**Title:** Retrofitting Post-Quantum Cryptography in Internet Protocols: A Case Study of DNSSEC

**Authors:** Moritz Müller, Jins de Jong, Maran van Heesch, Benno Overeinder, and Roland van Rijswijk-Deij

**Venue:** To be published in ACM SIGCOMM Computer Communication Review

# Retrofitting Post-Quantum Cryptography in Internet Protocols: A Case Study of DNSSEC

Moritz Müller
SIDN Labs and University of Twente
moritz.muller@sidn.nl

Jins de Jong
TNO
jins.dejong@tno.nl

Maran van Heesch
TNO
maran.vanheesch@tno.nl

Benno Overeinder
NLnet Labs
benno@nlnetlabs.nl

Roland van Rijswijk-Deij
NLnet Labs and University of Twente
r.m.vanrijswijk@utwente.nl

## ABSTRACT

Quantum computing is threatening *current* cryptography, especially the asymmetric algorithms used in many Internet protocols. More secure algorithms, colloquially referred to as Post-Quantum Cryptography (PQC), are under active development. These new algorithms differ significantly from current ones. They can have larger signatures or keys, and often require more computational power. This means we cannot just replace existing algorithms by PQC alternatives, but need to evaluate if they meet the requirements of the Internet protocols that rely on them.

In this paper we provide a case study, analyzing the impact of PQC on the Domain Name System (DNS) and its Security Extensions (DNSSEC). In its main role, DNS translates human-readable domain names to IP addresses and DNSSEC guarantees message integrity and authenticity. DNSSEC is particularly challenging to transition to PQC, since DNSSEC and its underlying transport protocols require small signatures and keys and efficient validation. We evaluate current candidate PQC signature algorithms in the third round of the NIST competition on their suitability for use in DNSSEC. We show that three algorithms, partially, meet DNSSEC's requirements but also show where and how we would still need to adapt DNSSEC. Thus, our research lays the foundation for making DNSSEC, and protocols with similar constraints ready for PQC.

## CCS CONCEPTS

• **Security and privacy** → *Digital signatures*; **Security protocols**;

## KEYWORDS

Security, DNS, Post-Quantum Cryptography

## 1 INTRODUCTION

Quantum computing has the potential to solve some computational problems that are currently considered infeasible for existing computers. This also includes problems that lay the foundation of current, state-of-the-art, public-key cryptography. With Shor's algorithm [54], future quantum computers can break *current* cryptographic algorithms such as RSA or Elliptic Curve Cryptography (ECC) in polynomial time, rendering them unusable. Today, many applications rely on these algorithms to provide message confidentiality and integrity, and authentication of the parties communicating. One example are the DNS Security Extensions (DNSSEC) that provide authenticity and integrity for messages exchanged in the Domain Name System (DNS). In its main capacity, DNS helps computers to translate human readable domain names like `example.com` to IP addresses like `93.184.216.34`. Without DNSSEC, recipients of the IP address cannot verify whether it has been tampered with, potentially misdirecting them to malicious content. Around 20% of Internet users rely on DNSSEC and adoption is rising [41]. Quantum computers threaten DNSSEC because insecure public-key cryptography could render DNSSEC ineffective.

Although a sufficiently powerful quantum computer that can break current public-key cryptography is not available yet, the field of quantum computing is evolving rapidly [8] and quantum algorithms that can be used to break cryptography are also being improved [32]. This means the need to replace conventional cryptography by quantum-safe alternatives is imminent. *Quantum-safe* algorithms are expected to neither be broken efficiently by today's computers nor by quantum computers. Even though experts expect it to take at least another ten to twenty years before the first quantum computers could break traditional algorithms [18], it is necessary to start transitioning already. Previous experience, such as the transition from 3DES to AES, teaches us that many years are needed to complete such a transition [49]. Since it is difficult to estimate the speed at which quantum computers will be developed, it is prudent to start as early as possible.

The National Institute of Standards and Technology (NIST) has initiated a process to test and standardize quantum-safe algorithms. Currently, four key encapsulation and three signing algorithms are evaluated in the third round of the process and are considered for standardization [3]. NIST expects to select candidates for standardization by early 2022. These quantum-safe algorithms differ in required computational resources for key generation, signing and validation, sizes of keys and signatures, as well as achieved security levels (we list these algorithms and their attributes in Table 3). From this it becomes clear, there will not be a single solution that fits all applications, establishing a need to examine which *quantum-safe* algorithms meet the requirements of existing security protocols.

In this paper, we discuss parameters to assess the readiness of existing security protocols for post-quantum cryptography. In particular, we study DNSSEC, a protocol that has not been assessed in the context of post-quantum cryptography before and for which concerns have been raised about the transition to PQC [52].

We use DNSSEC as a use case, because it has strict constraints on *(i)* message size and *(ii)* signature validation and generation throughput, both of which are challenges for many of the proposed quantum-safe algorithms. We analyze which algorithms *could*, at least partially, meet these requirements and propose potential changes to the DNSSEC protocol that can help find a middle ground between constraints of the quantum safe algorithms and of the protocol. Thereby, we take the first steps to prepare DNSSEC for post-quantum cryptography. These steps could also be applied to protocols with similar constraints and that rely on the same underlying transport protocol as DNSSEC (e.g. certain encapsulations of the Extensible Authentication Protocol (EAP) [1]).

