Measurement Survey of Server-Side DNSSEC Adoption

Matthäus Wander
University of Duisburg-Essen
Duisburg, Germany
matthaeus.wander@uni-due.de

***Abstract*—This paper answers the question how far DNSSEC signing has found adoption in practice. By applying zone enumeration techniques on all top-level domains we gather the number of 6.4 million signed second-level domains. This figure is a complete snapshot of the DNSSEC ecosystem as of January 2017. The adoption concentrates among a small number of top-level domains, some of them having half of their domains signed with DNSSEC, while most top-level domains have adoption ratios of 1%, or less. The majority of top-level domains use NSEC3 hashing to thwart zone enumeration, but GPU-based zone enumeration allows us to recover 79% of cleartext domain names.**

Most second-level domains use RSA as signing algorithm with a combination of 2048-bit and 1024-bit keys, but 512-bit keys are also common despite being demonstrably insecure. ECDSA adoption has grown to 8% within the last two years. 0.45% of domains are not signed correctly and fail to validate. However, there are fewer domains failing due to DNSSEC errors than due to other misconfigurations or network problems.

I. Introduction

Network attacks like spoofing threaten the security of the *Domain Name System* (DNS) [1], which serves as foundation for naming in most Internet applications. The *Domain Name System Security Extensions* (DNSSEC) [2] have been proposed to secure domain name lookups by providing data integrity and authenticity. Apart from protecting name resolution itself, DNSSEC offers the opportunity to tie application-level security measures to domain data with *DNS-based Authentication of Named Entities* (DANE) [3]. Use cases for DANE include putting security constraints on public-key certificates or associating email addresses with public keys. Despite its potential benefit, several challenges have been cited that hinder DNSSEC deployment in practice [4], including lack of motivation, interoperability problems and operational costs.

This paper studies the current state of server-side DNSSEC adoption, i.e. signed domains. We show that DNSSEC signing is widely spread and shed light on vital parameters like the choice of the signing algorithms and key sizes. The studies shown in this paper employ the following measurement methods (see Fig. 1):

a) Probing of all top-level domains (TLD) for their DNSSEC parameters over an almost 4-year period (Section III).
b) Zone enumeration of all TLDs to quantify adoption of DNSSEC signing for second-level domains (Section IV).

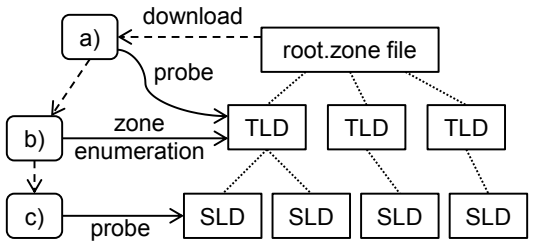


Fig. 1. Overview of measurement methods.

c) Probing of second-level domains (SLD) for their DNSSEC parameters (Section V).

Prior work typically relies on zone files or lists of popular domain names for large-scale DNS measurements, e.g. [5], [6]. Although convenient to use, this poses an incomplete picture because not all top-level domain operators offer zone files. Unlike when working with domain lists, zone enumeration guarantees to return the complete number of DNSSEC-signed second-level domains. Depending on whether the TLD uses NSEC [7] or NSEC3 [8] for authenticated denial of existence, the domain names gathered from zone enumeration will be returned as cleartext or as hash digest. We can recover the majority of domain names from NSEC3 hashes by using GPU-based hash breaking techniques [9].

Thus, this paper differs from earlier work by contributing a **complete** quantification of DNSSEC-secured second-level domains (6.4 million) and by contributing an analysis over a partial but so far **largest** set of signed second-level domains (5.1 million).

II. DNSSEC

The domain name space is cut into administrative entities called zones, where each zone comprises a set of resource records under a common domain name. DNSSEC uses public-key cryptography to add signatures over resource records (RRSIG records). Keys are authenticated via secure delegations (Fig. 2): the parent domain authenticates the public key of a subdomain via a parent DS record, which contains the fingerprint (hash value) of the public DNSKEY record of the subdomain. Like in classic DNS, NS records indicate the authoritative servers for a subdomain and glue records contain their IP addresses.

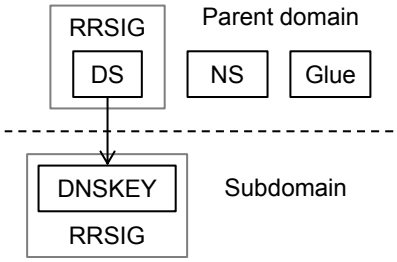


Fig. 2. Secure delegation in DNSSEC.

Different cryptosystems are specified for signing, identified via an algorithm number in DS and DNSKEY records. A common setup is to use two different key pairs for signing: the Key Signing Key (KSK) inherits trust from the parent domain and authenticates a Zone Signing Key (ZSK), which signs the actual domain data. The DNSSEC protocol does not require to use a separate KSK and ZSK; instead, one key pair can be used, which combines the purposes of a trust anchor and zone signing key.

When querying for a non-existent domain name or record type, DNSSEC uses the NSEC approach for authenticated denial of existence. The server returns the two closest existing names in lexicographical order to prove the non-existence indirectly. This allows to obtain a copy of all existing names by querying for non-existent names repeatedly (*zone enumeration*). To avoid disclosure of domain data, the NSEC3 hashed authenticated denial of existence attempts to hide domain names by returning hash values instead of cleartext names.

One of the use cases for authenticated denial of existence is the denial of a DS record, which indicates in a secure way that a subdomain does not use DNSSEC signing. This allows for gradual deployment of DNSSEC without the threat of downgrade attacks.

Definition: In this paper, we define DNSSEC as deployed for a domain if there is a complete chain of secure delegations from the root to the domain in question. This definition excludes domains that have been signed but did not publish a DS record at their parent domain. Such a state (known as *insecure* or *locally signed*) is common during a testing phase before fully activating DNSSEC and will be treated by validating resolvers in the same way as an unsigned domain without DNSSEC.

