Double-Signed Fragmented DNSSEC for Countering Quantum Threat

Syed W. Shah, Lei Pan

Deakin Cyber Research and Innovation Centre, Australia
Cyber Security Cooperative Research Centre, Australia
syed.shah,l.pan@deakin.edu.au

Dinh Duc Nha Nguyen, Robin Doss

Deakin Cyber Research and Innovation Centre, Australia
Cyber Security Cooperative Research Centre, Australia
dinh.nguyen,robin.doss@deakin.edu.au

Warren Armstrong

Quintessence Labs
Canberra, Australia
wa@quintessencelabs.com

Praveen Gauravaram

Tata Consultancy Services (TCS)
Brisbane, Australia
p.gauravaram@tcs.com

ABSTRACT

DNSSEC, a DNS security extension, is essential to accurately translating domain names to IP addresses. Digital signatures provide the foundation for this reliable translation, however, the evolution of *Quantum Computers* has made traditional digital signatures vulnerable. In light of this, NIST has recently selected potential *post-quantum* digital signatures that can operate on conventional computers and resist attacks made with *Quantum Computers*. Since these *post-quantum* digital signatures are still in their early stages of development, replacing pre-quantum digital signature schemes in DNSSEC with post-quantum candidates is risky until the post-quantum candidates have undergone a thorough security analysis. Given this, herein, we investigate the viability of employing *Double-Signatures* in DNSSEC, combining a post-quantum digital signature and a classic one. The rationale is that double-signatures will offer protection against quantum threats on conventional signature schemes as well as unknown non-quantum attacks on post-quantum signature schemes, hence even if one fails the other provides security guarantees. However, the inclusion of two signatures in the DNSSEC response message doesn't bode well with the maximum allowed size of DNSSEC responses (i.e., 1232B, a limitation enforced by MTU of physical links). To counter this issue, we leverage a way to do application-layer fragmentation of DNSSEC responses with two signatures. We implement our solution on top of OQS-BIND and through experiments show that the addition of two signatures in DNSSEC and application-layer fragmentation of all relevant resource records and their reassembly does not have any substantial impact on the efficiency of the resolution process and thus is suitable for the interim period at least until the quantum computers are fully realized.

KEYWORDS

PQC, post-quantum DNSSEC, digital signature, double-signatures

1 INTRODUCTION

The Domain Name System (DNS) is responsible for translating human-readable domain names into machine-understandable IP addresses. As can be imagined, any vulnerability in this system can potentially make such translation precarious, leading users to malicious servers instead of their intended destinations. The Security Extension of DNS - i.e., Domain Name System Security Extensions (DNSSEC) helps establish that the messages received by a client are from an authorized DNS server and have not been altered in transit. DNSSEC primarily leverages *digital signatures* for establishing the aforementioned properties of *authenticity* and *integrity*. However, the underlying security assumptions of conventional digital signature algorithms - i.e., the computational difficulty of integer factorization and discrete logarithms will not hold when an attacker has access to a Cryptographically Relevant Quantum Computer (CRQC) [1, 25]. To mitigate this problem, the National Institute of Standards and Technology (NIST) has recently shortlisted three post-quantum digital signatures - i.e., FALCON [23], CRYSTALS-DILITHIUM [13], and SPHINCS+ [9]. An important envisioned property of these post-quantum digital signature algorithms is that they can run on classical computers but also withstand the attacks conducted using CRQCs, thereby offering sufficient protection against the impending threat of CRQCs. For DNSSEC to continue offering the needed *authenticity* and *integrity* in the post-quantum era, it must be updated with post-quantum digital signatures. However, for successful transitioning to the post-quantum era, the *Interim Period* - i.e., from now until CRQCs are fully realized or until the security of post-quantum candidates is fully established, presents a particular challenge. Precisely, in preparation for the post-quantum era, simply switching pre-quantum cryptography (such as conventional digital signatures in DNSSEC) with post-quantum cryptography (such as the post-quantum candidates shortlisted by NIST) in the interim period may lower overall security [6]. The reason is that the shortlisted post-quantum candidates (both key encapsulation mechanisms and digital signatures) have yet not gone through a thorough cryptanalysis against both classical and quantum attacks. The shortlisted post-quantum candidates may be compromised even

using conventional classical computers in the near future. Therefore, there is an inherent risk in simply switching from pre-quantum cryptography to post-quantum cryptography alone in that overall security may be lowered not only against CRQCs but also against today's classical computers (see further details in Section 3).

Aligned with the aforementioned discussion and the recommendations from the European Union Agency For Cybersecurity (ENISA), we in this paper investigate the plausibility of *Double-Signed DNSSEC* by combining the pre-quantum and post-quantum digital signatures for the interim period. However, the size of signatures and public keys of post-quantum candidates shortlisted by NIST is very large (i.e., shortlisted post-quantum signatures are 11× to 122× and public keys are 14× to 20× of their pre-quantum counterparts). These sizes further increase when double signatures are incorporated into the DNSSEC resolution process to ensure authenticity and integrity. This increased size of signatures and public keys significantly impacts the DNSSEC resolution process [10]. Precisely, since UDP is the preferred transport protocol for DNSSEC because of it being lightweight, any response message that exceeds 1232B triggers IP-level fragmentation which is unreliable. The maximum size of DNSSEC response messages - i.e., 1232B is obtained as: (IPv6 MTU=1280)-(IPv6 Header=40)-(UPD Header=8). With double signatures, the sizes of DNSSEC response messages that carry different resource records for facilitating reliable address resolutions are much larger than the threshold size of 1232B. Therefore, these large DNSSEC messages necessitate IP fragmentation which is unreliable over UDP (i.e., some fragments may never reach the resolver) [28]. Due to this issue, the standard DNSSEC offers a fallback mechanism via TCP for sending large response messages without resorting to fragmentation. Precisely, when the response size exceeds the threshold (i.e., the resolver's advertised EDNS(0) buffer size), name servers send back a truncated response (i.e., by setting the TC flag), indicating the resolver to retry via TCP (and thus negatively contributing to the address resolution time). However, many name servers lack support for TCP in addition to high-resolution time with TCP due to it being not as lightweight as UDP [16, 28, 29].

To circumvent the aforementioned issues, a potential alternative is to perform the fragmentation at the application layer (i.e., within the DNS protocol itself). While there exist ways to tackle the application layer fragmentation (see [8, 24]), none of them can accommodate double signatures and public keys. Therefore, to accomplish this, first, we constructed a Docker-based testbed on an Amazon EC2 instance using commercial grade DNS software - i.e., *BIND9* (OQS-BIND, see full details on our testbed in Section 5). Second, we explored generating two signatures (pre-quantum and post-quantum) on the *Name Server* using the default BIND9 tools. Third, we devised a way to fragment the DNSSEC responses with two signatures and public keys on the name servers and accurately reassemble them on the resolver. We also modified the source code of the BIND9 resolver to enable the verification of both signatures before marking the response as authenticated. Finally, using our testbed, we conducted an empirical analysis to show that our implementation can handle two signatures in DNSSEC simultaneously and quantify the impact of double signatures on the DNSSEC resolution process. Our experiments show that double signatures have a negligible impact on the average resolution time of DNSSEC compared to only post-quantum digital signatures. Therefore, this analysis confirms the efficacy of double signatures in DNSSEC for the interim period until CRQCs are fully realized. Our main contributions are summarized as follows:

- We developed a fully functional Docker-based DNSSEC testbed over an Amazon EC2 instance using the widely used BIND9 software that can handle both pre-quantum and post-quantum signatures and public keys simultaneously over UDP in a single DNSSEC response message.
- We thoroughly investigated the source code for BIND9 and extensively modified the resolver component to enable the verifications of both pre-quantum and post-quantum digital signatures at the same time.
- Through empirical analysis we show that the addition of double signatures in the DNSSEC resolution process did not affect the performance. Hence, it is recommended for the interim period until CRQCs are fully realized.

The rest of the paper is organized as follows. Section 2 presents a brief overview of DNS, while Section 3 discusses the overall quantum threat and the motivation for using double signatures in DNSSEC. Section 4 presents the details of the maximum message size issue with double signatures and how to handle them in the resolution process. Section 5 details our test setup and experimental methodology. Results are discussed in Section 6 and related works and concluding remarks are contained in Sections 7 and 8, respectively.

2 DOMAIN NAME SYSTEM (DNS)

The Domain Name System (DNS) is a distributed database that plays a pivotal role in the functioning of today's Internet by facilitating the translation of human-readable domain names into corresponding machine-understandable IP addresses. In general, DNS is divided into various *zones* that correspond to a particular level of granularity in the aforementioned translation process. These *zones* contain various *types* of *resource record* (RR) that correspond to *labels*. These *resource records* are generally used to look up the IP addresses of domain names, name servers of various zones, and various other types of data needed for address translation.