## 2 RELATED WORK AND APPROACH

This is the first study that analyses the applicability of quantum-safe algorithms for protocols with strict constraints on signature length, focusing on message authentication, and the first that studies this for DNSSEC. Related work from Crockett et al. [20] applies quantum-safe algorithms to TLS and SSH and Heesch et al. [61] apply them to OpenVPN and HTTPS. None of these protocols, however, have the same constraints as DNSSEC. Van Rijswijk-Deij et al. [62] evaluate the performance of Elliptic Curve Cryptography in DNSSEC, but using PQC imposes additional size-requirements.

For our research, we derive the requirements of DNSSEC from standards [5–7], community best practices [27], our own active measurements covering a daily snapshot of the DNS for a representative set of over 220M domain names [63], and operational experience from running the Dutch ccTLD `.nl`.

For this study, we consider quantum-safe signing algorithms that are part of the third round of the NIST standardization process [3]. We consider both finalist and candidate algorithms – seven in total. Table 3 shows the key and signature sizes for each algorithm with estimated security level I [58, §4.A.5]. We first select candidate algorithms based on the signature size, meeting the requirements of DNSSEC explained in Section 4. Then, we measure the performance of the selected algorithms, using their optimized implementation as provided on the NIST website [59]. For each, we measure how many signatures we can create and verify in 10 seconds for a random message, repeat this 1,000 times, and report the mean performance. We choose a random 86-byte string as message to sign.[1] All measured algorithms rely on current hash functions (SHA256, SHAKE-256 and SHA3) to transform the signed records into a string with standard length. Therefore, the record size only affects the performance of the established hash-function and not the performance of the new signing algorithm itself. We perform the measurement on a single core of a machine equipped with an Intel Xeon Silver 4110 CPU (2.10GHz), 64GB RAM, running Ubuntu 18.04.3 LTS. In the interest of reproducibility, we make our measurement code public [46].

Implementations will likely be further optimized. For this reason, the performance metrics are only a rough estimate. The selection of suitable algorithms for DNSSEC will therefore mostly be based on key and signature size, which are inherent properties of the algorithms unlikely to change in the future.

We compare the record sizes and performance metrics with current algorithms that are commonly used or recommended for DNSSEC [66]: *RSA-2048* belonging to the most popular algorithm family in DNSSEC [63], *ECDSA-P256*, an elliptic curve algorithm, widely deployed because of its small signatures, and *EdDSA-Ed22519* an algorithm based on Edwards Curves which the IETF expects to become the future recommended default for DNSSEC [66]. We benchmark these reference algorithms using the integrated OpenSSL speed test on the same hardware as the PQC algorithms.

## 3 POST-QUANTUM CRYPTOGRAPHY

Post-quantum cryptography (PQC) is the group of algorithms that run on a classical computer and can withstand both a conventional and a quantum computer's attack. The 26 years since the invention of Shor's algorithm [54] dictate the time scale of main developments in the field. This is much less time than has been spent, e.g., on cryptanalysis of conventional public key algorithms such as RSA and ECC. For this reason, current standardization efforts (such as the NIST competition [3]) focus on standardizing quantum-safe algorithms based on multiple different mathematical problems. In this section we summarize some of the different approaches to PQC. Regarding the security of these algorithms, we note that many factors play a role in the cryptanalysis of new algorithms and discussing these in detail is out of scope for this document. Nevertheless, depending on the approach PQC schemes take, general observations can be made based on the current state of research.

There are currently five classes of PQC algorithms. Three of these (lattice-based, multivariate and hash-based) are considered for signature schemes in the NIST competition as finalist or alternate candidates [3] and we assess all three (see also Table 3). The specific algorithms chosen are the security level I variants, which corresponds in strength to a 128-bit classical key search [58]. This is equivalent in strength to commonly used 256-bit ECC keys and stronger than the current standard RSA-2048, which measures 112 bits in classical security [10] and is widely used for DNSSEC.

**The *multivariate* approach** (1980s) is based on systems of algebraic quadratic equations over finite fields. Typically, signature schemes are given by underdefined systems, meaning that there are several valid signatures for a public key. This is no problem as long as it is sufficiently difficult to find another valid signature. Although improvements to several attacks have been found recently [9, 43, 51], multivariate schemes have a good security track record. Generally, they have small signatures with fast verification.

***Lattice-based* cryptography** [2] (1996) builds on the hardness of finding short vectors in a high-dimensional lattice. It used to be impractical, but provably secure, or practical but with a security reduction. Newer schemes combine these and some have submitted provably secure signature schemes to the NIST standardization, such as qTesla-p-I. Like many cryptographic algorithms, they are vulnerable to side-channel attacks [34, 45, 53]. However, for DNSSEC this is not a major concern, since these attacks require physical access to signers. In general, lattice-based systems form good and allround algorithms with relatively small signatures and keys, combined with fast operations.

***Hash-based* signature schemes** build on the property that it is hard to find a pre-image (input message) for a certain digest or to find two elements with the same digest. A large advantage of these schemes is the solid security basis that only depends on the security of the chosen cryptographic hash function. For this reason, hash-based schemes are considered extremely conservative alternate candidates for standardization [3]. Typically, hash-based signature

[1] The median number of bytes covered by an `RRSIG` [7] of all signed `AAAA` records of domains in `.com` [63].

schemes have very small keys, large signatures and require significant computational overhead for signing and verification.

## 4 DNSSEC REQUIREMENTS

In this section, we explain the functionality of DNSSEC. From this, we derive requirements that modern DNSSEC set-ups demand from cryptographic algorithms. DNSSEC adds additional payload to DNS messages and requires additional computational operations. The choice of cryptographic algorithms has an influence on both.

### 4.1 DNSSEC

DNSSEC adds integrity and authenticity to the DNS. To achieve this, operators cryptographically sign information associated with their domain name (e.g. an IP address). The public key, signature and the signed information is published in resource records (RR) in a zone file at their authoritative name server.

