End-Users Get Maneuvered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks

Shuai Hao*
University of Delaware
haos@udel.edu

Yubao Zhang
University of Delaware
ybzhang@udel.edu

Haining Wang
University of Delaware
hnw@udel.edu

Angelos Stavrou
George Mason University
astavrou@gmu.edu

Abstract

The success of Content Delivery Networks (CDNs) relies on the mapping system that leverages dynamically generated DNS records to distribute client requests to a proximal server for achieving optimal content delivery. However, the mapping system is vulnerable to malicious hijacks, as (1) it is difficult to provide precomputed DNSSEC signatures for dynamically generated records, and (2) even considering when DNSSEC is enabled, DNSSEC itself is vulnerable to replay attacks. By leveraging *crafted* but *legitimate* mapping between the end-user and edge server, adversaries can hijack CDN's request redirection and nullify the benefits offered by CDNs, such as proximal access, load balancing, and Denial-of-Service (DoS) protection, while remaining undetectable by existing security practices including DNSSEC. In this paper, we investigate the security implications of dynamic mapping that remain understudied in security and CDN communities. We perform a characterization of CDN's service delivery and assess this fundamental vulnerability in DNS-based CDNs in the wild. We demonstrate that DNSSEC is ineffective to address this problem, even with the newly adopted ECDSA that is capable of achieving *live signing*. We then discuss practical countermeasures against such manipulation.

1 Introduction

Content Delivery Networks (CDNs) play an important role in the Internet ecosystem by delivering a large fraction of the Internet content to end-users with high availability, performance, and scalability. Typically, CDNs place a large number of edge servers (i.e., *surrogates*) at geographically distributed edge networks, enabling content caching and proximal access for end-users. User requests for content hosted by CDNs are served at the "edge" via request redirection to improve user-perceived performance and balance the load across server clusters. Moreover, CDNs are able to provide a security portal of protection mechanisms against distributed denial-of-service (DDoS) attacks by redirecting users from overwhelmed nodes [23, 84].

The majority of today's CDNs leverage the Domain Name System (DNS) as the core of their mapping systems to redirect a client's request to a nearby edge server. Based on real-time measurement of server and network conditions, a DNS-based mapping system can provide fast, accurate, and fine-grained control for request redirection and thus has been widely used in leading CDN vendors that operate a large number of edge servers such as Akamai. However, such a DNS-based mapping system requires DNS records to be very dynamic, which restrains CDN vendors from authenticating their mapping DNS records by using DNSSEC signatures. Due to its prohibitively high computational overhead, the traditional RSA-based DNSSEC was originally designed for static records and is not a feasible solution to secure dynamic DNS records in the context of CDNs.

In this paper, we conduct a large-scale empirical study to investigate security implications in the DNS-based CDNs, which can be exploited by adversaries to hijack the operation of request redirection in a stealthy manner. Our work makes the following contributions:

- **Illustration of Redirection Hijacking Attacks in CDNs:** We illustrate that an adversary can utilize a *legitimate* mapping record (i.e., a *replayed* message) to override a CDN's server selection and redirect a certain group of users to an edge server chosen by the adversary. Furthermore, even the newly adopted Elliptic Curve Digital Signature Algorithm (ECDSA) that is capable of providing real-time DNSSEC signatures is ineffective to detect and prevent such attacks.

[*] Currently with the Center for Applied Internet Data Analysis (CAIDA) at UC San Diego, performed this work entirely at University of Delaware.

- **Characterization of Operational Practices of Request Routing:** To assess the magnitude of this vulnerability, we characterize the content delivery operations of popular CDN vendors and perform the threat analysis to elaborate on the ineffectiveness of DNSSEC via detailed case studies. We find that 16 out of 20 popular CDNs suffer from various degrees of security threats posed by redirection hijacking. On the other hand, we also notice that CDNs using anycast are not susceptible to such a manipulation.
- **Measurement of Practical Impacts of Redirection Hijacking:** We quantitatively measure the practical impacts caused by redirection hijacking. Moreover, we examine more severe threats, by which adversaries could exploit redirection hijacking to direct end-users to unresponsive edge servers, resulting in the nullification of the CDN's benefits (e.g., DoS mitigation) and the violation of the CDN's service commitments.
- **Challenges and Practical Considerations of Countermeasures:** Finally, we present the challenges of addressing this redirection hijacking from different perspectives, and elaborate on corresponding countermeasures in practice and their limitations.

The remainder of this paper is organized as follows. In §2, we review the background of CDN operations and DNS security. In §3, we present the threat model and the redirection hijacking attack. In §4, we characterize the CDN's operations and perform a large-scale threat analysis, illustrating that DNSSEC is not an effective solution. We then discuss the impact of current practice and potential countermeasures in §5. We survey related work in §6 and finally conclude the paper in §7.

2 Background

2.1 Content Delivery Networks

2.1.1 DNS-based Mapping

The mapping system plays a critical role in the CDN's request routing for directing each client's request to an appropriate surrogate with low latency and sufficient resource capacity. Traditionally, the mapping system uses a client's local recursive DNS resolver (LDNS or rDNS) as the representation of the local area network to determine each client's location. However, this approach has become inaccurate due to (1) the poor location proximity between clients and their LDNSes [63, 73] and (2) the increasing usage of public DNS services. To this end, the EDNS-Client-Subnet (ECS) extension [38] has been proposed to rectify the problem of location discrepancy between clients and their recursive DNS resolvers.

EDNS-Client-Subnet (ECS). With ECS, the network prefix of a client's IP address is included in the option field of a DNS query to enable the DNS-based mapping system to use the direct knowledge of a client's location rather than its LDNS. A recent study by Chen *et al.* [34] showed that Akamai's end-user mapping[1] rolled out by ECS had been providing significant performance benefits for clients behind public DNS services.

Load Balancing. The load balancing module of DNS-based CDNs such as Akamai typically selects proper surrogates by a two-level assignment [34, 62]: global load balancing and local load balancing. The global load balancing relies on network measurements to select a server cluster, typically geographically close to a client's network. Then, the local load balancing assigns the individual server(s) from the chosen cluster, leveraging the combined information such as responsiveness and capacity.

2.1.2 Anycast Routing

The deployment of the DNS-based dynamic mapping requires extra infrastructure and operational support. Therefore, some new CDN providers then enable their CDN platforms with anycast routing, by which multiple distributed endpoints announce the same IP address. BGP routing protocol selects the shortest Autonomous System (AS) path to reach each advertised IP address block, and thus end-users located in different areas will be directed to different topographically-close locations via BGP routing.

Since anycast-based CDNs rely on Internet routing protocols for request redirection, conceptually they are immune to redirection hijacking attacks. However, we observe that in practice some anycast CDNs are also leveraging DNS-based mapping to improve accuracy and performance, making themselves vulnerable to request routing manipulation (§4.3.1).[2]

2.2 DNS Cache Poisoning Attack

The correctness of DNS resolution is the fundamental anchor for the operation and security of the Internet. There-

[1] In [34], the "end-user mapping" is used to dedicatedly describe ECS-based mapping (compared to the *NS*-based mapping which uses LDNSes). To be clarified, in this paper we use "DNS-based mapping" to include ECS-based and NS-based mapping. In most cases, unless specified, we do not differentiate the "DNS-based mapping" and "end-user mapping" since they have identical implications in the context of dynamic mapping.

[2] CDNs may leverage anycast in different strategies: anycasting nameservers or anycasting web servers (or both). Note that our study only involves the way in which a CDN directs users to web servers. Anycasting nameservers means that clients will connect to the nameservers via anycast addresses, but it does not affect the process of end-user redirection. In particular, if a CDN utilizes anycast DNS but DNS-based redirection, it will also be vulnerable to redirection hijacking.

fore, DNS has become an attractive target of adversaries who attempt to exploit DNS for various malicious purposes. One of the most serious threats to DNS is that adversaries trick a resolver to accept fraudulent DNS records as legitimate responses from authoritative nameservers, known as record injection or cache poisoning attacks [24, 30, 53, 77].

DNS cache is intrinsically vulnerable to record injection because a recursive resolver cannot ensure whether a received response is from a legitimate authoritative nameserver or a miscreant entity. The general practical approach for mitigating a cache poisoning attack involves the *challenge-response* defenses [51], including transaction-ID (TXID) randomization, source-port randomization, or the 0x20 encoding [40], in order to enable a resolver to validate the legitimacy of received responses via the randomized values within requests.

Although those countermeasures increase the difficulty of injecting fraudulent records, insufficient adoptions and deployment [44, 46, 74] have continued to make many rDNSes still vulnerable to cache poisoning attacks. Large-scale DNS poisoning attacks are still possible on the Internet [19, 22]. Furthermore, efforts aiming to increase the entropy of DNS queries are only effective against *off-path* attackers; an adversary, which can monitor network traffic and interpret transaction packets, is still able to construct a forged DNS response with correct parameters to bypass all of the challenge-response defenses and pollute the content of cache, i.e., a Man-in-the-Middle (MitM) attack.

2.3 DNSSEC

In order to secure the process of DNS resolution, especially defend against MitM attacks, DNSSEC [28] leverages the digital signatures to validate DNS responses. Within DNSSEC, each resource record set (*RRset*) is signed and verified by public key cryptography: a recipient of a signed RRset (i.e., *RRSIG* record) validates the signature via the public key (i.e., *DNSKEY* record) of the signer. The *trust of chain*, starting from *trust anchor* at root zone, ensures that each key is trusted and able to be validated (via the *DS* record provided by its parent zone to authorize the *DNSKEY* that is used to sign the RRset).