For an abstract-level understanding of the DNS translation process, imagine a client who queries the IP address of a domain name *example.com*. For this translation, the client will first send a query to a *(caching) resolver* which will handle all the subsequent translation process on the client's behalf. Assuming that the resolver has not previously cached this answer (i.e., the IP address of *example.com*), it will query the *root server* for the name server of *.com* domain names. Once the resolver receives the response, it queries this name server for the name server(s) of *example.com*. Finally, once the resolver knows the corresponding name server, it queries it for the IP address of domain name *example.com* and returns it to the client.

As specified in RFC 1035, the generic DNS messages exchanged between the resolver and name servers have five sections [17]. Figure 1 shows an overview of a generic DNS message. The *header* section is always present and its structure is depicted in Figure 2. The *Question* section encapsulates the resolver's query whose IP address needs to be resolved. This section has three main fields - i.e., *QName* (contains encoded query), *QType* (mentions the type

of needed resource recorded in response), and *QClass* (contains the class of query, mostly specifies IN). As shown in Figure 1, the last three sections have a similar format - i.e., concatenation of various RRs. The RRs are actual data entries in the DNS database that provide information related to the queried domain name and help the actual translation of domain names to corresponding IP addresses. The top-level format of RRs is shown below in Figure 3. Going back to our earlier example of *example.com*, an RR with *Name* 'example.com' and *TYPE* 'A' will contain an IP address of 32 bits in the RDATA field, and with *TYPE* 'NS', RDATA field will contain the domain name of the authoritative server for *example.com*. The *Answer* section in a response message contains RRs that answer the query from the resolver, while the *Authority* section contains RRs that point to the authoritative server. The *Additional* section contains RRs that somehow relate to the query but may not essentially contain the answers to the query.
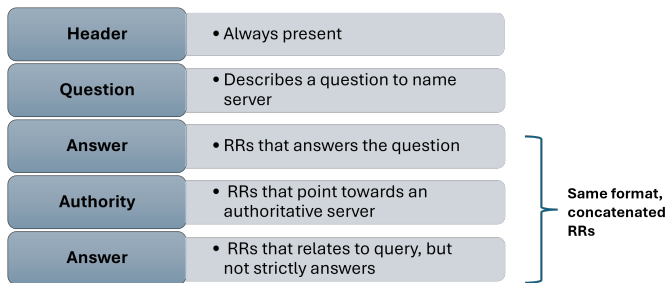

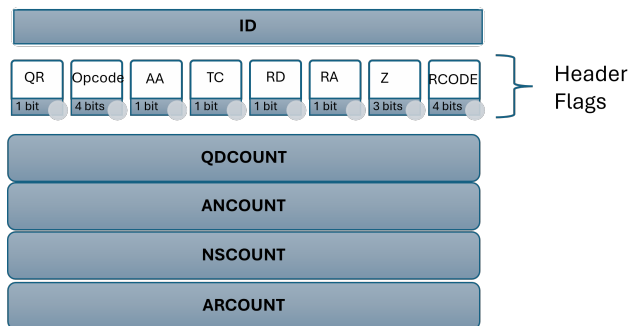**Figure 1: Generic DNS Message Format**

Figure 2: DNS Header

As discussed in Section 1, DNSSEC leverages digital signatures to ensure source authenticity and data integrity in DNS. The resource record labels are not required to be unique in DNSSEC resource records of a particular type. The RRs of a specific type and label are grouped into a *resource record set* (RRSet) that is signed using a specified digital signature algorithm. The corresponding digital signature is stored in *RRSIG*, and the corresponding public key is published in *DNSKEY*. In DNSSEC, for operational benefits, generally, two types of key pairs are generated - i.e., Zone Signing Keys (ZSK) and Key Signing Keys (KSK) (see details in RFC 6781 [12]). The ZSK pair is used for signing and verifying the resource records,
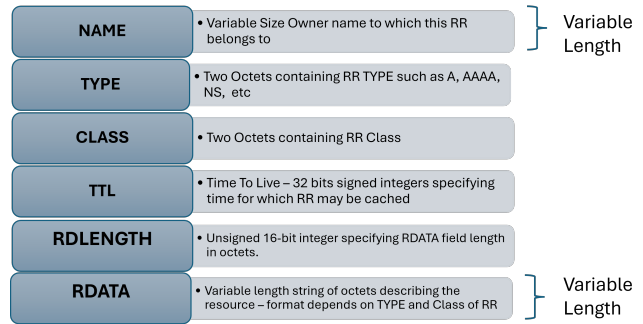

Figure 3: RR's Top-Level Format

while the KSK pair is used for signing the ZSK and helps enable the *chain of trust* in DNSSEC.

As explained above, the client initiates the translation process by querying the root server, then its children, and then its children's children, and finally reaching the name server that provides the queried IP address. This hierarchical process facilitates an important element of DNSSEC - i.e., the chain of trust that helps establish trust in public keys needed for signature verification. For the chain of trust, each zone that is queried as part of the resolution process must have a digest of public KSK kept as the *Delegation Signer* resource record in its parent zone; otherwise, the public ZSK sent by the name server cannot be trusted. Since the root server(s) has no parent zone, it does not publish any *delegation signer* (DS) resource record. Therefore, for the entire chain of trust, the public KSK of the root server is obtained out-of-band so that the root server's public KSK is pre-installed on most operating systems to eliminate the need for fetching and configuring the key(s). An overview of the chain of trust is shown in Figure 4.

Since RRSIG and DNSKEY RRs are the root cause of the maximum message size issue related to the double signatures (see discussion in the introduction), below we present an overview of RRSIG and DNSKEY RRs (see Figures 5, 6). Their thorough understanding is pivotal for the simultaneous fragmentation of two signatures at the application layer. We refer the reader to RFC 4034 for a more detailed discussion on these relevant RRs [2].

3 QUANTUM THREAT & QUANTUM-SAFE DNSSEC

In this section, we present an overview of the impending quantum threat, its repercussions for DNSSEC, and the problems associated with sticking to conventional digital signatures or switching to post-quantum digital signatures in DNSSEC. Then, we present a discussion on our proposed quantum-safe DNSSEC.

3.1 Quantum Threat

The assurances of source authenticity and data integrity in DNSSEC are underpinned by digital signatures. However, these conventional digital signatures used in DNSSEC rely on the complexity and difficulty of computational problems (e.g., integer factorization and discrete logarithms) which are difficult for attackers to solve using today's classical computers [21, 22]. There exists strong evidence
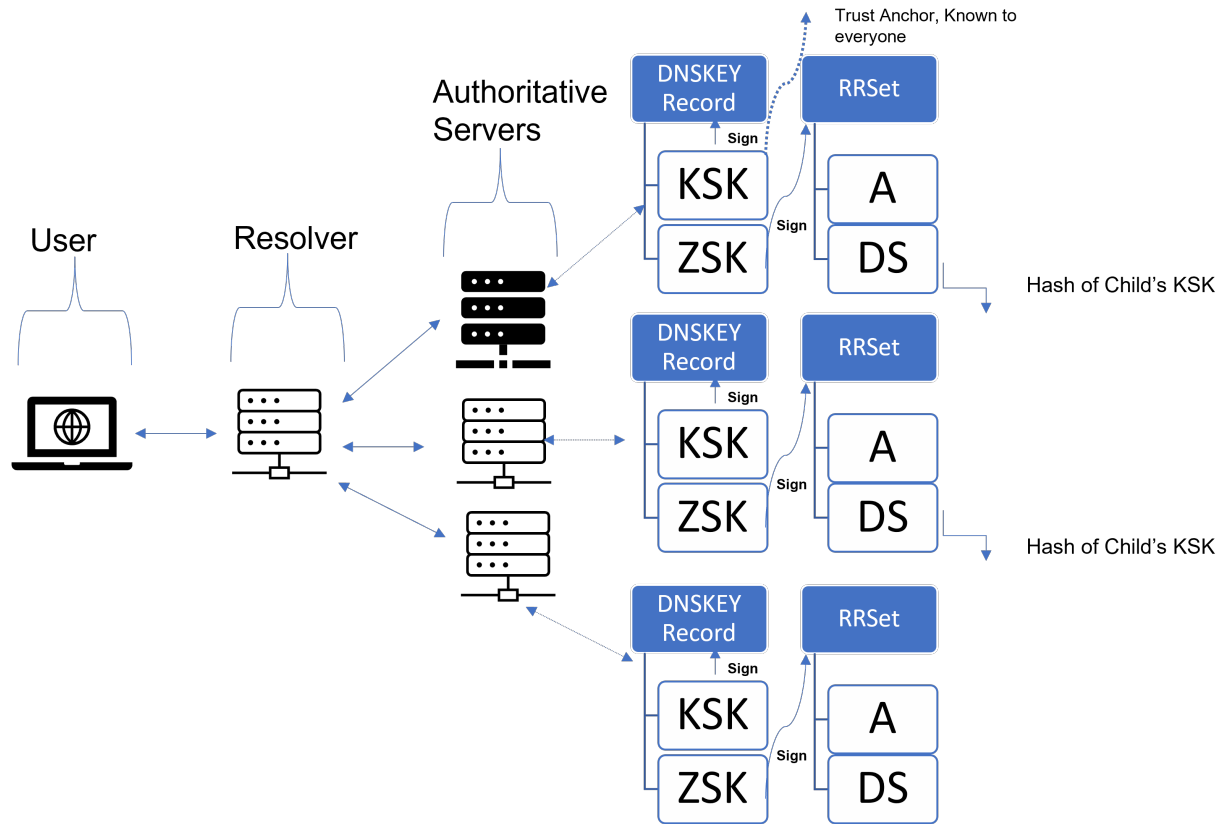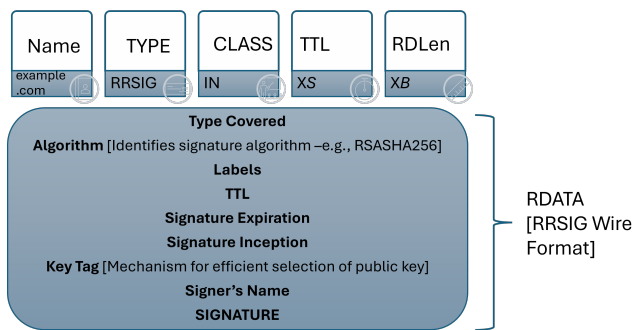
Figure 4: Chain-of-Trust in DNSSEC

