DNSKEY Management

Julien Perrochet [JP] Tobias Schlatter [TS]

December 12, 2012

Abstract

In this paper we recapitulate the current deployment status of DNSSEC and go into the main issues when managing its keys. This paper outlines the paper "Interadministrative Challenges in Managing DNSKEYs" [7] and updates some of its contents, especially with respect to the signing of the root zone on July 15, 2010. We will see that DNSSEC penetration in the Internet's zones is still low, but that the necessary foundations have been lain for full deployment.

1 Introduction

DNSSEC is gradually being deployed throughout the Internet. However, managing the cryptographic keys required for the secure delegation provided by DNSSEC is not always entirely trivial.

In this paper we will first remind of the concepts and operations in DNSSEC which are relevant for key management. Then we will give the reader an overview of the current status of DNSSEC deployment in the Internet to put the discussion about key management into the right context.

Finally, we will shortly outline the DNSSEC situation in Switzerland (i.e. the `ch.` zone) and how its authorities handle DNSKEY management issues.

1.1 Chain of Trust[JP]

The goal of DNSSEC is to allow resolvers – any person or process querying a DNS server – to ensure the authenticity of the reply. Namely, in standard DNS, nothing guarantees the resolvers that his query's answer has not been modified during the transmission or really originated from a legitimate server.

DNSSEC permits zone operators to *sign* the entries they serve to a resolver, enabling the latter to verify that what he receives is authentic.

This is achieved by establishing a trust chain between something known and the domain that is being queried: e.g., I trust this person to tell me the truth and to be who it claims to be, because a friend I trust trusts this person.

In the DNS world, the *friend* we trust is the so-called root authority – the root domain – of whom we know the public key in advance, so we can verify anything it signed.

By signing the keys of the TLD's, the root will put its trust into their keys, and

any keys the TLD's have signed, resulting in a trust hierarchy illustrated in figure 1. This way, a resolver querying the LACAL's DNS server for `lacal.epfl.ch.` will be able to verify the received query answer against the known root-authority in the following way:

1. Verify the reply's authenticity using `lacal.epfl.ch.`'s public key;
2. Check the authenticity of `lacal.epfl.ch.`'s public key by obtaining its signature from `epfl.ch.`'s DNS server;
3. Verify the query – containing `lacal.epfl.ch.`'s public key signature – returned by `epfl.ch.` with its public key;
4. Check `epfl.ch.`'s public key against the corresponding signature served by `ch.`;
5. Again, verify `ch.`'s responses by getting its public key signature from the root domain.
6. Finally, the root domain's responses can be verified with its already known public key.

DNSSEC's trust hierarchy is similar to the SSL Certificate infrastructure, where an entity marks its trust in another by *signing* its certificate[1]. However, DNSSEC theoretically only has one single top-authority, while the SSL Certificate hierarchy disposes of many.

[1]Which is nothing more than a public key with additional metadata.

1.2 Operations[TS]

In the following, every necessary operation for key management in DNSSEC is described as a reminder to the reader. This is by no means intended to give an introduction to DNSSEC but rather to remind the reader about specific details particularly relevant to this paper.

Enabling DNSSEC When enabling DNSSEC for a zone, after it has been signed, the parent zone has to establish and sign a DS record with the newly created public-key.

Key Rollover RFC4641 [5] suggests that the DNSKEY records for a zone should be changed somewhere between once every week up to once every month. While the detailed rollover procedure is irrelevant here, it is important to know that the parent zone has to update its DS record regularly and hence has to be signed again, too.

Disabling DNSSEC An operator might choose to discontinue DNSSEC for a zone. The DS records in the parent zone (or other means of signing DNSKEYs as we will see later) need to be removed.

2 DNSSEC Deployment[TS]

This section will expose the current deployment status of DNSSEC and give the reader an idea about the current DNSSEC landscape before going into the specific DNSKEY management issues. Please refer to section 4 for more details on the situation in the `ch.` zone.