III. Top-Level Domains

We first survey the use of DNSSEC in top-level domains over an almost 4-year observation period, including the key sizes and frequency of key rollovers.

Method (Fig. 1a): We download the public IANA root zone file [10] daily and probe the authoritative name servers of all TLDs for various record types. Timeouts are handled by resending the query to a randomly chosen server of the TLD under test for up to a total of 10 attempts. Truncated messages are handled by retrying over TCP.

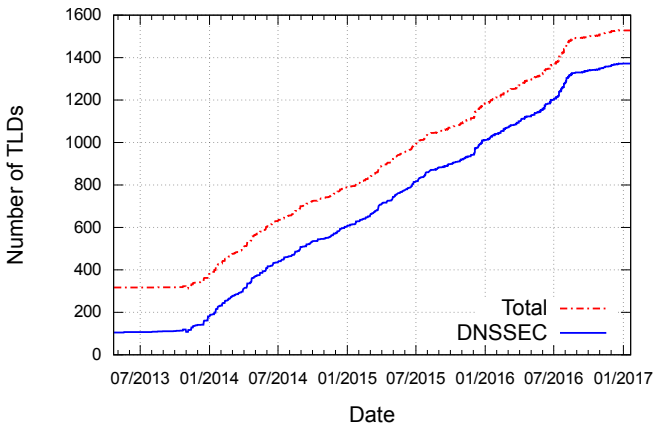


Fig. 3. Number of (signed) TLDs over time.

TABLE I
RSA key lengths of TLDs (January 2017).

Length	KSK	ZSK
1024	2	1610
1152	0	5
1280	5	526
1536	1	0
2048	2067	156
4096	18	0
Total:	2093	2297

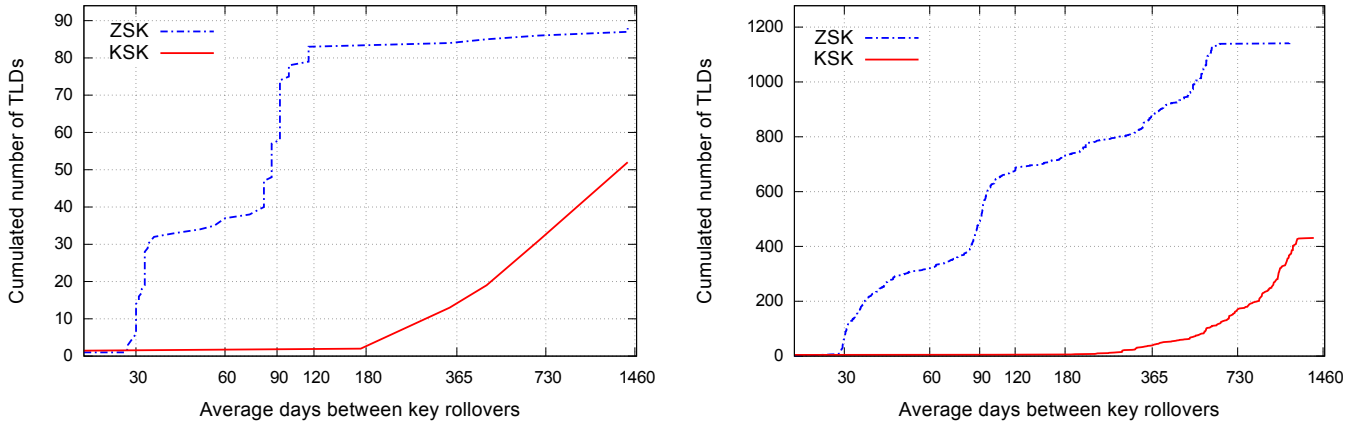
A. Quantification

The observation period covered in this analysis is from April 2013 to January 2017. Fig. 3 shows the number of TLDs over time. In April 2013, there were 317 TLDs, out of which 105 (33%) had deployed DNSSEC. The number has increased steadily since then, except for the removal of 13 TLDs. ICANN removed 11 test TLDs in October 2013 [11], as well as `an` (Netherlands Antilles) and `tp` (Portugese Timor) in 2015 because these country codes are now historic. DNSSEC adoption is growing among the remaining TLDs: out of those 304 TLDs still in existence, 158 (52%) have deployed DNSSEC by January 2017.

The total number of TLDs increased to 1528, out of which 1372 (90%) have deployed DNSSEC. The vast increase is mostly due to the introduction of new generic TLDs, which began in October 2013 and is still ongoing. 1210 new domains originate from the new generic TLD program; 14 more new domains are internationalized domain names (IDN) delegated in addition to country-code TLDs. All new generic TLDs are deployed with DNSSEC because registry operators are bound by contract with ICANN [12].

B. Public Keys

Of the 1372 TLDs that deployed DNSSEC, all domains use a KSK/ZSK scheme with RSA as cryptosystem. Table I shows the key sizes in use, separately for KSKs and ZSKs. Note that the total number of keys is larger than the number of TLDs due



(a) TLDs that were signed before April 2013 ("*old*").

(b) TLDs first signed after April 2013 ("*new*").

Fig. 4. Key rollovers intervals of TLDs (April 2013 to January 2017).

TABLE II
PUBLIC RSA EXPONENTS OF TLDS (JANUARY 2017).

Exponent	KSK	ZSK
3	165	165
65,537	1923	2123
$2^{32}+1$	5	9
Total:	2093	2297

to key rollovers or stand-by keys that are in the zone but not used for signing. The bulk of TLDs use 2048 bits for the KSK and 1024 bits for the ZSK, and this did not change during our observation period. However, there is the trend to use 1280-bit ZSKs for new generic TLDs, exercised by infrastructure provider Neustar and their subsidiary ARI Registry Services.