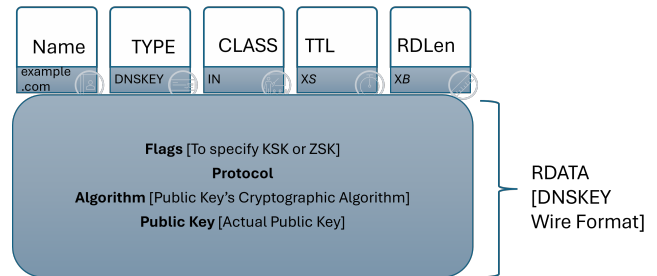**Figure 5: RRSIG RR Wire Format**

Figure 6: DNSKEY RR Wire Format

that these "computationally hard problems" will become solvable in "polynomial time" with the quantum computational model using the so-called CRQCs [1, 18]. These CRQCs will facilitate the implementation of Shor's Algorithm that will help solve the aforementioned computational problems in polynomial time [25], thereby rendering the conventional digital signatures insecure. For DNSSEC, it means that in the post-quantum era (i.e., when CRQCs will be fully realized), it would be difficult to establish the integrity and authenticity of all resource records exchanged as a part of the address resolution process. Therefore, in the post-quantum era, it will become impossible to trust the final IP address of a queried domain as the conventional digital signature will no longer offer the source authenticity and data integrity at an adequate level.

Given the aforementioned impending issues with conventional digital signatures (and also with existing key encapsulation mechanisms), NIST is currently undertaking a Post Quantum Cryptography (PQC) standardization project. It aims to standardize digital signature algorithms and key encapsulation mechanisms that can run on today's classical computers but can withstand the attacks conducted using the CRQCs (if they are realized in the future).

In this pursuit, NIST has recently shortlisted three post-quantum digital signature algorithms, including FALCON [23], CRYSTALS-DILITHIUM [13], and SPHINCS++ [9]. To make DNSSEC (or in general any other internet protocol) quantum-safe, it must be updated with the aforementioned post-quantum digital signature algorithms and be investigated for any issues that demand protocol-level changes to DNSSEC for the post-quantum era. However, simply switching the conventional digital signatures with post-quantum algorithms may be precarious. The reason is that the post-quantum digital signature algorithms as shortlisted by NIST are currently in their infancy and thus have been studied and analyzed for a shorter period (as compared with their classical counterparts). First, there could be flaws in the algorithm design itself that make them vulnerable to cryptanalytic attacks. Second, there could be side-channels in the different implementations that might leak keys via timing attacks, power analysis attacks, and other similar attacks. Third, there could be an algorithmic breakthrough that shows how the underlying problem can be solved efficiently not only on a CRQC but on a classical computer as well. Therefore, as envisioned by the European Union Agency For Cybersecurity (ENISA) in its recent report on PQC (see details in [6]), the cryptanalysts may have overlooked severe attacks on post-quantum candidates (digital signatures or key encapsulation mechanisms) that could be quickly conducted using today's classical computers. In view of this, simply updating global networking protocols such as DNSSEC on a large scale with post-quantum digital signatures algorithms may be problematic because any attacks on these post-quantum candidates revealed in the near future will render the DNSSEC-based address resolution process untrustworthy. Conversely, keeping the conventional digital signature algorithms intact in DNSSEC (and not integrating post-quantum digital signatures) can also be problematic because CRQC inception may not be publicly announced (especially when any state or a nefarious actor builds CRQC sooner than anticipated). In such a situation, the sole reliance on conventional digital signatures for DNSSEC will make data integrity and source authentication unattainable. Therefore, the quantum threat as envisioned above makes a challenging and complex situation as of today - i.e., either sticking only to conventional digital signatures or switching only to post-quantum digital signatures can potentially have severe security consequences.

3.2 Quantum-Safe DNSSEC

Simply replacing the conventional digital signatures with post-quantum candidates mentioned above may not make the DNSSEC quantum-safe and may even make DNSSEC susceptible to attacks conducted using today's classical computer (see the details mentioned above). Therefore, according to ENISA in [6], a better and more cautious way to proceed is to use post-quantum digital signatures as an extra layer of protection in conjunction with conventional digital signatures. The combination essentially leads to a *double-signed* DNSSEC (note that, the terms double-signed, combiners, or hybrid cryptosystems are used interchangeably). A well-designed double-signed DNSSEC will offer the benefit that it will remain *strong* if at least one of the used digital signatures is secure against a specific attack. In a double-signed DNSSEC, we will accept the IP address in a resolution process only if both signatures are verified. In particular, any attack revealed on post-quantum digital signatures in the near future that can be conducted with today's classic computer will almost always fail to comprise the address resolution process as the security of conventional digital signatures is established and assured against classical computers. Similarly, if the post-quantum digital signatures remain secure and no attacks can be conducted upon them, they will be secure against CRQC-driven attacks. Therefore, the double-signed approach is best suited for the *interim period*, as otherwise (i.e., with conventional or post-quantum digital signatures alone), DNSSEC may unknowingly be susceptible to attacks in the interim period as mentioned above.

As far as the double signing is concerned, existing literature specifies two common interfaces [6]. The first one, referred to as *detached-message interface* simply concatenates a pre-quantum signature X with a post-quantum signature Y. The verifier only accepts the results when the X verifier accepts the X signature and the Y verifier accepts the Y signature. The second interface, referred to as *signed-message interface*, typically signs with a pre-quantum algorithm to get the first signature X and then signs the result - i.e., X with the post-quantum algorithm to get the final signature (e.g, Z). The double-signature verifier in this interface normally applies the post-quantum verifier first followed by the pre-quantum verifier (or vice versa depending upon the signing agreement).

For the double-signed DNSSEC, we in this paper adopt an approach akin to the detached-message interface. Our reason is that this approach is relatively easier than others to integrate into our BIND9-based setup (see details on evaluation setup in Section 5). The overview of our double-signed DNSSEC is shown in Figure. 7. It essentially integrates two signatures - i.e., pre-quantum and post-quantum in the signed zone files on the authoritative server. Both these signatures are verified independently on the resolver before accepting any received resource record as verified and trusted to offer enhanced security.

4 DOUBLE-SIGNATURES AND MAXIMUM MESSAGE SIZE ISSUE

In this section, we discuss the message size issue when double signatures are incorporated in DNSSEC along with the fragmentation approach that we have devised and tested for accommodating the dual signatures in DNSSEC.