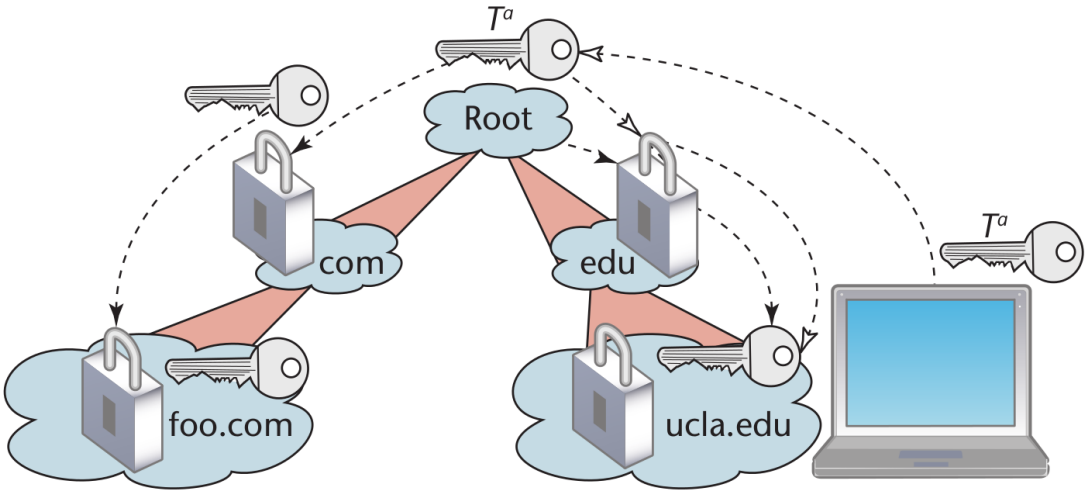

Figure 1: Example of a trust-hierarchy. Here, the resolver can establish a trust chain between the Root and the `ucla.edu.` zone.

2.1 Penetration in Production

In figure 2 you can see the development of deployed, production DNSSEC zones in time. The data comes from SecSpider [6] which crawls as many DNSSEC zones as possible, verifies whether they are properly signed and then applies a heuristic to determine whether it is actually a production zone. The goal is to rule out zones which are created for testing such as `bogussig.bogussig.test.jelte.nlnet.labs.nl`.

We can see that DNSSEC is gradually being deployed and the number of signed production zones grows rapidly (note the log-scale!). The rapid increase of crawled DNSSEC zones mid-2008, so interprets [7], is due to a cache poisoning attack discovered in summer 2008 [1]. During our research, we were unfortunately unable to find an estimate of the number of (production) DNS zones in the Internet in order estimate the percentage of DNSSEC penetration.

Contrary to the state at writing of [7], the DNS root-zone is signed since July 15, 2010 [3] and hence serves as the global trust anchor. However, as of this writing, data[2] gathered by [6] suggests that more than half of the deployed DNSSEC zones which are verifiable (i.e. are likely not to have a spoofed key, see SecSpider in the next section or [7] for details), do not have a fully linked certificate chain up to the root zone.

2.2 Trust Anchor Repositories

Not always can the certificate chain be ensured from the root-zone down to a DNSSEC zone. Before the signing of the root-zone, it was obviously impossible to do such a thing, after the signing – as formerly discussed data suggests – there are still orphaned zones whose parent isn't properly signed.

A Trust Anchor Repository (TAR) can be used in such a situation to supply resolvers with the trusted public key for a given zone during DNS resolution, there-