DNSSEC Zone Enumeration. With DNSSEC, to provide authentication for negative responses (i.e., authenticated denial of existence), a Next-SECure (NSEC) record lists and signs a pair of lexicographic consecutive names in the zone, indicating that no names exist between the NSEC's owner name and the "next" name. However, NSEC records expose the existence of names in the zone, which then allows adversaries to enumerate NSEC records and walk through the zone space to learn all of the (sub)domains and associated IP addresses (i.e., the zone enumeration attack), resulting in undesired policy violation or more complex attacks [59].

In order to make the zone enumeration more difficult, the alternative NSEC3 record [59] lists the cryptographically hashed names rather than valid (sub)domain names. However, it is still vulnerable when adversaries apply an dictionary attack by querying non-existent names and guessing real names [12, 43]. Thus, NSEC5 [43] is then proposed to replace the NSEC3's *unkeyed* hash with a new *keyed* hash generated by separate secondary keys.

Another technique to mitigate zone enumeration is "On-line Signing" [76, 86] (i.e., "White Lies" [42]). Instead of disclosing real domains or pre-computed hashes, on-line signing creates on-demand signature, proving non-existence for a specific name by listing its derived predecessor and/or successor. However, this approach has two major drawbacks [86]: (1) with the traditional RSA algorithm, it introduces significant computational load for authoritative nameservers to generate the real-time signatures, resulting in potential DoS attacks, and (2) the primary private keys must be distributed among nameservers, increasing the risk of key leakage.

Live Signing by ECDSA. To mitigate zone enumeration and DNSSEC amplification attacks [82], Elliptic Curve Digital Signature Algorithm (ECDSA) [47] has been employed as an alternative cryptosystem for DNSSEC [83]. Different from the traditional RSA-based scheme, ECDSA leverages the Elliptic Curve Cryptography (ECC) to generate signatures with reduced computational overhead and signature size. While the process of validating an ECDSA signature is slower than that of validating an RSA signature [47, 80], the significantly reduced computational overhead (about 10 times faster in signing [13]) enables ECDSA to sign all of the necessary RRSIG records "on-the-fly" (i.e., *live signing*), providing a practical solution in the context of dynamically generated records at the "edge" of the Internet. The support for the ECDSA signing algorithm in CloudFlare [13] has demonstrated a real case of global ECDSA-based DNSSEC adoption in large CDN platforms.

3 Threat Model

Attacker Model. The key feature of a redirection hijacking attack is that an adversary can inject *crafted* but *legitimate* records into a recursive DNS resolver to manipulate the dynamic mapping inside CDNs. Essentially, our attacker model is the same as that of DNSSEC. On one hand, an off-path adversary is able to bypass the challenge-response mechanism by guessing the authentication parameters (i.e., source-port number and TXIDs) via different effective techniques (e.g., fragmentation at-

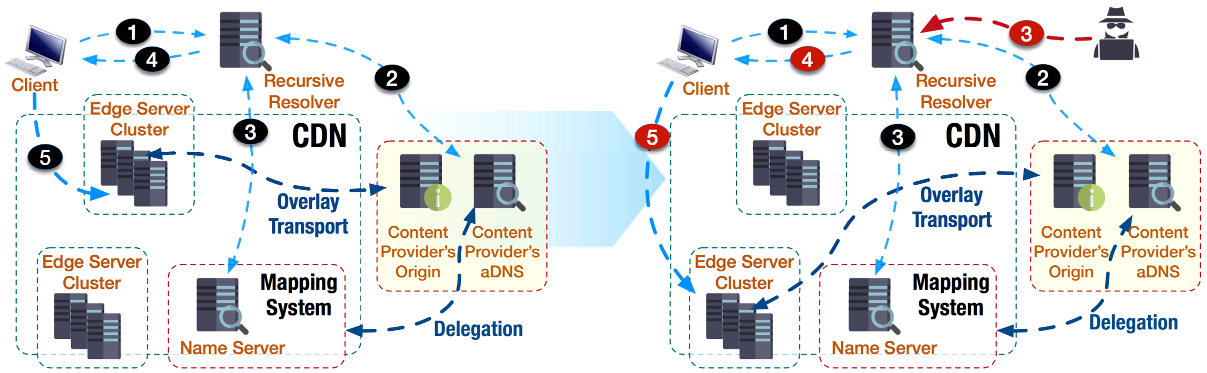

Figure 1: Illustration of a Redirection Hijacking Attack. (An adversary replays and injects a legitimate record associated with suboptimal or non-responsive edge servers, resulting in the maneuvered end-user redirection while still passing DNSSEC validation.)

tacks [45, 74] or socket overloading [46]) against the insufficient randomization or vulnerable implementations [44]. On the other hand, a MitM attacker can easily bypass the countermeasures of randomization by sniffing network packets and observing those parameters. Furthermore, we assume that an adversary can inject *legitimate* records into DNS caches, regardless of whether DNSSEC is used since DNSSEC itself is vulnerable to replay attacks [29]. A recent work [55] demonstrated that, with the feasibility of exploiting MitM attacks and parameter-guessing techniques, more than 92% of current DNS platforms on the Internet are still vulnerable to record injection; even popular public DNS platforms are vulnerable to *indirect injection*, in which a poisonous record is injected in advance and becomes effective after other records expires.

Within CDNs, we assume that adversaries do not need to harvest surrogate servers [33, 79] or profile CDN's mapping algorithm; they only need to use selective mapping records to override the CDN's server selection.

Redirection Hijacking Attack. In comparison to the normal operations of a DNS-based mapping system in CDN, Figure 1 illustrates how redirection hijacking attack works: an adversary exploits the dynamic end-user mapping to manipulate an end-user's access to edge networks. Normally, the content provider delegates its name resolution to the CDN vendor's mapping system, typically via either CNAME redirection as shown in Figure 1 or directly hosting the NS records in CDNs. When a client's request for a content object (❶) is redirected into a CDN's nameserver (❷), the mapping component examines the incoming queries (e.g., the client's IP prefix in ECS), performs real-time topological mapping based on network measurements, and returns an optimized assignment (❸ $\sim$ ❹) that directs the client to a close, responsive edge server [34] (❺).

Since the dynamic mapping between end-users and edge servers makes it impractical to pre-sign a mapping record with the traditional RSA-based DNSSEC, we also consider that the ECDSA could be used as an alternative solution to provide on-demand signatures for those dynamically generated records in CDNs. However, even mapping records with ECDSA signatures are still vulnerable to redirection manipulation. This is because (1) in operational practices, the validity period of a DNSSEC signature (including ECDSA) should be long enough[3] to enable easy administration and avoid query load peaks (see §4.4.1 in RFC 6781 [56]), and (2) the validation of the DNSSEC signature cannot detect whether a message is forwarded or replayed to a different recipient by a third party. An adversary can simply fetch a legitimately signed mapping record that was used or is being used for a different client's network and inject it into resolver's cache. Because the injected record, which is generated by a legitimate authoritative nameserver but for a different group of clients, carries a valid signature, the resolver will accept it for caching after a successful signature validation. Once the injected record is accepted, client requests will be redirected to a non-optimal edge server chosen by the adversary, typically heavily loaded and geographically distant from clients, or even to an unresponsive edge server to interrupt client access to the service hosted by CDNs. Also, an adversary could exploit the same record replayed for many clients to potentially mount DoS attacks on targeted edge servers (§4.4).

We further note that such an attack can be successful even in the environments with strong security settings. Due to the nature of replay attacks in redirection hijacking, neither the client end nor resolver signature validations can detect such manipulation.

[3] Cloudflare's ECDSA-based signatures have a validity period of two days. The expiration time of the traditional RSA-based DNSSEC signature in practice is normally set to one month [56].

4 Attack Assessment

To assess the magnitude of redirection hijacking in CDNs, we present the characterization of the CDN's request routing and conduct a detailed threat analysis to demonstrate the vulnerability of DNS-based CDNs to the manipulation, even with DNSSEC. Then, we quantitatively measure the practical impacts and explore the more serious threats posed by the redirection hijacking, which nullify the CDN's load balancing and DoS protection.

4.1 Methodology

In order to identify the CDN platforms that are vulnerable to redirection hijacking, we measure popular commercial CDNs across the Internet to characterize their configurations and operations. To do so, we set up virtual machines in different Amazon EC2 regions (us-east-1, us-west-2, ap-northeast-1, ap-southeast-2, ap-south-1, eu-central-1, eu-west-1, and sa-east-1, as shown in Figure 2) as a group of geographically distributed vantage points to retrieve DNS resolution results for customer websites hosted in each CDN provider. Then, we examine the request routing strategies and analyze practical impacts and more serious threats.

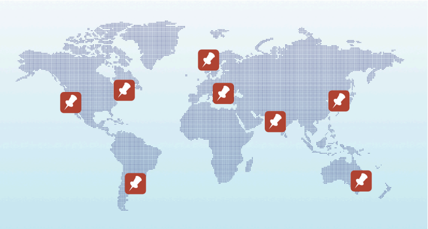

Figure 2: Vantage Points for Resolution

More specifically, we empirically investigate the patterns of content delivery for CDN vendors by taking the following steps:

- First, we simply search through official blog articles, technical documents, and announcements published by each CDN vendor as well as external technical blogs (e.g., [1, 2]) to learn the details of content delivery mechanisms.
- We then verify our findings by studying DNS configurations and resolution results from distributed vantage points for a list of customers of each CDN provider, which are gathered by available utilities (e.g., [3, 4]) and the customer list/case studies presented on CDN websites. For example, an identical A RRset should be fetched from different locations when global anycast routing is utilized, and diverse A RRsets are observed when DNS-based dynamic mapping is used.
- Finally, we crosscheck the information of domain names and IP addresses acquired from DNS resolution via publicly available passive DNS databases [5, 25] to validate if the patterns of content delivery inferred from resolution results are compatible with the records stored in passive DNS databases.

