Reliable and Decentralized Certificate Revocation via DNS: The Case for RevDNS

Protick Bhowmick
Virginia Tech
USA
protick@vt.edu

Dave Levin
University of Maryland
USA
dml@cs.umd.edu

Taejoong Chung
Virginia Tech
USA
tijay@vt.edu

Abstract

The Online Certificate Status Protocol's long slide—after 25 years of soft-fail rules, privacy leakage, and shaky infrastructure—exposes a deeper failure in web-PKI revocation. Certificate Authorities increasingly route OCSP traffic through CDNs for speed, yet this recentralizes trust: our measurements show Akamai serves 62 percent of all revocation responses, creating single points of failure and betraying PKI's decentralized ideals.

We present RevDNS, a DNS-based revocation scheme that drops CDN dependence while preserving real-time guarantees. Revoked serial numbers live in DNSSEC-signed `TXT` records; `NSEC` proofs allow aggressive negative caching, so recursive resolvers answer 99.8 percent of checks without bothering a CA. From 1.1 billion certificates and 5 million revocations, we find a large CA such as Let's Encrypt can publish data for 612 million certificates in a 345 MB zone, with resolvers shouldering nearly every lookup.

Because answers piggyback on ordinary DNS lookups, RevDNS adds no latency and discloses no more about users than standard DNS traffic. By keeping revocation authority with CAs and avoiding fragile hacks like short-lived certificates, RevDNS delivers a durable, decentralized path for TLS revocation—one that finally aligns operational practicality with the web's security ambitions.

CCS Concepts

• **Networks** → **Application layer protocols**; **Network measurement**; **Security protocols**.

Keywords

TLS, HTTPS, CRL, OCSP, DNS, DNSSEC, Network Measurement, RevDNS, Revocation

ACM Reference Format:
Protick Bhowmick, Dave Levin, and Taejoong Chung . 2025. Reliable and Decentralized Certificate Revocation via DNS: The Case for RevDNS. In *ACM SIGCOMM 2025 Conference (SIGCOMM '25), September 8–11, 2025, Coimbra, Portugal.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3718958.3754351

1 Introduction

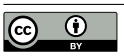
The demise of the Online Certificate Status Protocol (OCSP)—once a cornerstone of web PKI revocation—reflects a broader industry consensus: after 25 years of operational struggles, OCSP is no longer sustainable. Let's Encrypt, the largest Certificate Authority (CA), recently announced it will cease OCSP support in 2025 [32], citing untenable costs from 12 billion daily requests and negligible security benefits [5, 37]. This decision aligns with ecosystem-wide shifts: the CA/Browser Forum made OCSP optional in 2024 [1], while Microsoft and others now favor short-lived certificates (7–47 days) as a de facto revocation alternative [53]. Yet, as we demonstrate empirically, OCSP's failure stems not from technical irrelevance but from systemic operational challenges—a problem our work directly addresses.

Certificate revocation remains critical to mitigating key compromises (e.g., Heartbleed [58]) and CA breaches (e.g., DigiNotar [4]). Historically, OCSP promised real-time revocation but faltered under soft-fail policies, privacy leaks, and unreliable infrastructure. CAs increasingly offloaded OCSP to CDNs to improve performance, but at the cost of centralizing trust: Akamai now serves 62.29% of revocation data (§3), including Let's Encrypt's. While CDNs reduce latency, they create single points of failure and cede control to third parties—a precarious trade-off for PKI's decentralized ethos. Meanwhile, short-lived certificates shift rather than solve the problem, demanding automation many servers lack [10].

The Case for Revival. We argue that revocation need not be abandoned—it can be reimagined. Through the first large-scale analysis of OCSP hosting, we reveal that CDNs mask but do not resolve OCSP's core limitations: centralized trust, operational costs, and privacy risks. For instance, our measurements show that Akamai alone handles over 60% of all revocation requests, meaning a single CDN outage could disrupt revocation for hundreds of millions of certificates. These findings underscore a critical gap: the PKI ecosystem needs revocation that is both reliable *and* decentralized.

We address this gap with RevDNS, a DNS-based revocation system that eliminates OCSP's costs without sacrificing security. Our key insight is that DNS—already the internet's most scalable caching layer—can distribute revocation status efficiently. By encoding *revoked* certificates in DNSSEC-signed `TXT` records and leveraging `NSEC` proofs for negative caching, RevDNS reduces CA operational load by 99% compared to OCSP. It imposes no latency overhead (piggybacking on DNS queries) and leaks no more privacy than standard DNS. Crucially, it retains CAs' autonomy, avoiding reliance on CDNs or browser-managed filters like CRLite [46]. We make two key contributions:

Measurement of CDN-hosted revocation servers. Our first contribution is an extensive measurement study quantifying the performance, availability, and security implications of CDN-hosted revocation. On the *positive* side, we observe that CDN caching significantly improves global revocation reliability and reduces latency. On the *negative* side, we find severe centralization, with Akamai



SIGCOMM '25, Coimbra, Portugal

ACM ISBN 979-8-4007-1524-2/25/09
https://doi.org/10.1145/3718958.3754351

alone responsible for hosting 62.29% of *all* revocation data, including from Let's Encrypt. Such consolidation clashes with the PKI's diverse trust requirements and raises concerns over a single point of failure or undue external influence.

Reliability without centralization. Our second contribution is RevDNS, a new method that offers revocation data efficiently, reliably, and *without* surrendering operational control to a small set of CDN providers. To our knowledge, RevDNS is the first scheme to unify these properties. Its core insight is that DNS already supports globally distributed caching for small data objects.

RevDNS stores only *revoked* certificates in DNSSEC-signed `TXT` records. Since most certificates remain valid, RevDNS uses `NSEC` proofs to indicate non-revoked serials and enable *aggressive negative caching*, dramatically reducing load on both CAs and DNS infrastructure. By recasting OCSP within DNSSEC, RevDNS maintains low overhead without relying on third-party CDNs.

Our large-scale experiments confirm that RevDNS substantially reduces the cost and latency of revocation checks. In many environments, it adds *no extra delay* because revocation queries coincide with routine DNS lookups for the target domain. This low cost, combined with ease of deployment and preservation of user privacy, sets the stage for practical, comprehensive revocation coverage across the web.

2 Background

In this section, we provide a brief background on certificates, detail the protocols for certificate revocation, and DNSSEC.

2.1 Certificates

Digital certificates are essential in web security, establishing a link between subjects, typically domain names, and cryptographic public keys. These certificates are issued by CAs, whose authority is derived from a select group of self-signed root certificates. The validation process of a digital certificate involves a trust chain, encompassing the root certificate, any intermediate certificates, and the leaf certificate. Each link in this chain is authenticated by verifying signatures, assessing expiry status, and evaluating revocation status.

The de facto standard for web certificates is X.509, encoded using the ASN.1 [14, 26]. These certificates include fields like the subject's common name, public key, a unique serial number, a validity period, and instructions for checking if the certificate has been revoked.

2.2 Revocation Mechanism

During a TLS handshake, clients are responsible for checking the revocation status of all leaf and intermediate certificates presented to them. We discuss two standardized revocation protocols: Certificate Revocation List (CRLs) and Online Certificate Status Protocol (OCSP).

CRL. CRL is a collection of serial numbers from certificates that have been revoked, accompanied by a CA's signature. While CRLs are comprehensive, they suffer from inefficiencies as they can be very large and require clients to download the entire list, a considerable overhead [42], thus Chrome and Firefox do not use CRLs [20, 55].[1]

OCSP and its Extensions. OCSP [52] enables clients to query a CA's *OCSP responder* for a certificate's revocation status. Certificates include the Authority Information Access (AIA) extension, which specifies the responder's URL. A client submits an OCSP request containing the certificate's serial number and hashed representations of the issuer's name and public key, allowing the CA to verify issuance. The responder replies with a signed message indicating the certificate's status (`Good`, `Revoked`, or `Unknown`) and its cache-validity period. While OCSP provides timely revocation information, concerns about responder availability and client privacy persist [16].

To mitigate latency and privacy issues, OCSP Stapling allows web servers to fetch and attach OCSP responses directly in the TLS handshake [42], reducing reliance on external queries. However, adoption remains inconsistent, often due to deployment complexities and misconfigurations [16].

More recently, OCSP responses have been extended to support delegated certificates, which are signed by the CA but used exclusively for OCSP signing [43]. This OCSP Signature Authority Delegation enables CAs to separate certificate issuance from revocation checking, improving operational security. However, the inclusion of delegated certificates increases OCSP response sizes; since OCSP responses are rarely cached between clients and responders, this results in higher network overhead and an increased operational burden for CAs.

Moreover, delegating OCSP signing to an *external* operator—most commonly a CDN that terminates TLS at the edge—creates an additional supply-chain risk; A delegated responder certificate allows its holder to produce syntactically valid OCSP responses for *any* certificate issued by the parent CA. If that key is ever compromised, the attacker can forge GOOD status messages for a revoked certificate—or mark a legitimate certificate REVOKED to cause a denial of service—without touching the CA's own infrastructure [51].

2.3 Other Revocation Distribution Schemes

To address performance and privacy concerns associated with traditional CRLs and OCSP, browser vendors have implemented compact, partial revocation lists; Google's CRLSet, for instance, contains approximately 1,000 revocations and is updated daily for Chrome browsers [39]. However, the criteria for including certificates in these lists remain unclear [42]. Mozilla introduced OneCRL [49] in 2015, which focuses solely on intermediate certificates and currently holds about 1,600 entries [29]. In 2017, Larisch et al. proposed CRLite, a system using cascading bloom filters to compress revocation information for all certificates into roughly 10 MB [40]. While efficient, this approach requires users to regularly download updates and trust non-CA entities like browser vendors; presently, Mozilla manages the creation and daily distribution of these filters to Firefox browsers [46]. Similarly, Smith et al. [54] proposed a revocation scheme using a dynamically sized bit vector. However, it introduces a new field in the X.509 certificate and periodic updates to the clients, which makes deployment challenging.

