19.pdf`

[7] IANA
Interim Trust Anchor Repository, 2009
`https://itar.iana.org/`

[8] D. Atkins, R. Austein
Threat Analysis of the Domain Name System (DNS, 2004
`http://tools.ietf.org/html/rfc3833`

[9] Amit Klein
BIND 9 DNS Cache Poisoning, 2007
`http://www.trusteer.com/bind9dns`

[10] Alla Bezroutchko
Predictable DNS transaction IDs in Microsoft DNS Server, 2007
`http://www.scanit.be/advisory-2007-11-14.html`

[11] surfnet.nl
The impact and importance of DNSSEC, 2009
`http://dnscurve.org/dnssec.html`

[12] C. Kaufman
Domain Name System Security Extensions, 1997
`http://tools.ietf.org/html/rfc2065`

[13] Kolkman & Gieben
DNSSEC Operational Practices, 2006
`http://www.nlnetlabs.nl/downloads/publications/dnssec/draft-ietf-dnsop-dnssec-operational-practices-08.txt`

[14] dnscurve.org
DNSCurve: Usable security for DNS, 2009
`http://dnscurve.org/dnssec.html`

[15] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose
DNS Security Introduction and Requirements, 2005
`http://www.ietf.org/rfc/rfc4033.txt`

[16] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk
Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2005
`http://tools.ietf.org/html/rfc5280`

[17] Dave Coombs
Chain of Trust, 2006
`http://www.carillon.ca/library/chainoftrust_1.2.pdf`

[18] Gieben
DNSSEC in NL, 2004
`http://www.nlnetlabs.nl/downloads/publications/dnssec/dnssecnl/secreg-report.pdf`

[19] M. Andrews, S. Weiler
The DNSSEC Lookaside Validation (DLV) DNS Resource Record, 2006
`http://tools.ietf.org/html/rfc4431`

[20] International Civil Aviation Organization
Public Key Directory Website
`http://www2.icao.int/en/MRTD/Pages/icaoPKD.aspx`

[21] International Civil Aviation Organization
ICAO PKD Participant Contact List, 2009
`http://www2.icao.int/en/MRTD/Downloads/PKD%20Documents/PKD%20Board%20-%20Participants%20List.pdf`

[22] S. Weiler
DNSSEC Lookaside Validation (DLV), 2007
`http://www.rfc-editor.org/rfc/rfc5074.txt`

[23] IANA
How to use the trust anchors, 2009
`https://itar.iana.org/instructions/`

[24] University of California, Los Angeles
Secspider Website
http://secspider.cs.ucla.edu/

[25] Eric Osterweil
SecSpider: Distributed DNSSEC Monitoring and Key Learning, 2007
http://irl.cs.ucla.edu/talks/OARC_SecSpider.pdf

[26] Eric Osterweil, Dan Massey, Lixia Zhang
SecSpider and TAR (Expanding it), 2008
http://irl.cs.ucla.edu/talks/SecSpider-final-2008-04-02.pdf

[27] IKS GmbH
IKS-Jena Website
https://www.iks-jena.de/leistungen/dnssec.php

[28] M. StJohns
Automated Updates of DNS Security (DNSSEC) Trust Anchors, 2007
http://tools.ietf.org/html/rfc5011

[29] H. Eland, R. Mundy, S. Crocker, S. Krishnaswamy
Requirements Related to DNS Security (DNSSEC) Trust Anchor Rollover, 2007
http://www.ietf.org/rfc/rfc4986.txt

[30] Nominet
Signing the Root, 2007
http://www.nominet.org.uk/digitalAssets/25692_Signing_the_Root.pdf

[31] ICANN
Signing the root zone: A way forward toward operational readiness, 2008
http://www.icann.org/en/announcements/dnssec-paper-15jul08-en.pdf

[32] Monika Ermert, Craig Morris
Department of Homeland and Security wants master key for DNS, 2007
http://www.heise.de/english/newsticker/news/87655

[33] Ólafur Guómundsson, Steve Crocker, Shinkuro, Inc.
Overview of DNSSEC Trust Anchors Repositories (TAR), 2009
http://www.ripe.net/ripe/meetings/ripe-58/content/presentations/tars.pdf

[34] SPARTA Inc, Shinkuro Inc, National Institute of Science and Technology
Statement of needed internet capability, Trust Anchor Repositories, 2008
http://www.dnssec-deployment.org/tar/tarpaper.pdf

[35] ICANN
ICANN to Work with United States Government and VeriSign on Interim Solution to Core Internet Security Issue, 2009
`http://www.icann.org/en/announcements/announcement-2-03jun09-en.htm`

[36] Phosphorus
Phosphorus Project Website
`http://www.ist-phosphorus.eu/`

[37] Leon Gommans, Bas van Oudenaarde, Freek Dijkstra, Cees de Laat, Tal Lavian, Inder Monga, Arie Taal, Franco Travostino, Alfred Wan
Applications Drive Secure Lightpath Creation across Heterogeneous Domains, 2006
ISBN: 0167-739X
`http://www.nextgrid.org/download/publications/Applications_drive_secure_lightpath.pdf`

[38] Yuri Demchenko, Alfred Wan, Mihai Cristea, Cees de Laat
Authorisation Infrastructure for On-Demand Network Resource Provisioning, 2008
`http://staff.science.uva.nl/~demch/papers/0095anav-grid2008-crp-authz-08.pdf`

[39] Phosphorus Participants
AAA scenarios and test-bed experiences, 2008
`http://www.ist-phosphorus.eu/files/deliverables/Phosphorus-deliverable-D4.2.pdf`

[40] Phosphorus Participants
ForCES Token Based Switch, 2008
`http://www.ist-phosphorus.eu/files/deliverables/Phosphorus-deliverable-D4.3.2.pdf`

[41] Phosphorus Participants
Updated GAAA Toolkit library for ONRP, 2009
`http://www.ist-phosphorus.eu/files/deliverables/Phosphorus-deliverable-D4.5.pdf`

[42] Yuri Demchenko, Cees de Laat, Thierry Denys, Christian Toinard, 2009
Authorisation Session Management in On-Demand Resource Provisioning in Collaborative Applications, 2009
`http://staff.science.uva.nl/~demch/papers/colsec2009-authz-context-ticktok-v10.pdf`

[43] DNSSEC.org
DNSSEC Software, Libraries
`http://www.dnssec.net/software`

[44] NLnetLabs
ldns
`http://www.nlnetlabs.nl/projects/ldns/`