DNSSEC in NL

final report

R. Gieben

miekg@NLnetLabs.nl

version: 1.1.0

January 6, 2004

Abstract and conclusions

DNSSSEC; if it's ok, it is ok. If not, something *is the matter*

Abstract

With the advent of DNSSEC a registry will be confronted with a lot of new procedures and possibilities. The main change with DNSSEC is the introduction of cryptographic material to the DNS. To investigate the impact on registries, registrars and registrants a test bed was created. This test bed, called SECREG, is the first deployment of DNSSEC in a TLD.

Some prior knowledge of DNSSEC is assumed as this document will not provide a detailed explanation on the subject. But in a nutshell DNSSEC works like this. All answers in DNSSEC are digitally signed, by checking the signature a resolver is able to check if the info is identical (correct and complete) to the info on the authoritative server. More information on DNSSEC can be found at dnssec.net [1].

The purpose of this document is to serve as a guide to registries who want to deploy DNSSEC. It consists of two basic parts a policy part which explains how a DNSSEC registry should look like and what will be the changes in a DNSSEC world. The other part is technical and describes SECREG and its implementation.

The introduction section is geared to the explanation of DNSSEC and tries to explain what DNSSEC is not – this in order to downplay the mythical role DNSSEC is said to have.

Conclusions

This paragraph will briefly list the conclusions reached in this document.

DNSSEC is a technology to protect the DNS DNSSEC will allow for detection of spoof attacks in the DNS. It will not provide protection against anything else.

DNSSEC will change the registrant – registrar – registry -model With DNSSEC a new contact is introduced: the zone-c (zone contact) who holds the private key and it thus able to edit the zone file. With this key some administrative changes to a registry's database should be allowed. This leads to 2 changes, (1) the database must be opened up to allow modification based on a private key, (2) the registry may directly communicate with a registrant, bypassing the registrar.

More emphasis on security Although DNSSEC does not mandate more security by it self it is prudent for registries to ask them self some serious questions about their (other) security procedures. See section 3.7.

Importance of key material In DNSSEC a key is just “a key”. It is wise for registries to ”make a legal statement” about this fact; a registry must never be legally responsible when something happens with the DNSSEC key(s). This legal responsibility should be comparable to what currently happens when the TLD goes dark.

Deployment The protocol is finished, some software packages need to be updated (notably BIND9), key handling issues are solved; DNSSEC is ready. In short: there are currently no show stopper bugs known to the community.

Key compromises Be prepared to handle key compromises (section 3.13), which may require manual intervention.

Public key distribution A secure registry will have to distribute its public key. This key is the starting point of trust. This is a new responsibility for a registry to handle. (See section 3.12.)

Open issues

The following issues are not completely dealt with in this document. This is either because they are too TLD specific or that there currently is no correct answer.

How to actually implement the public key distribution The key must be made public and preferable via multiple channels (newspapers, radio, Internet). How to do this correctly and secure is not known.

DNSSEC in zones different than .nl The .nl zone is a clean, delegation only zone file. Other TLDs may have different zones. Details on how DNSSEC deployment is affected by the zone file are not given.

Acknowledgements

Thanks go to (in alphabetical order), Jaap Akkerhuis, Bart Boswinkel, Ben Geerlings, Marc Groeneweg, Rico Vermeulen (all SIDN), Olaf Kolkman (RIPE NCC), Ted Lindgreen (NLnet-Labs).

Without their help and proof reading this would still be a bug ridden document.

Special thanks go to Paul Wouter (Xtended Internet), Pim van Pelt and Alex Bik (BIT Internet) for helping to create the largest secure registry in the world. Without their domains and support SECREG would not have been as successfull.

Miek Gieben - October 2003

Contents

Chapter 1

Introduction to DNSSEC

DNSSEC does not “secure the Internet”

1.1. What DNSSEC is not

A lot of people think DNSSEC *will secure the Internet*. That it will make an end to script kiddies and other nuisances currently found on the Net. This will not be the case. DNSSEC is designed to do *one* thing and that is to enable detection of spoofing attacks in the DNS.

Other dreams about DNSSEC include use it as a PKI, a public key infrastructure. DNSSEC is not designed for this, and it therefore lacks core PKI operations, like key revocation. Building a PKI is something very different to managing the DNS. What *is* possible is to store keys in the DNS, which up until now was only possible with a hack. Notably technologies like IPSEC will benefit from this.

IPSEC secures the ip layer, it encrypts all trafic on this level. Currently it goes through a lot of effort to get keying material from the DNS (they use a hack to put keys in the DNS). Setting up this encryption will be greatly simplyfied with DNSSEC. A second benefit is that the with DNSSEC, the keys from the DNS are protected by a chain of trust.

1.1.1. DNSSEC and X.509

When comparing DNSSEC to a PKI infrastructure we see a lot of differences. For instance DNSSEC has no formalized Authority structure, there is no central CA (Certificate Authority). In DNSSEC there are just keys, signatures and a chain of trust – which delegates trust from the root down to the leaf nodes.

The elaborate structure of X.509 is not well suited to be molded into DNSSEC. X.509 wants to do *everything*, DNSSEC only wants to do one thing: secure the DNS. If you want a PKI you should deploy X.509.

1.2. What DNSSEC is

Hopefully with DNSSEC an increased security awareness will come to the Internet. This provides a great opportunity for registries and users alike to re-examine their current security practices and update them. The focus of DNSSEC is to add the ability to detect spoofing attacks in the DNS. DNSSEC will *not* prevent them, it only allows for detection. In the current DNS world spoofing attacks are difficult to detect.

DNSSEC itself does not magically add more security to a registry. But with DNSSEC registries will be forced to give security more attention. It is the hope, that this will increase the security of TLDs implementing DNSSEC. Section 3.7 and section 3.6 have more on the issues that should be raised. DNSSEC makes the policy of a registry more visible. Registries considering implementing DNSSEC *must* have a good policy in place.

1.2.1. How it works

The following is copied from Nominum's website FAQ on DNSSEC [3]

> **What is DNSSEC?**
> DNS Security (DNSSEC) is a method for supplying cryptographic verification information along with DNS messages. Public Key Cryptography is used in conjunction with digital signatures to provide a means for a requester of domain information to authenticate the source of the data. This ensures that it can be traced back to a trusted source, either directly, or via a chain of trust linking the source of the information to the top of the DNS hierarchy.

1.3. Why do it?

Why should a registry deploy DNSSEC? It costs money and time to convert the current administrative infrastructure to one that can handle DNSSEC.