Messge Size Limitations

Since DNS(SEC) is an application-layer protocol, the generated messages are first placed in UDP packets, which subsequently are placed in IP packets that make the payload of link layer frames. The size of these payloads has a limit that is dictated by the Maximum Transmission Unit (MTU) of the physical links that these frames traverse. The size of the frame's payload greater than the link's MTU generally enforces routers to fragment these frames into smaller IP packets that travel separately to the resolver. However, the underlying transport protocol - i.e., UDP is mainly unreliable for successful delivery of all these fragments and thus may lead to failure of address resolution. To circumvent this issue, either we need to resort to a reliable transport protocol - i.e., TCP or we need to formulate messages such that their size is less than the MTU of the majority of physical links over which these DNS(SEC) messages
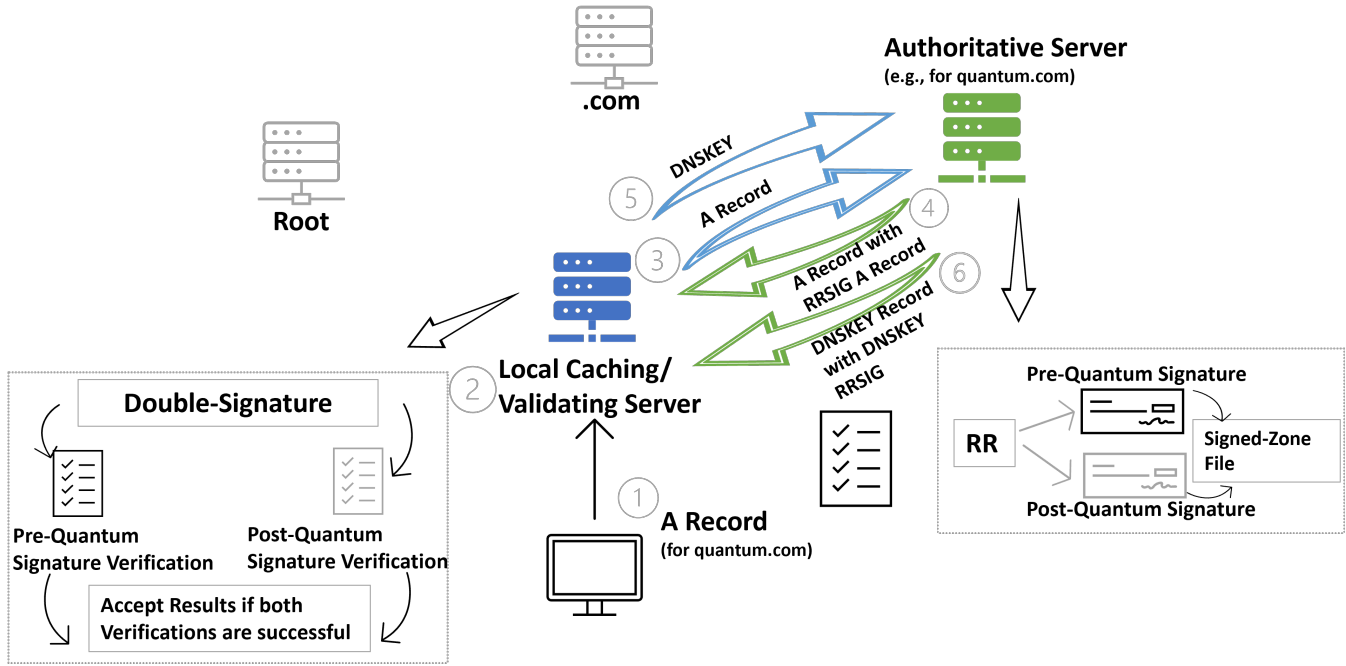
Figure 7: An Overview of Double-Signed DNSSEC

will likely travel. Since as per our discussion in Section 1, TCP has support issues in existing DNS infrastructure and also is not as lightweight as UDP [16, 28, 29], the only option is to restrict DNS message size such that IP-level fragmentation is not triggered. To accomplish this, initial RFCs on DNS [17] specify the maximum DNS message size to be less than (or equal to) 512B. However, the Extension Mechanism for DNS - i.e., EDNS(0) as defined in RFC 6891 [5], theoretically allows a maximum message size of up to 64KB. Abstractly, this mechanism introduces a pseudo-RR called OPTION (alluded as OPT) and allows resolvers to specify buffer size in the 'Class' field of this pseudo-RR in the 'additional section' of an RR (see RR format in Fig. 3). Therefore, through EDNS(0) it is theoretically possible to increase DNS message size to a maximum of XB if the MTU of the end-to-end path between resolver and name server is at least 'X+8(UDP Header)+40(IPv6 Header). However, as of now, there exists no standard way to estimate the MTU of end-to-end links. Therefore, taking the cautious approach, the maximum DNS message size is limited to 1232B so as to avoid IP-level fragmentation and frequent TCP re-transmission [15] (assuming MTU of 1280B).

The above-mentioned limit on the maximum DNS message size (i.e., 1232B) creates a particular challenge for double-signed DNSSEC as envisioned in this work. Precisely, any combination of pre-quantum and post-quantum signatures in double-signed DNSSEC will lead to response messages of size significantly larger than the limit discussed above. For example, our investigation shows that the message in response to a query of 'Type A' (see address resolution process depicted in Figure 7) is generally the smallest in size. However, our thorough analysis suggests that double signed DNSSEC with pre- and post-quantum algorithms that lead to the signatures

of the smallest sizes (e.g., pre-quantum ECDSA of 64B and post-quantum FALCON512 of 690B), has an approximate response to this query of size 2500B (note that exact size depends upon variable fields in an RR). The other messages such as those that carry 'DNSKEY RRs' are even larger. Therefore, to accommodate double signatures in DNSSEC, we need to fragment these large responses at the application layer - i.e., within the DNS(SEC) protocol itself.

Fragmentation at the Application Layer

There exist ways to fragment DNSSEC messages at the application layer in the post-quantum era. For example, ARRF (A Resource Record Fragment) mechanism [8] proposes a *request-based* fragmentation approach at the application layer. For response messages larger than the threshold limit, ARRF on name servers 'truncate' the first fragments indicating to the resolver that there are additional fragments before this response can be re-assembled and considered complete. To facilitate the resolver in efficient reception of fragments, ARRF introduces a new pseudo-RR called Resource Record Fragments (RRFRAGS, akin to OPT described above). This pseudo-RR adopts the format of a typical RR as shown in Figure 3, however, its fields are customized such that the name server can indicate to the resolver in the first fragment how the complete response would look like once all the fragments are received. Likewise, it also facilitates name servers in ascertaining whether the resolver requests a fragment (or is it a normal query). Although ARRF seems to perform fragmentation reliably, it has two main shortcomings. The first one is that it relies on a non-standard RR that will potentially cause issues with the middleboxes. Additionally, as acknowledged by the authors of ARRF, this approach may also be susceptible to memory exhaustion attacks.
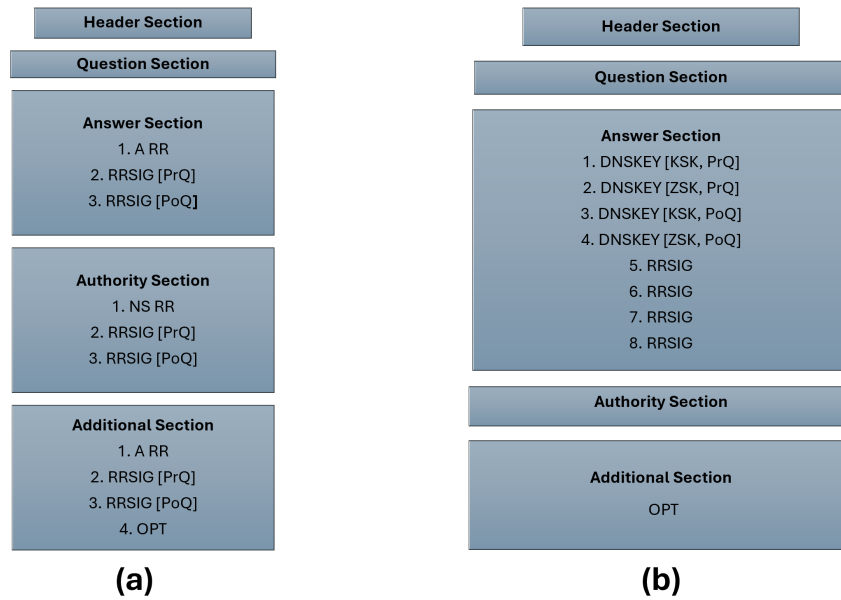
Figure 8: Structure of (a) 'A' DNSSEC Response (b) DNSKEY DNSSEC Response (PrQ - Pre quantum, PoQ - Post quantum)

Given the above-mentioned problems with ARRF, QName-Based Fragmentation (QBF) is proposed in [24]. Like ARRF, QBF also adopts a request-based fragmentation approach but leverages standard RRs, unlike ARRF which relies on customized RRs for its operations. When the response size is bigger than the threshold, QBF sends a fraction of the signature or public key in the first fragment and sets the 'TC' flag in the header of the response to '1', indicating to the resolver that this response is part of a fragmented message. Using the first fragment, the resolver deduces the size of the actual response and makes explicit requests (e.g., in parallel) for all other fragments (which can be deduced from the first fragments). To indicate to the name server that a particular query is for a fragment of a previous query, the resolver makes subsequent queries as '?n?domain_name', where '?' is a delimiter (a non-valid character for a domain, indicating that query or response is a fragment), 'n' is the number of the fragment, and 'domain_name' is the domain name of the actual query. Once the 'n' received by the resolver gets equal to the total number of fragments deduced by the resolver from the first fragment, all the fragments are re-assembled into the complete response and then verified.