Without DNSSEC, a resolver asking for e.g. a `AAAA` record would only need to query for the record itself. With DNSSEC, in addition to the requested record, a validating resolver also receives the corresponding signature (`RRSIG`) in the same response. Then, it sends an additional query, asking for the public key (`DNSKEY`) of the signed zone. This response is then also signed (two top rows in Table 1).

In some cases even multiple signatures are transmitted: If the requested information does not exist, then DNSSEC-signed zones provide the resolver with an *authenticated denial of existence* (NSEC(3)). The details of this proof are out of scope, but the response can contain three or more signatures [7, 44] (two bottom rows of Table 1).

When an operator uses multiple keys to sign a zone, a signature is attached for each key. This is for example the case during a key rollover, replacing one key with another. Also, operators can sign their zone with different algorithms, resulting in multiple signatures for each used algorithm and two or more `DNSKEY` records. Also, most zones split between a key signing the zone content (Zone-Signing-Key – ZSK) and a key signing just the keyset (Key-Signing-Key – KSK). As a consequence, queries for the public key contain both keys along with the signature (resulting in large messages).

After a validating resolver has fetched all necessary records, it can validate the signature. The result of the validation is *cached* as defined by the time-to-live (TTL) field in the signed RR. Only after the TTL has expired, will the results have to be validated again.

### 4.2 Cryptographic Requirements

The DNSSEC protocol allows adding new cryptographic algorithms relatively easily, and new algorithms have been proposed and integrated numerous times [38, 42, 55]. All algorithms, however, must adhere to boundaries and requirements set by the design and deployment of DNS, DNSSEC and the underlying transport protocols.

*Signature and key size.* Originally, DNS packets were limited to 512 bytes. With the introduction of the Extension Mechanisms for DNS (EDNS(0) [21]), this limit is, in theory raised to 64 kilobytes. Previous research and operational experience, however, have shown that sending large DNS packets is often problematic.

First, the maximum transmission unit (MTU) of the underlying networks can be a limiting factor. Packets larger than the MTU cause fragmentation or trigger a retransmission via TCP. In the best case, this causes additional round trip time (RTT) for transmitting the fragments or for establishing the TCP connection. In the worst case, fragments can never be transmitted and the TCP connection cannot be established because of interfering middle boxes or lack of support. As a consequence, end users, for example, are not able to visit their requested website. Van den Broek et al. [60] have shown that up to 10% of all resolvers might be unable to handle fragments.

Second, fragmented DNS responses can be misused to spoof the cache of recursive resolvers [35]. Both two problems, potential packet loss and the susceptibility to spoofing, encouraged DNS software developers and operators to recommend a maximum supported message size of 1,232 bytes [27].

Third, DNS is often misused in amplification attacks, where thousands of small queries from an attacker trigger large responses directed to a victim. The extra records DNSSEC adds to a response make this attack more effective [64]. With the introduction of elliptic curve based algorithms in DNSSEC, the signatures can be up to 64 bytes small, which partially mitigates this problem [65].

These three reasons lead us to conclude that small signatures are also preferred for quantum-safe algorithms and that *signatures* should not exceed 1,232 bytes. Signatures are transmitted in every DNSSEC message, for example every time an `A` or `AAAA` record is returned (around 55% of all queries [30]) or in response to a query for a non-existing record (around 15%). In the latter case, a response will even contain multiple signatures. Also, they are cached the shortest (see Table 1). Therefore, it is crucial that signatures are transmitted reliably, without the risk of packets being dropped or retransmitted. Signatures smaller than 1,232 bytes decrease these risks significantly. Preferably, even, signatures are far below this threshold leaving room for payload and multiple signatures. Public keys, on the other hand, need to be transmitted less frequently, so having larger keys may be acceptable. We explore this in Section 6.

*Validation.* Resolvers need to serve their clients as fast as possible. A medium size resolver today processes a few thousand queries per seconds resulting in a few hundred validations [62]. This is far below their maximum capacity. The underlying cryptographic libraries can validate thousands of signatures per second of current algorithms used in DNSSEC (see bottom of Table 3). The total number of DNSSEC-signed domain names is still rising and large resolvers likely need to validate ever more signatures. Therefore, we expect that at least 1,000 quantum-safe signatures should be validated per second in our evaluation. This is a conservative boundary and we can expect that future implementations and specialized hardware will also speed up post-quantum algorithms.

*Signing.* Zone operators sign records on five different occasions: (i) when the zone is signed for the first time, (ii) when the key is changed (*rolled*), (iii) when records change, (iv) when a signature expires or, (v) *on-the-fly*. The latter, obviously, is the most time critical. In this approach, signatures are created when a record is queried. This is for example necessary when records are created dynamically depending on the querying resolver and requires signing in milliseconds. This setup is usually only used at CDNs (e.g. Cloudflare [19]); typical operators only re-sign records when they change or when a key rollover takes place. The frequency depends on the zone. Zones of top-level-domains like `.com` and `.nl` change frequently. E.g. every time a new domain is registered new records

| Response Type | RRs in response | RRs added by DNSSEC (covered RR) | Alexa 1M median TTL (mean) |
|---|---|---|---|
| AAAA | ≥ 1 AAAA | 1 RRSIG (AAAA) | 5 min (0.6 h) |
| DNSKEY | ≥ 1 DNSKEY | 1 RRSIG (DNSKEY) | 60 min (8.3 h) |
| Non-existent domain (with NSEC) | SOA | 1 RRSIG (SOA)<br>2 NSEC<br>2 RRSIG (NSEC) | 60 min (2.0 h) |
| NSEC3 Closest-encloser proof (§5.5 of [33]) | SOA | 1 RRSIG (SOA)<br>≥ 3 NSEC3<br>≥ 3 RRSIG (NSEC3) | 10 min (2.8 h) |

