Collision Free Key Tags for DNSSEC

draft-huque-dnsop-keytags-01

Mark Andrews, **Shumon Huque**, Yorgos Thessalonikefs

IETF 122, Bangkok, Thailand

March 20th 2025

Collision Free Key Tags for DNSSEC

Draft -00 was published during IETF120 in Vancouver, July 2024, and generated some positive discussion on the mailing list.

KeyTrap was in the air at that time (ANRP talk)

Refreshed as -01 recently and is on the agenda for the DNSOP meeting at IETF 122.

And added Yorgos/NLnet Labs as a co-author.

Abstract (for reference only)

DNSSEC employs a Key Tag field in the RRSIG and DS resource records in order to efficiently identify the key that produced a DNSSEC signature and the key that should be used as secure entry point into a delegated zone. The Key Tag was not intended to be a unique identifier. Key tag collisions can occur in practice for keys in the same zone, though they are relatively rare in normal operation. Colliding key tags impose additional work on a validating resolver, which then has to check signatures for each of the candidate set of keys identified by the Key Tag. Furthermore, they open up resolvers to computational denial of service attacks by adversaries deploying specially crafted zones with many intentionally colliding key tags. This document specifies updates to the DNSSEC protocol and the process of key generation to avoid colliding keys and enforce the uniqueness of key tags.

Proposed Updates to DNSSEC spec

Proposes to update the DNSSEC specification to disallow colliding key tags for DNSSEC keys within the same zone (i.e. the DNSKEY RRset).

Enforce for all new DNSSEC algorithms.

For existing algorithms, we can have aliases with new algorithm numbers. Will require transition period, algorithm rollovers and associated work of course.

Or, perhaps we can propose a future flag day for enforcement of no collisions for existing algorithms?

Motivations

Eliminate the possibility of computational denial of service attacks like KeyTrap, that employed specially crafted zones with many intentionally colliding keys in the DNSKEY RRset.

But more generally, make DNSSEC validation more efficient. Resolvers should not be forced to do unnecessary work to authenticate signatures with an entire set of candidate DNSKEYs, when faced with colliding key tags, so let's outlaw collisions.

Updated Key Generation Procedures

For Single Signer implementations, this is straightforward (and some implementations do this already)

For any new generated key, compute its keytag, and if it is the same as any other key in the existing DNSKEY RRset, throw it away, and generate a new key.

(Note: need to check revoked forms too as updating flags results in new key tags)

Multi-Signer (RFC 8901) Considerations

Here, multiple distinct parties could be generating their own keys for the same zone.

So key generation needs to be coordinated.

Possibilities:

- Partition the key tag space between the multi-signer parties. Each party needs to generate a key that falls within their allocated partition, and re-generate otherwise.
- Use a central key broker to enforce uniqueness
- Cross party communication during key generation, with race condition avoidance.
- Other?

Partitioning the key space

Assign each signer a range.

Generate a candidate key and test if it and its revoked form have tags which fall within the assigned range. Test if either tag collides with an in use tag or it's revoked value.

Keep the key if both tags are in range and they do not collide with an in use key.

Discard the key if either tag doesn't fall within the range or if they collide with a in use key.

Is the Working Group interested?

Lorem ipsum.