[2] `http://secspider.cs.ucla.edu/islands.html`

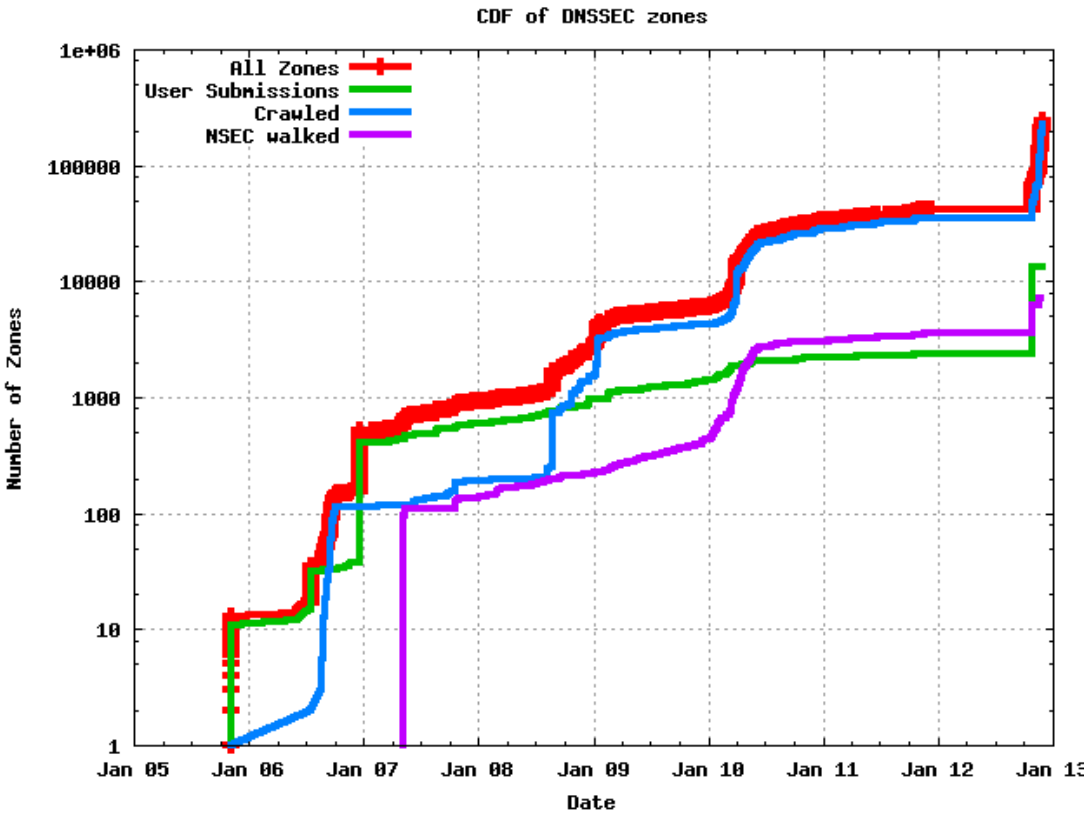

Figure 2: Number of deployed DNSSEC zones by SecSpider [6]

fore the name *inline repository*. One of these repositories is the Interim Trust Anchor Repository (ITAR) formerly operated by IANA and discontinued since the signing of the root-zone [2].

The Internet Software Consortium (ISC) still runs an inline TAR [4], using the DNSSEC Lookaside Validation standard [10]. When a resolver needs to check the key of a zone `<X>` with this TAR, it makes a DNS request of type DLV to `<X>.dlv.isc.org`. If it succeeds, the DLV record contains the signature for the (hopefully matching) DNSKEY. The ISC inline TAR requires manual addition of keys to the repository.

SecSpider [6, 7] is another inline TAR, but uses a different approach for key validation: rather then entering and signing public keys manually, it randomly queries DNSKEY entries from different point in the world simultaneously and only enters it into the TAR, if all the received keys match. This makes spoofing of a key very difficult due to the spatial distribution and the unpredictability of the queries.

Another type of TAR resides at the resolver: a statically configured TAR stores keys locally at the resolver and hence does not require to issue an additional query for keys when a new DNS request arrives. The statically configured TAR may automatically poll keys asynchronously, or may be just a bunch of manually configured trusted keys. As an example of a statically configured TAR, we have Vantages[3], a standalone daemon, [7] which can poll keys from different locations such as web pages, but also from other Vantage resolvers. Of course, the operator may also choose to add trusted keys manually.

[3] `http://www.vantage-points.org/`

3 Operations[JP]

This section covers the issues and quirks related to the management of DNSSEC.

3.1 3Rs

Regarding the management of the Internet namespace, three entities can generally be considered, namely:

- **R**egistries;
- **R**egistrants;
- **R**egistrars.