**Table 1: Records added by DNSSEC and the median time they are cached of the 1M most popular domains [4].**

| Prio | Requirement | Good | Accepted Conditionally |
|---|---|---|---|
| #1 | Signature Size | ≤ 1,232 bytes | – |
| #2 | Validation Speed | ≥ 1,000 sig/s | – |
| #3 | Key Size | ≤ 64 kilobytes | > 64 kilobytes |
| #4 | Signing Speed | ≥ 100 sig/s | – |

**Table 2: Requirements for quantum-safe algorithms.**

need to be signed. For *.nl*, zone files are published every 30 minutes, typically requiring around 11,000 new signatures to be created.

To support signing of larger zones, frequent zone file publication, and additional overhead, suitable quantum-safe algorithms must at least be capable of creating 100 signatures per second. Slower algorithms might be acceptable for zones that are less prone to change. For on-the-fly signing, obviously, higher signing speeds are required.

*Requirements summary.* The size of signatures is the most important criterion when selecting an algorithm, followed by the time it takes to validate signatures. Only if signatures can be transferred reliably between name server and resolver and the resolvers can validate the signatures timely, the basic protocol of DNSSEC can stay unchanged. The requirements are summarized in Table 2. The third column shows the requirements that we expect algorithms to fulfill and which are marked in blue. Under some circumstances or with some modification of the DNS protocol higher boundaries might be acceptable. These are listed in the last column, marked in orange and are discussed in Section 6.

## 5 EVALUATING ALGORITHMS

The previous section shows that signature size is the most crucial requirement. We mark the attributes of algorithms that fully or partially fulfil each requirement in blue or orange respectively and use this encoding also in Table 3. Attributes that do not fulfil the requirements are marked in pink. We pre-select *aspirant* algorithms that create signatures ≤ 1,232 bytes. This leaves us with three algorithms: Falcon-512, RedGeMSS128, and Rainbow-$I_a$ (marked light gray in Table 3). For those, we additionally evaluate signing and validation performance.

The remaining algorithms create signatures larger than 1,232 bytes. Their reliable transmission cannot be guaranteed and they make DNSSEC more attractive as an amplifier in a DDoS attack. For this reason, we do not consider them for DNSSEC any further.

*Falcon-512.* Falcon [31] is a signature scheme based on NTRU-lattices [39]. It stands out as a computationally efficient algorithm, with an optimized implementation already available. Falcon-512 has the smallest pair of public key and signature, which is particularly relevant for the DNSSEC case. It is the only algorithm where both signatures and public keys fall within the size limit, although both keys and signatures are considerably larger than current non-PQC DNSSEC algorithms. This may still cause problems during transmission, since it is neither possible to ship more than one key at a time, nor to ship more than one signature, or even only one signature and a payload that exceeds 523 bytes. In our test-bed, the performance of Falcon-512 is closest to the current algorithms and meets the requirements of DNSSEC. Further performance improvements are possible using a hardware FPU, AVX2 and FMA opcodes [31].
Its implementation and level-$I$ security strength are delicate; conversely more testing is required to gain trust in its security. NIST currently expects either Falcon or Crystals-Dilithium to be standardized as the primary post-quantum signature scheme at the conclusion of the third round [3].

*Rainbow-$I_a$.* Rainbow-$I_a$ [24] is a multivariate scheme. It is based on the Unbalanced Oil and Vinegar (UOV) scheme [50]. The signature size of Rainbow-$I_a$ matches the sizes of current recommended algorithms based on elliptic curves and is therefore a good fit for DNSSEC. The public keys, however, are significantly larger and do not fit in DNS packets. As with the signature size, the performance of Rainbow-$I_a$ is comparable to current algorithms and meets the requirements. Its performance can be improved further with AVX2 instructions. A version with a reduced public key size is Cyclic Rainbow, but this comes at the cost of an increase in computational requirements. We note that the adoption of Rainbow-$I_a$ could be hindered by royalties[25].

*RedGeMSS128.* GeMSS [15] is a multivariate signature scheme of the Hidden Field Equation type. RedGeMSS128 produces the smallest signatures in the GeMSS family, at security level $I$ even smaller than EdDSA-Ed22519. The public key, however, exceeds the maximum record size of the DNS. First measurements indicate GeMSS signs considerably slower than current algorithms. The usage of SSE2, SSE3 and the AVX2 CPU instructions could improve performance [15]. If new insights show that Rainbow is unacceptable, GeMSS forms an alternate candidate for standardization [3].

| Algorithm | NIST Verdict | Approach | Private key | Public key | Signature | Sign/s | Verify/s |
|---|---|---|---|---|---|---|---|
| Crystals-Dilithium-II [29] | Finalist | Lattice | 2.8kB | 1.2kB | 2.0kB | | |
| Falcon-512 [31] | Finalist | Lattice | 57kB | 0.9kB | 0.7kB | 3,307 | 20,228 |
| Rainbow-$I_a$ [56] | Finalist | Multivariate | 101kB | 158kB | 66B | 8,332 | 11,065 |
| RedGeMSS128 [16] | Candidate | Multivariate | 16B | 375kB | 35B | 545 | 10,365 |
| Sphincs$^+$-Haraka-128s [11] | Candidate | Hash | 64B | 32B | 8kB | | |
| Picnic-L1-FS [17] | Candidate | Hash | 16B | 32B | 34kB | | |
| Picnic2-L1-FS [17] | Candidate | Hash | 16B | 32B | 14kB | | |
| EdDSA-Ed22519 [12] | | Elliptic curve | 64B | 32B | 64B | 25,935 | 7,954 |
| ECDSA-P256 [12] | | Elliptic curve | 96B | 64B | 64B | 40,509 | 13,078 |
| RSA-2048 [12] | | Prime | 2kB | 0.3kB | 0.3kB | 1,485 | 49,367 |

