افزونه های امنیتی DNS، همه گیری و کارایی

مرتضی ملکی نژاد شوشتری

۴۰۰۵۲۲۲۱۱

درس امنیت سیستم های کامپیوتری – بهار ۱۴۰۴

فهرست مطالب

۱ چکیده

در سال ۱۹۸۴ میلادی که DNS به طور رسمی معرفی شد، امنیت از دغدغه های آن نبود. به همین علت مشکلات امنیتی زیادی در این پروتکل کشف شد و برای هرکدام از آن ها راه حل هایی بیان شد.

DNSSEC این مسئله را حل میکند که کاربر بتواند تایید کند که این رکورد[۱] توسط جای معتبری تنظیم شده است و یک مهاجم این را تنظیم نکرده است.

در این گزارش به شیوه کار DNSSEC، مسائلی که حل میکند، چالش ها و گستردگی آن میپردازیم.

۲ مقدمه

از آنجایی که DNS بخش جدایی ناپذیری از اینترنت است، بدیهی است امنیت آن نیز جزو مهم ترین مباحث امنیت اینترنت باشد. دسته ای از حملاتی که روی DNS رخ می‌دهد، از این نوع است که مهاجم تلاش دارد به طرق مختلف رکورد مورد نظر خود را تنظیم کند و کاربران دامنه هدف را به سمت آدرس IP خود هدایت کند. به عنوان مثال میتوان به حمله Kaminsky در سال ۲۰۰۸ اشاره کرد که با استفاده از Cache poisoning فرد مهاجم به راحتی میتوانست رکورد های مورد نظر خود را تنظیم کند [۱] یا در سال ۲۰۱۴ نیز محققان کشف کردند که ایمیل هایی که به سمت سرورهای Gmail، Yahoo و ... باید میرفتند، به سمت سرورهای ایمیل ناشناسی میرفتند [۹]

برای جلوگیری از این حملات DNSSEC ارائه شد تا با استفاده از زیرساخت های فعلی DNS بتواند این تضمین را بدهد که این رکورد DNS توسط سازمان یا شخص معتبری تعریف شده است.

DNSSEC نسبت به DNS پیاده سازی پیچیده تری دارد و همین امر منجر به این شده که تمایل جامعه نسبت به به کارگیری آن کمی کم باشد. البته با توجه به مزایای امنیتی ای که دارد این رشد مثبت است ولی چالش هایی در زمینه استقرار و نظارت دارد.

۳ شیوه

۱.۳ حملات

۱.۱.۳ حمله Kaminsky

این حمله در سال ۲۰۰۸ توسط دن کامینسکی کشف شد. در این حمله:

- مهاجم ابتدا درخواست‌هایی برای زیردامنه‌های تصادفی از دامنه هدف (مثال: abc123.example.com) ارسال می‌کند.
- همزمان با این درخواست‌ها، مهاجم تعداد زیادی پاسخ جعلی با شناسه‌های تراکنش تصادفی ارسال می‌کند.
- اگر یکی از این پاسخ‌ها با شناسه تراکنش واقعی مطابقت داشته باشد، سرور DNS رکورد جعلی را می‌پذیرد.
- از آنجا که حمله روی دامنه اصلی (example.com) انجام می‌شود، تمام زیردامنه‌ها تحت تأثیر قرار می‌گیرند.

این حمله به‌خصوص خطرناک است زیرا:

- نیاز به حدس زدن تنها یک شناسه تراکنش دارد
- اثر آن روی تمام زیردامنه‌ها اعمال می‌شود

[۱] record

- می‌تواند در مدت زمان کوتاهی انجام شود

۲.۱.۳ حمله Cache Poisoning سال ۲۰۱٤

در سال ۲۰۱۴ محققان امنیتی کشف کردند که مهاجمان توانسته‌اند با استفاده از آسیب‌پذیری در سرورهای DNS، ترافیک ایمیل‌های مربوط به سرویس‌های بزرگی مانند Gmail، Yahoo و دیگر ارائه‌دهندگان را به سرورهای تحت کنترل خود هدایت کنند. مشخصات این حمله شامل:

