# Proposal for a Tamper-Proof Root Zone Update Process

Steve Crocker

16 October 2017

## Introduction

This is a proposal for a modification to the architecture and operation of the root zone update process to address what is fundamentally a political issue, not a technical issue, viz providing credible assurance to governments around the world that the U.S. Government or ICANN cannot abruptly or arbitrarily remove or otherwise tamper with the contents of TLD entries in the root zone.

## “What if the U.S. Government takes us out of the root?”

There is a periodic concern raised that ICANN might abruptly remove a ccTLD from the root, possibly at the direction of the U.S. Government. No matter how often or how vigorously the U.S. Government and/or ICANN say this will never happen, the theoretical possibility – and hence the perceived vulnerability – remains. One related example is argued to provide a precedent. In the design of GPS, the U.S. made it clear that it could restrict access to the full precision of the system, and it did so during the Gulf War. This has been part of the justification in Europe for the Galileo project.[[1]](#footnote-1) Russia did not wait for the Gulf War and started GLONASS even earlier.[[2]](#footnote-2)

It would not be rational for the U.S. or ICANN to take any ccTLD out of the root without that country’s concurrence. If it did so, it would instantly undermine U.S. and ICANN credibility and likely cause a rapid dissolution of the IANA function. Moreover, doing so would have no useful value. The effect of removing an entry from the root is gradual, but the response would be immediate. There would be a massive response around the world to locally reconstruct the “damaged” root zone. Nevertheless, even though unlikely and irrational, it’s currently not impossible for this to happen. And even if this never happens, there’s no credible assurance that can be given to suspicious countries that such a thing will never happen.

## Failing Safe

This paper is based on one key idea: Among the various sorts of errors that might be possible, it is far better to occasionally delay a requested change to a TLD operator’s portion of the root zone than it would be to have a change that is not requested and authorized by the TLD operator.

The current process for editing the root zone is controlled by PTI in response to requests from TLD operators. PTI processes requests and sends them to Verisign. Verisign updates the master copy of the root zone database. Verisign generates a fresh copy of the root zone twice a day, which is then made available to all of the root server operators.

If one asks what sorts of errors might have the biggest negative effect, there’s a huge difference between mistakenly making a change to a working entry versus failing to make a requested change. And the most serious change that might be made improperly is to remove or disable the TLD’s entry completely.

If one were going to design a system that is very, very resistant to catastrophic error, it would make sense to build in extra safeguards against accidental change of a working entry. Taken together with the political concerns that removal might happen on purpose, I think there is a good argument in favor of redesigning the root zone update function so that it is verifiably protected against inadvertent or malicious change of working entries. Note that a mistake in a change may still be possible if the TLD requests an erroneous change; we are not guarding against that case here.

Improving the root zone update process in this way is not a trivial task, but it’s achievable. As a rough precedent, the U.S. has extremely strong controls on the use of its nuclear weapons. The specific technical issues regarding control of root entries are not identical to the control of nuclear weapons, but there are some similarities. Specifics aside, the kind of thinking that has gone into designing and fielding fail-safe systems is likely to be relevant here. See, for example, the literature on Permissive Action Links.[[3]](#footnote-3)

## A Conceptual Architecture

Here’s a very high-level view of how to design a Tamperproof Root Zone Update System (TPRZUS). What follows here is intended only to highlight the key ideas. A design detailed enough to be implemented remains to be developed.

### 1. The TPRZUS

The first piece of the conceptual design is the update process will be sealed in a tamperproof system, i.e. TPRZUS. Today’s system is spread across the TLD operators, Verisign and ICANN. The communication among these parts of the system is protected with authentication and encryption but within each organization trusted people have access to all of the moving parts. This proposal would reorganize the collection of pieces of the system be reorganized to be tamperproof. Any change to the design or implementation of the system would require special, coordinated access and the participation of an external control group.

### 2. Normal Operation

The second piece of the conceptual design is that in normal operation any change to the part of the root zone associated with a particular TLD would require active affirmation by the TLD operator. For purposes of discussion, let’s assume each TLD operator is given a token or box that is cryptographically keyed to the root zone update system, and that a positive message is required from that box before a proposed change can be made to that TLD operator’s portion of the root zone.

Thus, in normal operation, a TLD operator would request a change, the operators of the TPRZUS, i.e. ICANN and Verisign, would agree to the change, the change would be entered into the TPRZUS, and then an exchange of messages would take place between the TLD operator and the TPRZUS.