Key lifetime: We now analyze the average key lifetime to determine the time frame an attacker has for breaking the above keys. Fig. 4 shows the frequency of ZSK and KSK rollovers. Note that some TLDs did not perform a key rollover and thus the plots end without touching the top edge. The results are grouped into two different sets: a) TLDs that have been signed for the whole observation period ("*old*"), b) TLDs that have been signed for part of the observation period, in particular newly introduced TLDs ("*new*"). This bisection accounts for the possibility that newly signed TLDs may not have matured their DNSSEC operations yet. Most TLDs from the *old* set (88%) replaced the ZSK every 30 to 120 days; 55% also replaced the KSK at at least once during our 4-year observation. 6 TLDs from the *old* set did not replace their 1024-bit RSA ZSK.

Key rollovers were less common in the *new* set until around 2015, but are now approaching the figures of the *old* set. The average key rollover interval is slightly longer: 54% roll their ZSK within 120 days, 69% within 1 year, 89% within 2 years. The KSK has been replaced at least once by 34% of TLDs.

RSA exponents: Table II shows the public RSA exponents. The most common is $e = 65\,537$, which is also the most frequently used exponent in other RSA-based applications [13]. The choice of e affects the performance of RSA signature verification: modular exponentiation can be computed faster with small values that have a low hamming weight. $e = 3$ may offer an even better verification performance, but is susceptible to signature forgery on broken implementations that do not handle the message padding correctly [14]. Although this is an implementation weakness and $e = 3$ can be implemented securely, $e = 65\,537$ is a more conservative choice. $e = 2^{32}+1$ is also suitable, albeit less supported in cryptographic implementations according to anecdotal evidence, as it does not fit into a 32-bit integer.

IV. ZONE ENUMERATION

Method (Fig. 1b): In order to quantify DNSSEC adoption among second-level domains, we perform zone enumeration on all top-level domains. Zone enumeration exploits the disclosure of domain data in negative DNSSEC responses ("name not found" errors). It returns the cleartext list of all names within a zone that uses NSEC denial of existence [7] or a set of hash values of a zone that uses NSEC3 hashed denial of existence [8]. Depending on whether a zone uses the NSEC3 opt-out feature, we get either the hash values of all names (opt-out disabled) or the hash values of signed names only (opt-out enabled).

NSEC zone enumeration is trivially possible, whereas NSEC3 requires hashing power to find appropriate query names [9]. We use an OpenCL-based hashing implementation, which runs on a 4-core CPU to retrieve most NSEC3 hash values and switch to a GPU to close the last few gaps of large NSEC3 chains with $> 10\,000$ records.

A. Quantification

In January 2017, 176 TLDs were using NSEC and 1196 TLDs were using hashed NSEC3 for authenticated denial

of existence. The crawling finished after about ten days and yielded 14.5 million names from NSEC records and 13.5 million hash values from NSEC3 records. The NSEC3 count is lower than the NSEC count due to the use of the NSEC3 opt-out feature, which omits NSEC3 records for existing but unsigned domains. The types field in the NSEC/3 records allows us to classify the existing resource records within TLD zones.

Table III shows the top 15 TLDs with most secure delegations, i.e. DS records in column ①, and their NSEC/3 configuration. The parameter i represents the number of additional hashing iterations in NSEC3, which is used to adjust the hashing costs for server and attacker. The number of DS records gives us the number of second-level domains that adopted DNSSEC, which totals to 6.4 million. Most DNSSEC-signed second-level domains can be found under `nl`, `se`, `cz` and `com`. The accumulation of DNSSEC-signed domains under these TLDs is not a coincidence, because the operators provided an incentive for registrars to adopt DNSSEC [15]. SIDN (`nl`) has offered an 8% discount in registry fees for a two-year period. CZ.NIC (`cz`) has offered technical support and financial support of DNSSEC-related marketing campaigns. IIS (`se`) has offered a discount on DNSSEC-signed domains, which has been subsidized by the Swedish government. Although there was no such initiative for `com`, the incentives provided by country-code TLDs also fueled the deployment under `com` and other TLDs: a registrar, which adapted its business processes for DNSSEC support under one TLD, can add support for other TLDs with little extra cost.

Adoption Ratio: Each domain with a secure DNSSEC delegation is represented by one NSEC or NSEC3 record with the DS type set. As our crawler continues zone enumeration until it has retrieved the whole NSEC/3 chain, the DS count is guaranteed to be an accurate figure of DNSSEC domains. We can also count the total number of domains with or without DNSSEC, but this figure is incomplete for TLDs that use the opt-out feature, as it omits unsigned delegations. Instead, we estimate the DNSSEC adoption ratio (column ②) by comparing the DS count with domain statistics provided on registry websites. A few TLDs have adoption ratios of around 50%, which shows again registries that have endorsed DNSSEC implementation actively. The largest TLD `com` with around 126 million domains in total has a DNSSEC adoption ratio of 0.5%.

Address records: TLD zones consist mostly of domain delegations, but some TLD operators allow to put address records directly into the TLD zone, e.g. in `de`. This is reflected by column ④, which shows the number of signed, authoritative A, AAAA, CNAME or MX records. Although these domains are protected by DNSSEC, we do not include them in the adoption ratio for two reasons: 1) We cannot deduce reliably the number of signed second-level domains from the number of address records, because one domain typically consists of multiple address records. 2) These names are signed by the TLD operator without cooperation of the domain owner, which does not reflect domains whose owner chose deliberately to deploy DNSSEC.

Another cause for signed address records in a TLD are dangling glue records: when a delegation is deleted without also deleting the corresponding glue record, the dangling glue record becomes a signed, authoritative address record. This appears to have happened in many cases under `org` and `info`, which do not enforce removal of dangling glue records.

Empty non-terminals: Besides authoritative data, `org`, `de` and `info` have a significant number of NSEC3 records without any type set (column ⑤). This is an effect of NSEC3, which includes records for empty non-terminals, e.g. when an address record exists for `www.example.de` but not for `example.de`, or when a dangling glue record like `ns1.example.org` remains after the delegation of `example.org` has been removed. Empty non-terminals in a TLD zone are caused by the same reasons as authoritative address records: either by deliberate inclusion of an address record or by outdated glue records.