[1] It is worth noting that Firefox ingests CRLs to feed CRLite.

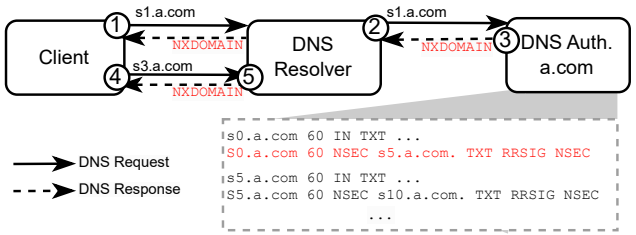


Figure 1: A diagram showing a DNS resolver's use of cached `NSEC` records. With the `NSEC` record for `s1.a.com` (③) cached, the resolver can immediately issue `NXDOMAIN` responses for queries within the `s0` to `s5` range (④), simplifying the display by omitting `RRSIG` records.

In 2016, Chariton et al. [19] proposed `DCSP`, a system for distributing revocation information via DNS. While `DCSP` appears similar to RevDNS at first glance, there are several key distinctions. In particular, `DCSP` and RevDNS store revocation data differently, resulting in substantially diverging performance and resilience characteristics: `DCSP` places all revocation data and metadata (including signatures) in `TXT` records. This approach inflates record size, and requires `DCSP` to serve `TXT` records for *all* queries, regardless of revocation status. Consequently, DNS resolvers must cache a greater volume of data, increasing both bandwidth usage and operational overhead.

Conversely, RevDNS uses compact DNS records that allow resolvers to respond to queries for non-revoked certificates, even if those serial numbers have not been previously requested. This approach eliminates the need for additional DNS records for valid certificates, substantially reducing the load on both DNS authoritative servers and resolvers, given that most certificates remain unrevoked.

This design leverages extensive "negative caching" for non-revoked certificates, a principle widely acknowledged as fundamental to the success of DNS [45].

2.4 DNSSEC

DNS utilizes *records* for mapping *domain names* to *values* (e.g., an `A` record links a domain name to an IPv4 address). DNS encourages caching with each record containing a TTL, defining the cache duration. Originally, DNS lacked security, enabling DNS response forgery and attacks. To counteract this, DNS Security Extensions (DNSSEC) were introduced [6–8, 30], providing integrity for DNS records primarily through two record types:

- `DNSKEY` records: Public keys in DNSSEC. Zones usually employ two `DNSKEY` records for signing: Key Signing Keys (KSKs), which signs `DNSKEY`s and Zone Signing Keys (ZSKs), which signs all of the other DNS records.
- `RRSIG` records: Signatures over records of a certain type and name, creating an RRSet. For instance, the `A` `RRSIG` for `example.org` authenticates all its `A` records. These are generated using the private key corresponding to a `DNSKEY` record's public key.

Hosts typically use a DNS resolver for domain name lookups; upon receiving a query from a host, the resolver iteratively identifies the authoritative name server for the domain and retrieves the required record. If DNSSEC is supported, it additionally fetches all necessary DNSSEC records (`DNSKEY`s and `RRSIG`s) for validation. The resolver then returns the validated record to the host. Notably, resolvers extensively utilize caching, avoiding repeated requests for any unexpired records already obtained.

NSEC:. In a scenario where a DNS resolver queries for a record that does not exist in a DNSSEC-signed zone, the response includes an `NSEC` record accompanied by an `NXDOMAIN` error code. The `NSEC` record specifies the *next domain name* in the zone file, along with the types of records that are available for the queried name. Essentially, the `NSEC` record creates a link from one domain name to the next within the zone, ordering them lexicographically to form a continuous chain. This feature enables DNS resolvers to cache the `NSEC` information, thus allowing them to directly respond to queries within the specified range, without the need to revert to the DNS authoritative server for each query, which is called *aggressive negative caching* [33].

Figure 1 illustrates this mechanism. When a client queries `s1.a.com`, the DNS authoritative server issues an `NXDOMAIN` response along with an `NSEC` record (colored in red) as a proof that the DNS records between `s0` and `s5` do not exist in the zone. Therefore, DNS resolvers can leverage the cached `NSEC` record to promptly return `NXDOMAIN` for any query within the specified range (e.g., `s3.a.com`), eliminating the need to forward these queries to the DNS authoritative server. A recent study [25] surveyed leading resolver software—including BIND, Unbound, Knot Resolver, and PowerDNS—and found that each supports aggressive negative caching.

RevDNS leverages NSEC to reduce the number of OCSP responses that a CA must manage, simultaneously accelerating the OCSP fetching process.

Deployment: DNSSEC has faced criticism for its low deployment rates among *domain name owners*, as individual users often encounter difficulties in deploying and managing DNSSEC correctly, primarily due to insufficient support from registrars or DNS operators, having only 3% of `.com` domains implemented DNSSEC [17]. Many users struggle with proper implementation, often hindered by limited support from registrars and DNS operators. As a result, only 3% of `.com` domains had deployed DNSSEC [17].

Despite the challenges in server-side deployment, DNSSEC validation by DNS resolvers has shown significant growth. This increase is largely attributed to the rising number of clients utilizing public resolvers that perform DNSSEC validation by default, such as Google [35] and Cloudflare [11], as well as large ISPs like Comcast [41] that have implemented proper DNSSEC support. APNIC, one of the five Internet registries, reports that over 42% of clients now use resolvers that conduct DNSSEC validation [9] in August 2024; notably, some European countries have achieved significant DNSSEC implementation, with Germany and the Czech Republic reporting 82% and 93% adoption rates, respectively.

RevDNS does not suffer from the challenges of DNSSEC deployment writ large—in particular, it does not need all websites to deploy DNSSEC (which is the greatest hurdle to DNSSEC deployment on the web today). For RevDNS to work for a CA, *only the CA needs to deploy DNSSEC at its servers.* CAs would have incentive to do

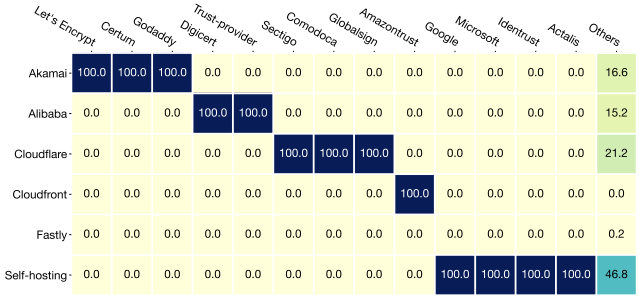


Figure 2: A heatmap showing CAs and the proportion of their OCSP servers facilitated by CDNs.

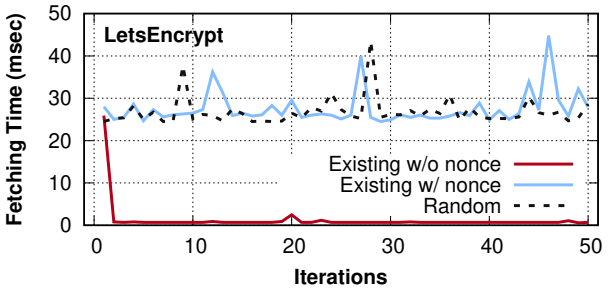


Figure 3: All OCSP responders (e.g., LetsEncrypt), utilizing Akamai operate in a reverse-proxy mode.

so, because—as we will show—RevDNS helps drastically reduce the number of OCSP requests it needs to provide directly from its servers. Clients and resolvers also need to support validation of DNSSEC, and to this end RevDNS leverages the trend of increasing DNSSEC-validating resolvers.

3 Measurement of CDN-hosted OCSP

In this section, we perform an extensive measurement and analysis of CDNs' roles in hosting revocation data. We evaluate (§3.1) who is responsible for hosting OCSP responders, (§3.2) how CDNs host them, and (§3.3) their performance.

3.1 Who hosts OCSP responses?

First, we aim to understand who are actually responsible for OCSP response delivery.

3.1.1 Methodology. From January 1st to 28th, 2025, we monitored all publicly available Certificate Transparency (CT) logs, parsing each certificate and extracting OCSP URLs from the Authority Information Access field. Our initial parsing identifies 192,728 unique OCSP responders. We then use the Censys API to collect all non-expired certificates referencing these responders, obtaining a total of 1.07 B certificates.

However, many responders share the same fully qualified domain name (FQDN) but have distinct URIs resolving to the same endpoint; for example, Google's OCSP infrastructure introduced 192,550 unique URIs, each mapped to small certificate subsets (tens to hundreds) but share the same FQDN. After merging such cases, we consolidated the dataset to 169 unique OCSP responders.

Identifying which organization actually serves a given OCSP response requires cross-referencing several data sources. First, we resolve the domain in the OCSP URL to its IP address and map that IP to an organization using CAIDA's AS-organizations dataset [21] (e.g., `ocsp.apple.com` resolves to `17.253.21.202`, which belongs to Apple). However, a distinct organization name does not necessarily mean that the CAs do not control their OCSP responders; in the CA ecosystem, mergers and acquisitions are common.

To distinguish first-party from third-party hosting, we first map each responder's IP address to an AS owner using CAIDA's AS-organizations dataset. For every case where that owner does not clearly coincide with the issuing CA, we perform a manual review: we inspect the RIR `whois` records for the IP prefix and the domain's `whois` data to determine operational control. We then cross-reference these findings with the public catalogue of CA mergers and acquisitions [3] and extend it with seven additional deals not covered there (Table 2, Appendix). Because the population of unique responders is modest, this hybrid automated–manual procedure reliably labels hosting relationships while avoiding the pitfalls of relying solely on automated IP-to-organization mapping.