4.2 Characterization Overview

Request routing in CDNs mainly consists of two consecutive steps[4]: *domain delegation* and *surrogate selection*. In the domain delegation, the Content Providers (CPs) delegate the domain resolution to CDN vendors. In the surrogate selection, CDNs redirect a client's request to a proximal edge server. In essence, these two steps determine how CDNs enable their service infrastructures to be located and accessed by end-users. Thus, we characterize CDNs' request routing with respect to these two redirection steps. Table 1 summarizes the request routing and DNSSEC provision in popular CDN vendors.

Domain Delegation. The domain delegation is used to forward each client's request from the origin of CPs to a CDN's platform. The most common domain delegation mechanisms are CNAME redirection and NS hosting.

- **CNAME Redirection**: The CNAME record enables a domain name to be resolved via an alias. By pointing a CP's domain to a domain provisioned by CDN via CNAME, a client's request will subsequently be redirected to a CDN's domain name and resolved by the CDN's nameservers.
- **NS Hosting**: An alternative approach of domain delegation is to designate CDN-provided authoritative nameservers in the NS records of a DNS referral response, which is generated by the CP's authoritative nameservers and then is received by clients. Consequently, the DNS resolution of the CP's domain will be fully operated by the CDN.

From Table 1, we can see that all CDN vendors provide CNAME redirection to enable CPs to delegate the DNS resolution to CDNs. Only three CDN vendors support the NS hosting for domain delegation. Given the prevalent use of CNAME in CDNs, however, we note that the integrity of CNAME records has been widely disregarded on the Internet. This is because (1) typically, the first-level front-end CNAME redirection occurs at the CP's authoritative nameserver, which is mainly out of the control of CDN vendors, (2) the CPs lack motivation to sign CNAME records at their authoritative nameservers

[4] The higher-level techniques of request routing [31] such as application-level request routing are only suitable for large-file delivery due to extra latency [34], and thus we only consider those techniques when discussing countermeasures (see Section 5.4).

Table 1: Characterization of CDNs' Request Routing and DNSSEC Provision. (The “`DNSSEC (A)`” column refers to the effectiveness of securing the records with DNSSEC (“✓”- providing DNSSEC signing to customers; “`Feasible`”- capable to secure non-dynamic DNS records with DNSSEC in anycast-based CDNs; “×”- unable to mitigate the replay attack with DNSSEC due to the dynamics of DNS mapping). The “●” indicates that adversaries may be able to manipulate end-user redirection, which results in serious damage (§4.4). The “○” indicates that the record suffers from limited forms of dynamic vulnerability that may not cause serious threats such as service interruption.)

CDN	Domain Delegation	Surrogate Selection	DNSSEC (A)	Dynamics	
				CNAME	A
Akamai	CNAME Chain	DNS-based Mapping (ECS)	×		●
Cachefly	CNAME/NS Hosting	Anycast Routing	Feasible		
CDN.net	CNAME	DNS-based Mapping	×		●
CDN77	CNAME	DNS-based Mapping (ECS)	×		●
CDNetworks	CNAME	DNS-based Mapping (ECS)	×		●
CDNlion	CNAME	DNS-based Mapping	×		●
CDNsun	CNAME	DNS-based Mapping	×		●
ChinaCache	CNAME/CNAME Chain	DNS-based Mapping (ECS)	×		●
CloudFlare	CNAME/NS Hosting	Anycast Routing	✓		
CloudFront (Amazon)	CNAME/NS Hosting	DNS-based Mapping (ECS)	×		●
EdgeCast (Verizon)	CNAME/CNAME Chain	Hybrid Type I	Feasible		○
Fastly	CNAME	Hybrid Type II	×		●
Highwinds	CNAME	Anycast Routing	Feasible		
Incapsula	CNAME	Hybrid Type I	Feasible		○
KeyCDN	CNAME Chain	DNS-based Mapping (ECS)	×	●	●
LeaseWeb	CNAME	DNS-based Mapping	×		●
Limelight	CNAME	DNS-based Mapping	×		●
MaxCDN/NetDNA	CNAME	Anycast Routing	Feasible		
Rackspace	CNAME Chain	DNS-based Mapping (ECS)	×		●
cedexis (*MultiCDN*)	CNAME Chain	N/A	×	●	

due to the dynamics of mapping records in the following surrogate selection, (3) in some cases, dynamic CNAME mapping exists in CDNs (see §4.3.1), and (4) many CDN vendors leverage multiple CNAME records (i.e., CNAME chain in Table 1) to facilitate their platform management (e.g., enabling customers to adopt various services by being mapped to different CNAMEs), which means that traversing signed CNAME records is significantly expensive for recursively validating DNSSEC signature for each CNAME record. We will discuss the technique of “CNAME Flattening” in §5.3 to mitigate the security threat of CNAME in CDNs.

Surrogate Selection. The surrogate selection falls into two fundamental approaches: DNS-based and anycast-based. Table 1 shows that the DNS-based mapping is still dominant in CDNs and the ECS has been widely supported, especially for those vendors operating a large-scale infrastructure, such as Akamai and Amazon. However, more recent vendors are more likely to build their platforms with anycast routing to leverage its easy and robust deployment. We also observe that some CDN vendors have employed a different hybrid system design by leveraging both DNS-based mapping and anycast routing to improve the performance of their global content deliveries. In the following section, we will elaborate on those different patterns for the operations of request routing and analyze the security threat of redirection hijacking caused by the dynamic surrogate selection and the ineffectiveness of DNSSEC via case studies.

4.3 Threat Analysis

4.3.1 DNSSEC (Live Signing) is NOT a Solution: Case Studies

DNSSEC is proposed as a foundational system-wide solution to DNS vulnerabilities, especially for the record injection by MitM attacks. Here we depict detailed case studies to analyze the vulnerability under different CDN deployment patterns. We demonstrate the infeasibility of providing pre-computed DNSSEC signatures in the dynamic context of DNS-based CDNs. As discussed in §2.2, the root cause is that the traditional RSA-based signature algorithm cannot achieve on-demand signature in real-time due to its high computational cost.

Subsequently, for these case studies, we also examine scenarios in which all necessary signature operations can be efficiently performed. To do so, we assume that (1) CNAME records would be secured by adding corresponding signatures and that (2) CDNs are able to generate on-demand DNSSEC signatures to sign dynamic mapping records efficiently, such as the ECDSA-based implementation that has been used in Cloudflare [13].

Case Study of End-User Mapping: Akamai. Exemplified by Akamai, Figure 3 shows a typical resolution chain by CNAME redirection and the end-user mapping system rolled-out by ECS [34]. Specifically, the CP's domain is first translated to a domain provisioned by Akamai's CDN via CNAME. Afterward, the CDN's nameservers take over the resolution, and finally an A record is dynamically generated by the end-user mapping subsystem to assign an edge server with optimized performance such as responsiveness and capacity, based on the location estimation of the end-user's IP address carried in ECS extension.

Due to the diversity of mapping records and more than 240,000 servers within more than 1,700 networks in Akamai's CDN [8], it is inefficient and impractical to pre-determine or predict the server assignment for each customer and provide a pre-computed DNSSEC signature, resulting in the fundamental vulnerability to record injection attacks. An adversary is able to exploit this vulnerability to hijack redirection and mislead end-users to a different domain controlled by the adversary. We note that such a threat can be mitigated by employing ECDSA-based signature, as ECDSA is capable of dynamically signing the records. However, given the adoption of ECDSA, the dynamic mapping is still vulnerable to redirection hijacking attacks as mentioned in §3.2.[5]

Case Study of Anycast: Cloudflare. Anycast announces the same IP address(es) from multiple locations and relies on BGP to perform front-end redirection. Therefore, the CPs leveraging anycast-based CDNs would have identical A record(s), which are static, and thus the anycast-based CDNs are able to secure the integrity of RRsets with either ECDSA-based or pre-computed RSA-based signatures. This makes the anycast-based CDNs immune to redirection hijacking.

The examples below show the configurations of Cloudflare with the domain delegation of CNAME and NS hosting, respectively. In both cases, the returned signed A records are with the global anycast addresses, and hence there is no risk of redirection hijacking. However, we also notice that although DNSSEC is enabled, the integrity of an upstream CNAME record, which is typically out of the CDN's control, has been widely disregarded by customers, leading to the risk of domain hijacking via CNAME.

```
$ DNS resolution for domain using NS Hosting
  filippo.io.        NS        beth.ns.cloudflare.com.
  filippo.io.        NS        jim.ns.cloudflare.com.
  filippo.io.        DS        ...
  filippo.io.        RRSIG     DS     [ECDSA signature]

  blog.filippo.io.   A         104.20.145.15
  blog.filippo.io.   A         104.20.144.15
  blog.filippo.io.   RRSIG     A      [ECDSA signature]
```

```
$ DNS resolution for domain using CNAME
  www.martindale.com.       CNAME     www.martindale.com.cdn.cloudflare.net.
  www.martindale.com.cdn.cloudflare.net .     A      104.18.60.26
  www.martindale.com.cdn.cloudflare.net.      A      104.18.61.26
  www.martindale.com.cdn.cloudflare.net.  RRSIG  A     [ECDSA signature]
```

Note that ECDSA provides Cloudflare with the solution to sign its records "on-the-fly" at the edge, but its invulnerability to end-user manipulation is mainly due to anycast routing rather than ECDSA signing.