The push for DNSSEC will come from end-users: registrars, users of online shops, banks and users of IPSEC and other security protocols. They demand more security than they currently have. Take for example an online bank. Currently a user (1) goes the website of the bank, (2) logs in and (3) performs some actions. Online banking has come a long way in securing step 2. They are using encryption over HTTP and user has password pads to secure log in. Step 1 however is not secured; a user can never be certain that he/she is visiting the correct site. No matter how good step 2 is implemented, if step 1 fails, step 2 does not guarantee you more security.

This is where DNSSEC comes in. With DNSSEC step 1 can be secured - or at least it is now possible to *detect* that a user is visiting the wrong site.

Another example is IPSEC. They need some kind of global key database from which they can pull keys which can then be used to encrypt traffic. Until now something like that did not exist. DNSSEC provides a solution for that. The same thing can be said for SSH (secure shell).

The first step in making the Internet more secure is DNSSEC.

1.4. DNSSEC lingo

A number of new words and phrases are frequently used in this paper albeit more in the Technical chapter. (This is our own interpretation unless stated otherwise.)

public key The public part of an asymmetric cryptographic key. This part can be published on the Internet. Data signed with this key can only be decrypted with the corresponding private key.

private key The private part of an asymmetric cryptographic key. This part must be kept secret. Data encrypted with this key can be verified with the corresponding public key. This is called signing.

zone key Public key that belongs to a zone and is published in the DNS.

key signing key (KSK) Master key that signs other zone keys.

delegation signer (DS) key Zone key that is uploaded to the parent zone, this is essentially the same as KSK.

zone signing key (ZSK) Zone key that signs the zone data.

key roll over The process of going from the current key to a new key.

zone-c Security contact for a zone. This person is considered to have control over the private key.

RR See resource record.

Resource record A resource record is one bit of DNS information. It consists of a name, a type and the data.

RRset Resource records with the same name and type are always transmitted together. DNS exchanges information by exchanging RRsets.

DS RR See delegation signer.

insecure zone A zone which is not carrying any keying information.[5]

secure zone A zone which is carrying keying information.

lame key sort of equivalent to a lame nameserver. The zone key exists in the DNS, but the corresponding private key is not in the hands of the current domain holder.

zone The whole of a domain name together with the information belonging to that name. To mind come: soa record, name server records and ip number information.

domain name The name of a zone.

Chapter 2

Technical

A key is just a key

2.1. Introduction to SECREG

During the .nl.nl DNSSEC experiment which at its peak had 17 zones delegated, it became clear that the DNSSEC protocol was not finished. The remaining problem(s) was not the core protocol, but the operational procedures to deal with (public) keys. Our nl.nl work resulted in a draft [6] which was later deprecated by the DS draft[4]. The DS draft also solves the operational issues. To test this NLnet Labs and SIDN started a new experiment called SECREG.

Since November 2002 SECREG is up and running. It is the first test bed of its kind that uses actual TLD domain names. People who have a .nl domain can participate in the experiment and have their zone key signed by SIDN directly. Currently (third quarter of 2003) some 140 zones are secured via SECREG. The goal is to define a set of questions (and possibly also answers) and what needs to be done before a registry is able to deploy DNSSEC.

The DNSSEC protocol specification, after 8+ years of discussion, is now for 99% finished. The .nl TLD wants to be ready to for DNSSEC when it is finished. Technically DNSSEC just adds signatures and (public) keys to the DNS. But with that change comes also the responsibility of handling the DNSSEC zone key of .nl. And the signing of the entire .nl zone, and not to forget the numerous child keys that will be introduced.

The purpose of SECREG is to test DNSSEC deployment and to gather hands on experience with the protocol and related (key) management issues. It can also serve as a playground for domainname holders and SIDN to test out different scenarios. Another goal is to describe how to handle keys and what guidelines can be offered to domainname holders. And finally we want to educate people on DNSSEC and its issues. Also we want to think through and test the implication DNSSEC has on registries and the current registration model.

The secured SECREG tree is kept completely separate from the normal .nl zone. Great care is taken to keep both zones synchronized. If at some point synchronization between the secure domain and its unsecured counterpart is lost the security attributes (DS, SIG's) of the domain are immediately dropped from SECREG.

2.2. The .nl zone key

The biggest change that comes with DNSSEC is the addition of keys and signatures in the DNS. For .nl this means there will be a .nl zone key that signs the other keys of the child zones that want to be secure. This key must be protected. Theft of this key or other key related incidents (accidental removal, key compromise) would seriously jeopardize SIDN's operation of the .nl zone. Not to mention the PR damage it would do.

It also must be very clear what SIDN's intentions are when it deploys DNSSEC. Any liability should be avoided. Theft of a child's key should never result in any charges against SIDN. Any liability issues outside SIDN's control must be avoided. Again such provisions are probably already in place, but it will not hurt to re-evaluate them in the light of DNSSEC.

2.3. SECREG design

SECREG is the first secure (DNSSEC ready) registry for a TLD level. It is not a full registry as it only implements the 4 DNSSEC operations (make secure, key roll over, block and change zone-c (still called zone-c in SECREG). The other "normal" operations are still handled by the .nl registry (SIDN). SECREG tries to keep the data from both registries synchronized. If that fails a zone is dropped from SECREG immediately. This section gives some details on some of the implementation issues.

2.3.1. Overview

The registry consists of several parts.

local database keeps track of who submitted a zone for DNSSEC. This is where each DS record and the key signing key for each zone is kept.

remote registry database this is the official domain database, in this case it is the database of SIDN.

website a web based user interface. All the initial requests must be done via this interface.

email robot some of the actions of the registry require user approval. Requests for approval are send out via email and received with this email robot.

signing machine this machine merges the DS list and the official zone and signs it using the private nl-key. To emphasize: the machine is on line and holds the private .nl key(s) (which is not very sensible, and this must be changed in the real deployment).

nameservers the signed .nl zone is loaded on these machines.

2.3.2. SECREG shadow setup

At first SECREG was implemented as a minimum state registry. Only the domain names and the zone keys were kept. This was done to make the design easier and to not duplicate the whois information in the local database. This setup was however dangerous. If SECREG does not keep state the following situation can occur. A zone is secured via SECREG and its KEY information is in the DNS. Then a domain transfer is conducted thereby transferring the zone to a new owner. The result of this is that there is a KEY in the DNS that does not belong to the new owner, but it is still *tied* to the domain. To prevent this out of sync situation SECREG now does store whois information. If the whois information at SIDN changes and this change is not reported to SECREG the domain is deactivated and the DS is removed from the DNS. Thereby making the zone verifiable unsecured.

There are currently three nameservers that carry the signed .nl zone. These are bakbeest.sidn.nl, alpha.nlnetlabs.nl and dnssec.nic-se.se. These nameservers are not listed in the "real" .nl zone. All the zone transfers between these servers are secured by using (TSIG) shared secrets. Although BIND9 still doesn't work correctly with DNSSEC and being recursive the general idea is to use one of these servers as an recursive caching forwarding. This will give a *normal* resolver a secure look on .nl and a non secured look on the rest of the Internet.