Other record types (column ⑥) indicate either unsigned domains when opt-out is not in use, e.g. under `se`, `cz` and `nu`, or special resource records in a negligible number of cases.

B. Limitations

The crawling of NSEC/3 records has finished for all but two TLDs: `pr` had a partly broken NSEC chain, probably due to configuration errors, and `by` changed the NSEC3 salt every 10 minutes, which was too frequent to retrieve the complete NSEC3 chain. Changing the salt creates a new NSEC3 chain and forces us to re-start the crawling process. As zone enumeration requires one network query for each NSEC/3 record, we would have needed to increase the query rate to retrieve an estimated amount of 60k NSEC3 records within a 10-minute window. Our crawler sends only one query at a time to avoid excessive load on TLD servers. In contrast, the NSEC3 hash iteration count does not have a significant impact on the crawling speed, because zone enumeration is network-bound for all but the last few gaps of the NSEC3 chain.

A disadvantage of counting second-level domains only is that it underestimates TLDs that subdivide their name space for domain registration on third level. For example, there are 1280 second-level domains in the `br` zone, but 3.8 million domain registrations when including third-level domains under `com.br` and others. Although most TLDs that used to enforce registration on third level have opened the second-level space, e.g. `tw` or `uk`, they continue to have significant registration numbers on third level as well, e.g. below `com.tw` or `co.uk`. While we could perform zone enumeration on second-level domains, we need a reliable method to discriminate whether a domain is open for public registration and whether it is useful to aggregate the second-level domain statistics below their respective TLD. We suggest for future work to consider Mozilla's Public Suffix List [16] for this purpose.

TABLE III
TLDs WITH MOST DNSSEC-SIGNED SECURE DELEGATIONS (JANUARY 2017).

TLD	NSEC/3	① DS	② Adoption	③ Cleartext	④ Address	⑤ Empty	⑥ Other
nl.	NSEC3, opt-out, $i = 5$	2,592,219	45%	78%	5	1	1
se.	NSEC	673,262	49%	all	10	0	711,757
cz.	NSEC3, $i = 10$	655,529	52%	82%	0	0	604,807
com.	NSEC3, opt-out, $i = 0$	614,209	<1%	75%	0	0	1
no.	NSEC3, opt-out, $i = 5$	409,416	57%	81%	4	2	2
eu.	NSEC3, opt-out, $i = 1$	355,157	9%	91%	7	1	1
fr.	NSEC3, opt-out, $i = 1$	304,663	10%	67%	0	2	9
be.	NSEC3, opt-out, $i = 5$	127,177	8%	82%	0	1	2
hu.	NSEC3, opt-out, $i = 5$	107,434	15%	70%	5	1	2
net.	NSEC3, opt-out, $i = 0$	101,872	<1%	88%	0	0	1
nu.	NSEC3, $i = 5$	78,443	26%	94%	0	0	226,618
org.	NSEC3, opt-out, $i = 1$	73,836	<1%	85%	14,495	7,075	1
de.	NSEC3, opt-out, $i = 15$	60,065	<1%	66%	257,728	72,433	3
pl.	NSEC3, opt-out, $i = 12$	27,821	1%	68%	8	0	1
info.	NSEC3, opt-out, $i = 1$	27,294	<1%	89%	22,156	10,804	1
		[1357 others omitted]					
	Total:	6,441,427			464,301	103,766	21,110,687

C. NSEC3 Hash Breaking

Although the NSEC/3 records suffice to quantify DNSSEC adoption, we require the cleartext second-level domain names to analyze additional parameters. Within the set of 13.5 million NSEC3 records, 5.7 million indicate a DNSSEC subdomain with the DS type set. We use GPU-based hash breaking [9] to recover names from these NSEC3 hash values with 7 graphic cards from hardware generations between 2011 and 2016. After about two weeks of brute-force and dictionary computation, we gather 4.5 million names (78.9%). The results are broken down per TLD in Column ③ of Table III.

We did not recover the cleartext name of all hash values because an exhaustive search is infeasible. The recovery ratio depends primarily on the amount of short names up to 8 characters, which can be exhausted by brute force, and on the quality of the dictionary for longer names. It is worth to note that the NSEC3 hash iteration count does not have a significant impact on the recovery ratio. Although the hash breaking takes indeed longer, it does not slow down the attack to a degree that would affect our recovery capability. For example, mx ($i = 100$), lat ($i = 100$) and la ($i = 150$) show recovery ratios of 80%, 79% and 96%, respectively. On the other hand, we recovered only 8% from xn--3e0b707e ($i = 10$), 39% from jp ($i = 8$) and 43% from name ($i = 0$), presumably because our dictionary is less suitable for these TLDs.

V. SECOND-LEVEL DOMAINS

In the following study, we survey the signing algorithms, key sizes and broken DNSSEC configurations among second-level domains. We compare the results from January 2017 with a previous measurement from March 2015, whose full figures are available in [17].

Method (Fig. 1c): We query the DS and DNSKEY records for each second-level domain name that we have retrieved with zone enumeration in Section IV. For an efficient and effective bulk retrieval, we use a threaded implementation that forwards all queries via DNS-over-TCP connections with keep-alive to a BIND9 recursive resolver. BIND9 takes care of handling errors, dealing with fragmentation and corner cases with uncommon or broken DNS server implementations.

A. Quantification

Our input domain list consists of 4.5 million domains from NSEC3 and 0.7 million domains from NSEC zone enumeration. Probing these domains yields 5.1 million results (2015: 3.4 million), which we analyze further. 90k domains (1.7%) are missing, most likely because they were removed within the few weeks when TLD zone enumeration commenced until when the second-level domain probing has finished. This is plausible, because the domain name space is changing constantly. We omit another 25k domains (0.5%) from the analysis because they failed to return a processible DNSKEY response, e.g. timed out repeatedly or returned a badly formatted DNS message.