Additionally, we notice that certain domains deploy `CNAME` to redirect incoming requests to a different entity. This redirection is a hallmark strategy of CDNs. To provide a concrete example, `ocsp.comodoca.com` uses a `CNAME` entry that directs to

`ocsp.comodoca.com.cdn.cloudflare.net`.

CDNs often incorporate specific CDN-tailored custom headers within the HTTP responses (e.g., `cf-ray` from `Cloudflare`), assisting us in confirming our domain mappings.

3.1.2 Results. We find that 64% (108) of OCSP responders are managed by third-party CDNs; only 36% (61) of the OCSP responders are fully hosted by the CAs themselves. Specifically, Akamai hosts 33 (20% of all) OCSP servers, Cloudflare 67 (40%), Alibaba 5, Fastly 2, and Cloudfront manages 1.

This corresponds to an even larger fraction of certificates; of the 1.07 B certificate we examined, *79.47% (921 M) are hosted by one of four CDNs*: Akamai (62.29%), Alibaba (7.61%), Cloudfront (4.85%), and Cloudflare (4.72%). Most alarming, we find extensive centralization in OCSP hosting.

Figure 2 illustrates CDN usage across the top 12 CAs, ranked by certificate issuance; Notably, we observe that CAs typically rely on a single CDN provider for their OCSP infrastructure; for example, Certum, with 11 OCSP servers, exclusively partners with Akamai.

These results reveal for the first time the extent to which OCSP hosting has become centralized around a small set of CDNs. On the one hand, this is a natural outcome of CDNs' economies of scale. On the other hand, it is surprising that the foundation of all secure communication on the Internet has become so reliant on single points of failure and compromise.

3.2 How do CDNs host OCSP responders?

There are several ways that a CDN could host OCSP servers. The safest way is by acting as a caching reverse proxy. Alternatively, a CA could delegate signing capabilities to a CDN; this can result in greater performance, but risks key compromise at the CDN. Here,

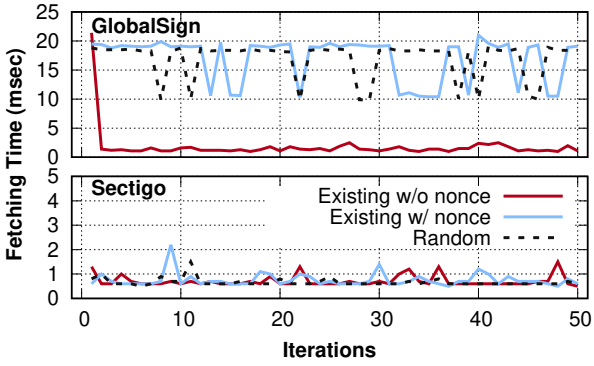


Figure 4: Cloudflare prefetches all OCSP responses from Sectigo, thus serving all responses without any significant delays. However, only GlobalSign uses it as a caching reverse proxy.

we seek to understand how CDNs host OCSP data, and if they are doing so safely.

3.2.1 Methodology. Determining whether a CDN merely acts as a reverse proxy or takes the additional step of signing the OCSP response is non-trivial. We use two techniques centered on OCSP requests that CDNs are incapable of responding to: a *nonce* extension and random serial number.

A nonce extension carries a unique random number, which acts as a unique identifier for each request; this ensures that the OCSP response cannot be cached thereby preventing reply attacks. Consequently, the request must be processed *directly by the CA* unless the CDN is actually signing the response. This results in higher latency when retrieving OCSP responses compared to conventional OCSP responses.[2]

We proceed with our investigation as follows: (1) From our measurement client, we dispatch 50 OCSP requests every second using the same unexpired serial number with and without *nonce* extension; each request has different nonce number so that the only entity possesses the private key can generate the OCSP response; this will help us to force the CAs to generate the response. (2) The nonce extension compels the CA to sign and generate the response. However, not all CAs support the nonce extension. Thus, there might be instances where we cannot measure the difference when the CDNs do not support the nonce extension. To account for this, we also dispatch 50 OCSP requests using various random serial numbers that do not exist. In such scenarios, if the CDN does not possess the private key, the response must originate from the primary CAs.

3.2.2 Results. We break down our results by how CDNs host OCSP servers:

Caching reverse proxy We first find that all CAs utilizing Akamai, Alibaba, and Fastly operate in a reverse-proxy mode. As depicted in Figure 3, the latency for the initial request is much higher

[2]An alternative explanation is possible: the signing component might reside within the CDN infrastructure but simply be slower to respond. While our measurements treat higher latency as evidence of delegation back to the CA, we acknowledge this ambiguity.

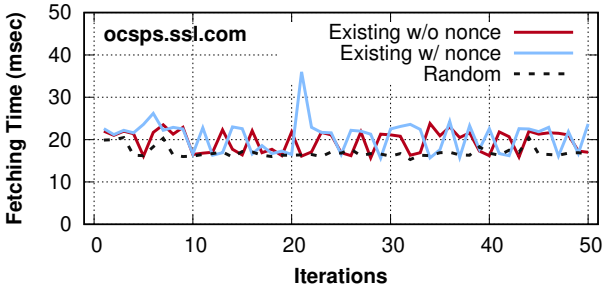


Figure 5: SSL.com runs OCSP responders on Amazon EC2 machines. Thus, there is no observable latency difference across the three types of OCSP requests.

than that of subsequent requests, indicating cache-based delivery. However, for all requests incorporating either a nonce or a random serial number, elevated latency is consistently observed, which implies a reverse-proxy. Interestingly, four Let's Encrypt OCSP responders routed through Akamai along with a set of five other OCSP responders (§A.1) administered by Alibaba, and an additional two OCSP responders managed by Fastly (§A.2), do *not* support for the nonce extension; this suggests that these CDNs forward the OCSP queries to the corresponding responders *regardless of whether they support the nonce extension*, thereby yielding the identical OCSP responses for all requests incorporating a nonce. We believe that this behavior is attributable to the cache implementation strategy within the CDN's reverse proxy. HTTP queries employing the GET/POST method may result in cache misses if the payload varies. Given that the OCSP request is encoded in ASN.1 format [43] and incorporated into the HTTP request, a distinct nonce within the OCSP request will trigger a cache miss. Thus, CDNs may consider tailoring their caching strategies based on the CA's nonce support capabilities.

Pre-fetch and caching reverse proxy. Cloudflare is known to prefetch OCSP responses from their customer CAs to mitigate latency while operating as a reverse proxy [50]. We observe that 28 out of 67 of their OCSP responders exhibit no discernible latency differences across three types of requests. For example, as shown in Figure 4 (bottom), when sending OCSP requests to Sectigo's OCSP responders, even the initial request for an existing certificate displays minimal fetching time, implying that OCSP responses are prefetched. Interestingly, this behavior persists even when OCSP requests for non-existent, randomly generated serial numbers are made; an immediate `error` response is returned. This suggests that CAs periodically push all OCSP responses to Cloudflare, enabling them to ascertain the existence or non-existence of certificates and consequently reject inappropriate requests without requiring consultation with the CAs.

Additionally, although Cloudflare responds almost instantly, we observe that none of its delegated OCSP responders honour the `nonce` extension. Cloudflare's infrastructure deliberately strips the nonce so that requests differing only in client-supplied nonce values map to a single cache key; retaining the nonce would create a unique object per query and defeat caching. Hence the absence of the nonce

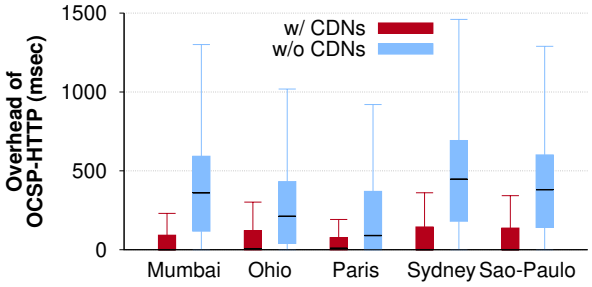


Figure 6: It reveals significant OCSP checking overhead across the vantage points, especially without CDN use; for example, in Sydney, 51.9% of the top 100K websites incur at least 100 msec delay due to OCSP verification.

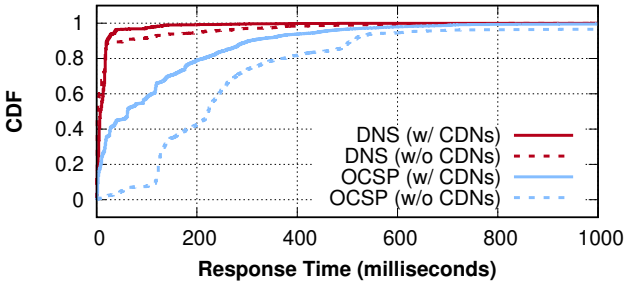


Figure 7: The distribution of time overhead incurred during DNS lookups for OCSP URLs and the subsequent fetching of OCSP responses over HTTP across the five regions. Note that 85% of OCSP URL lookups only take less than 18 milliseconds.

reflects an explicit design choice to maximise cacheability, not a lack of protocol awareness.

For the remaining 39 OCSP responders managed by Cloudflare, the service operates in a reverse-proxy mode; interestingly, all of them are administered by GlobalSign, the only CA served through Cloudflare that *supports the nonce extension*. Given that none of the other OCSP responders utilized by Cloudflare support the nonce extension, we suspect that Cloudflare intentionally acts as a reverse proxy solely for responders that support this feature.

