DNSSEC Key State Transitions

Yuri Schaeffer, Yuri@NLnetLabs.nl

June 6, 2011

1 Warning

This document is a rough explanation of an idea. It is based on earlier work and may use but not introduce terminology of an other document. Possibly introduce completely new notations, skip reasoning steps without warning (well, you are warned now) and be plain wrong. This document is considered unsuitable for young children and Liberal-Arts majors. Read at OWN RISK.

2 Introduction

I introduce a new method to decide which records of a key to publish or unpublish. It is largely based on my previous work. We still use the perspective of the validators, and have therefore the same statemachine as before.

To increase readability and confusability I write DS as A, Dnskey as B, Rrsig Dnskey as C, and Rrsig as D. I also introduce a new notation to indicate the state of a record which is used as a superscript.

direction	P ≤ 1	P 1
in	$\uparrow$	$+$
out	$\downarrow$	-

The subscript refers to the key from the current record. For example: B_x^+ Means there exist some key material x with DNSKEY (B) on introducing and fully propagated to all validators ($+$). or: $\exists x \in Keys \cdot Dnskeystate(x) = Omnipresent\ B^+$ (without the subscript refers to the key currently being evaluated. Furthermore, A_x^+ is stronger than $A_x^\uparrow$. When the latter is true the former is true as well. $A_x^+ \rightarrow A_x^\uparrow$.

3 Rules

The below rules *should* be true for each record. A detailed explanation follows later after the overview.

$$A_x^\uparrow \tag{1}$$

$$\neg A_x^\uparrow (A^- \vee B^+ C^+) \vee A_x^\uparrow B_x^+ C_x^+ \vee A_x^+ B_x^\uparrow C_x^\uparrow A_y^+ B_y^\downarrow C_y^\downarrow \tag{2}$$

$$\neg B_x^\uparrow (B^- \vee D^+) \vee B_x^\uparrow D_x^+ \vee B_x^+ D_x^\uparrow B_y^+ D_y^\downarrow \tag{3}$$

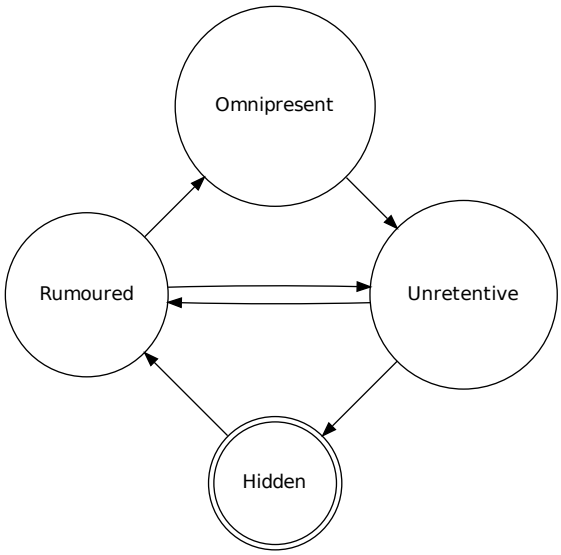

Figure 1: State diagram for individual records.

Each key material is traveling in a set direction (goal). Each record strives to a stable state which satisfies travel direction and maximizes certainty.

The method of advancing the states is the same. Look at each individual record for each key in the zone. Keep looping over this set while there are still changes. When no changes can be made return the time for a next update.

4 Evaluation

First we check the desired state of a record, when this is the same as the current state we stop (for this record) and continue to the next. Otherwise we will evaluate all rules for this record.

When evaluating a record we check which of the rules are true in the current situation. Next step is to apply the change (or pretend we did) and run the checks again. When all rules that are true pre change are still true post change we are allow, dnssec wise, to go to the next state. By allowing the checks to be false pre change we have a mechanism to repair an invalid DNSSEC state that somehow came into existence (user messed with database?).

The next step is to take a look at the timing. Given the transition we are planning to make and the type of record, are we allowed to make the next step? If so we do just that. If not, we give the time as feedback to the rest of the engine.

5 Rules explained

$$A_x^{\uparrow} \tag{4}$$

The notation is not sufficient here. What I mean to say is there must be a DS Rumoured or Omnipresent at all times. Does not matter which key or what algorithm.

$$\neg A_x^{\uparrow}(A^- \vee B^+C^+) \vee A_x^{\uparrow}B_x^+C_x^+ \vee A_x^+B_x^{\uparrow}C_x^{\uparrow}A_y^+B_y^{\downarrow}C_y^{\downarrow} \tag{5}$$

We can break this one up in three parts. At least one must be true.

1. If there is not any DS rumoured or omnipresent the key we are evaluating must have its DS hidden or its DNSKEY and RRSIGDNSKEY omnipresent.

2. or, there must be a key with its DS rumoured or omnipresent and its DNSKEY and RRSIGDNSKEY omnipresent

3. or, there must be a key with DS omnipresent and its DNSKEY and RRSIGDNSKEY rumoured or omnipresent and a key with DS omnipresent and its DNSKEY and RRSIGDNSKEY unretentive or hidden.

Note: Although the notation does not reflect it, it is stated implicit that each statement about any key is only applicable to keys of the same algorithm.

$$\neg B_x^{\uparrow}(B^- \vee D^+) \vee B_x^{\uparrow}D_x^+ \vee B_x^+D_x^{\uparrow}B_y^+D_y^{\downarrow} \tag{6}$$

We can break this one up in three parts. At least one must be true.

1. If there is not any DNSKEY rumoured or omnipresent the key we are evaluating must have its DNSKEY hidden or RRSIG omnipresent.

2. or, there must be a key with its DNSKEY rumoured or omnipresent and its RRSIG omnipresent

3. or, there must be a key with DNSKEY omnipresent and its RRSIG rumoured or omnipresent and a key with DNSKEY omnipresent and its RRSIG unretentive or hidden.

Note: Although the notation does not reflect it, it is stated implicit that each statement about any key is only applicable to keys of the same algorithm.