Case Study of Hybrid Type I – Regional Anycast: Incapsula. Incapsula enables a hybrid strategy for request routing, in which DNS-based mapping is used to preliminarily determine the geographic area of end-users and a *regional anycast* address is used to serve a specific region. A world-wide network is divided into different regions (typically 5-7 regions based on the continents) and within each region, identical anycast addresses are advertised and used to direct end-users in this region to a close point-of-presence (PoP).

Figure 4 illustrates an example of a global network using regional anycast and its susceptibility to redirection hijacking. Even with the adoption of DNSSEC, similar to DNS-based redirection, an adversary can inject a legitimate anycast record assigned to clients from a different region, directing victim users to edge servers that are located in another continent.

Case Study of Hybrid Type II – Separate Anycast and Unicast: Fastly. Instead of adding ECS support, Fastly addresses the problem of location discrepancy in a different hybrid strategy: (1) in a normal case, the traditional NS-based mapping is utilized to direct end-users to close PoPs; (2) anycast addresses are used to answer the queries from public DNS resolvers. Under such a strategy, end-users behind ISPs leveraging centralized DNS

[5] It is worth noting that DNS-based CDN vendors could also provide anycast-based DNS-hosting services and optional DNSSEC signature (e.g., Akamai's Fast DNS [9]). However, this type of service aims to protect the DNS infrastructure only; if a customer enables the content delivery, dynamic A records are still used to direct end-users to edge servers and thus cannot be protected by DNSSEC.

```
www.dell.com.                                     3600   IN  CNAME  www1.dell-cidr.akadns.net.
www1.dell-cidr.akadns.net                         3600   IN  CNAME  cdn-www.dell.com.edgekey.net.
cdn-www.dell.com.edgekey.net.                     21600  IN  CNAME  cdn-www.dell.com.edgekey.net.globalredir.akadns.net.
cdn-www.dell.com.edgekey.net.globalredir.akadns.net.  3600   IN  CNAME  e28.x.akamaiedge.net.
e28.x.akamaiedge.net.                             20     IN    A    104.117.80.33
```

Figure 3: An Example of DNS-based End-User Redirection by CNAME (Akamai)

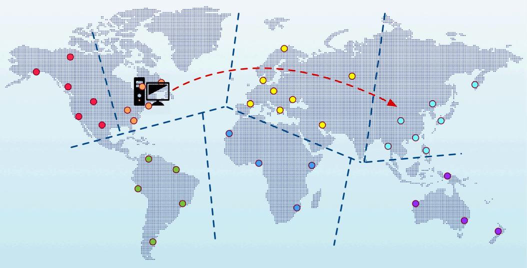

Figure 4: Illustration of Redirection Hijacking with Regional Anycast. (The global platform is divided into different regions, each of which leverages the anycast routing within the region. A redirection hijacking can force end-users to access the suboptimal or unresponsive edge servers located within a remote region.)

infrastructures will still suffer from the problem of location discrepancy. Moreover, clients that do not use public DNS services are vulnerable to redirection hijacking, as in the case of DNS-based mapping.

Case Study of Dynamic CNAME: KeyCDN. Unlike other DNS-based CDNs, KeyCDN leverages CNAME to map the CP's domain to a close PoP first and then assign an appropriate edge server within the PoP via `A` records.

```
$ DNS resolution from us-west
  ja.onsen.io.                CNAME   jaonsenio-4ecf.kxcdn.com.
  jaonsenio-4ecf.kxcdn.com.   CNAME   p-usse00.kxcdn.com.
  p-usse00.kxcdn.com.           A     76.164.234.2

$ DNS resolution from us-east
  ja.onsen.io.                CNAME   jaonsenio-4ecf.kxcdn.com.
  jaonsenio-4ecf.kxcdn.com.   CNAME   p-uswd00.kxcdn.com.
  p-uswd00.kxcdn.com.           A     107.182.231.101
```

The dynamic CNAME mapping introduces another potential attack vector for redirection hijacking via CNAME records. Similar to hijacking a dynamic `A` record, an adversary could inject a legitimate CNAME record associated with a remote non-optimal PoP to degrade the user-perceived performance, even under the availability of DNSSEC live signing enabled by ECDSA.

On the other hand, with the DNSSEC, redirection hijacking for dynamic `A` records would not cause significant performance degradation because all valid `A` records are being mapped to IP addresses within the nearby PoP assigned by CNAME. However, adversaries can still leverage legitimate records to redirect users to IP addresses of unresponsive edge servers within PoP to nullify the DoS protection and interrupt end-user access for the victim service.

Case Study of Multiple-CDN Deployment: Cedexis. We then investigate the deployment with multiple CDN providers (*a.k.a. CDN Brokers* [65, 66]). A typical deployment pattern of multiple CDNs leverages Global Traffic Management (GTM) as the first-level redirection, in which the GTM platform directs end-users to a selected appropriate CDN provider:

```
$ DNS resolution from us-east
  www.lequipe.fr.                     CNAME   2-01-273c-0023.cdx.cedexis.net.
  2-01-273c-0023.cdx.cedexis.net.     CNAME   lequipe-fr.lequipe.netdna-cdn.com.
  lequipe-fr.lequipe.netdna-cdn.com.    A     94.31.29.248

$ DNS resolution from ap-northeast
  www.lequipe.fr.                     CNAME   2-01-273c-0023.cdx.cedexis.net.
  2-01-273c-0023.cdx.cedexis.net.     CNAME   www.lequipe.fr.edgekey.net.
  www.lequipe.fr.edgekey.net.         CNAME   e7130.g.akamaiedge.net.
  e7130.g.akamaiedge.net.               A     104.116.83.6
```

In the example above, Cedexis's GTM platform [11] is responsible for choosing an appropriate CDN vendor according to the location of a client and the real-time performance of CDNs in this area. As such, the diversity of `A` records depends on the strategy of each CDN's request routing. Clients directed by NetDNA would not be vulnerable to redirection hijacking for `A` records due to the use of global anycast (assuming there are signed anycast `A` records), but clients directed by Akamai will be at the risk of hijacked redirection mappings.

Since the selection of CDN providers is performed via dynamic CNAME redirection, live-signing DNSSEC for CNAME cannot prevent adversaries from injecting legitimate records to redirect users to arbitrary non-optimal CDN providers, nullifying performance improvements offered by the GTM and CDN platforms.

Summary. The vulnerability of CDNs to redirection hijacking stems from the dynamics of DNS records used for request routing, which gives adversaries a chance to maneuver CDN's user redirection by injecting crafted but legitimate DNS records. We summarize the features of dynamic mapping for CNAME and `A` records in Table 1. The DNS-based CDNs are widely vulnerable to redirection hijacking, but CDNs using global anycast for request routing are immune to such an attack due to the static mapping of DNS records. Specifically, Cloudflare

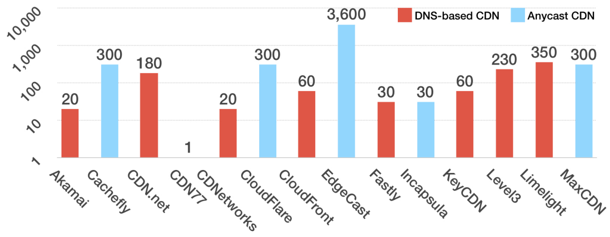

Figure 5: TTL (seconds)

is the only CDN vendor providing DNSSEC signatures for A records to its customers, by leveraging its global anycast routing and ECDSA-based DNSSEC implementation. Also, we consider other CDN vendors with anycast routing of being capable of supporting DNSSEC signatures without DNS dynamics, labeled as "Feasible" in Table 1.[6]

4.3.2 TTL

We list the TTL values of DNS records for surrogate assignment in Figure 5. The DNS-based CDNs use shorter TTL values in their dynamic A records for fast traffic redirection and load balancing, typically less than 300 seconds. Most of anycast CDNs have the TTL values of A records at 300 seconds while Edgecast has a larger value at one hour, and Incapsula leverages a short value at 30 seconds.

The length of TTL in a normal DNS record has a significant impact on the possibility of DNS poisoning because short TTLs force the recursive resolver to more frequently perform DNS lookups, which grants adversaries more chances (i.e., more frequent "windows of opportunity") to perform record injections [51]. With DNSSEC enabled, adversaries can craft records based on legitimate records with valid signatures that are re-used or replayed. Thus, the prevalent use of short TTL values in normal DNS records essentially increases the possibility of injecting replayed records.

On the other hand, since CDNs typically utilize short TTLs in dynamic mapping records and adversaries usually intend to use larger TTLs in injected records to cause more damage, intuitively, a dynamic record with a large TTL value may indicate that it is highly likely to be a crafted mapping. However, popular large-scale passive DNS databases do not enable their sensor servers to capture the TTL in the traces so that such a manipulation might not be detected via passive DNS databases.

4.3.3 Performance Impact

We analyze the performance impact caused by redirection hijacking in which adversaries inject crafted records to deliberately direct end-users to a geographically distant non-optimal site.

Performance matters. User experience is extremely important to the business of CPs, especially eCommerce sites [34, 10]. Thus, the performance benefits provided by CDNs become critical to CPs. A prior work [27] observes that even little differences in CDN's performance could cause significant financial gain/loss.

Performance metrics. Similar to the study [34], we measure the following metrics to characterize the potential performance impact when an end-user is diverted from optimal edge servers by redirection hijacking.

- Round-Trip-Time (RTT): RTT measures the propagation delay when a packet traverses the networks, which indicates the quality of the selected network path and is significantly dominated by the distance between two endpoints.
- Time-to-First-Byte (TTFB): TTFB measures the amount of time between when the first byte of requested content is received and when the client issues the request.
- Content Download Speed: Unlike the study [34] that leverages the Real User Measurement (RUM) system to measure the web page download time, we use the file download speed measured by the curl utility because curl does not support concurrent connections for embedded contents in web pages.