99% domains appear to use a regular KSK/ZSK scheme, i.e. two separate key pairs for signing. This shows that using a combined signing key is not a common practice. For the sake of readability, we continue to use the terms KSK and ZSK, even if a few zones actually sign their zone data with the KSK.

B. Public Keys

Table IV shows the cryptosystems used for signing, separately for KSK and ZSK. The most frequently used cryptosystem is RSA with one of the SHA hash functions. The deprecated RSA/MD5 is not in circulation at all. DSA is used on a minor scale and this has not changed significantly since 2015. Unlike the elliptic curve cryptosystem GOST, which is rarely used, 256-bit ECDSA is used by a significant share of second-level domains (8%). This is a recent trend; in 2015, ECDSA adoption was as rare as GOST.

Table V shows the RSA key lengths in use. The most common combination is 2048-bit RSA for KSK and 1024-

TABLE IV
SIGNING ALGORITHMS OF SECOND-LEVEL DOMAINS (JANUARY 2017).

Cryptosystem	KSK	ZSK
RSA/MD5	0	0
DSA/SHA-1	3,567	3,699
RSA/SHA-1	1,806,540	2,764,225
RSA/SHA-256	2,855,191	4,114,435
RSA/SHA-512	41,019	79,850
GOST R 34.10-2001	37	100
ECDSA P-256/SHA-256	418,207	559,006
ECDSA P-384/SHA-384	477	296
invalid	2	0
Total:	5,125,040	7,521,611

TABLE V
RSA KEY LENGTHS OF SECOND-LEVEL DOMAINS (JANUARY 2017).

Length	KSK	ZSK
512	6,145	275,240
1024	784,784	6,289,586
1280	1,617	231,019
1536	226,072	228
2048	3,571,293	152,909
4096	112,209	8,444
Other:	630	1,080
Total:	4,702,750	6,958,506

bit RSA for ZSK and this has not changed significantly since 2015. Furthermore, there is a worrying amount of 512-bit RSA keys in circulation, which do not offer a reasonable security benefit over unsigned DNS. While the number of 512-bit KSKs went down slightly since 2015, both in relative and absolute values, the number of 512-bit ZSKs has spiked from 14k (0.3%) to 275k (4%). The majority of them can be attributed to a hosting provider below `.cz`.

Common prime factors: In other applications, weak RSA keys have been found that share prime factors due to insufficient randomness during key generation [18]. These keys can be broken efficiently by computing the greatest common divisor with Bernstein's algorithm over a large set of RSA moduli [19]. Although our set of 8.9 million unique RSA keys from top-level and second-level domains yields 6 keys with common prime factors, actually none of them are vulnerable, because the DNSKEY records have been either truncated or the server returned invalid data. Truncation occurs for example when the DNS zone is stored in an SQL database with a column length too short to accommodate the whole DNSKEY record. This was a typical error with older versions of PowerDNS.

TABLE VI
RSA EXPONENTS OF SECOND-LEVEL DOMAINS (JANUARY 2017).

Exponent	KSK	ZSK
3	339	370
65,337	39	96
65,535	14	7
65,537	4,693,400	6,950,090
$2^{30}+3$	31	18
$2^{32}+1$	8,927	7,925
Total:	4,702,750	6,958,506

TABLE VII
DSA KEY LENGTHS OF SECOND-LEVEL DOMAINS (JANUARY 2017).

Length	KSK	ZSK
512	5	3
768	1	1
1024	3,561	3,695
Total:	3,567	3,699

TABLE VIII
VALIDATION RESULT OF SECOND-LEVEL DOMAINS.

Result	Count
No DNSKEY (dangling DS)	19,386
No trusted DNSKEY (dangling DS)	1,216
No RRSIG for trusted DNSKEY	380
Signature expired	1,799
Signature ahead of time	1
Signature verify failure	49
Validation failure	22,831
Validation success	5,092,022

The most common public RSA exponent is $e = 65\,537$, as shown in Table VI. The few occurrences of $e = 65\,337$ and $e = 65\,535$ are probably a typo but without security implications, as long as they are coprime with $\phi(n)$, where n is the RSA modulus.

The DSA key lengths are shown in Table VII. 1024 bits is the most common and maximum specified DSA key size.

C. Validation Result

All 5.1 million domains in this analysis ought to be signed, as this was indicated by the parent DS record set. We attempt to validate the DNSKEY record set by the trusted KSKs to determine whether the domains have a valid authentication chain. First, we check whether any DNSKEY matches any of the DS records in terms of key tag, algorithm number and fingerprint. Then, we check whether there is an RRSIG record created with any of the KSK that is authenticated by a parent DS record. We attempt to verify the signature of the RRSIG with the trusted KSKs and compare the validity period of the RRSIG with the time of retrieval of the DNSKEY response. If validation succeeds, all DNSKEY records are authentic and can be used for validation of other signatures of that zone.

Table VIII shows the validation results. For 19k domains (0.38%), the response was well-formed in principle but did not contain any DNSKEY record. This usually indicates a configuration error or lack of DNSSEC support on the server. 1216 responses (<0.1%) contained a DNSKEY record but none of the keys was authenticated by the parent DS record set. This is typically caused by an improper key rollover: the KSK is replaced in the DNSKEY set but the parent DS record remains unchanged. Automating key rollover and DS updates

would remedy this failure. In 380 cases there was a trusted KSK but the DNSKEY set was not signed by it. This is either caused by a lack of DNSSEC support, i.e. a legacy server returns the queried DNSKEY set but fails to include the corresponding RRSIG set, or by an integrity failure of the zone data, i.e. missing resource records.