Fragmentation with Double-Signatures

The issue with both aforementioned approaches is that none of them can accommodate double signatures as envisioned in this work for the security benefits discussed in Section 3. To accommodate two signatures while performing fragmentation at the application layer, herein, we devise an approach akin to QBF since it adopts standard RRs as discussed above. Fragmentation with double signatures poses a challenge because using a combination of pre- and post-quantum signatures and public keys results in 'A' and 'DNSKEY' responses that make it difficult for the name servers to create a first fragment with sufficient information. This first fragment is

needed by resolvers to determine how many more fragments to expect before the response can be re-assembled and verified.

As per our analysis, the typical responses involved in address resolution with double signatures look something like those shown in Figure 8. Since the sizes of pre-quantum signatures and public keys (e.g., for ECDSA256, RSASHA256) are much shorter than post-quantum candidates (i.e., FALCON, DILITHIUM, SPHINCS+) it is inefficient to fragment both types of RRSIGs and DNSKEYs included in typical DNSSEC responses. Therefore, we ensure that entire pre-quantum RRSIGs and DNSKEYs are always sent in the first fragment. However, our observation is that this approach works well when we are dealing with the responses to 'QTYPE' of 'A', leaving sufficient space in the first fragment to include a fraction of post-quantum RRSIGs. The inclusion of the entire pre-quantum and a fraction of post-quantum RRSIG in the first fragment helps the resolver in estimating the number of subsequent fragments and making explicit requests for them in 'parallel' for efficiency reasons. However, our thorough analysis of the sizes of different response messages that typically make part of the address resolution process suggests that this fragmentation approach does not bode well with DNSKEY responses in certain instances (such as when RSASHA256 is used as a pre-quantum signature in double-signed DNSSEC). For example, the size of each field of all sections of a DNSKEY response to a query for 'test.example' is shown in Figure 9. It can be observed that the total size of this response is 4462B. Note that, this response is when RSASHA256 is combined with FALCON512 - i.e., the algorithm with the smallest post-quantum signatures. The other combinations will lead to even larger response sizes. In this (and other similar) case(s), sending the pre-quantum RRSIGs and DNSKEYs along with the header, question section, and additional section (see marked fields in Figure 9) in the first fragment takes up around 1178B. This leaves only 54B for sending any post-quantum

RRSIGs and DNSKEYs, which is insufficient. In an instance like this, it becomes entirely difficult for a resolver to estimate the remaining number of fragments and thus can not make explicit requests for them.
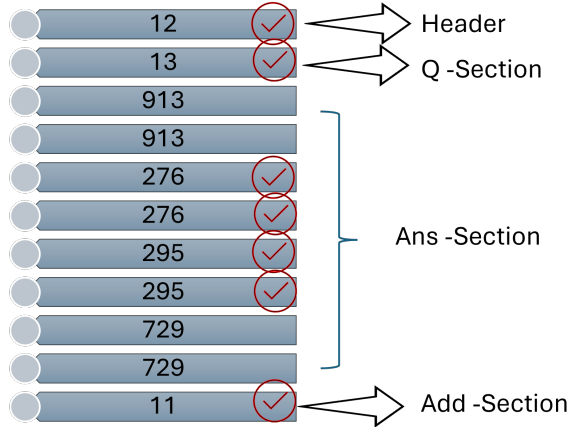

Figure 9: Sizes of different fields in a representative DNSKEY Response

To circumvent the aforementioned issue, we leverage z-bits (which are reserved as of yet) in the header of a DNSSEC response. Whenever a name server is unable to send any part of post-quantum signatures or keys in the first fragment (e.g., with RSASHA256), it sets z-bits in the header such that they represent one out of three post-quantum signature algorithms (i.e., FALCON, DILITHIUM, SPHINCS+). Upon receiving a response, the resolver first checks whether z-bits are set or not. If z-bits are set, it first deduces the post-quantum algorithm and then by leveraging pre-quantum RRs that it has already received, estimates the total size of the response and the number of fragments this response should be divided into. Subsequently, the resolver makes parallel requests explicitly for all remaining fragments.

Once all the fragments are received, the resolver first creates an updated message with both pre- and post-quantum RRSIGs and DNSKEYs and then combines all other fragments into this updated message. Our thorough analysis suggests that the aforementioned approach works accurately when various combinations of double signatures are incorporated in DNSSEC. However, we noticed that the re-assembled messages were not verified on the resolver (based on OQS-BIND). Our in-depth investigation suggested that this is due to the reason that DNS(SEC) uses a compression scheme for reducing the size of messages by eliminating the repetition of domain names in response messages. This is accomplished by replacing the domain names with a pointer to a previous occurrence of the same domain name (see RFC 1035 [17]). Due to this, our observation is that when we always send pre-quantum RRSIGs and DNSKEYs in the first fragment, it sometimes changes the order of RRs in the reassembled message (as compared with the original message created by the name server). Therefore, this issue leads to a bad compression pointer due to which the responses fail to verify on the resolver.

As a way of dealing with this issue, we use an offset to TTLs of RRs (see Fig. 3) in DNSKEY responses to keep track of the position of pre-quantum RRs in the original response messages on name servers. Once all the fragments are received by the resolver, it leverages this offset to create an updated message that places all the RRs in their original position. Our analysis suggests that this offset approach accurately re-assembles the messages such that they are correctly verified on the resolver.

The overall working of double-signed fragmented DNSSEC is shown in Figure 10. As discussed above, whenever the response size is greater than the advertised threshold, the name server first estimates how many fragments this particular response should be divided into, sets the z-bits if needed and includes an offset to TTL of relevant RRs, and finally sets TC flag to '1' to indicate to the resolver that this message is fragmented. The nameserver then sends the first fragment to the resolver and caches all other fragments. When the resolver receives this first fragment, it extracts the relevant information from this fragment to estimate how many additional fragments it should be expecting before the original message can be re-assembled. Once the fragments are estimated, the resolver then creates a separate query for all remaining fragments and sends these queries in parallel for efficiency reasons. For example, in Figure 10, we have assumed that the original response was divided into four fragments and then the resolver makes explicit queries in parallel for fragments 2 - 4 (i.e., ?2?test.socrates - ?4?test.socrates). Note that these fragments are formatted in the same way as in [24]. When the name server receives these fragment requests, it sends the fragments that it has previously cached to the resolver. Once the resolver receives all the fragments (i.e., 4 in this representative example), it starts creating an updated message by utilizing TTL offset and re-assembles all the fragments. Finally, it re-sets TC and z-bits to '0' and then attempts to verify the signatures to ensure the authenticity and integrity of the received IP addresses.

5 EVALUATION SETUP

To achieve the aforementioned goals - i.e., implementing and evaluating *Double-Signed Fragmented DNSSEC*, herein we present a Docker-based DNSSEC setup built on an Amazon EC2 instance. Our setup is comprised of a client, resolver, root server, and authoritative servers, all of which are running in separate docker containers. All these components - i.e., Resolver, Root Server, and Authoritative server leverage BIND9 as a DNS software tool that is widely used in the industry. These integrated components represent the following elements of a typical DNSSEC infrastructure.