2.3.3. Current operation

Every morning the (local) database machine of SECREG generates a list containing the DS records of every active zone in the database. Active means that the zone does not has any operations pending and is not blocked. This DS list is then securely sent to the signing machine. The signing machine holds the .nl private key. When the DS list has arrived, the signing machine waits for the .nl zone file to be uploaded. This uploading is a manual operation which is performed somewhere between the hours 7:30 and 10:00.

As soon as this upload has taken place the .nl zone is merged with the DS list: only zones that are delegated get their DS included. If a DS is found for a non delegated zone this is reported. During this merging process the .nl public key is also added. This merging takes about an hour.

Next the nl zone together with the DSs and the .nl key is signed with the .nl private key. This signing process takes another $1\frac{1}{2}$ hours. Around 11 'o clock/noon this whole process is finished. The zone can now be loaded onto the nameservers.

Because the signed zone is now about 300 MB large (800.000 delegations at the moment of writing) in size this loading also takes up some time. There are no good messurements on how long it takes exactly, but is probably in the order of 15 minutes. After this (local) load the secondaries are notified and they perform an AXFR. Around one 'o clock all the secondaries are synchronized and the loading process is finished.

2.3.4. One parental DS supported

SECREG only supports one delegation signer record (DS) per secured zone. This was an implementation choice. This means a child cannot have more than one KSK that has a valid chain of trust. For more info on the DS record read section 3.14.

2.3.5. Tools, languages and machines

The registry is written in a combination of MySQL, Perl (with the Net::DNS::SEC package) , shell scripts and (s)HTML. It further uses BIND9 for signing and loading the signed .nl zone and sendmail for all the email delivery. It took one person[1] about 2 months to implement the basic registry. It then took another month to perfect the system. All code is available at request and is licensed under the GPL.

2.4. DNSSEC Operations

With DNSSEC 4 new operations are defined, these are

securing Upload the zone key to SECREG and start being secure,

key rollover Uploading a new zone key to SECREG and phasing out the older one,

block Revoking the security of a domain (going back to DNS) in case of a key compromise or administrative reasons,

change zone-c Update the current zone contact.

All operations will be explained in the next sections.

2.4.1. Securing

This operation is currently called “Make Secure” in SECREG. It is implemented as follows. A person, this can be anybody, contacts SECREG with the request to secure a domain. SECREG asks for an email address, which will become the zone-c address, and a domain name. Suppose the domain that is to be secured is miek.nl and the zone-c contact address is info@miek.nl. Next SECREG will fetch the whois data from SIDN and will use the IP addresses of the nameserver to fetch the key data of “miek.nl”. The purpose of using the IP address is the circumvent the DNS (which cannot be trusted). After this step the person is confronted with the zone keys that were found in the zone. The person then chooses which key should be used by SECREG as the KSK and clicks “proceed” on the website. SECREG will now store all the information about “miek.nl” in its database. This information includes all current whois data and of course the KSK from miek.nl.

The next step is to verify the zone-c. SECREG will mail all the contacts for miek.nl, excluding the newly appointed zone-c. In particular the tech-c’s and admin-c will all receive an ACK email, the registrar will receive a NACK email. These emails serve the following purpose:

ACK mail This email asks for a confirmation of the current “make secure” operation. This email should be returned within 2 days. If a positive answer is received SECREG waits up to two days for any NACKs to show up. If a negative answer comes back the operation is aborted. If SECREG does not receive back *any* ACK mail, the operation is aborted.

[1] me :-)

NACK mail This email asks if the current operation should be blocked. This gives the registrar a final saying about a domain that is registered via him. This email should be returned within 2 days. If a positive answer is returned (the NACK is confirmed) the operation will be aborted, if a negative answer (the NACK is rejected) is received the operation is allowed to continue. If SECREG does not receive back a NACK mail the operation is continued.

Thus if after two days an ACKs is returned and no NACKs have been seen, SECREG assumes the “make secure” operation was OK. It will make the zone miek.nl active in the database. All active zones get included in the signed .nl zone.

2.4.2. Key rollover

Issues

When implementing a key rollover difficulties arise when the old key is not in the zone anymore but still is cached on the Internet. The consequence of this will be that fresh data from the zone (which is signed with the new key), will be verified with the old (still cached) key. This yields bad data and the zone is considered bad by this resolver. There are multiple solutions for this, ranging from keeping the old key longer in the zone to publishing the new key some TTL's earlier in the zone to allow pre-spreading of the new keyset. Currently SECREG does not care what client solution is chosen. It only replaces the DS record at the .nl side.

A rollover is a difficult operation. It should not only be performed frequently from a crypto viewpoint, but also to keep administrators up to speed on the procedure. Doing an emergency key rollover when not recently having done a scheduled rollover is bound to fail.

Current implementation

A key rollover is always initiated from the child. The zone key RRset must be signed by both the old key (which is also stored by SECREG) and the new zone key. Next a request is sent to SECREG (via the web interface) that a child zone wants to perform a key rollover. First the user of the website must authenticate himself against SECREG. See section 3.5 (Access to SECREG) information.

After a successful authentication SECREG retrieves the signed key RRset, verifies that signature with the current zone key. If that signature is okay the user is presented with a list of key that are all contained in the key RRset. SECREG can not automaticly determine which key will be the new zone signing key (this is changed with KSF flag draft[8]), so it lets the user pick one. This key is then used as new zone signing key and a DS record of it is created. The next day this new DS record is inserted into the signed .nl zone. After this update, the parent (.nl in this case) is finished with the rollover.

The old key could be used by SECREG as a fall back mechanism in the case of a key compromise. Currently this is not done and such emergency scenario's are not researched within SECREG.

Child side

A child zone is completely free to implement a key rollover. SECREG has no other constraint then a double (or more) signed key RRset at the time of the rollover. As said in Issues (see above), there are some ways to solve this problem at the child. These solutions can be divided into two groups:

1. publishing the new key ahead of the rollover,

2. keeping the old key around after the rollover.

Ad 1 A child can pre-publish a soon to be used key in the zone. By the time the key rollover is published in the parent zone all the resolvers already have cached the new key. The new key must be published at least 1 TTL before the key rollover will be initiated.

Ad 2 Another possibility is that a child zone will keep the old key around until some time after the key rollover. The cut off date (the date on which the old key can be discarded) is the date on which the signature of key old expires.