1799 domains ($<0.1\%$) returned an expired response. In a few cases one server returned stale zone data while another server would have given us a valid response, which is a server synchronization failure. However, it was more common that the signatures have not been renewed, which could be avoided with automatic signing. The actual signature verification failed in only 49 cases. This shows that the reliability of DNSSEC depends in practice on operational issues and not on the implementation of cryptographic primitives. In total, 23k DNSKEY responses failed to validate correctly, i.e. DNSSEC has degraded the availability of 0.45% domains in this study. Compared to 2015, this is a slight improvement (0.6% validation failures). Note that the number of domains failing due to DNSSEC validation is smaller than the number of domains failing due to other reasons (25k), e.g. server timeouts.

D. Limitations

With our measurement method, the recursive resolver will attempt to query the authoritative servers of the second-level domain until it retrieves a well-formed DNSKEY response (or gives up). We instruct the resolver with the CD flag (checking disabled) to omit DNSSEC validation, as we perform the validation in our analysis tool. In case of a validation error, we do not attempt to check whether another redundant server would have given us a valid response.

We assume in our analysis that GOST signatures validate correctly, but did not check this due to lack of GOST support in our implementation. Another limitation is that we did not repeat the measurement on a regular basis and thus cannot evaluate key rollover intervals of second-level domains.

VI. Discussion

RSA key length: We have seen an extensive use of 2048-bit and 1024-bit RSA keys in practice. While 2048-bit RSA is considered sufficiently secure, the use of 1024-bit keys is debatable. NIST deprecated the use of 1024-bit RSA in 2013 and recommends to use ≥ 2048 bits, with replacement of the private key after 1 to 3 years [20]. BSI recommends 2000-bit RSA with a predicted conformance until the year 2021 [21]. Kleinjung et al. factored a 768-bit RSA modulus in 2009 [22] and estimated that similar academic efforts could factor 1024-bit RSA by the year 2020 [23]. Although 1024-bit RSA has not been broken in public yet, general consensus is that it does not provide enough security margin for future use.

A strategy to compensate for short key lengths is to replace the key more often, as we have observed for most TLDs. Although key lifetime is an essential parameter in a security assessment, it is hard for domain administrators to make a reasonable choice with a key length on the brink of being vulnerable. Furthermore, as key rollovers increase the system complexity, frequent rollovers contribute to a brittleness due to implementation bugs or operational mistakes. We thus suggest to use long-term keys with conservative key lengths, e.g. according to the recommendations by NIST or BSI.

DSA key length: Considering DSA key lengths, algorithms for computing the discrete logarithm in DSA are similar to RSA integer factorization, though require more computational effort [24]. Adrian et al. [25] estimate that computing the discrete logarithm with 768-bit finite groups is within range of academic teams and 1024-bit groups within range of state-level attackers. The maximum DSA key length specified in DNSSEC is 1024 bits [26], which does not provide enough security for future use. Changing the DNSSEC protocol to support larger key sizes would be possible but is not reasonable. A 1024-bit public DSA key is represented with all necessary parameters by 405 bytes, whereas a 1024-bit public RSA key with $e = 65\,537$ is represented by 132 bytes. DNS is sensitive to message size and DSA has an inefficient ratio of security level to key size in a DNSKEY record. Although DSA signatures are shorter than e.g. RSA signatures, we consider the size of the public key as more important because DNSKEY record sets are larger than the average address record set. We thus recommend to deprecate the use of DSA in DNSSEC.

Message length: The drawback of long keys is the message size constraint in DNSSEC, which uses UDP transport primarily. A growing message size increases the probability for interoperability problems due to path MTU or fragmentation issues. Domain administrators have the following solution approaches:

1) Use **elliptic-curve cryptosystems** instead of RSA [27]. DNSSEC specifies the use of elliptic-curve ciphers GOST, ECDSA P-256 and ECDSA P-384. The security level of ECDSA P-256 is comparable to 3072-bit RSA [28], while the public key is only 64 bytes long. The signature verification performance of ECDSA is lower than RSA, though.

2) Use only **one key pair** instead of two with the KSK/ZSK scheme. This not a viable option for operators of high-value domains like TLDs, where keys are stored separately for security reasons. However, if both the KSK and ZSK are stored at the same location, e.g. on a signing server, then there will be no security benefit from splitting the keys. An automated update method, e.g. [29], helps to keep the parent DS record in sync.

3) Rely on **TCP** instead of UDP. This increases the lookup latency and will not work with clients that do not support DNS-over-TCP. Although the majority of clients supports TCP [30], requiring TCP degrades the domain availability for legacy clients.

Automation of the signing process is advisable in general. As suggested in Section V-C, most validation failures are caused by dangling DS records or expired signatures. An automated signing and monitoring system is capable to avoid these failures in most cases.

VII. Data Sets

The complete statistics and the list of domains failing DNSSEC validation are available on our website[1]. The list of domain names and NSEC/3 records acquired during zone enumeration is available on request for academic researchers [31]. The DNS records received during the measurements have been fed to the Farsight Security passive DNS database [32].

VIII. Related Work

The SecSpider project [33] uses web crawling, manual user submissions and has used NSEC zone enumeration in the past to find DNSSEC-signed domains. In 2008, SecSpider located 871 signed domains, though most of them did not have a complete authentication chain [34]. Deccio et al. [35] used a similar approach and counted 2242 signed domains in 2010. As of February 2017, SecSpider tracks 1.6 million signed zones [36]. Dai et al. [5] surveyed the DNSSEC adoption among four TLDs and the Alexa top 1 million domains and found 35k signed second-level domains in mid 2016. As we have shown in this paper, we can use NSEC and NSEC3 zone enumeration [9] to provide a complete quantification of second-level domains.

Valenta et al. [37] surveyed the RSA key lengths in DNSSEC over time from domain list scans. The number of 512-bit keys found remained around 10k (0.35%) between 2014 and 2015.