- Client — this imitates any real-world user trying to resolve a domain name and access it.
- Resolver — this imitates any real-world cache server that will communicate with the DNS hierarchical servers (e.g., root server, authoritative server) on behalf of the client and facilitate the domain resolution process. Note that, since the process of caching and recursion are intimately connected, the terms recursive server and caching server are used synonymously in the literature.
- Root Server — this imitates a real-world root server but contains the zone file that will only contain the NS record of our (docker-based) authoritative server.
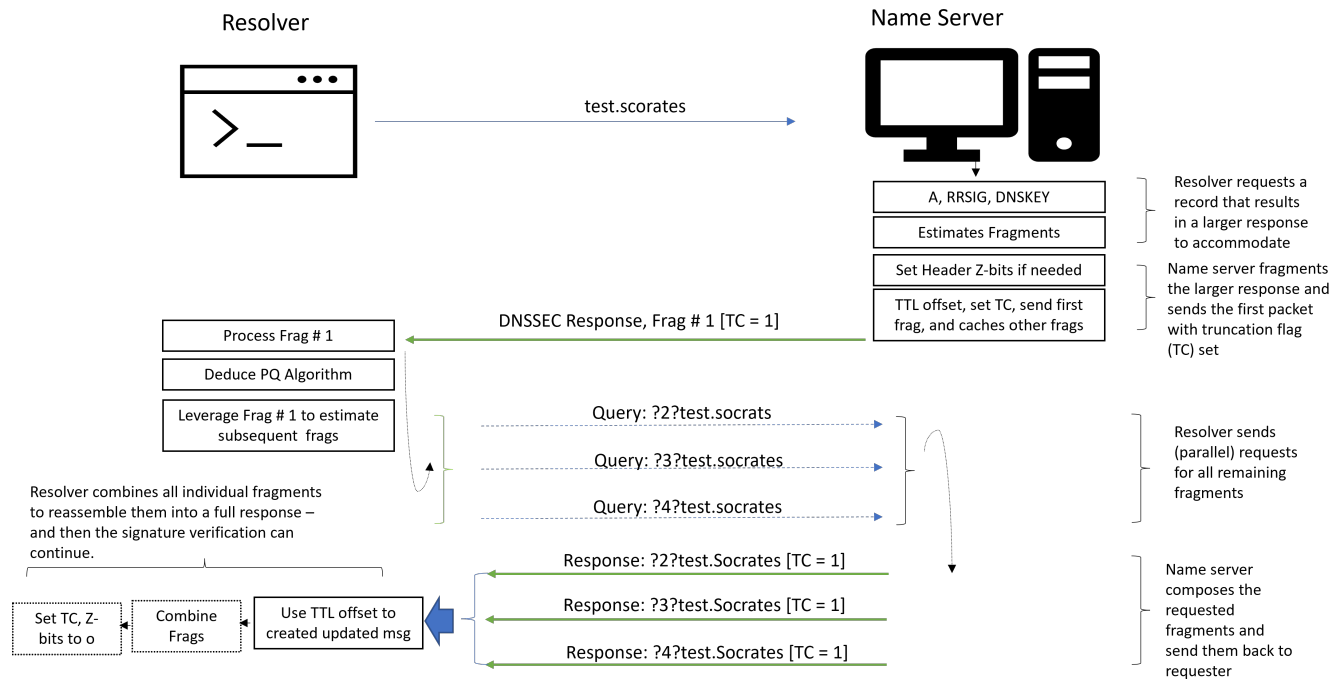
Figure 10: Work-Flow of Double-Signed Fragmented DNSSEC

- Authoritative Server — this imitates a real-world authoritative server that contains a zone file that holds the 'A' record of a dummy domain (e.g., test.socratescrc, that we keep in zone files of our authoritative server for experimental purposes).

For experimental purposes, our 'client' is a Ubuntu OS (22.04) that we simply use for querying our resolver and for analyzing the DNS response message. All of the other components - i.e., 'Resolver', 'Root Server', and 'Authoritative' servers run instances of Internet System Consortium's BIND9 as the DNS software (see details of BIND9 below). We create a simple 'Docker Network' (using Docker Version 25.0.3) to run all of the aforementioned components in a separate Docker container on an Amazon EC2 t2.xlarge instance which provides 4 cores of 2.3GHz Inter Xeon CPU and 16GB of RAM instance. Since by default, BIND9 does not support NIST's quantum-safe digital signature algorithms, we adopt the prototype implementation of OQS-BIND from [7] that adds the support of FALCON512, DILITHIUM2-AES, and SPHINCS+-SHA256-128S (with parameters set for level 1 - i.e.. 128-bit-security) using OpenSSL 1.1.1 the Open Quantum Safe (OQS) *liboqs* library 0.7.2 [27].

The overall interaction of all these components as a representative DNSSEC infrastructure is depicted in Figure. 11. A step-by-step description of communication among these components for a successful resolution of a domain name by our DNS(SEC) setup is presented below.

Step 1 Our client (a simple docker container with Ubuntu OS) wants to resolve 'test.socratescrc' (note that, for the experimental purpose we have assigned an 'A' RR to this subdomain on our authoritative server). It queries the 'Resolver' for address resolution.

Step 2 Once the 'Resolver' (a separate docker container that uses BIND9) receives this query and checks whether this resource record is already cached. If yes, it simply responds to the client with the corresponding IP (along with the other DNSSEC RRs). If not, it queries the 'Root Server' to find the IP address of the server that is authoritative for the queried domain.

Step 3 The 'Root Server' (a separate docker container that uses BIND9) responds with the IP address of the corresponding 'Authoritative Server' that can resolve the initial query.

Step 4 The 'Resolver' queries the 'Authoritative Server' whose IP address is returned by the 'Root Server' (e.g., for 'test.socratescrc').

Step 5 The 'Authoritative Server' (a separate docker container that uses BIND9) responds with the corresponding IP address along with other RRs such as RRSIG and DNSKEY needed for establishing the chain of trust.

Step 6 The 'Resolver' validates the received response by verifying the RRSIGs and responds to the client with appropriate DNS flags set to indicate whether this response is validated by the resolver and can be trusted.

To incorporate the fragmentation strategy depicted in Figure 10 in this evaluation setup, we utilized the QBF daemon since it adopts the standard RR format unlike ARRF that relies on custom-built RRs [24]. We included all the steps needed for accommodating double signatures in the daemon. This daemon then intercepts all the incoming and outgoing DNS packets on the resolver, root server,
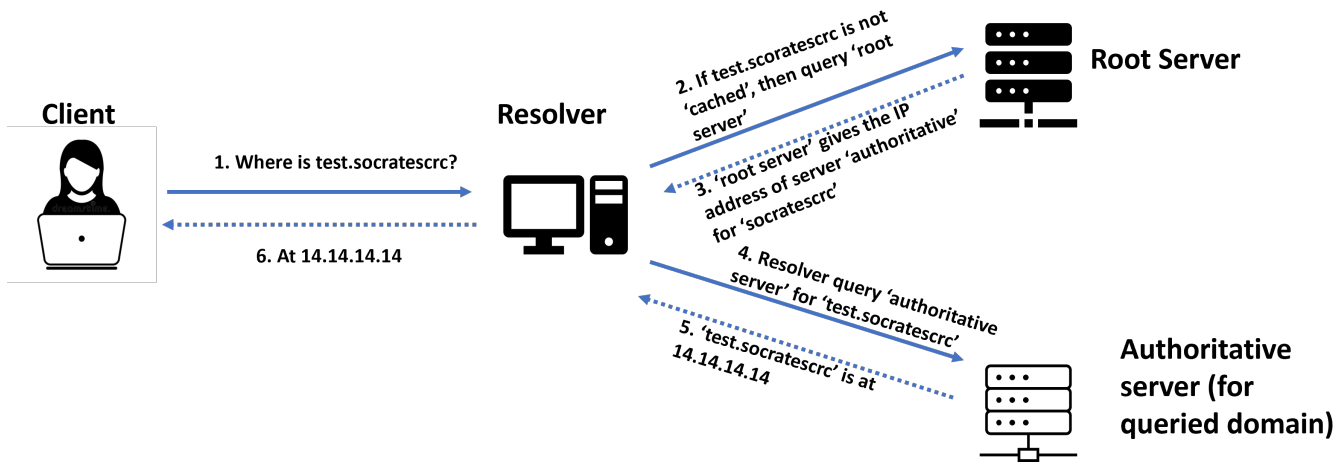
Figure 11: Step-by-Step Interaction of Testbed Components

and name server and performs the relevant steps as needed (see details in Section 4 and Fig 10). Therefore, practically, each component - i.e., resolver, root server, and authoritative server in our evaluation setup has an OQS-BIND software and daemon that incorporate our fragmentation strategy for double signatures (see overview in Figure 12). On the name servers, the daemon intercepts the incoming query, updates the maximum advertised UDP size to 64KB (i.e., max possible), and forwards it to the OQS-BIND software. The OQS-BIND creates a DNSSEC response to this query with double signatures, which is again intercepted by our daemon that executes all the steps depicted in Fig. 10. Upon receiving the response from the name server, the resolver's daemon intercepts it and goes through all the steps needed for accommodating and re-assembling double signatures (see Section 4 and Fig. 10). Once the resolver's daemon receives all the fragments, it reassembles them and finally sends the reassembled message to the OQS-BIND, which initiates the verification process for both signatures.
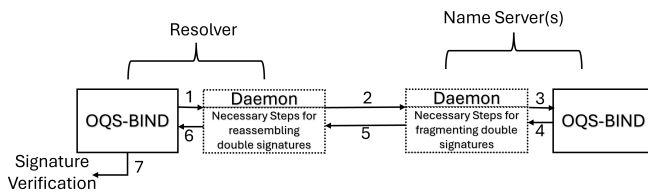

Figure 12: Interaction of components of evaluation setup

Berkely Internet Name Domain (BIND). BIND offers a complete suite of the DNS protocol, and the BIND9 software distribution contains a name server (named) and a set of associated tools such as for key and signature generation that are needed for our analysis. It is one of the most widely used software on DNS servers on the Internet [11]. Since BIND9 can be configured (using *named.conf*) as a resolver or name server, it allowed us to run 'Resolver', 'Root Server', and 'Authoritative Server' in separate containers using

the same BIND9 docker image. Although BIND9 allows flexible configuration features and supports multiple functions such as a single DNS server acting as both an authoritative name server and a resolver, we in our testbed have assigned one dedicated container to the resolver and authoritative name server as typically preferred by large operators [15]. Therefore, our testbed represents a realistic small-scale DNS infrastructure for assessing the impact of quantum-safe algorithms.