CDN-signed OCSP responses. Surprisingly, we found that 2 OCSP responders (ocsps.ssl.com and ocsp.wisekey.com) that always return the response immediately as if they act like the origin server; for example, as shown in Figure 5, SSL.com runs their infrastructure on Amazon EC2 instances and we notice that the fetching times across all types of OCSP requests are aligned with each other, which strongly suggests that two CAs use Virtual Private Server (VPS) and sign the OCSP responses. In case of ocsp.wisekey.com always carry the delegated certificates [43], which for the CA to delegate response signing to other entities without sharing their signing keys. Given that Wisekey has issued relatively few certificates (3 K active certificates respectively as of January 2025 [22]), this limited scale might not justify the use of a CDN.

3.2.3 Takeaways. These results are encouraging; they show that, although centralized hosting has become commonplace, it has at least been done in a safe manner by relying on CDNs only for implementing large geo-caching. Reassuringly, we identify only two CAs that store private keys at CDNs to generate OCSP responses; while this reduces latencies, the delegated certificates increase the size of the responses. Collectively, these results indicate that CAs do not need to relinquish their keys or signing abilities in order to improve performance and reliability.

3.3 Client Performance

Our findings thus far indicate that 65% of OCSP responders, covering 70.6% of certificates, rely on CDNs. While this approach can alleviate latency by leveraging globally distributed caching, it also raises trust concerns by centralizing revocation infrastructure under a few third-party providers. In this section, we evaluate how this CDN usage (or lack thereof) impacts OCSP performance from the client perspective, employing two complementary methodologies.

3.3.1 Controlled Firefox Experiments. To quantify OCSP-induced overhead under real browsing conditions, we instrumented Firefox to benchmark TLS handshakes across five AWS vantage points (Mumbai, Ohio, Paris, Sydney, and São Paulo). Our testbed attempts to load 100K domains from the Tranco list, performing an OCSP request for each TLS connection.

Overhead Evaluation. We define OCSP overhead as the extra latency incurred by OCSP fetching and categorize OCSP responders into two groups: (1) CDNs and (2) non-CDNs; Figure 6 shows the overhead in milliseconds across five regions, revealing several insights.

Figure 6 illustrates the distribution of added latency in milliseconds across the five vantage points. Two primary observations emerge. First, an appreciable fraction of TLS handshakes are delayed by OCSP, regardless of region. In Mumbai, for instance, OCSP checking prolongs the handshake in 58.1% of connections when responders are not using CDNs. Second, CDNs do reduce latency: the share of delayed connections in Mumbai drops to 33.6% when OCSP is served via a CDN. These measurements help explain why many Certificate Authorities offload their OCSP responders to CDNs, and simultaneously underscore the potential risks in centralizing revocation infrastructure.

As discussed, the OCSP checking penalty can be attributed to two factors: (1) DNS delay for OCSP URL lookup, and (2) actual OCSP transfer time over HTTP.

To explore these latencies, we measured individual delay components, with Figure 7 illustrating their distribution. As expected, most of the performance penalty arises from retrieving the actual OCSP response: for instance, 85% of the DNS lookups for OCSP URLs complete within 18 ms. This occurs because queries for distinct serial numbers utilize the same DNS OCSP URL, allowing resolvers to answer from cache. Consequently, such caching underscores the potential of RevDNS in minimizing reliance on CDNs.

While previous proposals [19, 48] advocate embedding entire OCSP responses in DNS to shift from HTTP-based delivery, this risks packet fragmentation (1,232 bytes for EDNS [27] and 1,472 bytes for IPv4), inhibiting deployment in practice.

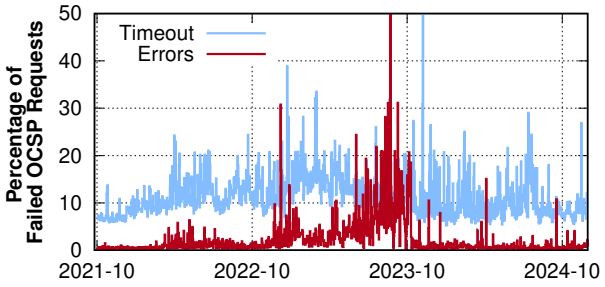


Figure 8: The ratio of failed OCSP requests have been increasing; in the latest snapshot, only 62.9% of OCSP requests from Firefox clients made successful. Each point on the x-axis represents a single day.

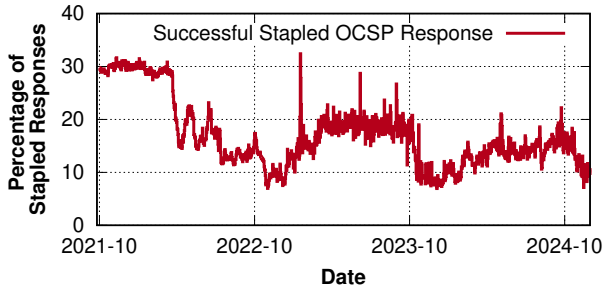


Figure 9: Firefox Telemetry shows the prevalence of OCSP Stapling among web servers; in our latest snapshot, only 24.04% of TLS web servers provide stapled OCSP responses.

3.3.2 Firefox Telemetry Observations. Beyond our controlled experiments, we also examine Firefox Telemetry [36] to gain broader insights into OCSP performance under real-world conditions. Firefox collects OCSP metrics by default, including success ratios and response latencies, providing a global perspective on responder behavior. We use Firefox Telemetry data [36] to gain insights into OCSP responder behavior from the viewpoint of clients. The dataset spans from October 2021 to October 2024, covering all clients that use Firefox Nightly versions 60 to 131.

Figure 8 presents two main failure categories: `Errors` (HTTP responses other than 200 or malformed data) and `timeout` (no response within a defined threshold). Over the measurement period, combined failure rates increased from 7.83% to 13.04%. While some of these issues may stem from client-side factors like poor network connectivity, the rising trend suggests persistent reliability challenges with OCSP responders. Equally concerning is that 12% of these failures are due to timeouts, preventing TLS handshake completion until the timeout elapses and thus significantly delaying its TLS handshake.

One possible way to mitigate such failure is OCSP Stapling, which allows servers to include OCSP responses within the TLS handshake. We monitor the `SSL_OCSP_STAPLING` signal in Firefox Telemetry and measure its adoption rate as illustrated in Figure 9. First of all, we observe a steady decline in stapling adoption, dropping from 29.8% in 2021 to 18.3% in our latest snapshot. A plausible explanation for low stapling deployment lies in the administrative complexity it introduces and the inadequate support offered by leading web servers. Prior work [16] highlights misconfigurations in Apache and NginX, including failures to prefetch OCSP responses promptly or to cache them correctly.

3.3.3 Takeaways. Our dual approach reveals critical insights into OCSP performance from the client perspective. We find that OCSP checking introduces significant latency overhead, affecting 43.32% of TLS connections to the top 100K Tranco websites when using CDN-backed responders, and this figure escalates to 91.93% without CDN support. However, our analysis also shows that DNS lookup times for OCSP URLs remain nearly identical regardless of CDN usage, highlighting the effectiveness of RevDNS that utilizes DNS. Additionally, telemetry data indicates a sharp increase in OCSP failure rates and a declining trend in OCSP Stapling adoption, further emphasizing the need for more decentralized and resilient OCSP infrastructures.

4 RevDNS

In this section, we introduce RevDNS, a system specifically engineered to enhance OCSP checking efficiency. RevDNS employs DNSSEC to maximize caching effectiveness while ensuring the integrity and authenticity of OCSP responses. To enhance clarity, we refer to the retrieval of OCSP responses through DNS as `OCSP-DNS`, in contrast to the traditional `OCSP-HTTP` mechanisms.

4.1 RevDNS: Design Overview

DNS has been considered an attractive method for conveying revocation information, as demonstrated by previous works [19, 48]. These approaches exploit the ability of DNS resolvers to cache responses, potentially addressing privacy concerns and enabling faster response delivery if cached. However, these methods face fundamental challenges, such as the need for CAs to generate DNS responses equal in number of certificates and size to OCSP responses, and the burden placed on DNS resolvers to cache extensive revocation data. RevDNS aims to overcome these limitations by utilizing DNSSEC.

4.1.1 Query Format. RevDNS employs DNS as a medium to transmit revocation information, similar to previous works [19, 48]. However, our intuition is based on the fact that *the number of unrevoked certificates vastly exceeds those revoked.* In RevDNS, each CA now serves OCSP responses through *their DNS authoritative servers*; clients issue OCSP requests over DNS with the following query format after parsing the certificate:

```
serial-number.ocsp.CA.com
```

CAs dedicate a subdomain for OCSP responses (e.g., ocsp.digicert.com). CAs now register *only the revoked serial numbers* with associated `TXT` records, which may include revocation metadata like the date or reason; given that most revoked responses do not specify the reason [42], the focus may be predominantly on the revocation date. For example, Digicert can serve a `TXT` record for

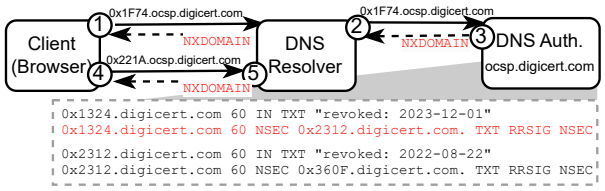


Figure 10: Overview of RevDNS: A client sends an OCSP request via DNS with a serial number (①, ②). For unrevoked certificates, an NSEC response providing adjacent revoked serial numbers (③) allows the DNS resolver to confirm non-revocation for any serial within the NSEC range (④, ⑤).

the revoked certificate of which serial number is 0x1324 with their TTL (e.g., 60) as follows:

```
0x1324.ocsp.digicert.com 60 TXT "revoked:20231201"
```

4.1.2 DNSSEC-signing. The DNS zone designated for OCSP responses must be secured with DNSSEC. This effectively eliminates the need for signatures in OCSP responses that ensure integrity. To minimize the attack surface, a CA may opt to separate their DNSSEC-signing keys (i.e., ZSK and KSK) from their base domain (e.g., CA.com). This separation serves a similar purpose to using delegated certificates in `OCSP-HTTP` responses. In such cases, CAs can delegate their zone to a distinct DNS server they manage, allowing the OCSP response zone to utilize its own signing keys (i.e., ZSK), while still preserving its authenticity.