Methodology. We leverage the DNS records obtained via the probes from distributed Amazon regions as shown in Figure 2, and use the same technique for launching a cache penetrating attack presented in [79], in which the curl utility is used to bypass CDN's server assignment by replacing the normal host header with a (distant) non-optimal IP address in HTTP requests. A recent work [35] verifies that such a technique still works for all CDNs in their study. For example, to fetch a content object from an edge server located in Asia as the representation of end-users on the east coast of the United States, we issue the following request at a host in the Amazon region of us-east-1:

```
curl -H Host:i.dell.com -O http://104.78.87.26/
     sites/imagecontent/products/...inspiron
     -15-7000-gaming-pdp-polaris-01.jpg
```

Our experiments are specifically performed based on Akamai's CDN platforms. We manually obtain a list of

[6] Note that the DNSSEC provision summarized in Table 1 involves only the capacity of signing the CDN-issued records for request routing; CPs may still be able to sign their records for origin sites, but request routing would not be protected by their signatures since the mapping records will be provided by CDNs. We argue that this has been a foundational obstacle for the DNSSEC adoption on the Internet, especially for the top websites leveraging (DNS-based) CDNs to provide worldwide services.

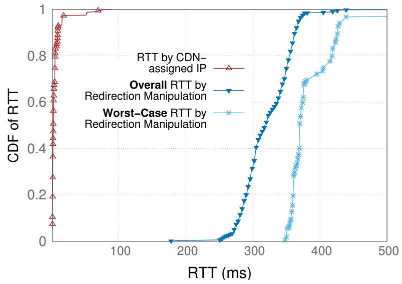

Figure 6: CDF for the RTT

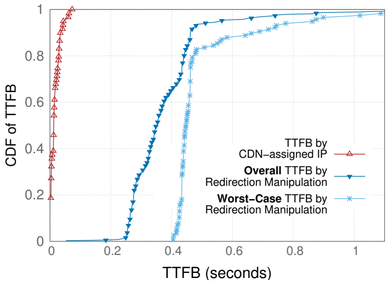

Figure 7: CDF for the TTFB

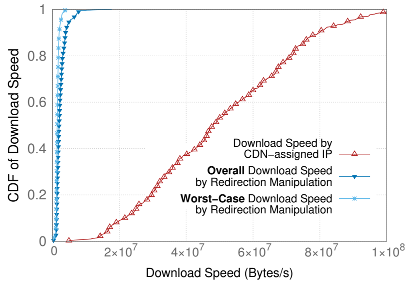

Figure 8: CDF for Download Speed

content objects from popular CDN-hosted sites (dell.com, apple.com, and walmart.com), including static web pages (.html and .css), dynamically generated web pages (embedded search keywords in URLs), images, documents, and medium-sized download files, with a variety of sizes from 500K to 50M. We download those web contents by using the `curl` utility to evaluate the performance impact experienced by end-users under redirection hijacking.

For each metric presented above, we report measured results associated with the optimal surrogate assignment and redirected non-optimal surrogates, respectively. In addition, we identify a redirected site with the most significant performance degradation for each vantage point, plotted as the worst cases in Figures 6-8.

Round-Trip-Time. RTT is a purely underlying network latency and the most straightforward performance metric of a network connection and user experience. Figure 6 shows that for the optimal assignment of the CDN's mapping system, RTTs are mostly less than 20ms; but hijacked redirections typically significantly increase RTT latency to around 300ms, and in the worst case, RTTs are increased to around 350 to 450ms.

Time-to-First-Byte. Since TTFB involves the network latency and aspects that are not affected by mapping decisions (e.g., the construction and compression of a web page), we only include the results for web pages. Figure 7 illustrates similar impacts of TTFB in comparison to RTT. Note that our results show lower TTFBs than the results reported in [34], probably due to the web pages we requested being less dynamic.

Download Speed. Figure 8 shows measured speeds for file downloads. Results from optimal mapping decisions vary, but the cases under redirection hijacking show a significant decrease in their file download performance.

4.3.4 Scope of Impact

As discussed before, both CNAME and `A` records for the CDN's request routing could be exploited by redirection hijacking. We then study whether hijacking a single record can cause collateral damage for other domains. Table 2 summarizes the scope of impact for those CDNs vulnerable to redirection hijacking. If CNAME records are unsigned, hijacking a CNAME record itself will just affect the domain associated with this record in all cases, since in these CDNs, there is no canonical name being reused among CPs. In other words, there is no shared name appeared on the "left-side" of a CNAME record. However, if CNAME could be signed, only KeyCDN's dynamic CNAME poses the threat of hijacking a single domain. Meanwhile, in some CDNs, there could be multiple (sub)domains being mapped to the same CNAME alias (i.e., a shared name appears on the "left-side" of an `A` record), and thus hijacking such `A` records would have collateral damages for those "co-resident" (sub)domains.

4.3.5 Domain Sharding

The *domain sharding* (or *content segregation*) [7] technique is typically used to increase the amount of simultaneous connections by utilizing multiple domains. For example, www.dell.com is directed to e28.x.akamaiedge.net, but all embedded images are served via i.dell.com, which is directed to e28.g.akamaiedge.net. Although this technique also distributes connections to different domains among multiple edge servers, in such a case, poisoning a portal domain (i.e., www.dell.com) is sufficient to affect the accessibility of most end-users.[7]

4.3.6 Impact of CDN Caching

In addition to the issues discussed above, we are aware of that redirection hijacking may also have a subtle impact on the caching system. The caching system is an important building block of a CDN's infrastructure, providing accelerated access for static and popular content. The cache-hit ratio is a critical metric to the CDN's performance, since a cache miss may cause extra latency for fetching requested content from a remote origin server as well as induce more network traffic and server workload.

[7] Note that domain sharding would become unnecessary under the adoption of HTTP/2 (SPDY) which supports concurrent requests.

Table 2: Impact of a single record hijacking (CDNs with global anycast that are immune to the redirection hijacking have been excluded).

CDN	CNAME		A (signed)	
	Single Domain (unsigned)	Single Domain (signed)	Single Domain	Multiple Domain
Akamai	✓		✓	
CDN.net	✓		✓	
CDN77	✓		✓	
CDNetwork	✓		✓	
CDNlion	✓		✓	
CDNsun	✓		✓	
ChinaCache	✓		✓	✓
CloudFront	✓		✓	
EdgeCast	✓		✓	✓
Fastly	✓		✓	✓
Incapsula	✓		✓	
KeyCDN	✓	✓	✓	✓
LeaseWeb	✓		✓	
Limelight	✓		✓	✓
Rackspace	✓		✓	

The popularity of requested contents on the Internet shows strong localization. In other words, redirected end-user groups may be highly likely to have totally different interests in web content. Thus, manipulated redirection would cause previously cached content to be rapidly expelled and limited caches at edge server to be frequently updated, consequently resulting in degraded performance and user experience. Also, the decreased cache-hit ratio will significantly increase the bandwidth costs of CPs for delivering content to numerous clients [21]. Finally, increased back-end connections to origin servers for fetching requested content will further slow down server responsiveness.

4.4 More Serious Threats

We further explore more serious threats of redirection hijacking for maneuvering end-user access in CDNs. Technically, CDNs have the natural capability to absorb and diffuse attack traffic with geographically distributed edge networks, and thus they become an ideal infrastructure to integrate enhanced security mechanisms, in which the edge servers can (1) act as reverse proxies to inspect incoming traffic and apply the rules of Web Application Firewalls (WAFs) to filter out malicious traffic and (2) perform load balancing and DoS protection by diverting users from overwhelmed edge servers via DNS-based dynamic mapping or anycast routing.

Adversaries could exploit redirection hijacking to launch a (or parts of a) DoS attack by directing requests from a large number of clients to a single IP address of the victim edge server. WAFs cannot discard those legitimate traffic from real clients. By selectively injecting the DNS records associated with different popular contents, more clients are connecting to the victim edge server, and then the server must maintain more back-end connections to different origin servers to fetch the content. Also, cached contents are quickly being replaced due to a high volume of traffic for massive contents. Sooner or later, the victim edge server become overloaded and unresponsive to client requests. More importantly, load balancing cannot appropriately distribute the traffic since clients are bypassing the mapping system. Subsequently, all clients that are redirected to the overloaded edge server will not be able to access the contents or services hosted by the CDN anymore.

Furthermore, adversaries can leverage the system failure or outage to significantly amplify their attacks. For example, we sent `ping` probes to monitor the liveness of edge servers for two weeks with IP addresses that have been obtained from our experiments for DNS resolution presented in Section 4.1. We found that 4.5% of IP addresses become unresponsive during the tests, around half of which do not come back online by the end of our experiments. With the easy detection for unresponsive edge servers, adversaries do not need to perform the actual DoS attack and can simply interrupt end users' accessibility by replaying legitimate mapping records associated with those unresponsive edge servers to resolvers.

5 Countermeasures

In this section, we discuss the practical factors affecting vulnerability and countermeasures for detecting or mitigating redirection hijacking attacks.

5.1 ECS Considerations

The introduction of EDNS-Client-Subnet provides DNS-based CDNs an attractive scheme to improve the accuracy of their mapping systems and user-perceived performance for clients using public DNS or the resolvers distant from their locations. As mentioned before, the presence or absence of the ECS option does not affect the vulnerability we studied in this paper. The standardized document [38] does not discuss the difficulty of signing dynamic mapping records. Also, according to the document, the EDNS0 extension does not change the behavior of data authentication, i.e., the ECS data will not be signed by DNSSEC.