2.4.3. Key compromise

The (un)thinkable has happened. A zone key has been compromised and an attacker can forge signatures of a zone. There must be a new zone key uploaded to the parent as quickly as possible and the old zone key must be discarded. In SECREG this operation is called "block". A user of the SECREG website must authenticate himself against SECREG. After this step a zone is blocked. After the next .nl zone update (which is performed at around 8 o' clock each morning), the domain is dropped from the secure tree.

When DNSSEC is deployed this situation will probably arise often. See section 3.13 on some estimates.

2.4.4. Change security contact

The security contact is the spokes person for the security of a zone to SIDN. From SIDN's viewpoint it doesn't really matter who is behind this email address – it assumes it is the person controlling the private keys of a zone.

This contact can be changed with this operation. The keys of zone stay the same during this operation. Only the email address pointing to the zone-c is updated. Again the user of the website must authenticate himself to SECREG. After this authentication has taken place the user is free to change the email address.

2.5. Problems encountered

The following problems were experienced during the experiment.

2.5.1. Out of sync secondary

During the experiment, one of the .nl secondaries (dnssec.nic-se.se) was not updated for more than two weeks due to a memory overflow. This causes severe problems. The signature lifetime for .nl is 1 week, thus the signatures on nic-se.se were *all* bad. This means that whoever was using this server gets bad (authoritative) data for .nl, which would in turn mean that .nl would drop of the earth for all users of this secondary.

With DNS a secondary that was not up to date was bad, but it was still sort of usable. With DNSSEC a secondary that is longer out of date than the signature lifetime is disastrous - it causes the local removal of a TLD (in this case).

2.5.2. Problems with how the experiment is setup

New nameservers added and not deleted by the scripts, causing the listing of a nameserver which carries the non secured .nl zone.

2.5.3. Timing in DNSSEC

A short quote that sums it right up:

> DNS has relative timers, DNSSEC has absolute timers.

With DNSSEC timing becomes an issue. Currently data in the DNS does not care about time, when there is a problems with the DNS operation the old data can still be usable. With DNSSEC timing is a critical factor, when problems are not fixed within the signature lifetime, zones get bad. And bad zones drop of the Internet. The BCP should delve deeper in this subject.

2.6. NL zone setup

The .nl zone is going to change with DNSSEC, there will be new records to be published, most important are the .nl zone key(s). It is wise to prepublish keys in the keyset. If an event of failure the new key is already distributed to the resolver and can be used immediately. Therefor the following keys should be in the .nl apex:

Two KSKs One KSK will be actually used. The other is there for emergencies. The private part of the unused KSK can be put in a safe, because it is not used to sign the keyset. The bit length of these keys can be quite large. A 2048 length seems to be in place. KSK keys should be changed every 2 years. Larger key sizes should be used when the years progress.

One ZSK A rollover of this key can be accomplished relatively easy. There is no need to prepublish it. It can also be shorter and changed more often. A bit length of 1024 and a lifetime of 6 months seems to be in order. Larger key sizes should be used when the years progress.

2.6.1. Zone signing

The current .nl zone is generated from a database and then uploaded to the nameserver systems in Amsterdam. With DNSSEC some extra steps are required. First the zone must be sorted – this speeds up the signing process considerable. After the zone is sorted it must be signed. This signing process requires access the private key parts of the ZSK and the current used KSK. Needless to say, only a few people should have access to those private keys.

After the zone has been signed – the SIG and NXT records have been added – it can be uploaded to the nameserver. Currently the .nl zone is signed daily.

2.6.2. Zone size

The signed .nl zone is significantly larger than the unsigned one. We're talking a factor 8 to 9 here. Of course this also depends on the keysize of the KZK. In 2003 the unsigned zone was around 700.000 delegation (around 40 MB), the signed one reached 300+ MB. This increase in zone sizes should not pose a big problem, even .com can handle the increase in size (that zones grows to 10 Gigabyte).

If the signed zone is larger than 3 Gigabytes a 64 bit machine should be used to serve it. It would be wise start experimenting with those 64 bits machine well before the actual DNSSEC deployment in a TLD.

2.6.3. Signature lifetime

As said, a signature in DNSSEC has a limited life time. After some date the signature is invalid. There are some constraints on this signature expiration date. It should not be too nearby and it should not be too far in the future. It should be *just right*.

A short lifetime protects the child's DS. In case of a child's key compromise, the signature on the DS determines how long the child is vulnerable for spoof attacks with the old key. The secure .nl zone is now signed daily. A signature lifetime shorter than a day, implies signing twice or more a day.

A longer lifetime is needed when there are operational problems. If the signature lifetime is one hour and something goes wrong SIDN has one hour to fix it and republish the new zone. If that cannot be done then the chain of trust for every secured domain in .nl gets severed

A suggested tradeoff might be a signature lifetime of a week. The minimum lifetime should not get under two days but that is also related to how often the .nl zone is signed. A lifetime for more than a month is also seen as being to long. When DNSSEC is first deployed a signature lifetime of one month is advisable, to help ease the transaction and give enough time to fix problems. Off course this should then be throttled back to 1 week (or shorter) as soon as possible.

2.7. Further reading

Currently the author and Olaf Kolkman (RIPE) are working on a draft[9] which specificies operational guidelines for DNSSEC administrators. This work is expected to be finished within 6 months.

Chapter 3

Policy

DNSSEC does not *mandate security – it just makes it more visible*

3.1. Specifics of the .nl zone

This document is written with the specifics of the .nl zone in mind. The .nl zone is a delegation only zone. There are no records from a child zones present, except for the NS records of course.

Being a delegation only zone makes life somewhat easier for DNSSEC - as no child information is signed with the key of the parent[1] Questions like "How to handle child zone information in the TLD zone?" are therefor not tackled.

3.2. Introduction

Current DNS administration is structured in a three-tier architecture: at the top we have the registry, which gives out domain names. Below that are the registrars who register names on behalf of their clients, the registrants. The registrant is the actual owner of a domain name.

It is normally the case that a registry only deals with the registrars. The registrars in turn deal with the registrants. It is this delegation of work that keeps the registry from being overwhelmed. This delegation follows the structure of the DNS.

It is also important to see where the contacts – tech-c, admin-c – are located. The following separation can be seen in most of the delegated domains:

- The tech-c, the technical contact, is usually situated at the provider. This contact handles technical matters concerning a domain name. Think of: restarting nameservers, interacting with the registry for technical reasons.

[1] NS records and glue at the parent are not signed with DNSSEC.

- The admin-c, the administrative contact, is per default (in the Netherlands) the holder of the domain. Email messages send to the admin-c email address are assumed to reach the holder of the domain name. Thus the admin-c is located at the registrant level.

