

Below is a summary of the process Verisign "invented", as it is described in the [patent application](#), page 5, paragraphs [0051]..[0060]:

Operation	Step	Description
new zone	S1020	Gaining registrar creates a new zone and signs it with a new set of keys
new ZSK -> old zone	S1030a	Domain owner sends the new DNSKEY and the DS records to the losing registrar. Losing registrar adds the new DNSKEY to the zone (and re-signs the new DNSKEY RRset with the old KSK),
new DS	S1030b	New DS record is added to the parent zone
lower old DNSKEY TTL	S1030c	TTL of the losing registrar's DNSKEY records is lowered ( <i>Verisign's mistake - lowering TTL at this step has no effect, it has to be lowered in advance, well before the step S1020</i> )
lower NS TTL	S1030c	TTL of the NS records is lowered
DS delay, DNSKEY delay	S1050	Wait for DS and DNSKEY TTL values to expire. ( <i>the same mistake is made - they ask to wait for expiration of the <b>new</b> DNSKEY TTL, while in reality it is necessary to wait for expiration of the <b>old</b> DNSKEY TTL</i> )
NS change	S1060	Change the NS records to the gaining registrar servers.
NS delay	S1070	Wait for the NS TTL to expire
domain transfer	S1080	Transfer the domain ownership.
old DS removal	S1090	Remove the old DS records.

And this is how this process is described in the "[Changing DNS Operators for DNSSEC signed Zones](#)" IETF draft, sections 3.1 and 3.2:

Operation	Step	Description
new zone	S1020	Gaining registrar creates a new zone and signs it with a new set of keys
<b>old ZSK -&gt; new zone</b>	-	<b>Gaining registrar adds the old DNSKEY to the new zone (and re-signs the DNSKEY RRset with the gaining registrar's KSK).</b>
domain transfer	S1090	Transfer the domain ownership.
new DS	S1030b	Add the DS record for the new DNSKEY
new ZSK -> old zone	S1030a	Registry passes the new DNSKEY records to the losing registrar. Losing registrar adds the new DNSKEY to the zone (and re-signs the DNSKEY RRset with the losing registrar's KSK).
DNSKEY delay, DS delay	S1050	Wait for the old DNSKEY to expire. DS expiration is not explicitly mentioned in the draft.
NS change	S1060	Change the NS records to the gaining registrar servers.
NS delay	S1070	Wait for the NS TTL to expire
old DS removal	S1080	Remove the old DS records and remove the old DNSKEY from the new zone.
<b>old ZSK removed from new zone</b>	-	<b>Gaining registrar removes the old DNSKEY from the new zone (and re-signs the DNSKEY RRset)</b>

What Verisign actually proposes is essentially the same process described in the IETF draft, but with **one step omitted**.

The problem with Verisign "invention" is that after we change the NS records (step S1060) and some resolver have an old RRSIG in it's cache and the DNSKEY is already expired from the cache, what will happen is that the recursive resolver will query the **new** authoritative servers for the DNSKEY records and will get the new DNSKEY RRset *without the old ZSK* (that is *the invention*). What we end up with is an RRSIG signed with the **old** DNSKEY, and the **new** DNSKEY RRset (which have **no** old DNSKEY - that is *the invention*). Recursive resolver will not be able to validate the RRSIG, and DNSSEC validation will fail. The domain will not validate and therefore it will not be accessible.

The same story repeats itself for a case where only the DNS hosting provider is switched (and not the registrar) - paragraphs [0070]..[0075], adding another mistake at [0075] c. - Verisign suggests to change the nameservers and then immediately "*Remove losing hosting provider DS record*". But what if some recursive resolver have RRSIG and DNSKEY in it's cache, while the DS record is already expired? - The resolver will receive the **new DS** only and the domain will fail to validate again.

Maybe Verisign should not invent DNSSEC procedures after all.

*Corrections and any other feedback will be greatly appreciated - ag@net-me.net*

*"...Systems and methods of transferring a DNSSEC enabled domain ... are described in which **the transfer of the domain may be achieved without disruption to a DNSSEC validation of the domain...**" (from the patent abstract)*