On the other hand, ECS indeed provides another attack vector for DNS abuse. For example, the *scope netmask* carried in ECS indicates the specific IP block associated with a reply. An adversary may be able to selectively poison a resolver's cache to impact only a specific IP range [54] via a fraudulent record directing clients to a malicious address. However, such an activity can be detected if the record is signed by DNSSEC (assuming that either ECDSA is used or only a limited number of mapping records exist so that the signatures can be pre-computed). Furthermore, if adversaries exploit redirection hijacking to maneuver end-user mapping for tussling the CDN's performance or interrupting a service, they could arbitrarily designate ECS data to impact more clients by using a less detailed network prefix.

Countermeasures. As discussed in §4.3, the root cause for why even the live-signing DNSSEC is not effective against redirection hijacking is that the resolvers cannot detect a legitimate but replayed mapping that is supposedly used for a different group of clients. Thus, assuming the ECS is enabled, one potential mitigation is to include ECS data in DNSSEC when signing RRsets. With ECDSA, the records generated by the end-user mapping can be dynamically signed on demand. Then, the signed ECS can guarantee that the IP address is assigned to the specified user group (ECS data) since adversaries cannot craft a valid record with an arbitrary client-subnet.

Limitations. ECS is suggested to be enabled only when clear advantages can be seen by resolvers [38], e.g., open DNS resolvers or a centralized DNS infrastructure serving clients from a variety of geographically distributed networks. Meanwhile, in current practice, CDN vendors typically enable ECS by whitelisting resolvers that explicitly support ECS, and vice versa. Thus, as only limited adoption of ECS can be expected, signing RRsets with ECS authenticates the records in the resolvers that enable ECS.

5.2 DNSSEC Considerations

The inclusion of ECS extension as additional information when signing a record with DNSSEC provides an effective countermeasure against the record replay in redirection hijacking, but its effectiveness is limited by the deployment of ECS. Inspired by this, we then consider a more general scheme that leverages existing additional data elements in DNSSEC.

Note that adversaries cannot generate a valid signature since they are unable to obtain the private key. Moreover, the replay attack of redirection hijacking can be successful because the validity period of DNSSEC signatures is typically long enough to be reused by adversaries to launch the record injection. However, only using a shorter validity period is not sufficient since the signature inception and expiration could also be fabricated by adversaries. Consequently, we consider that one possible mitigation is to secure the validity period by including additional timestamp information when signing a record. Combined with a short validity period in RRSIG (e.g., only slightly longer than the TTL of mapping records), this would significantly increase the difficulty of record injection, as the validity period cannot be altered and adversaries only have a short time window to perform the record injection.

Therefore, a straightforward approach is to include the validity period (i.e., signature inception and expiration) when signing a record. However, since the validity period is associated with the RRSIG record rather than the record being signed, it breaks away from normal operations of signing a record (but in a harmless manner): inception and expiration timestamp will be generated first, and then the RRSIG signature is computed according to both the responded RRset and validity period associated with the RRSIG record itself. Correspondingly, the resolver's software needs to be modified to include the validity period when computing the message digest. An alternative approach is to define a new extension representing the validity period in the additional section of DNS messages and sign the RRsets, including such extension data.

Note that the mechanisms we discuss here have similarities to TSIG/SIG(0) [68, 39], which sign complete DNS request/response with timestamps. However, TSIG requires a symmetric key and thus is most commonly used for authorizing dynamic updates and zone transfers. The SIG(0)'s functionality has been fundamentally replaced by DNSSEC. We argue that it may be worth enhancing the operations of DNSSEC to mitigate the threat of replay attacks due to the prevalence of dynamic mapping in CDNs.

5.3 CNAME Flattening

One of the foundational obstacles for CDN vendors to achieve the integrity of redirection records is the prevalent use of CNAME records, especially the dynamic CNAME mapping and chained CNAME records. A possible solution is to hide the CNAME chain from resolvers and leave the CNAME traversing to the CDN's authoritative nameservers, i.e., *CNAME Flattening* [14].[8]

CNAME Flattening implemented by Cloudflare was originally designed to enable the CNAME at the root domain while complying RFC's DNS specification [64], which requires that there should be no other record types if the type of a record is CNAME. With CNAME flattening, the CDN's authoritative nameserver acts as a re-

[8] A similar functionality has also been implemented by DNS-hosting providers, such as the ANAME record [17]. Here we focus on the discussion of such a feature provided by CDNs.

solver by recursively resolving the CNAME chain and finally constructs an A record to substitute for the original CNAME record.

We therefore suggest that CNAME flattening should also be leveraged by CDNs for security purposes. That is, instead of iteratively replying with multiple CNAME records, the CDN's authoritative nameserver takes full responsibility for the CNAME resolution, typically within the CDN's mapping infrastructure, and finally returns an A record, which can be signed with DNSSEC (live signing). This approach significantly reduces the computational overhead of signing CNAME records as well as the cost of multiple rounds of signature validation.

Note that CNAME flattening is mainly associated with the records for redirection operated by CDNs. The first level of CNAME delegation occurs at the CP's authoritative nameservers, which may be out of the control of CDNs. However, CPs can easily secure CNAME redirections by enabling (traditional) DNSSEC signatures at their authoritative nameservers, since those records are typically static mappings for domain delegation. Also, when enabling the CNAME flattening in DNS-based CDNs, the CDN's authoritative nameservers may need to employ ECS when retrieving mapping results as the representation of client networks.

Overall, CNAME flattening provides CDN vendors with a potential solution to secure CNAME records at an acceptable cost by avoiding iterative signature validation for multiple CNAME records, while retaining the flexibility of using a CNAME chain to facilitate platform management.

5.4 Request Re-Mapping

In addition to performing the request routing via DNS or anycast, CDNs also leverage the high-level re-mapping mechanism to remedy non-optimal server assignments in some cases. For example, when a request for content objects arrives at an edge server assigned by the mapping system, the edge server first performs an RTT measurement for the client. If the RTT is acceptable, the edge server immediately serves the content to the client based on normal content retrieval strategies; otherwise, the edge server requires the mapping system to reassign an optimal server and direct the client to a different server (e.g., via HTTP status code 3xx for redirection). Due to the extra server selection and redirection operations, the re-mapping introduces additional high latency penalty. Moreover, it is worth to note that, with the wide support of ECS, the accuracy of DNS-based mapping has been significantly improved for those clients impacted by the location discrepancy of LDNSes. That is, clients are rarely being assigned to a non-optimal edge server. Thus, the request re-mapping is typically only suitable for large-file transfers, such as video streaming and software distribution [18, 34].

Nevertheless, CDNs can still enable their Real User Measurement (RUM) system to monitor the performance from a large set of clients and aggregate the monitoring results with geographic locality or client-LDNS pairing to recognize the group of clients affected by anomalous redirections. In general, a more fine-grained performance monitoring and a more active request re-mapping could be useful to mitigate severe performance degradation in some cases. However, any high-level re-mapping mechanism still faces the threat of nullifying load balancing and DoS mitigation when unresponsive edge servers are exploited in redirection hijacking by adversaries, as discussed in §4.4.

5.5 Encryption and DNS-over-TLS

DNSCrypt [15] and DNSCurve [16] use ECC to encrypt DNS packets. Google Public DNS offers DNS-over-HTTPS [20] to enable the DNS resolution over encrypted connections. However, DNSCrypt and DNS-over-HTTPS can only secure connections between stubs and recursive resolvers. DNSCurve aims to authenticate the DNS packets between recursive resolvers and authoritative nameservers, but to date, it has only been supported by OpenDNS. Subsequently, DNS over Transport Layer Security (DNS-over-TLS) [88, 49] has been proposed to fundamentally address the weakness of DNS connectionless transmissions in security and privacy. Using TLS, the channels between stubs and recursive resolvers, as well as optionally between recursive resolvers and authoritative servers, would be protected from eavesdropping and MitM attacks. Recently, Cloudflare launched its new public DNS service that supports DNS-over-TLS (as well as DNS-over-HTTPS) [6].

DNS-over-TLS indeed addresses most security and privacy issues of DNS, including the vulnerability we showed in this paper (when applied to optional deployment between recursive resolvers and authoritative nameservers), because adversaries would be unable to know the content of DNS queries. However, due to the high performance impact and expensive costs of deployment, the adoption of DNS-over-TLS is still currently limited on the Internet.

6 Related Work

Disrupting CDN's server assignment has been recently proposed to circumvent Internet censorship [48, 89], whereby arbitrary edge servers rather than optimal servers assigned by the CDN's mapping system are used to bypass DNS-based/IP-based censorship and obtain censored content. The focus of such censorship circumvention is to retrieve censored content from edge servers with acceptable performance levels. In contrast, we explore the attack scenarios in which an end-user's access would be significantly degraded or interrupted, resulting in potential financial losses for both CDN providers and content providers.

DNS and CDN. The discrepancy of location proximity between end-users and their LDNSes has been observed for more than a decade [63, 73]. Pang *et al.* [69] characterized the responsiveness of DNS-based network controls according to the behaviors of end-systems and LDNSes. Huang *et al.* [50] proposed a solution called FQDN extension, in which clients obtain a location-aware cluster identifier and add this identifier to hostnames, to tackle the client-LDNS mismatch problem in Global Traffic Management (GTM). In order to improve the efficiency of content delivery, Krishnamurthy *et al.* [57] proposed a method by which HTTP interactions are piggybacked on DNS responses. Krishnan *et al.* [58] built a system to diagnose inflated latencies using active measurements to improve the effectiveness of the CDN's indirection and user performance. Scott *et al.* [72] built a tool chain for understanding the web deployment and footprints of CDNs by collecting DNS resolution results and probing the IPv4 address space. In addition, Pearce *et al.* [70] developed a tool to measure and study the global DNS manipulation exploited for the purpose of Internet censorship.