- مهاجمان از آسیب‌پذیری در الگوریتم‌های تولید شناسه تراکنش DNS سوءاستفاده کردند
- با مسموم کردن کش DNS، رکوردهای MX (مربوط به سرورهای ایمیل) را تغییر دادند
- این حمله منجر به انحراف ترافیک ایمیل‌های حساس به سرورهای مخرب شد
- مدت زمان قابل توجهی طول کشید تا این حمله کشف و رفع شود

این واقعه نشان داد که حتی زیرساخت‌های مهم اینترنتی نیز در برابر حملات DNS آسیب‌پذیر هستند و لزوم استفاده از DNSSEC را بیش از پیش آشکار کرد. در این حمله:

- مهاجمان توانستند به مدت چندین ساعت ترافیک ایمیل را منحرف کنند
- امکان سرقت اطلاعات حساس مانند اعتبارنامه‌های ورود وجود داشت
- بسیاری از سازمان‌ها و ارائه‌دهندگان سرویس پس از این واقعه به سرعت پیاده‌سازی DNSSEC را آغاز کردند

۲.۳ نحوه کار DNSSEC

DNSSEC با استفاده از چندین نوع رکورد جدید کار میکند. تا بتواند Integrity و Authenticity را ارائه بدهد. در DNSSEC ما دغدغه Confidentiality نداریم و همچنان افراد میانی میتوانند ببینند چه رکوردی درخواست داده شده است.

DNSSEC از زیرساخت موجود در خود DNS بهره میبرد و مفهوم جدیدی در لایه زیرساخت اضافه نمیکند. به همین منظور چندین نوع جدید رکورد DNS اضافه شده که لیست آن ها آمده.

۱.۲.۳ انواع رکورد های جدید

- **RRset** یک دسته از رکورد ها که همه یک نوع دارند میتوانند یک RRSet را تشکیل بدهند.
- **RRSig** محتوی امضای یک RRSet است.
- **DNSKEY** کلید عمومی امضا را دارد که کاربر ها با آن بتوانند صحت امضا را چک کنند.
- **DS** هش یک رکورد DNSKEY را دارد. برای اینکه اختیار یک دامنه به یک کلید دیگر منتقل شود استفاده می‌شود. (برای مثال Cloudflare امکان صدور امضای رکورد های example.com را به صاحب دامنه می‌دهد و اگر کسی کلید Cloudflare را قبول داشته باشد می‌تواند از روی آن به صحت کلید صاحب سایت نیز برسد.)
- **NSEC, NSEC3** برای اینکه یک امضا را نامعتبر بنامیم استفاده می‌شود.
- **CDS, CDNSKEY** برای اینکه در یک ناحیه [۲] فرزند از ناحیه والد بخواهیم تا رکورد DS خود را به روز کند.