The registries are the entities serving the records for a zone. For example, SWITCH is the registry for the `ch.` zone. Registrants, on their side, are the entities willing to obtain a certain domain: the registrant of `epfl.ch.` is the school in itself (its IT-department, to be precise[4]), for example.

These two entities can eventually be linked by the third one, the registrar: this is where registrants will obtain the actual registration service, namely, the linking between a zone name and their authoritative DNS servers. Figure 3 illustrates the relations among the three R's.

In Switzerland, for example, SWITCH plays both the roles of the registry and a registrar: other registrars also exist but one can purchase a domain name directly from SWITCH. Registries will generally delegate part or whole of the namespace registration management to registrars[5]

Registrants are also free to play the registry and registrar roles for their own subdomains: groups, laboratories and associations within EPFL may apply for an

[4] As a *whois* query on the epfl.ch domain will show you.

[5] For example, one can also buy a .ch domain from `http://gandi.net`

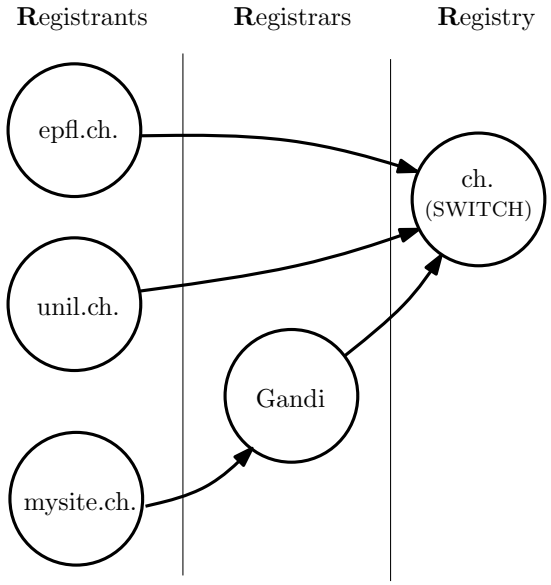

Figure 3: Example of relations between the three R's.

`epfl.ch.` sub-domain, like the LACAL and its `lacal.epfl.ch.` subdomain, for example.

The entities behind the three R's are the exact same whether DNSSEC is used or not, but DNSSEC modifies the relations among them: e.g., DNSKEY updates will now also require communication between the three R's, even if no practical information like DNS server addresses or billing data was actually updated.

3.2 Signing a zone

Whenever an entity wishes to sign its zone, it should ideally turn to its already signed parent zone, in order to create a continuous trust chain. Said parent should then serve a DS record containing the child zone's key signature.

Signing new zones will have different administrative implications depending on whether the parent zone is in the same administrative entity or not: e.g., `www.epfl.ch.` is managed by the same entity as `epfl.ch.`, so signing the `www` sub-domain will imply a different (and very likely easier) procedure than signing `epfl.ae`, EPFL's domain for its middle-east campus.

Enabling DNSSEC for a zone isn't a *do once and forget* operation, but requires additional coordination among the three R's. Let us see what managerial consequences arise, depending on whether we cross or stay within certain administrative boundaries.

3.2.1 Single administrative domain

Basically, once an organisation has its main domain signed, it is relatively easy for any sub-entity within said organisation to get a sub-domain signed, and to coordinate any operations relative to DNSSEC like key rollover: staying with our EPFL example, it can suffice to contact the IT-department and discuss about the procedure with them[6]. The bottom line is that it is relatively easy to handle the additional work and information updates required by DNSSEC as long as every concerned party lies within the same organisation.

3.2.2 Multiple administrative domain

DNSSEC's true challenges arise when we begin to consider several administrative entities (the three R's, mainly). As stated earlier, once the `epfl.ch.` zone is signed it is easy for the school to sign `www.epfl.ch.`. However, if the domain to be signed was `epfl.ch.` or `epfl.ae`, the school would have to deal with the entity or entities managing the TLDs (`ch.`, managed by

[6]EPFL does not actually use DNSSEC, but we use the school as an example for the sake of simplicity.