With DNSSEC the notion of public key cryptography is introduced in the DNS. The private key – which is used to sign a zone – must be kept secret. This private key belongs to the holder. Although a provider can "take care" of the key, it is something that belongs to the holder. This public key must be communicated to the registry, and in case of loss that fact must also be communicated to the registry. Both facts indicate that the zone-c – the security contact, must be able to communicate with the registry. Thus the zone-c will cut through the registrant - registrar - registry layer and bypass the registrar.

3.3. Zone Contact

For DNSSEC a new contact is needed. This contact, the zone contact or zone-c is to be considered the owner of the domain's private key. The contents of the zone file can only be changed by virtue of this private key.

The zone-c could be located at the registrant level. This in itself is not a reason for a registry to communicate directly with the registrant. A procedure could be setup that requires the registrar to do updates at the registry on behalf of the registrant. The registrar and the registrant would then be forced to setup procedures to allow the registrar to use the holder's private key. But this is a kludge - private keys are meant to be kept secure. It is more easy to allow the zone-c to communicate directly with registry. This is a change to the model, but is is a small change and it is only needed for a small subset of operations. Further more, most DNSSEC operations can be automated.

Thus DNSSEC leads to the following modifications in the model:

direct access to registries' administrative database The zone-c needs access to the database to perform (key related) operations for the domains under its control,

access based on DNS key info The access to the database is authenticated by using the key information stored in the DNS. This means that administration/authentication issues are being moved to the protocol. The current practice of keeping the DNS administration completely separate from the protocol will diminish.

These modifications can however be implemented gradually. Thus to summarize. The admin-c is the official contact for a domain name, the tech-c controls the nameservers and the zone-c controls the private key information of a zone. This separation of roles is not distinct. But it makes talking about who does what for a specific domain easier.

It is expected that in 99% of the cases the zone-c is located at the registrar's level. However in those other 1% the holder will have a good reason to do its own signing. Any new registration system will have to be able to talk to the zone-c thereby potentially bypassing the registrar.

3.4. Difference between the zone-c and tech-c

The tech-c and zone-c seem to cover the same ground, but there are important differences between to the two.

- The zone-c deals directly with the zone file. The tech-c deals with the nameserver(s) which use the zone file.
- The tech-c is located at the ISP (registrar) and the zone-c will sometimes be the holder (registrant).

These contacts can be the same person. That person can be even be working at a registrar. But that doesn't change the underlying point: the zone-c has access to the zone file and needs to be able to communicate with the registry.

A rule of thumb is: whenever information in a child zone changes the zone-c is needed to resign the zone. This could be as trivial as updating the SOA serial number. If this change affects data which is also stored at the parent (currently only nameservers and key signing keys) then parent/child interaction is needed.

3.5. Access to a DNSSEC registry

In SECREG all access to the registry is authenticated by the private key of the domain holder (= the key the zone-c controls). There are currently efforts underway to move some registry function (like updating nameserver records) to the DNS protocol. See section 3.15 for more on this subject. The authentication procedure in SECREG goes as follows:

1. a user wants to make a change in SECREG,
2. the user types in the domainname of the zone he/she wants to access,
3. the user is then presented with a random DNSSEC text record, using the domain name from 2,
4. the user must now sign this text record with the private key *belonging* to the zone from 2,
5. this signature is then cut and pasted into the website,
6. SECREG will now check the signature by using the public key corresponding to the domain from 2,
7. access is granted if the signature is ok. Otherwise no changes can be made.

"The user" can *only* be the zone-c if the authentication succeeds (or the key must be compromised). Currently registries will only allow registrars to update information in their databases. As described above it will be necessary to also allow access to the database, based on the private key of a domain. New domain registration systems for registry will need to take this into account.

If people mess up their DNS and/or loose their keys they can no longer update their information. As a backup, the provider (or registrar) must also be able to perform these updates in case of an emergency.

When trust is re-established access based on the private key can be granted again. People losing their private key are in no way different of providers losing their passwords to access the registry - procedures handling those events should easily be adjusted to handle the DNSSEC cases. To put the whole matter in a nutshell, a fall back mechanism must be in place, *if only for* key compromises.

3.6. Initial trust

Somehow a registry needs to know that a DNSSEC key is from the person holding the domain name. As one hundred percent certainty about someone's identity is an unachievable fact, this is a difficult step. One can however try to be as certain as possible and to verify as much as is reasonable possible. This 'initial trust' step is one of the most important steps in the whole of DNSSEC. Weak spots in this procedure will be found and exploited.

SECREG implements the following procedure:

First contact Someone approaches SECREG and claims to be the new zone-c for domain X. Domain X is signed and loaded on the nameservers.

Check 1. SECREG checks the nameservers to try to extract the key material of the zone.[2] [3]

Check 2. If there are signatures and they are valid, that 'someone' is *assumed* to be the zone-c. Acknowledgement emails are now sent to the registrar, the tech-c and admin-c for this domain name. Note: the (would be) zone-c does not receive those emails.

Verify If one the emails sent to the tech-c or admin-c returns with a 'yes' the domain will be secured and the 'someone' becomes the official zone-c. The registrar receives a nack email which – if returned – aborts the operation.

This 'no' or 'nack' mechanism is to prevent an unauthorized person to initiate the operation against the intention of the domain holder or registrar.

After a waiting period of two days and only if one of the ack mails and none of the nack mail is returned, SECREG considers the request valid. Otherwise the operation is aborted.

3.7. Security in a DNSSEC world

Having security and policies in place is always a good thing. Sending passwords in plain text over the wire is not a good idea. Having good policies and tight security should happen regardless the deployment of DNSSEC. DNSSEC emphasizes these topics, no registry can claim to provide adequate DNSSEC security when their registration procedure lacks security.

[2]This action is performed without consulting the DNS. All (IP) information is stored in the registry's (whois) database.

[3]Currently only one nameserver is checked. A better implementation would check them all and drop the request if there are inconsistencies.

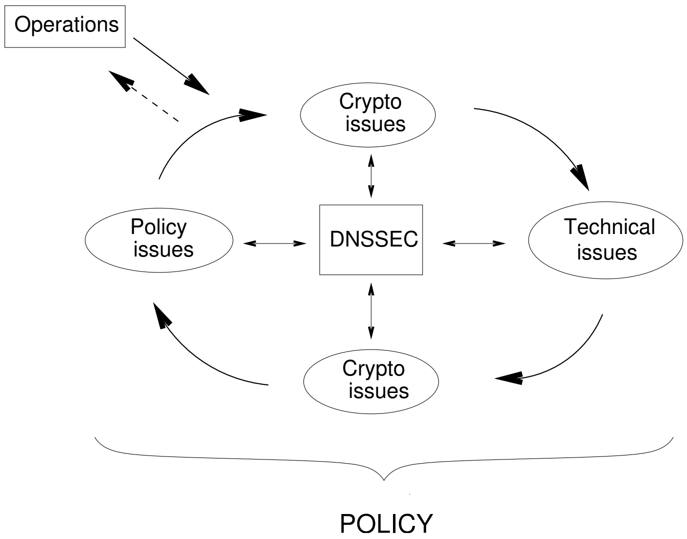

FIGURE 3.1. : Defining the DNSSEC policy.