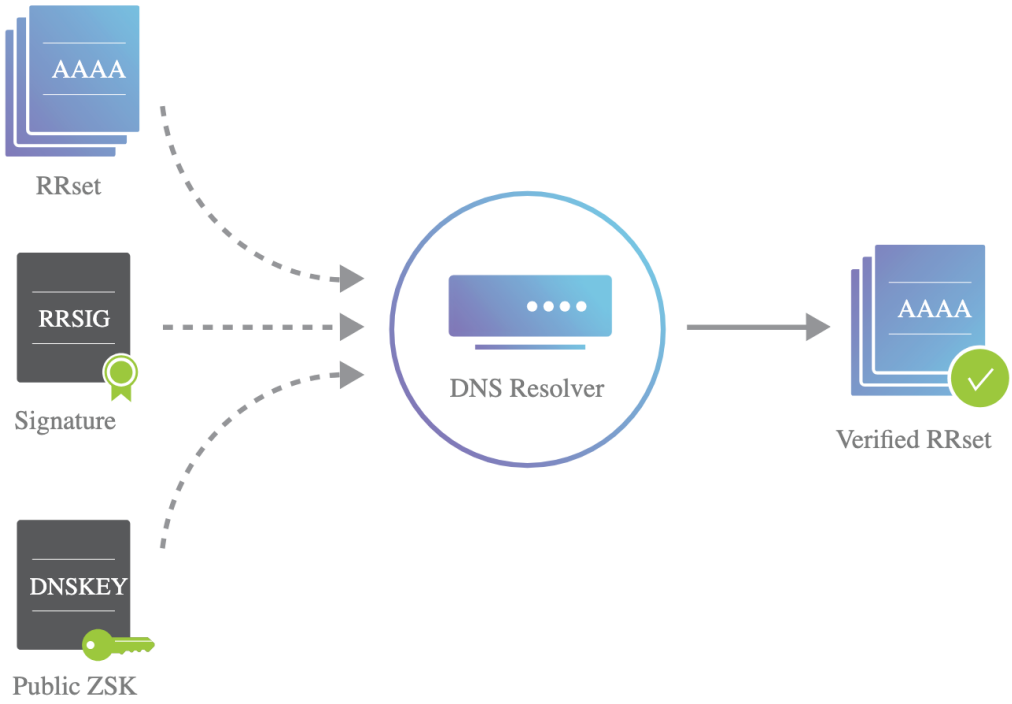
[۲] zone

۳.۳ امضا کردن

۱.۳.۳ ZSK

هر سرور یک جفت کلید ZSK دارد که کلید عمومی آن رکورد DNSKEY را میسازد و کلید خصوصی آن جهت امضا کردن RRSet های آن ناحیه استفاده می‌شود.

هنگامی که کاربر یک رکورد DNS را میخواهد،سرور RRSig و DNSKEY متناظر با آن را نیز میفرستد تا کاربر بتواند اصالت این رکورد را تایید کند.



شکل ۱: برای تایید اصالت یک رکورد به هر سه مورد RRSet، DNSKEY، و RRSIG نیاز است. [۳]

۲.۳.۳ KSK

خود رکورد DNSKEY توسط کلیدی به اسم KSK در فرایند مشابهی امضا میشود. رابطه بین KSK و ZSK مشابه رابطه بین Refresh Token و Access Token در Session Authentication است.

همچنان مشابه قسمت قبل،بخش عمومی KSK در یک رکورد DNSKEY قابل دسترس است و برای احراز اصالت امضاهای آن مورد نیاز است.