For readers knowledgeable about the details of the root zone and/or the idea of cryptographically keying a token to a system, a couple of questions come quickly to mind:

* What about portions of the root zone that might be shared among multiple TLDs, viz the address (“glue”) records for name servers that are used by more than one TLD?
* It’s relatively easy to see how to associate a token with a central system and to empower it to make changes. It’s less obvious how to make sure no other token is similarly empowered.

We skip the first question for now. The second question is touched on below.

### 3. Special Operation

The third piece of the conceptual design deals with special operation as opposed to normal operation. Special operation applies to the assignment or reassignment of control to a specific portion of the root zone.

The second piece of the design, above, is based on a token assigned to each TLD operator that gives the TLD operator veto power over any changes to that portion of the root zone. There has to be a process for creating those tokens, ensuring that a particular token can control only a specific portion of the root zone, and then delivering that token to the appropriate TLD operator. Here we skip over the creation and inventory control of the tokens and sketch just the assignment of a token to an operator.

This part of the design is partly technical and partly political. The determination of who is the appropriate operator for a TLD and hence who should receive the token requires some form of transparent due process. In today’s operation, the corresponding action is delegation or re-delegation[[4]](#footnote-4) of the TLD. The current process for delegations and re-delegations is carried out by the IANA group with approvals and authorization by both the ICANN Board and NTIA. The ICANN Board and NTIA focus on whether the process has been carried out properly, notwithstanding the perception of some that one or both groups may insert political considerations.

As part of strengthening the entire system, it may be appropriate to make the delegation and re-delegation process more open and to have multiple parties from the ccTLD community participate. For purposes of discussion, let’s postulate the creation and operation of a Root Zone System Oversight Board (RZSOB) that is empowered to

* approve the creation of new portions of the root zone corresponding to new TLDs,
* approve correspondence between a token in the inventory of tokens and the portion of the root zone corresponding to a particular TLD, and
* approve the assignment and delivery of the token to the operator of that TLD.

In addition to initial delegations and assignments of tokens to TLD operators, the RZSOB would also handle lost or broken tokens and re-delegations. The RZSOB would also have the power to disable tokens.

The RZSOB would necessarily be composed of a significant number of independent parties who would have to agree on each action. Fleshing out the structure and operation of the RZSOB remains to be developed.
 **4. Transition**

The fourth piece of the design deals with transition. Quite obviously it will not be possible to transition to this type of operation all at once. One way to orchestrate the transition is put the TPRZUS into operation inside of the current IANA/Verisign operation. That is, the interface between IANA and each of the TLD operators would remain the same as it is today, but, after appropriate testing and parallel operation, the TPRZUS would be put into operation. IANA would hold the tokens for each of the TLDs.

Whenever a TLD operator is ready to take over its operation, the RZSOB would assign it a token and move the control from IANA to the TLD operator. Operationally, this would be quite similar to a re-delegation.

## 5. Next Steps

The design process can be split into three parts. One is the design of the core of the system, i.e. a tamperproof system encompassing the ICANN and Verisign portions of the system. The second is the protocol between the TLD operators and ICANN. The third is the set of procedures and operating rules for the Root Zone System Oversight Board. Each of these can be done more or less independently, so there are multiple options for exploring the concepts, design and modes of operation.

In particular, it would be possible to design and test tokens to be used by TLD operators without building the complete system, and it would be possible to build the sealed tamperproof system but with the same people controlling the process as today.

The critical question is how much Org should do in research mode before seeking community consensus on whether this is an important idea to pursue.

1. <http://en.wikipedia.org/wiki/Galileo_%28satellite_navigation%29#GPS_and_Galileo>

One of the reasons given for developing Galileo as an independent system was that position information from GPS can be made significantly inaccurate by the deliberate application of universal [Selective Availability](http://en.wikipedia.org/wiki/Selective_Availability) (SA) by the US military; this was enabled until 2000, and can be re-enabled at any time. GPS is widely used worldwide for civilian applications; Galileo's proponents argued that civil infrastructure, including aeroplane navigation and landing, should not rely solely upon a system with this vulnerability.

On 2 May 2000, SA was disabled by President of the United States [Bill Clinton](http://en.wikipedia.org/wiki/Bill_Clinton); in late 2001 the entity managing the GPS confirmed that they did not intend to enable selective availability ever again. [↑](#footnote-ref-1)
2. http://en.wikipedia.org/wiki/GLONASS [↑](#footnote-ref-2)
3. http://en.wikipedia.org/wiki/Permissive\_Action\_Link [↑](#footnote-ref-3)
4. I believe a new terminology is now in use instead of re-delegation but I don’t remember it as I type this. This document should be revised to use the new terminology. [↑](#footnote-ref-4)