With the OCSP zone containing only records for revoked certificates, *other DNS requests for non-revoked certificates can be effectively managed using NSEC records, which provide proof of non-existence.* This strategy significantly reduces the size of the zone that the CA needs to manage, while concurrently enhancing the caching efficiency of DNS resolvers. Thus, DNS requests for non-revoked certificates result in `NXDOMAIN` responses with `NSEC` records, which are bracketed by two adjacent revoked serial numbers.

Figure 10 illustrates this technique; when a client queries the revocation status of a non-revoked certificate (e.g., 0x1F74), the authoritative DNS server for ocsp.digicert.com returns the `NSEC` record that contains two adjacent revoked serial numbers (i.e., 0x1324 and 0x2312). This allows the client to ascertain that the certificate is not revoked. It is worthwhile noting that the `NSEC` records also carry `RRSIG`s, which can be verified using the `DNSKEY`s of ocsp.digicert.com.

4.1.3 Aggressive Negative Caching. An intrinsic feature of DNS resolvers, which is known as *Aggressive DNSSEC Negative Caching* [33], can further mitigate the overhead for both DNS authoritative servers and resolvers, particularly queries for non-revoked certificates.

Queries for non-revoked certificates return an `NSEC` record that denotes the two closest revoked serial numbers; DNS resolvers with *aggressive negative caching* capabilities can cache the `NSEC` record and efficiently respond to any subsequent queries within the NSEC range during its TTL. For example, as shown in Figure 10, a DNS resolver, upon querying the serial number 0x1F74, can cache the NSEC record that shows two serials (0x1324 and 0x2312) are revoked. Consequently, any subsequent OCSP DNS queries falling

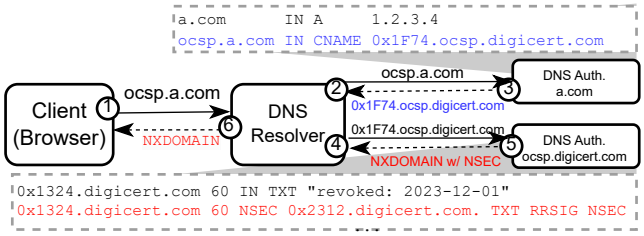


Figure 11: Stapling mechanism in RevDNS: A client issues a `OCSP-DNS` and an `A` domain query simultaneously (①). A `CNAME` record, indicated in blue, reroutes the `OCSP-DNS` request to the CA's DNS server (② to ⑤), a process known as *CNAME expansion* that enables internal lookup by the DNS resolver without client redirection [31, 44]. This streamlines certificate revocation checks (⑥).

within these NSEC ranges (e.g., 0x221A) can be immediately answered by the DNS resolver without needing to relay the query to the CA's DNS authoritative server since it is between 0x1324 and 0x2312. *This strategy not only significantly improves performance (e.g., reducing query latency) and decreases the load on both DNS resolvers and authoritative servers but also enhances privacy*; as more DNS queries are handled locally by DNS resolvers, fewer queries are transmitted to the authoritative servers, thereby reducing exposure and potential privacy risks.[3]

4.1.4 Stapling Support. Clients send `OCSP-DNS` requests after parsing the certificate. As indicated in Figure 7, the response time for DNS is generally faster than that for HTTP requests; however, there might be additional delays if the server cipher spec exchange completes before the `OCSP-DNS` response is received.

This latency can be effectively reduced if clients are able to send `OCSP-DNS` requests prior to obtaining the serial number information from the certificate. As illustrated in Figure 11, this can be achieved by following a specific naming convention for additional `OCSP-DNS` requests so that the client sends `OCSP-DNS` requests when looking up `A` record for the domain: ocsp.a.com. Thus, to support stapling, domain name owners can add a `CNAME` record that redirects this query to the corresponding serial number:

```
ocsp.a.com 3600 IN CNAME 0x1F74.ocsp.digicert.com
```

Because the TLS server (i.e., `a.com`) already knows its own certificate chain, publishing the above `CNAME` record is trivial. Modern recursive resolvers perform the ensuing *CNAME expansion* locally; the OCSP–DNS lookup therefore shares the same round-trip as the `A`-record query and never enters the critical path.

Stapling does, however, depend on DNSSEC for authenticating the TXT and `NSEC` records. Client-side validation remains uneven: recent APNIC measurements show that only about 40% of end-users sit behind validating resolvers [2]. On the *supply side* the picture is brighter; all major registrars—including GoDaddy, Namecheap,

[3]Because NSEC records are precomputed and not synthesized on the fly, a CA could in principle pre-generate the necessary data and rely on a third-party authoritative provider to serve it. However, as we discuss in §5, our evaluation indicates that the authoritative-side load is not high. Thus, CAs can reasonably operate their own authoritative infrastructure without needing to depend on CDNs or external DNS providers.

Google Domains, and Cloudflare—offer one-click DS-record provisioning so that it can sign zones easily [18]. Yet empirical scans still find DNSSEC enabled on barely 5% of .com domains, suggesting that registrar support alone is not enough [17, 28].

In short, limited DNSSEC penetration is a *deployment* hurdle, not a technical one. High-traffic sites can enable stapling immediately, and coverage will grow organically as validating resolvers and signed zones continue to proliferate.

4.1.5 Hard-Fail Revocation via DNSSEC and Encrypted DNS. RevDNS eliminates OCSP's soft-fail vulnerabilities by enforcing *hard-fail revocation*: clients abort connections unless revocation status is validated through DNSSEC. Unlike OCSP—where browsers often ignore errors—RevDNS ties TLS validity to cryptographic evidence of DNS integrity.

A straightforward deployment model is to rely on the recursive resolver's DNSSEC validation: clients require the AD bit and use DNS-over-HTTPS (DoH) [12] or DNS-over-TLS (DoT) to protect the channel, preventing on-path attackers from stripping signatures or injecting false negatives. However, this approach still places trust in the recursive itself, which could misbehave or be malicious, which creates two attack surfaces: (1) the recursive resolver itself, and (2) the client–resolver channel. DoH/DoT secures the channel, but does not protect against a dishonest recursive.

To mitigate this, clients can optionally fetch DNSKEYs/RRSIGs directly and perform DNSSEC validation locally. While not universally deployed, this hybrid strategy—DNSSEC validation on the client when possible, falling back to resolver-validated responses otherwise—reduces reliance on the recursive without eliminating scalability. For example, if a resolver returns an invalid or unencrypted TXT/NSEC response, or if client-side validation fails, the TLS handshake is terminated. This closes OCSP's soft-fail loophole by ensuring that failures lead to connection aborts rather than silent acceptance.

It is worth noting that major public resolvers such as Google [35] and Cloudflare [23] already perform DNSSEC validation by default, and Firefox deploys DoH natively [34]. Nevertheless, the critical distinction is that security-critical guarantees cannot rely on the AD bit alone when the recursive is untrusted; client-side validation offers a stronger model at the cost of added complexity.

4.1.6 Fate-Sharing with DNS. Critically, if a DNS resolver fails to respond, clients cannot resolve the server's address either, rendering revocation checks moot. This *fate-sharing* aligns security and connectivity: if DNS is unreachable, the client cannot establish a TLS connection regardless of revocation status. Thus, RevDNS introduces no new failure modes—unlike OCSP, which risks exposing users to revoked certificates during network outages.

4.1.7 Comparison with other DNS-Based Revocation Schemes. The key practical gulf between RevDNS and earlier DNS–based proposals (i.e., DCSP [19] and ODIN [48]) lies in *who learns what* and in *how much data the ecosystem must move*. In RevDNS, revocation status is conveyed through cache-friendly TXT/NSEC pairs that never embed certificate-specific labels. Once the first resolver in a region has cached a negative NSEC proof for a valid (i.e., *non-revoked*) serial, the CA vanishes from the traffic loop; subsequent clients obtain the answer locally and reveal nothing about their browsing patterns. Detailed measurements in §5 confirm that this design shrinks both CA traffic and recursive-resolver cache footprint by orders of magnitude.

CA	# of Certs		OCSP Resp. (B)	Zone Size (MB) ECDSA 256
	Total	Revoked		
LetsEncrypt	612,524,633	655,216	503	345
Google	148,668,485	524,759	471	273
Godaddy	107,581,381	1,464,592	1,777	712
Digicert	87,409,159	1,171,433	471	608
Microsoft	78,133,680	1,011	1,777	0.56
Amazon	56,171,777	30,282	314	16
Sectigo	52,496,781	784,298	471	408
Identrust	3,817,643	179,800	1,491	94
GlobalSign	1,581,068	270,300	1,442	139
Actalis	837,418	14,696	1,693	7.7

Table 1: The top 10 CAs ranked by the number of issued certificates in our latest snapshot; their number of revoked certificates, average OCSP response size, and estimated zone file size are shown.

By contrast, DCSP leaks far more: every cache miss exposes at least a group identifier—and often the serial number itself—via its two-stage TXT lookup. Those TXT sets also balloon quickly, regularly pushing responses beyond the 512-byte UDP limit and forcing TCP fallback, which doubles round-trip latency precisely when privacy is lost.