Double-Signed DNSSEC Setup. Our evaluation setup as described above is capable of performing the address resolution steps of conventional DNSSEC with both pre-quantum and post-quantum digital signatures. However, since we want the support of double signatures added to our testbed, we conducted experiments to ascertain whether BIND9 by default supports two signatures. Our observations are as follows:

Signature Generation on Name Servers: Our analysis revealed that we can create two key pairs and signatures - i.e., both pre-quantum and post-quantum on name servers (i.e., root and authoritative servers) by using the BIND9 utilities (see dnssec-keygen and dnssec-signzone tools in BIND9 manuals). Both signatures can be added directly to the corresponding signed zone files generated in BIND9. Figure. 13 depicts a portion of the signed zone file generated on the authoritative server. It is evident that the signed zone file contains two signatures for a representative record -i.e., *test0.socratescrc.*.

Double Signature Verification on Resolver: Our analysis of the verification process revealed that although we can send both signatures to the resolver and re-assemble them correctly in our test setup, the BIND9 signature validation by default checks only one signature before marking any response from the name server(s) as secure. Therefore, we modified the BIND9 source code (especially the code corresponding to the validation of the received responses) so that both the re-assembled signatures must be valid before accepting any response. Fig. 14 depicts a portion of the log file of our modified resolver. It is evident that the modified resolver in our setup verifies both signatures, unlike the default BIND9 resolver.

Figure 13: A portion of Signed Zone File depicting the addition of Double Signatures (Note, PQ signature - i.e., with ID 249 is shortened for brevity reasons)

Figure 14: A portion of the 'log-file' of our modified resolver that supports double-signatures

Experimental Methodology. For analyzing the impact of double signatures in the DNSSEC address resolution process, we kept 10 A resource records with unique labels on our authoritative server (e.g., test0.socratescrc – test10.socratescrc). Then, using the BIND9 key and signature generation tools, we generated two signatures with various possible combinations of pre-quantum (ECDSA256,

RSASHA256) and post-quantum (FALCON512, DILITHIUM2, SPHINCS+-SHA256-128S) digital signature algorithms. Once both signatures were generated, we moved the DS RR of the authoritative server to the root server and the DS RR of the root server to the resolver as a trust anchor so that we could establish the chain of trust identical to the conventional DNSSEC. Once all of this process is complete, using our client, we query the resolver iteratively for 10 resource records that we kept on our authoritative serve in our test setup. For each query, we estimate the query resolution time under simulated network conditions of bandwidth of 50Mbps and delay of 10ms (note that, we use the Linux tc utility for simulating these bandwidth and latency values). Finally, we compare the average resolution time of double signatures with post-quantum signatures to find any issues in migrating from only post-quantum signatures to double signatures for the interim period until CRQCs are fully realized as discussed in Section 1. A similar experimental approach is also used in [24] but with only one signature for the post-quantum era. Additionally, we also investigate how many additional fragments our daemon needs to compute and re-assemble with double signatures in comparison with post-quantum signatures alone.

6 EXPERIMENTAL RESULTS

In this section, we present the results of our experiments described in the previous section. Our thorough analysis suggests that our approach successfully incorporates all possible combinations of double signatures (pre-quantum + post-quantum) in DNSSEC, fragments them appropriately on name servers, and re-assembles the responses correctly on the resolver side. Additionally, we also observed that all the double signatures were correctly verified on the resolver. Table 1 shows the number of fragments when various combinations of pre- and post-quantum signatures are employed in our DNSSEC setup. Analysis suggests that our approach only adds one additional fragment in response to QTYPE 'A' and 'DNSKEY' queries when FALCON and DILITHIUM2 are combined with any of the pre-quantum signatures - i.e., ECDSA256 and RSASHA256 in comparison with post-quantum signatures alone. Furthermore, we observed that combining SPHINCS+ with any of the pre-quantum signatures does not add additional fragments. This is due to the reason that after sending the initial fragment, the daemon on name servers removes all RRs except RRSIG, DNSKEY, and OPT, thereby accommodating double-signatures created with SPHINCS+ in same number of fragments as with SPHINC+ alone. Mean resolution times with various combinations of double-signatures are shown in Tables 2, 3, 4. The results show that the impact of double signatures is insignificant on the resolution times of queries when compared with post-quantum signatures alone. In any double-signature combination that we have tested, resolution time only increases by less than 9% as compared with the post-quantum signatures alone. For instance, the resolution time for a combination of FALCON + ECDSA is 205.9ms as compared with 190.1ms for FALCON alone (i.e., an increase of approximately 8.3%). The lowest resolution time for double signatures was observed for any combination of FALCON with either RSA or ECDSA-based pre-quantum digital signatures. This is because FALCON has the shortest signature size compared with other post-quantum digital signatures. Similarly, RSA combined with any of three post-quantum candidates results in a lower

Table 1: Total Number of Fragments with Various Combinations of Double-Signatures

Digital Signature	Fragments in Respons to QTYPE A	Fragments in Respons to QTYPE DNSKEY
FALCON	2	3
FALCON+ECDSA	3	4
FALCON+RSA	3	4
DILITHIUM	7	7
DILITHIUM+ECDSA	8	8
DILITHIUM+RSA	8	8
SPHINCS	23	15
DILITHIUM+ECDSA	23	15
DILITHIUM+RSA	23	15

resolution time as compared with ECDSA (e.g., DILITHIUM + RSA results in lower resolution time in comparison with DILITHIUM + ECDSA, and likewise for other two post-quantum digital signatures). This is because RSA has faster signature verification than ECDSA [4].

The aforementioned observations suggest that double signatures can reliably be incorporated into DNNSEC infrastructure and have a minimal impact on performance as compared with post-quantum digital signatures alone. Therefore, their incorporation is ideally suited for DNSSEC during the interim period from now until CRQCs are fully realized. Particularly, the double signatures created with either FALCON or DILITHIUM with RSA or ECDSA have a resolution time that is 14-22% less than that of SPHINCS+ alone. Since both FALCON and DILITHIUM are based upon lattices, SPHINCS+ was only added as a post-quantum candidate by NIST to avoid the over-reliance on the lattices (see [20]). However, as observed in our experiments, SPHINCS+ has a significantly large signature size that results in much higher resolution time in DNSSEC. Therefore, the double-signature created with either FALCON or DILITHIUM is the optimal solution for the specified interim period while avoiding the envisaged problems of SPHINCS+ (see results described above and depicted in Tables 2 - 4). These double-signatures will essentially offer additional security to DNSSEC during the interim period if any attacks on lattice-based signatures are revealed in the near future that could be perpetrated through today's classical computer.