**Table 3: Signature algorithms in round three of the NIST competition [3] (security level I). DNSSEC candidate algorithms are shaded gray. Attributes meeting DNSSEC's requirements fully or partially are marked blue or orange, others in pink.**

## 6 DISCUSSION

The previous section shows that no algorithm fits all requirements perfectly. Falcon, Rainbow and GeMSS come closest, but each has shortcomings: Falcon-512 *technically* meets all requirements but its larger signatures may cause problems, e.g. during rollovers, and make DNSSEC an even more attractive tool for DDoS attacks. In comparison, signatures of Rainbow-$I_a$ and RedGeMSS128 are on par with current recommended algorithms, but their public keys go beyond the supported payload size. All algorithms perform signing and validation fast enough for today's use cases.

We therefore expect changes to the DNSSEC protocol are required before PQC algorithms can be deployed. We now sketch what changes may be needed, setting an agenda for future research.

### 6.1 Increased TCP support

The greatest bottleneck to deploying Falcon-512 is the large size of keys and signatures. Operators can reduce the size of the key set by relying on CSKs (Combined-Signing-Key – combining the ZSK and KSK), but signed messages might still exceed the threshold of 1,232 bytes in case of larger payloads or if multiple algorithms are used. Nevertheless, keys and signatures could still be safely transmitted using TCP. Today, not every name server supports TCP: we still observe 11% of name servers lacking TCP support [63]. Two developments, however, could help decrease this.

DNS Flag Day [27] is a recurring initiative by software vendors and operators. In 2020, it promotes, among others, the support of TCP. The previous flag day, promoting the support of EDNS, had a positive impact [57], and we expect the same for the upcoming. Also, *encrypted* DNS could increase TCP support. DNS-over-TLS (DoT) [40] and DNS-over-HTTPS (DoH) [36] both rely on TCP as transport and see some traction already [22]. TCP mitigates the threat of DDoS amplification attacks but requires more resources at recursive resolvers and name servers and its impact still needs to be thoroughly measured.

### 6.2 Out-of-band key distribution

Increased TCP support is still not sufficient for transmitting the *public key* of Rainbow and GeMSS, since both exceed the maximum DNS payload size of 64 kbytes [23].

This problem can be solved in two ways, both modifying the existing DNSKEY RR. One approach is to divide the public key into chunks small enough that they can be transmitted in one RR. Each chunk is published at a new label of the signed domain and chained with each other. The initial DNSKEY RR would then refer to the first chunk of the actual public key. The advantage of this approach is that it can likely be implemented in a manner that is backward compatible to existing implementations. The disadvantage, however, is that resolvers would need to send multiple queries to fetch a key, increasing the risk of transmission failures.

Alternatively, we propose transmitting the key *out-of-band*. Instead of directly providing the resolver with the key through a DNS record, name servers could serve a URI, instructing the resolver to fetch the key from a web server using HTTP. Because of the *chain of trust*, resolvers can still use the public key of the root to verify the key published on the web server. Resolvers not supporting this mechanism would already today either consider the zone not signed (*insecure*) or fall back to a supported algorithm if the zone is signed with multiple algorithms. Because of the higher TTL (see Table 1), an out-of-band transmission of DNSKEY RRs would only occur occasionally. This approach comes with two caveats: first, resolvers would need to support HTTP to fetch the key and second, zone-operators would need to maintain a web server. Whereas the former might be addressed by the rise of DoH (see previous section), the latter might be an additional barrier for operators rolling out DNSSEC and could create additional potential points of failure.

### 6.3 Performance

The two aforementioned measures address the challenges of large keys and signatures. If, however, it turns out that the candidate algorithms are not secure and faster hardware not affordable, then it might be necessary to use algorithms *too slow* for current DNSSEC deployments. One workaround might lay in the fact that resolvers only need to validate signatures if the signed record is not cached. If validating signatures is an expensive operation, decreasing the number of validations may be a solution. AAAA records of the 1M most popular domain names have a median TTL of 5 minutes (see Table 1). Increasing the TTL of these RRs to 1 hour would already reduce the workload for resolvers 12 times. Note, however that we expect that optimized implementations and specialized hardware could improve performance, rendering higher TTLs unnecessary.

### 6.4 Other Considerations

*Algorithms for high-security zones.* In this paper, we only considered PQC algorithms at security level I (128-bit security). In the future, however, some zones may have stronger requirements. Consider, for example, the DNS root zone, which has very long-lived keys – the root KSK was only changed for the first time 8 years after its introduction [47]. This long lifetime may increase the risk of a successful attack against the key and may thus require choosing schemes with higher security levels. Fortunately, the remaining PQC algorithms in the third round of the NIST competition leave room for this. For example, RedGeMSS256 offers security level V with very modest signatures (76B). The public key, however, is significantly larger at 3135kB making changes to the way keys are distributed in DNSSEC inevitable.