Ager *et al.* [26] compared local DNS resolvers against public DNS resolvers (Google Public DNS and OpenDNS) to study the responsiveness and diversity of resolvers. Subsequently, Otto *et al.* [67] examined the performance cost when clients use public DNS services to access CDNs. With the emergence of EDNS-Client-Subnet, Streibelt *et al.* [78] and Calder *et al.* [33] leveraged ECS with specified client prefixes to infer and profile large-scale Internet service infrastructure such as Google. Kintis *et al.* [54] investigated the potential privacy risk of ECS for surveillance, and revealed a cache poisoning threat for a highly selective group of clients.

Cache Poisoning and DNSSEC. Schomp *et al.* [71] assessed the vulnerabilities of diverse record injection attacks, particularly Kaminsky's attack and Bailiwick attack. Duan *et al.* [41] proposed a "Hold-On" period before accepting a reply to mitigate DNS poisoning attacks by also allowing a legitimate reply to arrive. Lian *et al.* [60] measured the practical impact of DNSSEC deployment and found that DNSSEC-signed domains may create collateral damage in resolutions of valid domains. van Rijswijk-Deij *et al.* [80, 81] studied the ECDSA deployment in CloudFlare and the .nl TLD and examined the computational overhead induced by the validation of ECC-based signatures. Yan *et al.* [87] proposed a revised DNSSEC signature that constructs a hash chain to limit replay vulnerability windows when the master server has failed. Their study tackles the problem of malicious slave servers and has a different scope than our study. Bau *et al.* [32] summarized the inherent vulnerabilities in DNSSEC with NSEC3, such as faulty resolver logic that enables adversaries to modify unsigned packet contents to introduce forged information into reply packets. Chung *et al.* [37] studied the DNSSEC support of registrars to understand the difficulties and challenges when domain owners try to deploy DNSSEC. Our study reveals another essential dimension of the insufficient DNSSEC deployment, especially for top domains, in which the dynamics of DNS records in DNS-based CDNs prevents the domains from creating pre-computed DNSSEC signatures.

Recent studies also reveal the pervasive mismanagement of DNSSEC. Shulman *et al.* [75] developed a validation engine to identify vulnerable keys in DNSSEC-signed domains. Chung *et al.* [36] performed a longitudinal study into how well DNSSEC's PKI is managed.

Security Issues in CDNs. Liang *et al.* [61] studied the practical impact of the CDN's HTTPS deployment. Composing HTTPS with CDN introduces the complexity of authentication delegation since CDN cuts the secure communication paths offered by HTTPS. Wählisch *et al.* [85] investigated the Resource Public Key Infrastructure (RPKI) deployment on the routing layer and reported that CDNs are the main cause of insufficiency in RPKI deployment. While the focus of these studies is on the vulnerability of CDN's backend, our study explores the frontend issue of CDN's service delivery.

Chen *et al.* [35] presented the forwarding-loop attacks, in which malicious customers may be capable of creating forwarding loops inside one CDN or across multiple CDNs to launch potential DoS attacks. The root cause of this threat is that CDNs lack control over customers' (mis)configurations. Vissers *et al.* [84] studied the "origin-exposing" attacks to identify the IP address of a service origin and to bypass the cloud-based security infrastructure, typically provided by CDNs. Jin *et al.* [52] revealed a new vulnerability of CDNs integrated with DDoS Protection Services (DPS), called residual resolution, in which a CDN may leak the origin IP address of its customer when the customer terminates the existing service and switches to another DPS platform.

7 Conclusion

In this paper, we present redirection hijacking, a new vulnerability of CDNs that stems from the dynamic characteristics of DNS records used for CDN's request routing.

In a redirection hijacking attack, adversaries can easily maneuver CDN's mappings between end-users and edge servers by injecting crafted but legitimate DNS records. We reveal that DNSSEC is ineffective to address such a hijacking attack, even with the new ECDSA-based signatures that are capable of achieving live signing for dynamically generated DNS records. This is mainly due to the reusability of signed legitimate records, which can be exploited by adversaries to override CDN's surrogate assignment and redirect client requests to inappropriate edge servers. We assess the magnitude of this vulnerability in the wild by characterizing the operations of the request routing for popular CDN vendors and analyzing the threats via multiple case studies. We quantify the practical impacts of redirection hijacking, especially on performance, and present more serious threats that could nullify CDN's load balancing and DoS protection. Finally, we discuss the countermeasures against redirection hijacking in CDNs from different aspects.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful and valuable comments that helped us improve the quality of this work. This work was partially supported by the U.S. NSF grant CNS-1618117, ONR grant N00014-17-1-2485, and DARPA XD3 Project HR0011-16-C-0055. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] `https://www.cdnplanet.com/blog/which-cdns-support-edns-client-subnet`.

[2] `https://www.cdnoverview.com`.

[3] `https://trends.builtwith.com/cdns`.

[4] `https://wappalyzer.com/categories/cdn`.

[5] `https://www.bfk.de/bfk_dnslogger_en`.

[6] `https://blog.cloudflare.com/announcing-1111/`.

[7] Akamai, *Inc*. Customized Caching Rules. `https://developer.akamai.com/learn/Caching/Customized_Caching_Rules.html`.

[8] Akamai, *Inc*. Facts & Figures (retrieved on June 2018). `www.akamai.com/us/en/about/facts-figures.jsp`.

[9] Akamai, *Inc*. Fast DNS. `https://www.akamai.com/us/en/solutions/products/cloud-security/fast-dns.jsp`.

[10] CDN.net. Why low latency CDN is important for eCommerce stores. `https://cdn.net/low-latency-cdn-important-ecommerce-stores`.

[11] Cedexis. `https://www.cedexis.com`.

[12] CloudFlare, *Inc*. DNSSEC Complexities and Considerations. `https://www.cloudflare.com/dns/dnssec/dnssec-complexities-and-considerations`.

[13] CloudFlare, *Inc*. ECDSA: The missing piece of DNSSEC. `https://www.cloudflare.com/dns/dnssec/ecdsa-and-dnssec`.

[14] CloudFlare, *Inc*. Introducing CNAME Flattening: RFC-Compliant CNAMEs at a Domain's Root. `https://blog.cloudflare.com/introducing-cname-flattening-rfc-compliant-cnames-at-a-domains-root`.

[15] DNSCrypt. `https://dnscrypt.org`.

[16] DNSCurve. `https://dnscurve.org`.

[17] DNSMadeEasy. Breakthrough in DNS Records. `https://www.dnsmadeeasy.com/services/anamerecords`.

[18] E. Zhang. Intelligent User Mapping in the Cloud. `https://blogs.akamai.com/2013/03/intelligent-user-mapping-in-the-cloud.html`.

[19] F. Assolini. Massive DNS poisoning attacks in Brazil. `https://securelist.com/blog/incidents/31628/massive-dns-poisoning-attacks-in-brazil-31`.

[20] Google Public DNS. DNS-over-HTTPS. `https://developers.google.com/speed/public-dns/docs/dns-over-https`.

[21] Imperva, *Inc*. The Essential Guide to CDN: CDN Caching. `https://www.incapsula.com/cdn-guide/cdn-caching.html`.

[22] J. Spring and L. Metcalf. Probable Cache Poisoning of Mail Handling Domains. `https://insights.sei.cmu.edu/cert/2014/09/-probable-cache-poisoning-of-mail-handling-domains.html`.

[23] P. Gilmore. Serving at the edge: Good for performance, good for mitigating DDoS. `https://blogs.akamai.com/2013/04/serving-at-the-edge-good-for-performance-good-for-mitigating-ddos-part-ii.html`.

[24] S. Friedl. An Illustrated Guide to the Kaminsky DNS Vulnerability. `http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html`.

[25] VirusTotal. `https://www.virustotal.com`.

[26] Ager, B., Mühlbauer, W., Smaragdakis, G., and Uhlig, S. Comparing DNS Resolvers in the Wild. In *ACM IMC* (2010).

[27] Alam, S. M. N., and Marbach, P. Competition and Request Routing Policies in Content Delivery Networks. In *CoRR*, 2009. `http://arxiv.org/abs/cs/0608082`.

[28] Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S. DNS Security Introduction and Requirements. *IETF RFC 4033* (2005).

[29] Ariyapperuma, S., and Mitchell, C. J. Security vulnerabilities in DNS and DNSSEC. In *International Conference on Availability, Reliability and Security (ARES)* (2007).

[30] Atkins, D., and Austein, R. Threat Analysis of the Domain Name System (DNS). *IETF RFC 3833* (2004).

[31] Barbir, A., Cain, B., Nair, R., and Spatscheck, O. Known Content Network (CN) Request-Routing Mechanisms. *IETF RFC 3568* (2003).

[32] Bau, J., and Mitchell, J. A Security Evaluation of DNSSEC with NSEC3. In *NDSS* (2010).

[33] Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., and Govindan, R. Mapping the Expansion of Google's Serving Infrastructure. In *ACM IMC* (2013).

[34] Chen, F., Sitaraman, R. K., and Torres, M. End-User Mapping: Next Generation Request Routing for Content Delivery. In *ACM SIGCOMM* (2015).

[35] Chen, J., Jiang, J., Zheng, X., Duan, H., Liang, J., Li, K., Wan, T., and Paxson, V. Forwarding-Loop Attacks in Content Delivery Networks. In *NDSS* (2016).