ODIN, meanwhile, maps each certificate to a unique DNS label that stores a full DER-encoded OCSP response, so any cache miss unambiguously signals the exact certificate being validated. The payload delivered is roughly an order of magnitude larger than the minimal TXT/NSEC pair in RevDNS, yet still has to be refreshed before the OCSP *nextUpdate* timestamp. The net effect is substantial authoritative-server bandwidth and enlarged resolver caches—while still exposing certificate-specific look-ups.

In short, RevDNS provides strong privacy guarantees while relieving both CAs and resolvers of most revocation traffic; DCSP and ODIN retain the privacy shortcomings of certificate-specific queries and impose significantly higher network overhead.

5 Evaluation

We now evaluate the performance of RevDNS by examining zone size, caching efficiency in both vanilla and stapling modes, and the benefits of aggressive negative caching.

5.1 Datasets

To gather revocation information for these certificates, we use the Common CA Database (CCADB) [24], which publishes a list of CRLs for each CA, along with the CA's name and any existing OCSP URLs. From this resource, we compile 12,879,839 revoked certificates from 7,255 CRLs managed by 186 different CAs.

We then utilize the authority_key_identifier field in CRL, to group the revoked certificates that managed by the same CA; this mapping facilitates the creation of a comprehensive list of serial numbers and their revocation statuses for each CA.

Zone Construction: To implement our technique, we extract the domain name of each CA from the CCADB (e.g., digicert.com) and create a corresponding subdomain by appending `ocsp` (e.g., ocsp.digicert.com). Each revoked serial number is then appended to this zone (e.g., 0x123.ocsp.digicert.com). Adopting this methodology, we are able to generate zone files for each CA and subsequently sign the entire zone.[4]

5.2 Zone Size

We now focus on the potential size of the DNS zone that a CA's authoritative server needs to manage, particularly in relation to OCSP responses. In traditional `OCSP-HTTP` setups, OCSP responders must sign and generate responses for all certificates. This process often results in a considerable amount of data that needs to be managed and served. However, with our proposed technique, CAs are required to consider only revoked serial numbers for their DNS zone, which significantly reduces the volume of data that needs to be managed.

Table 1 presents statistics for the top 10 CAs in terms of the number of issued certificates, along with their average `OCSP-HTTP` response sizes. We first observe that five CAs generate OCSP responses larger than 1 KB, primarily due to the inclusion of delegated certificates, as discussed in §3.2.

In RevDNS, the size of a DNS zone that a CA must manage is significantly determined by the choice of the signing algorithm used to produce `RRSIG`s and `NSEC` records. To assess this, we conduct experiments with a popular algorithm, `ECDSAP256SHA256`, which is required for DNSSEC-supporting resolvers according to RFC 8624 [57]; upon signing the zone with each algorithm, we analyze the overall zone size and derived several key insights. Our findings demonstrate that zone sizes can be efficiently managed using RevDNS; for example, we observe that a zone size of *345 MB is enough to encompass all revocation information to handle 612 M certificates managed by LetsEncrypt* when using the ECDSA algorithm[5]. Notably, many popular zones, including `.com`, `.net`, and `.edu`, have recently transitioned to the ECDSA algorithm, motivated by its enhanced efficiency and improved cryptographic robustness [56].

The efficiency of our approach is particularly notable when a CA manages a smaller number of revoked certificates; for example, consider the case of Microsoft, which has only 1,011 revoked certificates. We observe that merely 0.55 M bytes are sufficient to serve all revocation information for their 78 million certificates. Moreover, a single DNS packet for `OCSP-DNS` is typically smaller than an `OCSP-HTTP` response; for example, with the ECDSA256 algorithm, we found that 242 and 463 bytes are needed for revoked certificates (i.e., `TXT` records) and non-revoked certificates (i.e., `NSEC` responses), respectively. Considering that OCSP responses are rarely cached towards the client and CAs need to generate OCSP responses for *all* their issued certificates, we expect the traffic reduction achieved by RevDNS could be significant.

[4]The source code is publicly available at https://revdns.netsecurelab.org
[5]While not a direct comparison, for reference, the `.com` zone, containing approximately 160 million domains, is about 27GB in size.

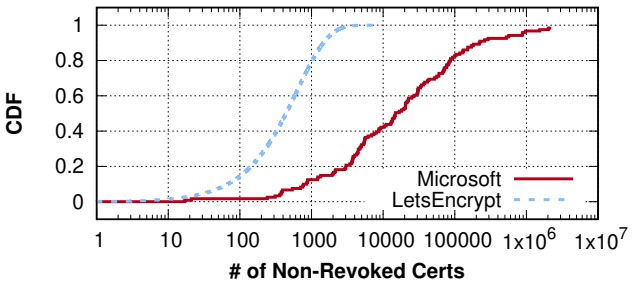


Figure 12: CDF of the number of non-revoked certificates served by a single `NSEC` record; for Microsoft, 20% of its `NSEC` record is capable of serving at least 85,000 non-revoked certificates.

5.3 Caching Efficiency

DNSSEC-supporting DNS resolvers can efficiently handle `OCSP-DNS` requests for non-revoked certificates, *even when these requests have not been previously made*, by leveraging *aggressive negative caching* with `NSEC` ranges.

This functionality is particularly advantageous for CAs, as it enables them to offload a significant portion of their workload onto DNS resolvers. The efficiency is greater for CAs with fewer revoked certificates, as the two revoked serial numbers in the `NSEC` records define a range of non-revoked serial numbers, allowing a single `NSEC` record to cover a vast number of non-revoked certificates.

Figure 12 illustrates the distribution of active non-revoked certificates that can be served by a single `NSEC` response. For example, for LetsEncrypt, we find that 10% of `NSEC` records can represent over each 1,000 `OCSP-DNS` non-revoked certificates. This efficiency is magnified for CAs maintaining a relatively small number of revoked certificates; taking Microsoft as an example, with 78 M active certificates and only 1,011 revoked certificates, 60% of their `NSEC` records can cover more than 10,000 non-revoked certificates. In some cases, a single `NSEC` record can represent as many as 2.1 million non-revoked certificates, highlighting the exceptional efficiency of this approach.

5.4 Experiments

Performance Enhancement with RevDNS. To evaluate the performance improvement of RevDNS relative to `OCSP-HTTP`, we use Firefox browsers to establish TLS connections with the top 100 K websites from the Tranco list and perform revocation checks via `OCSP-HTTP` and log all traffic in an Ohio AWS vantage point. We configure it to use Google's public DNS resolver.

Our experiment explores two RevDNS deployment scenarios: (1) 'Vanilla mode', where clients extract certificate serial numbers during the TLS handshake and send `OCSP-DNS` requests, and (2) 'Stapling' mode, involving simultaneous `OCSP-DNS` and `A` DNS requests, assuming `CNAME` record redirection by the website's DNS server to the CA's server. We can simulate the performance gains by comparing DNS latency in domain and OCSP URL resolutions, assuming equal resolution times for `A` and `TXT` records. We launch our experiments across five vantage points to measure TLS handshake

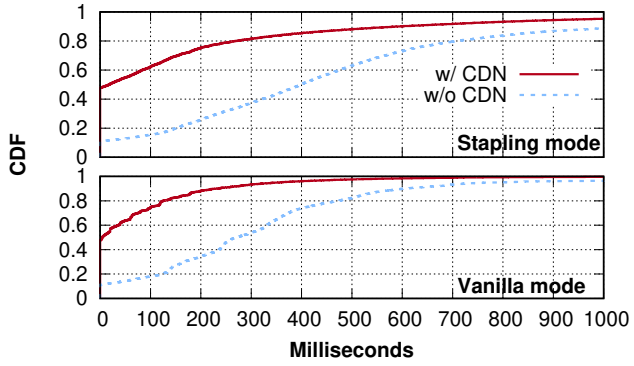


Figure 13: Performance difference in milliseconds between RevDNS and `OCSP-HTTP`; this indicates how much faster RevDNS is—shown in vanilla mode and with stapling mode—relative to `OCSP-HTTP`; more-positive values mean larger speed-ups. The x-axes are truncated; in Stapling mode, the x-axis extends to 8,217 milliseconds, and in vanilla mode, it extends to 3,939 milliseconds.

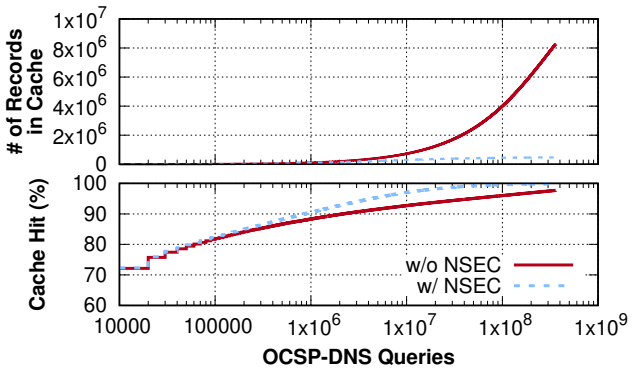


Figure 14: With aggressive negative caching, RevDNS serves 99.8% of OCSP requests from the cache, while only necessitating 5.6% of the caching space compared to previous work [19, 48].

completion times using `OCSP-HTTP` versus `OCSP-DNS` and categorize them based on whether the OCSP responder of the certificate uses CDN or not.

Figure 13 shows the result. We first confirm the performance gains in both modes; for example, when OCSP responders do not use CDNs, clients experience faster TLS handshakes for the 89% of domains compared to `OCSP-HTTP`. With CDN-utilizing OCSP responders, RevDNS find it beneficial for 52% of the Tranco domains; interestingly, the percentage of domains benefitting from RevDNS was similar across both CDN and non-CDN scenarios. However, the stapling mode offered greater time savings – up to 800 milliseconds for 20% of domains, compared to 700 milliseconds in the vanilla mode. This implies that stapling mode can yield superior performance gains, especially when latency is high between the client and the CA's DNS server or between the client and the web server, suggesting that earlier fetching of OCSP responses before TLS handshake can lead to enhanced performance.