*Alternatives to DNSSEC.* The measures described above show that quantum-safe algorithms *can* be applied to DNSSEC, but not without additional effort. Therefore, one could propose to abandon DNSSEC altogether and to find other solutions to guarantee authenticity in the DNS. DNS-over-TLS and DNS-over-HTTPS, for example, rely on TLS and HTTPS and earlier studies have shown that both can support quantum-safe algorithms [20, 61]. On the other hand, neither provides a full replacement for DNSSEC and the trust model is different, making it impossible to realise some of the newer applications of DNSSEC such as DANE [37]. Other alternatives, like DNSCrypt, claim to provide quantum-safe implementations [28], but deployment of DNSCrypt has not gained significant traction, and this seems unlikely to change in the future as there is not IETF specification for the protocol.

### 6.5 Transitioning to PQC

Especially in the early days of their development, trust in new PQC algorithms may still be low. Instead of only signing records with a quantum-safe algorithm, a combination of a conventional and a PQC algorithm can be used. This *hybrid* model [14] is especially valuable for long-lived signatures and keys, since there is a greater risk that an attack against one of the algorithms is successful over their lifetime. Our data [63], shows the average lifetime of a signature is around 34 days (median 21 days), making the risk small. Keys, however, can be much longer lived and replacing a crucial key, such as the one for the DNS root, is non-trivial and takes time [47].

The problem with a hybrid model is that it requires signing with two algorithms concurrently. While this is possible within the specifications of the DNSSEC protocol, it doubles the number of keys and signatures. Realistically, this means that such a model can only be deployed within the constraints discussed in Section 4 if both the conventional and the PQC algorithm have small signatures. The best combination would in this case be an elliptic curve algorithm (e.g. ECDSA P-256) with either Rainbow-$I_a$ or RedGeMSS128.

## 7 NEXT STEPS

We have identified three algorithms, currently under evaluation in the NIST competition, that show great potential to be applied in DNSSEC: *Falcon 512*, *Rainbow-$I_a$*, and *RedGeMSS128*. Falcon, in principle, could even be deployed in DNSSEC without protocol modifications, but still has the shortcoming of significantly larger keys and signatures than current algorithms. To address these and other challenges, we have proposed extensions and modifications that could make DNSSEC ready for quantum-safe algorithms. These algorithms may also be a fit for protocols with similar strict requirements on key and signature sizes.

Nevertheless, we need to keep in mind that standardizing quantum-safe algorithms for DNSSEC and getting them deployed takes time. If NIST standardizes one or more algorithms, they still need to be standardized in the IETF for the use in DNSSEC. Even for a rather uncontroversial algorithm like EdDSA-Ed22519, this effort took almost a year [55]. Fourteen months after its standardization we see the first resolvers supporting this algorithm, from roughly 10,000 vantage points [48]. Today, and more than two years later, still 30% of observed validating resolvers lack support and more than 99% of 7M signed domains do not use this algorithm [63]. Furthermore, the DNS root is still signed with an algorithm that was standardized more than 10 years ago, and even though its key has been successfully replaced for the first time in 2018 [47] it is still not clear when the algorithm gets updated.

From these experiences we conclude that making DNSSEC fit for quantum-safe algorithms needs to start as soon as possible; the results in this paper can help make a start to this process.

We will continue observing the standardization process and adapt our recommendations if necessary. Already during the course of writing this paper new developments influenced our algorithm selection: the LUOV [13] scheme, which also creates small signatures suitable for DNSSEC, was considered not secure enough anymore after an attack was published [26] and dropped out after the second round. NIST, however, finds its approach still promising and we will assess if future implementations might be a fit for DNSSEC as well. Meanwhile, we plan to evaluate our candidate algorithms and suggested modifications in practice, taking the next step towards retrofitting PQC in DNSSEC.

## ACKNOWLEDGMENTS

The authors would like to thank (in alphabetical order): Bart Gijsen, Chris Faber, Jelte Jansen, our shepherd Marco Mellia, Niek Willems, Steve Uhlig and the anonymous CCR reviewers. This work was partially funded by EU H2020 CONCORDIA (#830927).

[1] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz (Ed.). 2004. Extensible Authentication Protocol (EAP). RFC 3748 (Proposed Standard). , 67 pages. https://doi.org/10.17487/RFC3748 Updated by RFCs 5247, 7057.

[2] M. Ajtai. 1996. Generating Hard Instances of Lattice Problems (Extended Abstract). In *In Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*. ACM, 99–108.

[3] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. 2020. *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process.* US Department of Commerce, National Institute of Standards and Technology.

[4] Alexa. 2020. Top 1M sites from 2020-07-14. http://s3.dualstack.us-east-1.amazonaws.com/alexa-static/top-1m.csv.zip.

[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. DNS Security Introduction and Requirements. RFC 4033 (Proposed Standard). , 21 pages. https://doi.org/10.17487/RFC4033 Updated by RFCs 6014, 6840.

[6] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. Protocol Modifications for the DNS Security Extensions. RFC 4035 (Proposed Standard). , 53 pages.

https://doi.org/10.17487/RFC4035 Updated by RFCs 4470, 6014, 6840, 8198.

[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard). , 29 pages. https://doi.org/10.17487/RFC4034 Updated by RFCs 4470, 6014, 6840, 6944.

[8] F. Arute, K. Arya, and R. et al Babbush. 2019. Quantum Supremacy Using a Programmable Superconducting Processor. *Nature* 574 (oct 2019), 505–510.

[9] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. 2020. Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems. arXiv:cs.CR/2002.08322

[10] Elaine Barker. 2020. *Recommendation for Key Management: Part 1 – General.* Special Publication 800-57 P.1 Rev.5.