[36] Chung, T., van Rijswijk-Deij, R., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., and Wilson, C. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. In *USENIX Security* (2017).

[37] Chung, T., van Rijswijk-Deij, R., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., and Wilson, C. Understanding the Role of Registrars in DNSSEC Deployment. In *ACM IMC* (2017).

[38] Contavalli, C., van der Gaast, W., Lawrence, D., and Kumari, W. Client Subnet in DNS Queries. *IETF RFC 7871* (2016).

[39] D. Eastlake 3rd. DNS Request and Transaction Signatures (SIG(0)s). *IETF RFC 2931* (2000).

[40] Dagon, D., Antonakakis, M., Vixie, P., Jinmei, T., and Lee, W. Increased DNS Forgery Resistance Through 0x20-bit Encoding: Security via Leet Queries. In *ACM CCS* (2008).

[41] Duan, H., Weaver, N., Zhao, Z., Hu, M., Liang, J., Jiang, J., Li, K., and Paxsonh, V. Hold-On: Protecting Against On-Path DNS Poisoning. In *SATIN* (2012).

[42] Gieben, R., and Mekking, W. Authenticated Denial of Existence in the DNS. *IETF RFC 7129* (2014).

[43] Goldberg, S., Naor, M., Papadopoulos, D., Reyzin, L., Vasant, S., and Ziv, A. NSEC5: Provably Preventing DNSSEC Zone Enumeration. In *NDSS* (2015).

[44] Herzberg, A., and Shulman, H. Security of Patched DNS. In *ESORICS* (2012).

[45] Herzberg, A., and Shulman, H. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *IEEE CNS* (2013).

[46] Herzberg, A., and Shulman, H. Socket Overloading for Fun and Cache-poisoning. In *ACSAC* (2013).

[47] Hoffman, P., and Wijngaards, W. Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC. *IETF RFC 6605* (2012).

[48] Holowczak, J., and Houmansadr, A. CacheBrowser: Bypassing Chinese Censorship Without Proxies Using Cached Content. In *ACM CCS* (2015).

[49] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and Hoffman, P. Specification for DNS over Transport Layer Security (TLS). *IETF RFC 7858* (2016).

[50] Huang, C., Batanov, I., and Li, J. A Practical Solution to the Client-LDNS Mismatch Problem. *ACM SIGCOMM Computer Communication Review* (Mar. 2012).

[51] Hubert, A., and van Mook, R. Measures for Making DNS More Resilient against Forged Answers. *IETF RFC 5452* (2009).

[52] Jin, L., Hao, S., Wang, H., and Cotton, C. Your Remnant Tells Secret: Residual Resolution in DDoS Protection Services. In *IEEE/IFIP DSN* (2018).

[53] Kaminsky, D. It's The End Of The Cache As We Know It. *BlackHat* (2008).

[54] Kintis, P., Nadji, Y., Dagon, D., Farrell, M., and Antonakakis, M. Extended Abstract: Understanding the Privacy Implications of ECS. In *DIMVA* (2016).

[55] Klein, A., Shulman, H., and Waidner, M. Internet-Wide Study of DNS Cache Injections. In *IEEE INFOCOM* (2017).

[56] Kolkman, O., Mekking, W., and Gieben, R. DNSSEC Operational Practices, Version 2. *IETF RFC 6781* (2012).

[57] Krishnamurthy, B., Krishnamurthy, E., Liston, R., and Rabinovich, M. DEW: DNS-Enhanced Web for Faster Content Delivery. In *WWW* (2003).

[58] Krishnan, R., Madhyastha, H. V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., and Gao, J. Moving Beyond End-to-End Path Information to Optimize CDN Performance. In *ACM IMC* (2009).

[59] Laurie, B., Sisson, G., Arends, R., and Blacka, D. DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. *IETF RFC 5155* (2008).

[60] Lian, W., Rescorla, E., Shacham, H., and Savage, S. Measuring the Practical Impact of DNSSEC Deployment. In *USENIX Security* (2013).

[61] Liang, J., Jiang, J., Duan, H., Li, K., Wan, T., and Wu, J. When HTTPS Meets CDN: A Case of Authentication in Delegated Service. In *IEEE Security & Privacy* (2015).

[62] Maggs, B. M., and Sitaraman, R. K. Algorithmic Nuggets in Content Delivery. *ACM SIGCOMM Computer Communication Review* (2015).

[63] Mao, Z. M., Cranor, C. D., Douglis, F., Rabinovich, M., Spatscheck, O., and Wang, J. A Precise and Efficient Evaluation of the Proximity between Web Clients and their Local DNS Servers. In *USENIX ATC* (2002).

[64] Mockapetris, P. Domain Names - Implementation and Specification. *IETF RFC 1035* (1987).

[65] Mukerjee, M. K., Bozkurt, I. N., Maggs, B., Seshan, S., and Zhang, H. The Impact of Brokers on the Future of Content Delivery. In *ACM HotNets* (2016).

[66] Mukerjee, M. K., Bozkurt, I. N., Ray, D., Maggs, B., Seshan, S., and Zhang, H. Redesigning CDN-Broker Interactions for Improved Content Delivery. In *ACM CoNEXT* (2017).

[67] Otto, J. S., Sánchez, M. A., Rula, J. P., and Bustamante, F. E. Content Delivery and the Natural Evolution of DNS: Remote DNS Trends, Performance Issues and Alternative Solutions. In *ACM IMC* (2012).

[68] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. Secret Key Transaction Authentication for DNS (TSIG). *IETF RFC 2845* (2000).

[69] Pang, J., Akella, A., Shaikh, A., Krishnamurthy, B., and Seshan, S. On the Responsiveness of DNS-based Network Control. In *ACM IMC* (2004).

[70] Pearce, P., Jones, B., Li, F., Ensafi, R., Feamster, N., Weaver, N., and Paxson, V. Global Measurement of DNS Manipulation. In *USENIX Security* (2017).

[71] Schomp, K., Callahan, T., Rabinovich, M., and Allman, M. Assessing DNS Vulnerability to Record Injection. In *PAM* (2014).

[72] Scott, W., Anderson, T., Kohno, T., and Krishnamurthy, A. Satellite: Joint Analysis of CDNs and Network-Level Interference. In *USENIX ATC* (2016).

[73] Shaikh, A., Tewari, R., and Agrawal, M. On the Effectiveness of DNS-based Server Selection. In *IEEE INFOCOM* (2001).

[74] Shulman, H., and Waidner, M. Fragmentation Considered Leaking: Port Inference for DNS Poisoning. In *ACNS* (2014).

[75] Shulman, H., and Waidner, M. One Key to Sign Them All Considered Vulnerable: Evaluation of DNSSEC in the Internet. In *NSDI* (2017).

[76] Sisson, G., and Laurie, B. Derivation of DNS Name Predecessor and Successor. *IETF RFC 4471* (2006).

[77] SON, S., AND SHMATIKOV, V. The Hitchhiker's Guide to DNS Cache Poisoning. In *SecureComm* (2010).

[78] STREIBELT, F., BÖTTGER, J., CHATZIS, N., SMARAGDAKIS, G., AND FELDMANN, A. Exploring EDNS-client-subnet Adopters in Your Free Time. In *ACM IMC* (2013).

[79] TRIUKOSE, S., AL-QUDAH, Z., AND RABINOVICH, M. Content Delivery Networks: Protection or Threat? In *ESORICS* (2009).

[80] VAN RIJSWIJK-DEIJ, R., HAGEMAN, K., SPEROTTO, A., AND PRAS, A. The Performance Impact of Elliptic Curve Cryptography on DNSSEC Validation. *IEEE/ACM Transactions on Networking* (Sept. 2016).

[81] VAN RIJSWIJK-DEIJ, R., JONKER, M., AND SPEROTTO, A. On the Adoption of the Elliptic Curve Digital Signature Algorithm (ECDSA) in DNSSEC. In *CNSM* (2016).

[82] VAN RIJSWIJK-DEIJ, R., SPEROTTO, A., AND PRAS, A. DNSSEC and Its Potential for DDoS Attacks: A Comprehensive Measurement Study. In *ACM IMC* (2014).

[83] VAN RIJSWIJK-DEIJ, R., SPEROTTO, A., AND PRAS, A. Making the Case for Elliptic Curves in DNSSEC. *ACM SIGCOMM Computer Communication Review* (Oct. 2015).

[84] VISSERS, T., VAN GOETHEM, T., JOOSEN, W., AND NIKIFORAKIS, N. Maneuvering Around Clouds: Bypassing Cloud-based Security Providers. In *ACM CCS* (2015).

[85] WÄHLISCH, M., SCHMIDT, R., SCHMIDT, T. C., MAENNEL, O., UHLIG, S., AND TYSON, G. RiPKI: The Tragic Story of RPKI Deployment in the Web Ecosystem. In *ACM HotNets* (2015).

[86] WEILER, S., AND IHREN, J. Minimally Covering NSEC Records and DNSSEC On-line Signing. *IETF RFC 4470* (2006).

[87] YAN, H., OSTERWEIL, E., HAJDU, J., ACRES, J., AND MASSEY, D. Limiting Replay Vulnerabilities in DNSSEC. In *NPSec* (2008).

[88] ZHU, L., HU, Z., HEIDEMANN, J., WESSELS, D., MANKIN, A., AND SOMAIYA, N. Connection-Oriented DNS to Improve Privacy and Security. In *IEEE Security & Privacy* (2015).

[89] ZOLFAGHARI, H., AND HOUMANSADR, A. Practical Censorship Evasion Leveraging Content Delivery Networks. In *ACM CCS* (2016).