SWITCH, or `ae.`[7], respectively), which will have their own terms and conditions, that might very well vary among different registrars and TLDs.

While standard DNS can perfectly live with some of the temporary imprecisions caused by non-optimal interactions between the three R's, DNSSEC is far less tolerant to such jitter: typically, if the `epfl.ch.` zone updates its keys but SWITCH fails to reflect this change fast enough in its records, query responses from `epfl.ch.`'s DNS servers will appear as illegitimate whereas they perfectly are, and this will last as long as SWITCH's servers do not reflect the update.

As suggested by RFC4641 [5] key updates should happen at least every few months. Hence, it is crucial that a streamlined and reactive process is available for child-zones to notify their parent-zone when required. In the three R's setup, this means that the registry (or the registrars) must provide means for registrants to easily inform them of any changes regarding their DNSSEC keys.

Furthermore, as these communications concern the DNSSEC setup, security must be guaranteed: if a registry's update tools allow an attacker to update the key signatures, the whole point of DNSSEC is moot.

Entity Roles Overlap Another issue arises when we consider the fact that registrants may outsource part or whole of the hosting burden, like:

Key Signing Services that will handle the zone signing and key rollovers are available[8].

Hosting Countless hosting services are available, many of which are proposed by actual registrars. Some of them can also propose DNSSEC to their customers, in addition to the standard DNS service they also provide.

While having the registrars also handle DNSSEC will ease managing key rollovers and updates, it creates another problem: registrants, who generally are the registrars or registry's customers, might very well decide to switch from one service provider to another.

If standard DNS simply provides zone transfers as a mean enabling such migrations, DNSSEC must be treated differently. Whether a new private-public key pair is generated or whether the existing pair is transferred, new hurdles are introduced:

- The transferred private key is not totally *private* anymore, as now both the old and new service provider know it;
- Newly generated keys will require an update of the parent zone

In both cases, this *transfer* scenario will require a certain level of cooperation between two competing companies.

Another issue is that hosting and key signing services that are not registrars don't have to comply to any of ICANN's rules, increasing the difficulty to enforce common procedures, as none of these services is contractually bound to do so.

3.3 Computing power

DNSSEC does not only require a higher coordination among the three R's, it also

[7] epfl.ae has been registered at 'Instra Corporation Pty Ltd', a registrar for the ae.

[8] Like this registrar: `http://registrars.nominet.org.uk/registration-and-domain-management/dns/dnssec-signing-service`

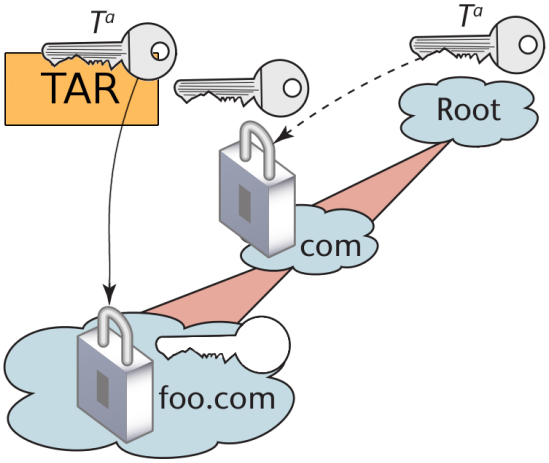

Figure 4: Illustration of a domain having disabled DNSSEC with a TAR still serving a record for it.

increases the strain on the computing infrastructure of a registrar when the size and quantity of the zones it manages grow: updates to a zone or key rollovers require cryptographic signing operations that are orders of magnitude more expensive than simply updating a DNS record. Even with improvements like incremental signing [8], the computational requirements are significant and must be considered when a registrar wishes to enable DNSSEC.

3.4 Disabling DNSSEC

The final and somewhat unintuitive managerial issue with DNSSEC is what needs to be done once a zone administrator wishes to turn it off.

In such a case, it is absolutely insufficient to only disable the service on the devices serving the aforementioned zone: in doing so, resolvers won't be able to distinguish between spoofed DNS replies that contain non-DNSSEC data and legitimate replies that simply stopped using DNSSEC.