[11] D.J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.L. Gazdag, A. Hülsing, P. Kampanakis, S. Kölbl, T. Lange, M.M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, and P. Schwabe. 2017. *Sphincs+.* https://sphincs.org/data/sphincs+-specification.pdf.

[12] D. J. Bernstein and Tanja Lange. 2020. eBACS: ECRYPT Benchmarking of Cryptographic Systems. https://bench.cr.yp.to/results-sign.html. Skylake (506e3)-sand.

[13] W. Beullens, B. Preneel, A. Szepieniec, and F. Vercauteren. 2019. *LUOV; Signature Scheme proposal for NIST PQC Project.*

[14] Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila. 2017. Transitioning to a Quantum-Resistant Public Key Infrastructure. In *Post-Quantum Cryptography*, Tanja Lange and Tsuyoshi Takagi (Eds.). Springer International Publishing, Cham, 384–405.

[15] A. Casanova, J.C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. 2019. *GeMSS: A Great Multivariate Short Signature.* www-polsys.lip6.fr/Links/NIST/GeMSS{_}specification{_}round2.pdf.

[16] A. Casanova, J.C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. 2020. *GeMSS: A Great Multivariate Short Signature.* https://www-polsys.lip6.fr/Links/NIST/changes{_}round2{_}V2.pdf.

[17] M. Chase, D. Derler, S. Goldfeder, J. Katz, V. Kolesnikov, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, X. Wang, and G. Zaverucha. 2019. *The Picnic Signature Scheme, Design Document v2.1.* https://github.com/microsoft/Picnic/blob/master/spec/design-v2.1.pdf.

[18] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. 2016. *Report on Post-Quantum Cryptography.* US Department of Commerce, National Institute of Standards and Technology.

[19] Cloudflare. 2019. ECDSA: The missing piece of DNSSEC. https://www.cloudflare.com/dns/dnssec/ecdsa-and-dnssec/.

[20] Eric Crockett, Christian Paquin, and Douglas Stebila. 2019. Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH. In *NIST 2nd Post-Quantum Cryptography Standardization Conference 2019.*

[21] J. Damas, M. Graff, and P. Vixie. 2013. Extension Mechanisms for DNS (EDNS(0)). RFC 6891 (Internet Standard). , 16 pages. https://doi.org/10.17487/RFC6891

[22] Casey Deccio and Jacob Davis. 2019. DNS privacy in practice and preparation. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies.* 138–143.

[23] J. Dickinson, S. Dickinson, R. Bellis, A. Mankin, and D. Wessels. 2016. DNS Transport over TCP - Implementation Requirements. RFC 7766 (Proposed Standard). , 19 pages. https://doi.org/10.17487/RFC7766 Updated by RFC 8490.

[24] J. Ding, M.S. Chen, A. Petzoldt, D. Schmidt, and B.Y. Yang. [n. d.]. *Rainbow.* https://http://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/Rainbow-Round2.zip.

[25] J. Ding, M.-S. Chen, Albrecht Petzoldt, and B.-Y. Schmidt, Dieter Yang. 2017. Intellectual property statements Rainbow. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/ip-statements/Rainbow-Statements.pdf.

[26] Jintai Ding, Joshua Deaton, Kurt Schmidt, Vishakha, and Zheng Zhang. 2019. Cryptanalysis of The Lifted Unbalanced Oil Vinegar Signature Scheme. *IACR Cryptol. ePrint Arch.* 2019 (2019), 1490.

[27] DNS-Violations. 2020. DNS flag day 2020 . https://dnsflagday.net/2020/.

[28] DNSCrypt. 2020. DNSCrypt: Frequently Asked Questions. https://dnscrypt.info/faq.

[29] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. 2019. *CRYSTALS-Dilithium, Algorithm Specifications and Supporting Documentation.* https://pq-crystals.org/dilithium/data/dilithium-specification-round2.pdf.

[30] Pawel Foremski, Oliver Gasser, and Giovane CM Moura. 2019. DNS Observatory: The Big Picture of the DNS. In *Proceedings of the Internet Measurement Conference.* 87–100.

[31] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. [n. d.]. *Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU.* https://falcon-sign.info/falcon.pdf.

[32] Craig Gidney and Martin Ekerå. 2019. How to Factor 2048 bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits. arXiv:1905.09749

[33] R. Gieben and W. Mekking. 2014. Authenticated Denial of Existence in the DNS. RFC 7129 (Informational). , 30 pages. https://doi.org/10.17487/RFC7129

[34] L. Groot Bruinderink, A. Hülsing, T. Lange, and Y. Yarom. 2016. Flush, Gauss, and Reload – A Cache Attack on the BLISS Lattice-Based Signature Scheme. Springer Berlin Heidelberg, 323–345.

[35] A. Herzberg and H. Shulman. 2013. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *2013 IEEE Conference on Communications and Network Security (CNS).* 224–232.

[36] P. Hoffman and P. McManus. 2018. DNS Queries over HTTPS (DoH). RFC 8484 (Proposed Standard). , 21 pages. https://doi.org/10.17487/RFC8484

[37] P. Hoffman and J. Schlyter. 2012. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698 (Proposed Standard). , 37 pages. https://doi.org/10.17487/RFC6698 Updated by RFCs 7218, 7671.

[38] P. Hoffman and W.C.A. Wijngaards. 2012. Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC. RFC 6605 (Proposed Standard). , 8 pages. https://doi.org/10.17487/RFC6605

[39] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. 1998. NTRU: A Ring-Based Public Key Cryptosystem. In *Lecture Notes in Computer Science.* Springer-Verlag, 267–288.