Prior work has also considered the availability of domains [34], [35], [38], [39], including but not limited to validation failures of signed domains. Van Adrichem et al. [40] counted that 4% of signed domains show a form of DNSSEC misconfiguration, though with a broader metric than our validation failure metric. Not all misconfigurations result in a failure, especially when only part of the server responses are erroneous. Automated domain monitoring systems help to locate misconfigurations before they manifest in an outage. Fukuda et al. [41] assessed the impact that the `jp` TLD would have after a signing respectively validation failure: 18% of validating clients would fail to resolve domains below `jp` in the first 10 minutes after the incident.

Van Rijswijk-Deij et al. [6], [27] argue to use elliptic-curve cryptography to solve DNSSEC's message size problems. We support this conclusion and showed that ECDSA has found significant adoption within the last two years.

Prior research [42], [43] demonstrated the additional server costs that are associated with DNSSEC deployment. NSEC3 in particular can severely degrade the server throughput with high hashing workloads [44].

IX. Conclusions

In this paper, we contribute an NSEC and NSEC3 zone enumeration of all top-level domains. This allows us to quantify the complete number of second-level domains that adopted DNSSEC, which totals to 6.4 million. The adoption varies significantly between top-level domains: 10 TLDs account for 92% of all DNSSEC-signed second-level domains. This uneven distribution is caused by the effort of a few TLD operators to promote DNSSEC and provide deployment incentives.

The crawling of NSEC and NSEC3 records finished for all but two top-level domains: one had a broken NSEC chain and one re-salted the NSEC3 chain every 10 minutes, which would have required us to increase our query rate to retrieve the complete chain. This shows that frequent re-salting protects from NSEC3 zone enumeration, but at high signing costs and at the risk of provoking an attacker to indeed raise the query rate beyond good manners.

NSEC3 hashing hides a portion of cleartext names, but we were able to recover 79% of names with GPU-based NSEC3 hash breaking. One of our findings is that the hash iteration count of the TLD does not have a significant impact on the recovery ratio, but rather the quality of the dictionary used to break the hashes. Thus, server operators suffering from high CPU costs should consider lowering the NSEC3 iteration count.

RSA is the dominant signing algorithm, most often with 2048-bit Key Signing Key and 1024-bit Zone Signing Key. 512-bit RSA keys are still in circulation, and in fact the number of 512-bit Zone Signing Keys has risen within the last two years. We discourage from using short RSA keys to cope with message size limitations. Instead, operators should consider the use of only one combined KSK/ZSK with $\geq$ 2048-bit RSA, which is replaced on the scale of years. A viable alternative to RSA is 256-bit ECDSA, which has taken off from almost zero adoption in 2015 to 8% in 2017. Discrete DSA should not be used in the future for DNSSEC due to insufficient key lengths.

Domains failing DNSSEC validation are quite rare (0.45%), but future work should attempt to minimize this number with automated signing and monitoring. Zone enumeration turns out to be a useful debugging tool, as it helps to identify broken NSEC/3 chains or servers returning erroneous NSEC/3 responses on real zone data. For future deployment studies or other measurement efforts of top-level domains, we suggest to consider the Public Suffix List [16] to get a better picture of those domains with public registration below second-level domain suffixes.

[1] http://dnssec.vs.uni-due.de/domains/

References

[1] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034 (INTERNET STANDARD), Internet Engineering Task Force, Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936. [Online]. Available: http://www.ietf.org/rfc/rfc1034.txt

[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," RFC 4033 (Proposed Standard), Internet Engineering Task Force, Mar. 2005, updated by RFCs 6014, 6840. [Online]. Available: http://www.ietf.org/rfc/rfc4033.txt

[3] R. Barnes, "Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE)," RFC 6394 (Informational), Internet Engineering Task Force, Oct. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6394.txt

[4] A. Herzberg and H. Shulman, "Retrofitting Security into Network Protocols: The Case of DNSSEC," *IEEE Internet Computing*, vol. 18, no. 1, pp. 66–71, Jan 2014.

[5] T. Dai, H. Shulman, and M. Waidner, *DNSSEC Misconfigurations in Popular Domains*. Cham: Springer International Publishing, 2016, pp. 651–660. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-48965-0_43

[6] R. van Rijswijk-Deij, M. Jonker, and A. Sperotto, "On the Adoption of the Elliptic Curve Digital Signature Algorithm (ECDSA) in DNSSEC," in *2016 12th International Conference on Network and Service Management (CNSM)*, Oct 2016, pp. 258–262.

[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Resource Records for the DNS Security Extensions," RFC 4034 (Proposed Standard), Internet Engineering Task Force, Mar. 2005, updated by RFCs 4470, 6014, 6840, 6944. [Online]. Available: http://www.ietf.org/rfc/rfc4034.txt

[8] B. Laurie, G. Sisson, R. Arends, and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence," RFC 5155 (Proposed Standard), Internet Engineering Task Force, Mar. 2008, updated by RFCs 6840, 6944. [Online]. Available: http://www.ietf.org/rfc/rfc5155.txt

[9] M. Wander, L. Schwittmann, C. Boelmann, and T. Weis, "GPU-Based NSEC3 Hash Breaking," in *Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on*, Aug 2014, pp. 137–144.

[10] Internet Assigned Numbers Authority, "Root Zone File." [Online]. Available: http://www.iana.org/domains/root/files

[11] ICANN, "Removal of Eleven Test Internationalized Top-Level Domains from the Root Zone," https://www.icann.org/news/announcement-2-2013-10-02-en, Oct. 2013.

[12] ICANN, "Base Registry Agreement," http://newgtlds.icann.org/en/applicants/agb/base-agreement-contracting, Nov. 2013.

[13] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter, "Public Keys," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 626–642.

[14] Y. Oiwa, K. Kobara, and H. Watanabe, "A New Variant for an Attack Against RSA Signature Verification Using Parameter Field," in *Public Key Infrastructure*, ser. Lecture Notes in Computer Science, J. Lopez, P. Samarati, and J. Ferrer, Eds. Springer Berlin Heidelberg, 2007, vol. 4582, pp. 143–153. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73408-6_10

[15] ICANN, "DNSSEC workshop held at ICANN 45 meeting," Oct. 2012, http://toronto45.icann.org/node/34375.