۳.۳.۳ واگذاری امضا

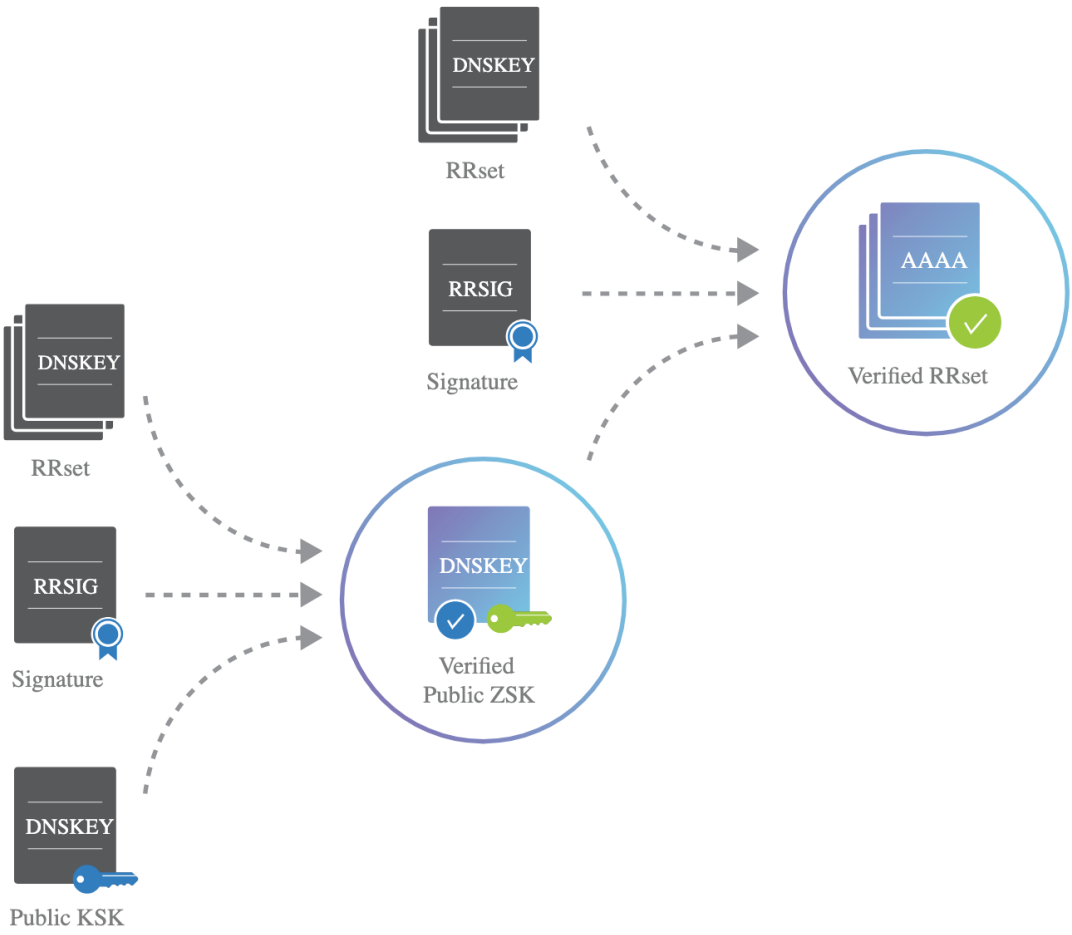
اگر هر سرور بخواهد به صورت آفلاین [۳] کلید KSK خود را به کاربر برساند، مقادیر این کلید ها بسیار زیاد می‌شود. پس این راه مناسبی نیست.

راه مناسب تر این است که بتوان کلید های محدودی را در دستگاه های کاربران هاردکد [۴] کرد و سپس صاحبان این کلید ها کنترل دامنه را به دست صاحبان دامنه بسپارند.

---

[۳] Offline

[۴] Hardcode



شکل ۲: رکورد DNSKEY از روی KSK بدست می‌آید و سپس با آن می‌شود RRSet های مختلفی را امضا کرد. [۳]

این کار به این صورت انجام می‌شود که در DNS Provider های سطح بالا یک رکورد DS تنظیم می‌شود که حاوی هش DNSKEY برای DNS Provider سطح پایین تر است. اصالت این رکورد نیز همچون سایر رکورد ها با استفاده از RRSIG و DNSKEY مربوط به DNS PROVIDER سطح بالاتر قابل احراز است.

از آنجایی که رکورد DS هش رکورد DNSKEY مربوط به KSK سطح پایین تر را دارد، پس هر تغییری در KSK نیاز به تغییر رکورد DS نیز دارد. پس تغییر KSK به نسبت تغییر هزینه بر تری است نسبت به ZSK.

٤.٣.٣ نبود رکورد و چالش های امنیتی آن

در صورتی که پاسخ DNS SERVER خالی باشد، پیامی برای امضا نداریم، از این رو رکورد NSEC معرفی شد.