[40] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. 2016. Specification for DNS over Transport Layer Security (TLS). RFC 7858 (Proposed Standard). , 19 pages. https://doi.org/10.17487/RFC7858 Updated by RFC 8310.

[41] Huston, Geoff. 2019. The State of DNSSEC Validation. https://blog.apnic.net/2019/03/14/the-state-of-dnssec-validation/.

[42] J. Jansen. 2009. Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC. RFC 5702 (Proposed Standard). , 10 pages. https://doi.org/10.17487/RFC5702 Updated by RFC 6944.

[43] D. Kales and G. Zaverucha. 2019. Forgery Attacks on MQDSSv2.0. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/official-comments/MQDSS-round2-official-comment.pdf.

[44] B. Laurie, G. Sisson, R. Arends, and D. Blacka. 2008. DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. RFC 5155 (Proposed Standard). , 52 pages. https://doi.org/10.17487/RFC5155 Updated by RFCs 6840, 6944.

[45] Sarah McCarthy, James Howe, Neil Smyth, Seamus Brannigan, and Máire O'Neill. 2019. BEARZ Attack FALCON: Implementation Attacks with Countermeasures on the FALCON signature scheme. Cryptology ePrint Archive, Report 2019/478. https://eprint.iacr.org/2019/478.

[46] Moritz Müller, Jins de Jong, Maran van Heesch, Benno Overeinder, and Roland van Rijswijk Deij. 2020. Measurement Code: Custom Performance Test. https://github.com/SIDN/pqc-dnssec-measurement.

[47] Moritz Müller, Matthew Thomas, Duane Wessels, Wes Hardaker, Taejoong Chung, Willem Toorop, and Roland van Rijswijk-Deij. 2019. Roll, Roll, Roll your Root: A Comprehensive Analysis of the First Ever DNSSEC Root KSK Rollover. In *Proceedings of the Internet Measurement Conference.* 1–14.

[48] NLnet Labs. 2020. DNSThought Live Measurements. https://dnsthought.nlnetlabs.nl/.

[49] D. Ott, C. Peikert, and other workshop participants. 2019. Identifying Research Challenges in Post Quantum Cryptography Migration and Cryptographic Agility. arXiv:1909.07353

[50] J. Patarin. 1997. The Oil and Vinegar Signature Scheme. https://ci.nii.ac.jp/naid/10028073837/en/. *Dagstuhl Workshop on Cryptography* (Sep 1997).

[51] Ray Perlner and Daniel Smith-Tone. 2020. Rainbow Band Separation is Better than we Thought. Cryptology ePrint Archive, Report 2020/702. https://eprint.iacr.org/2020/702.

[52] Rod Rasmussen. 2019. SSAC Comment to NIST on Quantum Cryptography Algorithms. https://www.icann.org/en/system/files/files/sac-107-en.pdf.

[53] P. Ravi, M. Prasad Jhanwar, J. Howe, A. Chattopadhyay, and S. Bhasin. 2018. Side-channel Assisted Existential Forgery Attack on Dilithium - A NIST PQC candidate. Cryptology ePrint Archive, Report 2018/821. https://eprint.iacr.org/2018/821.

[54] Peter W Shor. 1994. Polynomial Time Algorithms for Discrete Logarithms and Factoring on a Quantum Computer. In *International Algorithmic Number Theory Symposium.* Springer, 289–289.

[55] O. Sury and R. Edmonds. 2017. Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC. RFC 8080 (Proposed Standard). , 7 pages. https://doi.org/10.17487/RFC8080

[56] Rainbow team. 2020. *Modified Parameters of Rainbow in Response to a Refined Analysis of the Rainbow Band Separation Attack by the NIST Team and the Recent New MinRank attacks.* https://sites.google.com/site/jintaiding/nist-papers.

[57] Willem Toorop, Moritz Müller, and Taejoong Chung. 2019. Measuring the impact of DNS Flag Day. https://blog.apnic.net/2019/08/19/measuring-the-impact-of-dns-flag-day/.

[58] National Institute of Standards US Department of Commerce and Technology (NIST). 2016. Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[59] National Institute of Standards US Department of Commerce and Technology (NIST). 2020. Post-Quantum Cryptography: Round 2 Submissions. https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[60] Gijs Van Den Broek, Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. 2014. DNSSEC meets real world: dealing with unreachability caused by fragmentation.

*IEEE communications magazine* 52, 4 (2014), 154–160.

[61] M. van Heesch, N. van Adrichem, T. Attema, and T. Veugen. 2019. Towards Quantum-Safe VPNs and Internet. Cryptology ePrint Archive, Report 2019/1277. (2019). https://eprint.iacr.org/2019/1277.

[62] R. van Rijswijk-Deij, K. Hageman, A. Sperotto, and A. Pras. 2017. The Performance Impact of Elliptic Curve Cryptography on DNSSEC Validation. *IEEE/ACM Transactions on Networking* 25, 2 (April 2017), 738–750. https://doi.org/10.1109/TNET.2016.2605767

[63] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. 2016. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1877–1888.

[64] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. 2014. DNSSEC and Its Potential for DDoS Attacks. In *Proceedings of ACM IMC 2014*. ACM Press, Vancouver, BC, Canada. https://doi.org/10.1145/2663716.2663731

[65] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. 2015. Making the Case for Elliptic Curves in DNSSEC. *ACM SIGCOMM Computer Communication Review* 45, 5 (2015), 13–19.

[66] Paul Wouters and Ondřej Surý. 2019. Algorithm Implementation Requirements and Usage Guidance for DNSSEC. RFC 8624. https://doi.org/10.17487/RFC8624