Impact on DNS Resolver Caching. We investigate the caching benefits of `NSEC` records in RevDNS by simulating real-world traffic patterns. Considering LetsEncrypt OCSP servers process approximately 360 million OCSP requests hourly [15], we generate an equivalent number of `OCSP-DNS` requests for 317 million non-expired certificates, of which 499,935 are revoked (Table 1). Given the lack of specific data on OCSP request patterns and domain popularity, we adopt a web traffic distribution model following a Zipf distribution [38, 47] with a parameter α set to 1.2 to reflect that a relatively small number of highly popular websites account for the bulk of web traffic.

Assuming that the DNS responses are cached more than an hour, we generate `OCSP-DNS` requests and focus on two metrics: (1) the percentage of `OCSP-DNS` requests served from DNS resolvers' cache, and (2) the number of cached entries in DNS resolvers for 360 million `OCSP-DNS` requests. We also compare RevDNS with prior work [19, 48], which simply use DNS for caching OCSP responses regardless of certificate's revocation status: it caches every OCSP response, raising concerns about the impact on caching space, as every response is stored. In contrast, RevDNS implements the `NSEC` mechanism and utilizes aggressive negative caching, which potentially reduces the amount of caching space required by DNS resolvers.

Figure 14 (bottom) shows that with an increase in `OCSP-DNS` requests, a larger proportion of OCSP responses are served from the cache. For example, when all requests are made, RevDNS achieves a 99.8% rate of cache-served responses, a slight improvement over the 97.7% rate achieved without `NSEC` records, which rely solely on exact response matches for caching. Although this difference might appear small, we observe a substantial reduction in required cache space when using RevDNS (Figure 14, top). Specifically, RevDNS requires only 471 K cache entries, significantly fewer than the 8.3 M entries required without `NSEC`; this underscores the enhanced caching efficiency of RevDNS, particularly in reducing the DNS resolver's cache storage burden.

This strategy not only slashes latency and shrinks resolver-side state, it also strengthens privacy: with ≈ 99.8% of RevDNS lookups satisfied directly from resolver caches, *the CA never sees most serial-number requests at all*, sharply limiting the amount of per-site browsing information that reaches the revocation infrastructure.

6 Concluding Discussion

Certificate revocations are a necessary component of any PKI, and all clients are required to check for them, yet most do not due to the bad performance and poor reliability that CA-hosted revocation servers have. In this paper, we performed extensive measurement studies to understand the role that CDNs play in hosting certificate revocation data. Our results were partly encouraging, finding that CDNs' caching helps improve the reliability and performance of OCSP, but that this comes at a cost too high for the PKI: extensive centralization of hosting. We found that 70.6% of all certificates' revocation information is hosted by one of five CDNs, with Akamai alone responsible for hosting 56.1%. It would seem that among performance, reliability, and decentralization, we must pick two, but

we present a system, RevDNS, that shows that all can be achieved in tandem.

We stress, however, that today's DNS ecosystem is *operationally* less decentralised than its protocol design suggests: a handful of large public resolvers (e.g., Google 8.8.8.8, Cloudflare 1.1.1.1) handle a significant share of all queries. Thus RevDNS *reduces, but does not eliminate*, central points of control. It shifts dependency away from a single CA-selected CDN to the diverse resolver ecosystem, whose concentration varies with deployment and user choice. Future work should explore resolver-selection strategies, multi-homed querying, and incentives for enterprises and ISPs to operate independent resolvers so as to realise the full decentralisation potential of RevDNS.

We now conclude by discussing additional techniques CAs can employ to further enhance OCSP delivery through the use of RevDNS.

Serial Number Assignment: Serial numbers are already globally unique within each CA's namespace, so intra-CA collisions are impossible by construction. RFC 5280 places no randomness requirement on those values because a 160-bit space makes hash-collision attacks impractical [14]; nevertheless, the CA/B Forum now requires at least 64 bits of entropy to blunt any future collision or pre-computation attacks [13]. Under RevDNS, CAs can exploit the sparsity of the 160-bit serial-number space to reduce `NSEC` overhead. A practical tactic is to dedicate a rarely used high sub-range (e.g., 2^{120}-2^{150}) to certificates that are both short-lived and low-risk (e.g., , issued to lightly trafficked domains). By clustering these serials into just a few contiguous blocks, only a small set of `NSEC` records is needed to prove the absence of all other serials, thereby dramatically shrinking the cost of "not-revoked" proofs.

A complementary tactic is to allocate consecutive serial numbers to certificates that share operational dependencies (same hosting platform, same HSM, etc.). If one of those certificates is ever compromised, others in the sequence are statistically more likely to be revoked as well; placing them in an adjacent block means a single NSEC span can vouch for the entire cohort.

For any gaps that remain, DNSSEC's NSEC chain supplies a cryptographic absence proof: two adjacent NSEC records bracket each empty interval, showing that no other `TXT` records-hence no other revoked serials-exist between them. Together, these practices preserve both uniqueness and completeness without extra machinery while significantly reducing the number of NSEC ranges clients must fetch or cache.

OCSP Validity Period: OCSP responses include two values, `thisUpdate` and `nextUpdate`, which enable clients to cache the OCSP response; in the context of RevDNS, this caching period can be naturally aligned with the validity period of the `RRSIG`, specifically the `inception` and `expiration` dates. This alignment allows for a seamless integration of caching mechanisms within the DNS framework, enhancing both efficiency and reliability.

Zone Signing Keys: `OCSP-DNS` responses are signed using a ZSK, *distinct from the CA's certificate-signing key*; this separation reduces the CA's attack surface because the same key is no longer used for both certificate and OCSP signing. While CAs currently rely on delegated certificates, that approach inflates OCSP responses and often forces dependence on CDNs to handle the extra traffic. Using ZSKs both separates privileges and simplifies key rollover, allowing CAs to rotate their `DNSKEY`s regularly and thereby strengthen overall ecosystem security.

Practical deployment: client-side DNSSEC validation with OCSP hard-fail fallback The main security concern with resolver-assisted validation is trust in the recursive resolver. Even with encrypted transport (DoH/DoT), an attacker-controlled or misconfigured resolver can lie. A more realistic hard-fail deployment therefore *moves DNSSEC validation to the client and treats resolver assistance as an optimization rather than a trust anchor.* In this hybrid, the client first attempts a DNSSEC-validated RevDNS check and falls back to OCSP with hard-fail if DNS retrieval or validation fails. This does not replace OCSP; it reduces load and latency where DNSSEC works, while preserving security under failure or attack.

- Mechanism: The client sets the `DO` bit, requests the relevant `TXT` and validates the response locally against the zone `DNSKEY` and the parent `DS` chain to a trust anchor. If validation succeeds, the client accepts a not-revoked outcome or aborts on a revoked outcome. If any step fails (e.g., fetch error, missing signatures, invalid chain, timeout), the client performs a conventional OCSP check and treats failure to obtain a valid OCSP response as a connection failure (hard-fail).
- Performance: Cold paths can require several records such as `TXT` or `NXDOMAIN` & `NSEC`, `DNSKEY`s, `DS`, and `RRSIG`s). In practice, `DS` and `DNSKEY`s are often cached at large resolvers, and clients can parallelize lookups over a single DoH/DoT connection. Once keys are cached at the client, steady-state checks are typically one RTT to the resolver plus local signature verification. OCSP traffic is limited to paths where DNSSEC is blocked or broken. Importantly, client-side validation still benefits from resolver caching: hot `TXT` and `NSEC` records and key material reduce most checks to a single RTT. OCSP traffic is limited to paths where DNSSEC is blocked or broken.
- Resolver use without trusting it: DoH/DoT to a public resolver with solid DNSSEC support improves reachability and latency, but the client does not rely on the `AD` bit for security decisions. The `AD` bit can be used as a performance hint; acceptance still depends on local DNSSEC validation.
- Capability probing: Clients can cheaply detect DNSSEC-impaired paths using a non-critical probe to a known signed name. This probe is used only to choose the fast path vs. direct OCSP and never to decide trust. When the network changes, the client re-probes.
- Scope: Given that major browsers already ship centralized revocation (e.g., Firefox [46]) and that short-lived certificates further reduce reliance on revocation, this hybrid is unlikely to change browser behavior. It may be better suited to operator-controlled environments (e.g., enterprise mTLS) where clients and resolver choices can be managed and hard-fail policies are acceptable.

We believe this hybrid can take substantial pressure off OCSP responders: clients that can retrieve and validate DNSSEC records complete revocation checks without contacting OCSP, so only

DNSSEC-impaired paths fall back to OCSP where we hard-fail. In steady state, this should reduce overall OCSP query volume and smooth demand spikes, while preserving security on failure. As we discuss in the evaluation section, the authoritative-side DNS load appears modest; thus CAs can reasonably operate their own authoritative infrastructure rather than relying on CDNs or third-party DNS providers.

7 Ethics

All of our scans of public services were done at low rate limits, and to popular servers, thereby making it very likely that we did not adversely affect any other users' ability to obtain revocation information in a timely manner. Nonetheless, we ran our measurements from machines that were running web pages explaining that we are performing experiments, with information on how to opt out (none did).

Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. We are especially grateful to Eric Rescorla for his critical feedback and thoughtful discussions on hard-fail revocation, which significantly strengthened this work. This research was supported in part by NSF grants CNS-2339378, CNS-2247306, and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2023-00215700, Trustworthy Metaverse: blockchain-enabled convergence research].