Table 2: Means Resolution Time (ms) - Various Combinations of FALCON512, ECDSA256, and RSASHA256

FALCON	FALCON+ECDSA	FALCON+RSASHA
190.1	205.9	204.5

Table 3: Means Resolution Time (ms) - Various Combinations of DILITHIUM2, ECDSA256, and RSASHA256

DILITHIUM	DILITHIUM+ECDSA	DILITHIUM+RSASHA
214.5	225.6	220.7

Table 4: Means Resolution Time (ms) - Various Combinations of SPHINCS+-SHA256-128S, ECDSA256, and RSASHA256

SPHINCS+	SPHINCS+ECDSA	SPHINCS+RSASHA
245.6	263.3	256.7

7 RELATED WORK

Our literature review suggests that, historically there have been a few proposals for dealing with large message size issues. For example, authors in [14, 26] proposed potential ways for conducting fragmentation at the application layer. However, both of them can potentially lead to DoS amplification attacks since they send many packets in response to a single query. Additionally, there are a few recent research works that focus on transitioning toward quantum-safe DNSSEC and related issues. However, unlike this work, none of these prior research works consider double-signatures for DNSSEC. For example, authors in [3, 8, 19] investigated NIST Round 3 and 4 PQC candidates, respectively, for their utilization in DNSSEC. Müller et al. [19] specifically analyzed NIST Round 3 candidates and established the various requirements that different candidates either meet or do not for their incorporation in DNSSEC. Similarly, Beernink et al. [3] presented a way of delivering larger keys out-of-band from DNSSEC. Goertzen et al. [8] focused on the fragmentation issue in DNSSEC for the post-quantum era. However as discussed before, it adopts a custom-designed non-standard pseudo-RR for performing the fragmentation at the application layer. The usage of non-standard RR makes its practicability questionable. Similarly, an improved version of the fragmentation approach for quantum-safe DNSSEC is also presented in [24], which leverages only standard RRs and thus seems more practical. However, unlike these existing works, we investigate the possibility of using double signatures for DNSSEC, and how the fragmentation issue can be dealt with in the simultaneous presence of pre and post-quantum signatures in DNSSEC. Through our implemented testbed and empirical analysis, we show the efficacy of double signatures in DNSSEC for the interim period until CRQCs are fully realized.

8 CONCLUSION & FUTURE WORK

In this work, we investigated the possibility of using double signatures in DNSSEC. For this purpose, we first investigated the maximum message size issue resulting from the incorporation of double signatures in-depth and then devised an approach that can

successfully incorporate double signatures on name servers and deliver them to the resolver over UDP by conducting the fragmentation of response messages at the application layer. We then built a functional Docker-based testbed by leveraging OQS-BIND9 and a daemon that handles double signatures and the fragmentation strategy. Our investigation suggests double signatures can be generated and added to signed zone files on the authoritative name servers. However, the source code of BIND9 needs a moderate level of changes to facilitate the verification of two signatures for the same resource record. With these required changes made to the BIND9 resolver and clubbing our daemon with it, we comprehensively analyzed the resolution time for various combinations of pre and post-quantum digital signatures. Our analysis revealed that double signatures do not have any significant overhead as compared with post-quantum digital signatures alone. Particularly, our investigation suggests that lattice-based signatures - i.e., FALCON and DILITHIUM combined with any of the pre-quantum digital signatures results in a shorter resolution time than SPHINCS+ alone which was shortlisted (despite having a very large public key and signature size and a slower verification) by NIST to avoid over-reliance on the lattice-based signatures for the post-quantum era. Our analysis suggests that double signatures generated with FALCON and DILITHUM are well-suited for DNSSEC and offer performance benefits compared with SPHINCS+ alone. The double signatures approach will also offer additional security in case any attacks on lattice-based PQ candidates are revealed in the near future. While the analysis conducted in this work establishes the overall efficacy of double signatures for DNSSEC, we believe that resolution time can further be reduced by incorporating our fragmentation strategy directly into BIND9 rather than in the form of a daemon. We plan to make these advancements and relevant analyses in the future.

ACKNOWLEDGMENT

REFERENCES

[1] National Security Agency. 2021. *Quantum computing and post-quantum cryptography*. Retrieved August 28, 2024 from https://bit.ly/3Huvuaql

[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. Resource Records for the DNS Security Extensions. https://www.ietf.org/rfc/rfc4034.txt.

[3] G.J. Beernink. 2022. *Taking the quantum leap: Preparing DNSSEC for Post Quantum Cryptography*. Ph. D. Dissertation. University of Twente.

[4] S. Brady. 2020. JWTs: Which Signing Algorithm Should I Use? https://www.scottbrady91.com/jose/jwts-which-signing-algorithm-should-i-use.

[5] J. Damas and P. Vixie. 2013. *Extension Mechanisms for DNS (EDNS(0))*. Retrieved August 20, 2024 from https://datatracker.ietf.org/doc/html/rfc6891

[6] ENISA. 2022. *Post-Quantum Cryptography, Integration Study*. Retrieved April 20, 2024 from https://www.enisa.europa.eu/publications/post-quantum-cryptography-integration-study

[7] Jason G. August, 2021. OQS-BIND9. https://github.com/aditya-asr/OQS-bind.

[8] Jason Goertzen and Douglas Stebila. 2022. Post-Quantum Signatures in DNSSEC via Request-Based Fragmentation. arXiv:2211.14196 [cs.CR]

[9] A. Hulsing, D. J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, P. Kampanakis, S. Kolbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, J.-P. Aumasson, B. Westerbaan, and W. Beullens. 2020. *SPHINCS+. Technical Report, National Institute of Standards and Technology*. https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[10] P. Kampanakis and T Lepoint. 2023. Vision Paper: Do We Need to Change Some Things?. In *Security Standardisation Research. SSR*, F. Günther and J Hesse (Eds.).

[11] Jonathan Knudsen. Nov 11, 2021. A stitch in BIND saves nine. https://shorturl.at/jlmoD.

[12] O. Kolkman, W. Mekking, and R. Gieben. 2012. RFC 6781: DNSSEC Operational Practices, Version 2. https://datatracker.ietf.org/doc/html/rfc6781.

[13] V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, D. Stehlé, and S. Bai. 2020. *CRYSTALS-DILITHIUM. Technical Report, National Institute of Standards and Technology*. https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[14] L. Song M. Sivaraman, S. Kerr. 2015. *DNS message fragments*. Retrieved August 25, 2024 from https://datatracker.ietf.org/doc/draft-muks-dns-message-fragments/00/

[15] BIND9 Manual. 2023. *Introduction to DNS and BIND9*. Retrieved August 27, 2024 from https://bind9.readthedocs.io/en/v9_18_6/chapter1.html

[16] J. Mao, M. Rabinovich, and K Schomp. 2022. Assessing Support for DNS-over-TCP in the Wild?. In *Passive and Active Measurement*, O. Hohlfeld, G. Moura, and C. Pelsser (Eds.). https://doi.org/10.1007/978-3-030-98785-5_22

[17] P. Mockapetris. 1987. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. https://datatracker.ietf.org/doc/html/rfc1035.

[18] Michele Mosca. 2018. Cybersecurity in an Era with Quantum Computers: Will We Be Ready? *IEEE Security & Privacy* 16, 5 (2018), 38–41. https://doi.org/10.1109/MSP.2018.3761723

[19] Moritz Müller, Jins de Jong, Maran van Heesch, Benno Overeinder, and Roland van Rijswijk-Deij. 2020. Retrofitting post-quantum cryptography in internet protocols: A case study of DNSSEC. *ACM SIGCOMM Computer Communication Review* 50, 4 (2020), 49–57.

[20] NIST. 2022. PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates. https://csrc.nist.gov/news/2022/pqc-candidates-to-be-standardized-and-round-4.

[21] Mohammad Reza Nosouhi, Syed W. Shah, Lei Pan, Yevhen Zolotavkin, Ashish Nanda, Praveen Gauravaram, and Robin Doss. 2023. Weak-Key Analysis for BIKE Post-Quantum Key Encapsulation Mechanism. *IEEE Transactions on Information Forensics and Security* 18 (2023), 2160–2174. https://doi.org/10.1109/TIFS.2023.3264153

[22] Mohammad Reza Nosouhi, Syed Wajid Ali Shah, Lei Pan, and Robin Doss. 2023. Bit Flipping Key Encapsulation for the Post-Quantum Era. *IEEE Access* 11 (2023), 56181–56195. https://doi.org/10.1109/ACCESS.2023.3282928

[23] T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. 2020. *FALCON. Technical Report, National Institute of Standards and Technology*. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

[24] Aditya Singh Rawat and Mahabir Prasad Jhanwar. 2024. Post-quantum DNSSEC over UDP via QNAME-Based Fragmentation. In *Security, Privacy, and Applied Cryptography Engineering*, Francesco Regazzoni, Bodhisatwa Mazumdar, and Sri Parameswaran (Eds.). Springer Nature Switzerland, Cham, 66–85.

[25] Peter W Shor. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. IEEE, 124–134.

[26] L. Song and S. Wang. 2019. *ATR: Additional Truncation Response for Large DNS Response*. Retrieved August 25, 2024 from https://datatracker.ietf.org/doc/draft-song-atr-large-resp/03/

[27] D. Stebila and M. Mosca. 2017. Post-quantum Key Exchange for the Internet and the Open Quantum Safe Project. In *Selected Areas in Cryptography*, R. Avanzi and H Heys (Eds.). https://doi.org/10.1007/978-3-319-69453-5_2

[28] 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. https://doi.org/10.1109/MCOM.2014.6828880

[29] 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. https://doi.org/10.1109/JSAC.2016.2558918

https://doi.org/10.1007/978-3-031-30731-7_4