[16] Mozilla Foundation, "Public Suffix List." [Online]. Available: https://publicsuffix.org/

[17] M. Wander, "The Impact of DNSSEC on the Internet Landscape," Ph.D. dissertation, University of Duisburg-Essen, 2015. [Online]. Available: http://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=38851

[18] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices," in *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*. Bellevue, WA: USENIX, 2012, pp. 205–220. [Online]. Available: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger

[19] D. J. Bernstein, "How to find smooth parts of integers," 2004. [Online]. Available: https://cr.yp.to/papers.html#smoothparts

[20] E. B. Barker, W. C. Barker, W. E. Burr, W. T. Polk, and M. E. Smid, "SP 800-57. Recommendation for Key Management, Part 1: General (Revisision 3)," National Institute of Standards & Technology, Tech. Rep., Jul. 2012.

[21] BSI TR-02102, "Kryptographische Verfahren: Empfehlungen und Schlüssellängen," Bundesamt für Sicherheit in der Informationstechnik, Tech. Rep., Feb. 2015.

[22] T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, E. Thomé, J. Bos, P. Gaudry, A. Kruppa, P. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev, and P. Zimmermann, "Factorization of a 768-bit RSA modulus," Cryptology ePrint Archive, Report 2010/006, 2010, http://eprint.iacr.org/.

[23] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery, "On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography," *IACR Cryptology ePrint Archive*, vol. 2009, p. 389, 2009.

[24] O. Schirokauer, D. Weber, and T. Denny, "Discrete Logarithms: The Effectiveness of the Index Calculus Method," in *Algorithmic Number Theory*, ser. Lecture Notes in Computer Science, H. Cohen, Ed. Springer Berlin Heidelberg, 1996, vol. 1122, pp. 337–361. [Online]. Available: http://dx.doi.org/10.1007/3-540-61581-4_66

[25] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta *et al.*, "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice," https://weakdh.org/imperfect-forward-secrecy.pdf, May 2015.

[26] D. Eastlake, "DSA KEYs and SIGs in the Domain Name System (DNS)," RFC 2536, Internet Engineering Task Force, Mar. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2536.txt

[27] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "Making the Case for Elliptic Curves in DNSSEC," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 13–19, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2831347.2831350

[28] P. Hoffman and W. Wijngaards, "Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC," RFC 6605, Internet Engineering Task Force, Apr. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6605.txt

[29] W. Kumari, O. Gudmundsson, and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance," RFC 7344, Internet Engineering Task Force, Sep. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7344.txt

[30] N. Weaver, C. Kreibich, B. Nechaev, and V. Paxson, "Implications of Netalyzr's DNS measurements," in *Proceedings of the First Workshop on Securing and Trusting Internet Names (SATIN), Teddington, United Kingdom*, 2011.

[31] Matthäus Wander, "Domain names with DNSSEC." [Online]. Available: http://imdc.datcat.org/collection/1-0728-B

[32] Farsight Security, "DNSDB — Passive DNS Database," https://www.farsightsecurity.com/DNSDB/.

[33] E. Osterweil, D. Massey, and L. Zhang, "Deploying and Monitoring DNS Security (DNSSEC)," in *Proceedings of the 2009 Annual Computer Security Applications Conference*, ser. ACSAC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 429–438. [Online]. Available: http://dx.doi.org/10.1109/ACSAC.2009.47

[34] E. Osterweil, M. Ryan, D. Massey, and L. Zhang, "Quantifying the Operational Status of the DNSSEC Deployment," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008, pp. 231–242. [Online]. Available: http://doi.acm.org/10.1145/1452520.1452548

[35] C. Deccio, J. Sedayao, K. Kant, and P. Mohapatra, "Quantifying and Improving DNSSEC Availability," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, Aug. 2011, pp. 1 –7.

[36] SecSpider, "Global DNSSEC deployment tracking," http://secspider.verisignlabs.com/, May 2016.

[37] L. Valenta, S. Cohney, A. Liao, J. Fried, S. Bodduluri, and N. Heninger, "Factoring as a Service," Cryptology ePrint Archive, Report 2015/1000, 2015, http://eprint.iacr.org/2015/1000.

[38] G. V. D. Broek, R. V. Rijswijk-Deij, A. Sperotto, and A. Pras, "Dnssec meets real world: dealing with unreachability caused by fragmentation," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 154–160, April 2014.

[39] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang, "Impact of Configuration Errors on DNS Robustness," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 319–330, Aug. 2004. [Online]. Available: http://doi.acm.org/10.1145/1030194.1015503

[40] N. L. M. van Adrichem, N. Blenn, A. R. Lúa, X. Wang, M. Wasif, F. Fatturrahman, and F. A. Kuipers, "A measurement study of dnssec misconfigurations," *Security Informatics*, vol. 4, no. 1, pp. 1–14, 2015. [Online]. Available: http://dx.doi.org/10.1186/s13388-015-0023-y

[41] K. Fukuda, S. Sato, and T. Mitamura, "Preliminary evaluation of potential impact of failure in dnssec validation," in *DNS EASY Workshop, 3rd Global Annual Symposium on DNS Security, Stability and Resiliency*, 2011.

[42] B. Ager, H. Dreger, and A. Feldmann, "Predicting the DNSSEC overhead using DNS traces," in *40th Annual Conference on Information Sciences and Systems*, March 2006, pp. 1484–1489.

[43] D. Migault, C. Girard, and M. Laurent, "A Performance view on DNSSEC migration," in *2010 International Conference on Network and Service Management*, Oct 2010, pp. 469–474.

[44] Y. Schaeffer, "NSEC3 Hash Performance," Mar. 2010, NLnet Labs document 2010-002.