As figure 3.1 shows, defining new procedures for DNSSEC is a complicated matter. Policy includes the new (DNSSEC) operations, the old operations, guidelines and legal contracts. A change in the DNSSEC protocol will have its impact on the policies of a registry.

3.8. DNSSEC Operations

With DNSSEC a new set of operations has to be supported. These operations are completely separate from the current DNS operation. In SECREG all DNSSEC operations are using a different database. The newly introduced operations in DNSSEC are:

securing Upload the zone key to SECREG and start being secure,

key rollover Uploading a new zone key to SECREG and phasing out the older one,

block Revoking the security of a domain (falling back to a unsecured delegation) in case of a key compromise or administrative reasons,

change zone-c Update the current zone contact.

In section 2.4 on page 7 this is explained more verbose.

securing "Someone" will start this procedure. This "someone" is promoted to the zone-c if, and only if the operation is completed successfully. After the initial checks email are sent out to the admin-c, tech-c and registrar notify addresses. From an admin-c/tech-c a positive acknowledgment is expected. If this acknowledgement is not received the operation is aborted in the next two days. The registry can send in a negative acknowledgement. If this negative acknowledgement is received the operation is also aborted (no matter if a positive acknowledgement was received).

Again after a successful completion, "someone" is now the official zone-c and the uploaded key can be used as an authentication mechanism in future communications.

key rollover The zone-c uses this operation to go from an old key the a new one. The zone-c and the new key are all authenticated by the old key before the operation is allowed to proceed.

block When the key is lost this operation can be used to block all further updates to a zone. This operation can only be performed once. If successful the domain's security status is dropped.

change zone-c With this operation the security contact's email address can be changed. This operation is purely administrative.

As said above the operations for DNSSEC are separate and can be deployed along side the current operations. When a new registrations system is build they can be folded in with the rest of the operations.

3.8.1. DNS Operations

Currently once the password is checked an operation is carried out. With DNSSEC some operations are more complicated as there is now a key in play. All updates to the zone file need to be signed before they are valid. For instance, a domain is sold to a third party. But the current owner does not hand over the private key. This leads to the problem where the new owner is unable to update zone. When it *does* update the zone it becomes invalid. To solve this either the domain has be to un-secured (i.e. no DNSSEC), or the key must be updated.

There are five ways a registry can deal with this:

1. somehow facilitate a (private) key exchange between the old holder and the new one,
2. drop the security status of zone; perform the operation; re-initiate the security (a new "make secure" operation).
3. require a new key; no matter the operation, just require a new key after it.
4. A "we don't care stance": just do the operation no matter the impact on the key material,
5. Keep the DNS and DNSSEC operations separate.

Facilitate private key exchange In this case the security status of a domain is not dropped, ie. the zone stays signed and the key also remains the same. The advantage is that from a DNSSEC viewpoint the zone does not change. The major drawback is that the old holder must be willing to give up a private key – something that violates rule number 1 of public key encryption: "Do not *ever* disclose your private key". Further more the new holder will receive a key that is used and created by someone else and the new owner has no guarantee what so ever that the old holder has deleted its copy of the key. This makes a private key exchange a *highly unattractive* method for use when DNSSEC is deployed.

Dropping a zone's security status In this case whenever a operation is performed the zone security status is moved back to the DNS level. Then the operation is performed. No key exchange is needed, because there is no zone key. After a successfully performed transaction the zone is secured again. Although this last step is optional. Currently is looks like this method is the only useful method for performing administrative operations with DNSSEC secured zones.

Requiring a new key This is a slight variation on the above mentioned method. After each operation a key rollover is required. If the key rollover does not happen the DS is removed from the .nl zone, and the domain stops being secured.

We don't care Now just perform the operation as it would work in the DNS world. If a domain moves to a new holder, just do that and don't care (yet) on the repercussions it has on the key material. After the operation is completed check to see if the zone is still signed with the same key. If so continue to keep the DS in the .nl zone. If not, delete the DS. This could lead to the situation where old key material still litters the DNS space even with a valid signatures from .nl, while this should long be deleted. This is a situation that should be avoided.

SECREG approach In SECREG the DNSSEC operations are separated from the DNS operations. This was done because SECREG does not have direct access to the .nl database. This is like the "We don't care" stance but with some extra checking. A soon as something is not right the DS is removed and the chain of trust is broken.

3.8.2. How to handle operations

Table 3.1 sheds some light in the responsibilities of the zone contacts.

	apex zone changes	local zone changes	administrative changes
involved			
admin-c	no	no	yes
tech-c	yes	maybe	maybe
zone-c	yes	yes	no
parent-child interaction	yes	no	yes

TABLE 3.1. : DNS operations and reponsibilities matrix.

apex zone changes Changes in a zone which affect both parent and child. Currently this is a nameserver change. DNSSEC adds an update to the key to it.

local zone changes Changes in a zone which do not affect the parent. Adding an new MX record falls in this category.

administrative changes Updating the address of the domain holder is an example of this.

3.9. Current situation

With the arrival of DNSSEC a lot more emphasis is put on security. Registries should ask them self the following questions:[4]

- What kind of security is in place right now?
  - Physical nameserver security,
  - Network security for the nameservers,
  - Zone upload procedures,
  - etc.
- Procedures for registrars
  - How are they validated and authenticated?
  - How are passwords generated and distributed?
  - How are those passwords used by registrars?
- What is the likelihood of an successful attack on the TLD nameservers?
- What is the status of the secondaries of this TLD?
- Procedures for personnel
  - Revoking root access
  - Layed off people
  - Disgruntled ex-employees

These are questions that should be answered before truly implementing DNSSEC in a registry. If a registry cannot or will not answer these questions they *should not* deploy DNSSEC in their zone.

Registries should not lose sight of the fact that they are securing the DNS. It makes little sense to protect the .nl zone key more than for instance the nameservers itself. Take for example the uploading of the .nl zone file. If for some reason this has not happened, it is rather pointless that the .nl key is in a fault somewhere – the Dutch DNS is still down. On the other hand some measure must be taken to prevent, for example, an employee of SIDN to hijack the (private part) of the .nl zone key, but this no different than disallowing root access to the nameservers for that person. Each TLD registry should have internal procedures in place to handle this.

[4] And the answers to these questions must of course be documented.