References

[1] Ballot SC063v4: Make OCSP Optional, Require CRLs, and Incentivize Automation. https://cabforum.org/2023/07/14/ballot-sc063v4-make-ocsp-optional-require-crls-and-incentivize-automation/.
[2] Use of DNSSEC Validation for World (XA). https://stats.labs.apnic.net/dnssec/XA.
[3] WebPKI and Digital Signature related M&A + Investment + Public Offerings. 2023. https://gist.github.com/rmhrisk/c0afbb7e444dab9cf76936e24d4b32e8.
[4] C. Arthur. DigiNotar SSL certificate hack amounts to cyberwar, says expert. *The Guardian*. http://www.theguardian.com/technology/2011/sep/05/diginotar-certificate-hack-cyberwar.
[5] J. Aas. OCSP systems at scale are complex. 2024. https://news.ycombinator.com/item?id=41047832.
[6] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, IETF, 2005. http://www.ietf.org/rfc/rfc4033.txt.
[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, IETF, 2005. http://www.ietf.org/rfc/rfc4035.txt.
[8] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, IETF, 2005. http://www.ietf.org/rfc/rfc4034.txt.
[9] APNIC DNSSEC validation rate. https://stats.labs.apnic.net/dnssec.
[10] Announcing Six Day and IP Address Certificate Options in 2025. https://letsencrypt.org/2025/01/16/6-day-and-ip-certs/.
[11] Announcing Universal DNSSEC: Secure DNS for Every Domain. https://blog.cloudflare.com/introducing-universal-dnssec.
[12] T. Boettger, F. Cuadrado, G. Antichi, E. L. Fernandes, G. Tyson, I. Castro, and S. Uhlig. An Empirical Study of the Cost of DNS-over-HTTPS. *IMC*, 2019.
[13] Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, CA/Browser Forum. https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-2.0.2.pdf.
[14] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, IETF, 2008. http://www.ietf.org/rfc/rfc5280.txt.
[15] K. Christofferson. Let's Encrypt improves how we manage OCSP responses. https://letsencrypt.org/2022/12/15/ocspcaching.
[16] T. Chung, J. Lok, B. Chandrasekaran, D. Choffnes, D. Levin, B. Maggs, A. Mislove, J. Rula, N. Sullivan, and C. Wilson. Is the Web Ready for OCSP Must Staple? *IMC*, 2018.
[17] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. *USENIX Security*, 2017.
[18] T. Chung, R. van Rijswijk-Deij, D. Choffnes, A. Mislove, C. Wilson, D. Levin, and B. M. Maggs. Understanding the Role of Registrars in DNSSEC Deployment. *IMC*, 2017.
[19] A. A. Chariton, E. Degkleri, P. Papadopoulos, P. Ilia, and E. P. Markatos. DCSP: Performant Certificate Revocation a DNS-based approach. *EuroSec*, 2016.
[20] CA/Revocation Checking in Firefox. https://wiki.mozilla.org/CA/Revocation_Checking_in_Firefox.
[21] CAIDA ASOrganizations Dataset. http://www.caida.org/data/as-organizations/.
[22] Censys. https://censys.io/.
[23] Cloudflare 1.1.1.1 (DNS Resolver). https://developers.cloudflare.com/1.1.1.1/faq/.
[24] Common CA Database. https://www.ccadb.org/.
[25] C. Deccio and B. Tessem. On Aggressive Negative Caching in DNS Resolvers. *TMA*, 2025.
[26] O. Dubuisson. *ASN.1 communication between heterogeneous systems*. Morgan Kaufmann, 2001.
[27] DNS flag day 2020. https://www.dnsflagday.net/2020/.
[28] DNSSECDeploymentStatistics. https://stats.dnssec-tools.org/.
[29] Default OneCRL Data. https://firefox.settings.services.mozilla.com/v1/buckets/security-state/collections/onecrl/records.
[30] D. Eastlake. Domain Name System Security Extensions. IETF RFC 2535, IETF, 1999.
[31] R. Elz and R. Bush. Clarifications to the DNS Specification. RFC 2181, IETF, 1997.
[32] Ending OCSP Support in 2025. https://letsencrypt.org/2024/12/05/ending-ocsp/.
[33] K. Fujiwara, A. Kato, and W. Kumari. Aggressive Use of DNSSEC-Validated Cache. RFC 8198, RFC Editor, 2017.
[34] Firefox DNS-over-HTTPS. https://support.mozilla.org/en-US/kb/firefox-dns-over-https.
[35] Y. Gu. Google Security Blog: Google Public DNS Now Supports DNSSEC Validation. https://security.googleblog.com/2013/03/google-public-dns-now-supports-dnssec.html, 2013.
[36] GLAM: Glean Aggregated Metrics Explorer - Mozilla. https://glam.telemetry.mozilla.org/?
[37] S. Helme. Let's Encrypt to end OCSP support in 2025. 2024. https://scotthelme.co.uk/lets-encrypt-to-end-ocsp-support-in-2025/.
[38] S. A. Krashakov, A. B. Teslyuk, and L. N. Shchur. On the universality of rank distributions of website popularity. *Computer Networks*, 50(11), 2006.
[39] A. Langley. Revocation checking and Chrome's CRL. 2012. https://www.imperialviolet.org/2012/02/05/crlsets.html.
[40] J. Larisch, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. CRLite: a Scalable System for Pushing all TLS Revocations to Browsers. *IEEE S&P*, 2017.
[41] J. Livingood. Comcast Voices: Comcast Completes DNSSEC Deployment. http://corporate.comcast.com/comcast-voices/comcast-completes-dnssec-deployment, 2012.
[42] Y. Liu, W. Tome, L. Zhang, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, A. Schulman, and C. Wilson. An End-to-End Measurement of Certificate Revocation in the Web's PKI. *IMC*, 2015.
[43] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. IETF RFC 2560, IETF, 1999.
[44] P. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034, IETF, 1987.
[45] P. V. Mockapetris and K. J. Dunlap. Development of the Domain Name System. *SIGCOMM*, 1988.
[46] Mozilla: CRLite: Speeding Up Secure Browsing. https://blog.mozilla.org/security/2020/01/21/crlite-part-3-speeding-up-secure-browsing/.
[47] J. Nielsen. Zipf Curves and Website Popularity. https://www.nngroup.com/articles/zipf-curves-and-website-popularity/.
[48] M. Pala. OCSP over DNS (ODIN). draft-pala-odin-03, IETF, 2017.
[49] Revoking Intermediate Certificates: Introducing OneCRL. *Mozilla Security Blog*. http://mzl.la/1zLFp7M.
[50] N. Sullivan. High-reliability OCSP stapling and why it matters. CloudFlare, 2017. https://blog.cloudflare.com/high-reliability-ocsp-stapling/.
[51] R. Sleevi. SECURITY RELEVANT for CAs: The curious case of the Dangerous Delegated Responder Cert. 2020. https://groups.google.com/g/mozilla.dev.security.policy/c/EzjIkNGfVEE/m/XSfw4tZPBwAJ?pli=1.
[52] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 6960 (Proposed Standard), IETF, 2013.
[53] SC-081: Introduce Schedule of Reducing Validity and Data Reuse Periods. https://github.com/cabforum/servercert/pull/553.
[54] S. Trevor, D. Luke, and S. Kent. Let's revoke: Scalable global certificate revocation. *NDSS*, 2020.
[55] The Chromium Projects. https://www.chromium.org/Home/chromium-security/crlsets/.
[56] Verisign will help strengthen security with dnssec algorithm update. https://blog.verisign.com/security/dnssec-algorithm-update/.

[57] P. Wouters, O. Sury, and I. S. Consortium. Algorithm Implementation Requirements and Usage Guidance for DNSSEC. IETF, 2019.
[58] L. Zhang, D. Choffnes, T. Dumitraş, D. Levin, A. Mislove, A. Schulman, and C. Wilson. Analysis of SSL certificate reissues and revocations in the wake of Heartbleed. *IMC*, 2014.

A OCSP URLs

A.1 Alibaba

This is a list of URLs managed by Alibaba that outsource its OCSP service to Akamai.

- ocsp.dcocsp.cn
- ocsp.digicert.cn
- ocsp.trust-provider.cn
- ocsp.global.sheca.com/dvscag5
- ocsp.global.sheca.com/ovscag5

A.2 Fastly

This is a list of URLs managed by Fastly that outsource its OCSP service to Akamai.

- ocsp.int-e1.certainly.com
- ocsp.int-r1.certainly.com

CA	Parent/Peer	Comment	Link
Sucuri	GoDaddy	Sucuri is acquired by GoDaddy and operates as a separate brand	https://wptavern.com/godaddy-acquires-sucuri
Starfield	GoDaddy	Starfield tech is a spin-off company from GoDaddy	https://en.wikipedia.org/wiki/Starfield_Technologies
Actalis	Aruba	Actalis is part of Aruba group	https://www.actalis.com/company.aspx
Geotrust	DigiCert	Geotrust is powered by Digicert	https://digicert.com/resources/DigiCert_GeoTrustGuide_030420.pdf
Thawte	DigiCert	Thawte is now a Digicert brand	https://cheapsslsecurity.com/p/thawte-vs-digicert-comparison
Harica	Aristotle University	Harica is a CA maintained by Aristotle University	https://pki.auth.gr/documents/CPS-EN-4.0.pdf
Telesec	T-systems	Telesec PKI is owned by T-systems	https://telesec.de/assets/downloads/PKI-Repository/PKS-CPS_05.00.pdf

Table 2: To track CA acquisitions over time, we use a public dataset of acquisitions [3]; we also manually identified these seven additional acquisitions not captured in the public dataset.