For DNSSEC's shutdown to be clean and complete, the parent zone and any TAR (figure 4) must be cleaned from whatever DS record they contain for the zone which is being withdrawn from DNSSEC.

Again, this teardown must happen in a synchronous way – requiring additional efforts compared to standard DNS – as resolvers will perceive the domain to be *broken* as long as all involved parties are in an incoherent state.

4 Case-Study: `ch.` Zone^{TS}

The `ch.` zone is managed by SWITCH, the institution which also employs the Swiss educational network. In this section we briefly point out the current deployment status of DNSSEC and how SWITCH handles the discussed key management issues.

Certificate Chain According to a query on SecSpider [6], the `ch.` TLD zone is signed and fully verifiable from the root zone anchor on downwards. The certificate chain can hence fully be established and no separate TAR is required.

Penetration SWITCH states on their website that as of 30. September 2012, 1'734'170 child-zones of `ch.` have been registered[9]. Upon enquiry, SWITCH stated that there are around 370 secure delegations from the `ch.` zone. This is less than 0.02% penetration rate.

Key Management The validity period of a zone signing key (for `ch.`) is 37 days [9]. For child zones, DNSSEC and DNSKEY entries can be enabled and disabled through SWITCH's web-interface which is able to pull the keys automatically

[9] `https://www.nic.ch/reg/cm/wcm-page/statistics/index.html`

from the authoritative DNS servers. Partners (as Registrars are called in Switzerland) may also use EPP for key rollover.

Therefore, it is relatively simple to enable DNSSEC for a `<X>.ch.` zone. While using the web-interface is certainly cumbersome when having to deal with rollovers for a large number of domains, any major provider may become a SWITCH Partner and can then switch over to EPP which solves this issue.

Cost DNSSEC secure delegation is currently provided by SWITCH with no additional cost to normal DNS delegation.

5 Conclusion[JP,TS]

We have exposed the administrative challenges that arise with DNSSEC: while being far from trivial, these issues are not insurmountable, especially with regard to the gained benefits. Further, the modifications required from side the three Rs to their interaction process finally only consists in adaptations of already existing communication channels.

From this point of view, DNSSEC's requirements and goals seem to be reachable.

SWITCH's case study showed that the existing issues can be harnessed at least at a TLD level. While a wider adoption is required to see if the infrastructure can indeed scale, the foundations that have been lain seem ready to take on the challenge.

References

[1] CERT. Vulnerability note VU#800113 – multiple DNS implementations vulnerable to cache poisoning, july 2008. `http://www.kb.cert.org/vuls/id/800113`.

[2] IANA. Interim trust anchor repository, dec. 2012. `http://www.iana.org/domains/itar/`.

[3] ICANN and VeriSign. Information about DNSSEC for the root zone, dec. 2012. `http://www.root-dnssec.org/`.

[4] ISC. ISC's DLV registry, dec. 2012. `https://www.isc.org/solutions/dlv`.

[5] O. Kolkman and R. Gieben. DNSSEC operational practices. *IETF RFC 4641*, sept. 2006. `http://www.ietf.org/rfc/rfc4641.txt`.

[6] E. Osterweil. SecSpider – global DNSSEC deployment tracking, dec. 2012. `http://secspider.cs.ucla.edu/`.

[7] E. Osterweil and L. Zhang. Inter-administrative challenges in managing DNSKEYs. *Security Privacy, IEEE*, 7(5):44 –51, sept.-oct. 2009.

[8] N. L. R.Gieber. `.ca` Signing Metrics, May 2006. `http://www.nlnetlabs.nl/downloads/ca-reg.pdf`.

[9] SWITCH. DNSSEC key management practice statement, may 2011. `https://www.nic.ch/reg/cm/wcm-page/dnssec/keys.html`.

[10] S. Weiler. DNSSEC lookaside validation (DLV). *IETF RFC 5074*, nov. 2007. `http://www.ietf.org/rfc/rfc5074.txt`.