3.10. Who should own the private zone key?

With DNSSEC each secured zone in the DNS has to have a zone key. An important component of this key is the private key. This private key is used to sign the zone. The security contact is (from SIDN's viewpoint) the owner of the private key. This zone-c can be someone reporting to the domain holder or somebody reporting to the provider. There is a slight, but important, difference between these options.

Domain holder holds the private key. If this is the case the domain holder must have a direct link to the registry in order to update a key signing key. The ISP cannot do this, because it has no access to the key(s). This is a departure from the current registrant-registrar-registry model. In SECREG this model is chosen. The zone-c directly communicates with the registry. Almost all of these operations can automated, as SECREG has shown.

ISP holds the private key. The advantage here is that nothing changes to the current model. The registry can still communicate with the ISP. And a trust relationship between the registry and the ISP already exists. This relationship can be used to implement a procedure whenever a zone key is compromised.

3.11. .NL private key

With DNSSEC SIDN will own the .nl private key. This key could either be the starting point of trust in .nl (in case the root is not signed) or it could be signed by the root, in which case trust is delegated from root down to the .nl zone.

This private key is as important as the .nl nameservers, without the key .nl is not secured and without the nameserver .nl is down. Theft, or other key related accidents should not happen. Only trusted employees should have access to the private key. Procedures concerning this key should be comparable on how SIDN handles root passwords of important servers (i.e. the nameservers).

Key compromises, especially from child zones, are likely to happen and require more work from SIDN. A good and fast procedure to handle this might be in order.

3.12. NL public key distribution

With the signing of .nl also comes the burden on distributing the public .nl key to the resolvers of interest. This distribution cannot be done within the DNS. This is the starting point of trust. Publication in news papers, press releases, publication on the website could be considered. Together with the DS, because that is easier to check by humans.

There are two different possibilities which impact the key distribution problem: the root (".") can be signed or not.

Root is signed When this is the case, some root zone key will be the most important key in the DNS. Some, if not all, of the burden of distributing the zone key will then shift to the root operators. Via this root key the .nl DS record can be checked. The need for .nl to publish it own key will then not be so pressing.

Root is *not* signed When .nl introduces DNSSEC the root will probably not be signed. It is up to SIDN to distribute the .nl public key as widely as possible. It would be wise to also have procedures in place that can be used when the .nl key is compromised, lost or when a rollover is performed.

No matter if the root is signed "a" key needs to be distributed. This should happen via multiple channels; via the Internet, newspapers, television. This way end users can verify the authenticity of the key.

3.13. Key compromises

The only operation which is difficult to automaticly handle, is a key compromise. This operation will also be a prime candidate for attacks. If an attacker can fake a key compromise and then misuse the registry's procedure to upload a new key, a domain is essentially hijacked.

When a compromise has happened the child's zone key must be updated ASAP. As the zone key is useless some out of band mechanism is needed. The biggest threat here is that an attacker can roll over the current key at the registry by using the compromised key from the zone. Two things need to happen (1) the registry must be notified that the key is compromised. This must happen quickly and securely, and (2) a new key must be uploaded to the registry. For (1) a out of band update mechanism is needed and for (2) a choice can be made: restart the 'make secure' operation (i.e. pretend the zone was never secured) or utilize the out of band mechanism to get the new key and leave the zone secured.

How often will these compromises occur? Let's assume .nl holds 1 million delegated zones. One percent (1%, I'm optimistic here. . .) of those zones will experience a key compromise on a yearly basis. This comes down to $(1.000.000/36500)$ 27 compromises each day. Taking into account that most people don't work 365 day a year, but only up to 200, this boils down to about 50 compromises per working day.

Next to compromises, keys also can be accidently deleted or lost, so the 1% is probably a low figure. SIDN has to be prepared to handle these situations with ease.

3.13.1. NL key compromise

When the .nl key is compromised two things need to happen:

1. Contact the root (if it is signed) as soon as possible and replace the .nl key there,

2. Notify all users of the .nl public key of the fact that the key is compromised.

For both events a procedure must be in place. Also any liability issues outside SIDN's control should be avoided.

3.14. New records in .nl

This section explains each record. A short rational is given if it should be included in the .nl zone. The names of these records are likely to be changed[5], the new names are typeset in parentheses.

DS (DS) The DS record holds a hash value of a child's KEY record. It acts as a pointer to the key in the child's zone. In SECREG only one DS per child is supported. The actual DNSSEC deployment for .nl must support multiple DS records per child, but it should set a limit. A limit of three (3) seems to be a good starting point. This way child zones can have: 1. their current KSK, 2. a spare KSK, 3. an extra KSK for whatever purpose the child wants.

Only child keys that have a corresponding parental DS (plus the signature) in the parent zone, can use that key to sign their keyset. Ie. use the key as a KSK. There must of course also be a valid chain of trust from .nl down to their zone.

KEY (DNSKEY) The .nl zone keys must be included in the apex of the Dutch zone. See section 2.6 for more information on the keys.

NXT (NSEC) Every name in the .nl zone will get a NXT record indicating what the next name is. DNSSEC uses this to precompute information on the non existence of data (which can then be signed). NXT records are the predominate factor for larger zones when using DNSSEC, because every NXT also carries a signature. Zones in DNS can be up to 10 times as large as they are now. Experiments have shown that this does not pose a problem.

CERT (CERT) The CERT RR carries a certificate (key+signature). Currently they are not used much. They probably have no place in the .nl zone

SIG (DNSSIG) This records hold the signature data. Every RRset (dname, qname, qtype)-tuple gets a signature in DNSSEC.

3.15. Future developments

When deploying DNSSEC a first step is taken to a new sort of registry operation. The zone-c will have to be able to directly communicate with the registry to perform certain operations. An implication of this is that the key stored in the DNS could be used for authentication purposes. We probably are going to witness more of these "move some administrative issues into the protocol" actions. St Johns' secure notify draft [7] is an example of how administrative items are already being pushed in the protocol.

Technically this is a good development which could lead to a more robust DNS (less lame nameservers, for instance). But on the policy side this is much harder to implement. There we have ISPs making a living of registering domain names and fixing mistakes. But the features

[5] To prevent backward compatiblity problems with the old style DNS *and* the old style (RFC 2535) DNSSEC new type codes and names for the DNSSEC records are needed.

will probably make it to the protocol, after that registries will need to figure out for them selves how to use them.

3.16. Overall conclusion

In the introduction section the main conclusion(s) are already given. I want conclude with the observation that with DNSSEC we are going to see a few changes in DNS and in the DNS registration process. The most important change (for .NL at least) is the addition of a zone contact.

The SECREG experiment has shown that even with relatively little effort (the registry was written by one person) DNSSEC can be deployed and with the help of cluefull ISPs (Xtended Internet and BIT Internet) we have shown that it can scale. Finally the protocol looks robust enough to deployed.

2005 looks to be a great year for DNSSEC. . .

Bibliography

[1] Jacco Tünnissen
http://www.dnssec.net

[2] version v1.0, Stichting Internet Domainregistratie Nederland

[3] Nominum DNSSEC FAQ
http://www.nominum.com/getOpenSourceResource.php?id=8

[4] Delegation Signer Resource Record
Olafur Gudmundsson,
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-delegation-signer-15.txt
Work in progress

[5] DNS Security Extension Clarification on Zone Status
E. Lewis, Request for Comments: 3090
http://www.ietf.org/rfc/rfc3090.txt

[6] Parent's SIG over child's KEY,
R. Gieben, T. Lindgreen
http://www.nlnetlabs.nl/dnssec/dnssec-parent-sig-01.txt

[7] Using DNSSEC-secured NOTIFY to trigger Parent Zone Updating,
M. StJohns
http://www.ietf.org/internet-drafts/draft-stjohns-secure-notify-00.txt
Work in progress

[8] KEY RR Secure Entry Point (SEP) Flag,
Olaf Kolkman, Jacob Schlyter, Edward Lewis
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-keyrr-key-signing-flag-08.txt
Work in progress

[9] DNSSEC key operations,
O. Kolkman, R. Gieben
http://www.ietf.org/internet-drafts/draft-kolkman-dnssec-operational-practices-00.txt
Work in Progress

Appendix A

Scaling experiments

As of October 1^{st} 2003 15.000 domains were added to SECREG. Initially is was anticipated that having 50 domains in SECREG would classify it as a succes. In the design of SECREG as a registry scalability was not a requirement. Having this many domain meant that some changes to the registry were in order. Most of these changes have nothing to with DNSSEC, but merely on how the registry itself was build. This is really a good indication of how the DNSSEC protocol has progressed over the years. It finally looks ready to be deployed.
Also in this chapter two accounts are given from participating ISPs.

A.1. Issues with SECREG

For technical reasons `whois` serves as an interface between SECREG and the database from SIDN. Every day SECREG checks the consistency of its database with SIDN by doing a whois query for every domain in its database. This scales only up to a certain point. With 15.000 domains this check took too long (36+ hours or more). Therefore this check was completely eliminated from the day to day operations. The check now takes places after the .NL zone has been signed and loaded. This had the undesired effect of SECREG not being entirely synchronized with the official NL zone. This came apparent when signing the NL zone. Sometimes a zone was deleted from the NL zone but SECREG still had the DS record for that zone. Upon seeing a DS record without the NS records the signer would stop signing. After patching the signer to siliently discard this DS, the problem was gone.

It is better to integrate to two databases or find a way to make the seperate database communicate in an efficient manner. If some TLD is going to deploy a shadow zone of their own I *really* urge them to find better ways to synchronize the zones, (scripted) `whois` really doens't cut it.

A.1.1. Batch processing in SECREG

There was no provisioning for batch processing in SECREG. The initial aim was to have up to one hundred domains from various ISPs. It turned out this was a missing feature. If you want people to really help out in these kinds of experiments there must be some kind of mechanism to allow for batch operation.

Some interesting experiments were not conducted due the this missing functionality. Most notably a mass-key-rollover of an ISP was not tested.

A.2. Account from Xtended Internet

Xtended Internet moved about 150 domains to SECREG to see how that would work out. Every domain had two keys, so 300 keys had to be generated. The registration in SECREG and other operational items works flawlessly. Mass rollover has not been tested. Single zone key rollovers however were tested and they worked.

A.2.1. Snort sees DNSSEC reponses as hacks

Large dnssec responses seem to trigger a snort DNS alert. Xtended Internet experienced some false positives from a (still unknown) intrusion dectection system.

A.2.2. Overall experience

Positive, no real problems were experienced. However one problem remains below the service. . . key generations. Each generated key needs a certain amount of (truely) random data. With only a few keys the random data source on most computers is depleted. Currently some USB key devices have special hardware which is capable of generating a lot of random bits for a very low price and do this quite fast. These devices could be used to speed up this key generation.

A.3. Account from BIT

BIT has choosen to use one zonekey and one key signing key for all their domains. Signing all the zones with the same key may yield some penalties from a cryptographic viewpoint. But the operational benefits are huge. There are only 2 keys to worry about, in stead of the 30.000 keys that would have been required in a normal setup. Of course it is still possible for domains to use other keys then these so called *überkeys*.

Another reason to use just two keys is the above mentioned problem: each key generation requires some random data, with 30,000 keys this would need to much random bits and would thus take too long. That's the reason BIT deceided to just use 2 keys for all their domains. Of course nothing prevents them from using different keys for a small subset of their domain. These domain could then be called 'highly secured' or whatever.

A.3.1. Initial bootstrapping in SECREG

As there were too many domains to add them one by one, the registration system of SECREG was bypassed. All the domains were added to the database by use of a perl script. This also meant no additional checking was done on the keys and signatures. This *may* (we are not sure) caused problems that popped up at BIT.

A.3.2. Problems

In short: BIND9 crashed a lot, several times a day. It was discoverd that these crashes were DNSSEC related, disabling DNSSEC resulted in stable service.

After some investigation it turned out that all the signatures were made with a key from root (.), in stead of a key from the domain itself. If the domains were added via the mechanism used by SECREG this would have been detected, as the signatures would not been valid. Note that keys from the root are not illegal, and even if they were, BIND9 should not crash on this.

As the operations of BIT are more important than the testing of DNSSEC no follow ups of the experiment were done. Ie. DNSSEC remains disabled at BIT.

A.3.3. Overall experience

Due to the crashing servers BIT was unable to perform additional operations, but other than this problem the internal switch to DNSSEC proved not to be very difficult. The interface to SECREG was understandable and internally at BIT the domain name administrative was changed to incorperate DNSSEC. DNS administrator are now able to turn on DNSSEC and signatures and keys magicly appear in the zone on the nameserver. This shows that even the provider site of DNSSEC can be automated.

A.4. Scaling conclusion

The general conclusion drawn from this scalability experiment is that with relatively little effort we could scale SECREG to 15000+ domains. SECREG handled it ok and the participating provider could work well with the system and with DNSSEC. However some software bugs prevented us from realy testing all the interesting tidbits (mass key rollover). When deploying DNSSEC in NL these items still need testing.

All in all SECREG and DNSSEC proved not to difficult to be implemented, the DNS control software at BIT can now automaticly produces DNSSEC zones, the local administrators only need to know the bare basics of DNSSEC all other stuff is automated. The extension to 15.000+ zones proved that is feasible to deploy DNSSEC in a TLD.