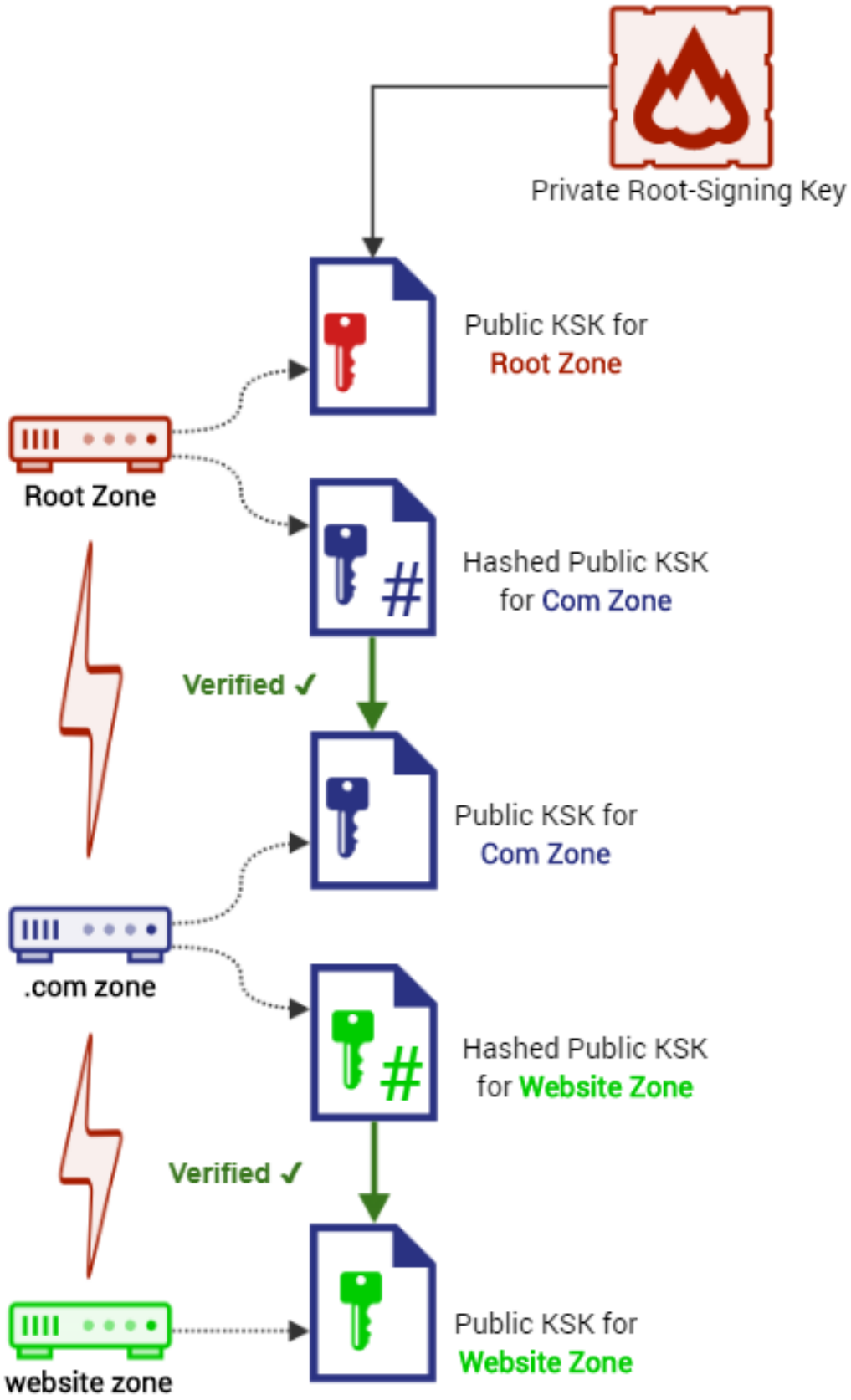
این رکورد وقتی دامنه درخواستی را نداشته باشد،دامنه بعدی را می‌دهد و به این ترتیب کاربر ما می‌فهمد آدرس درخواستی او وجود ندارد.

البته با این رکورد یک فرد میتواند همه زیردامنه های یک دامنه را بدست بیاورد.

٥.٣.٣ زنجیره اعتماد

جندین کلید ریشه در کلاینت ها هاردکد می‌شود و سایر کلید ها اعتبار خود را از این کلید ها بدست می‌آورند.

اینکار با استفاده از رکورد DS که بیان شد انجام می‌شود.



شکل ۳: به صورت زنجیری هر کلید اعتبار خود را از کلید قبل می‌گیرد. به غیر از کلید ریشه. [۴]

٤ چالش ها

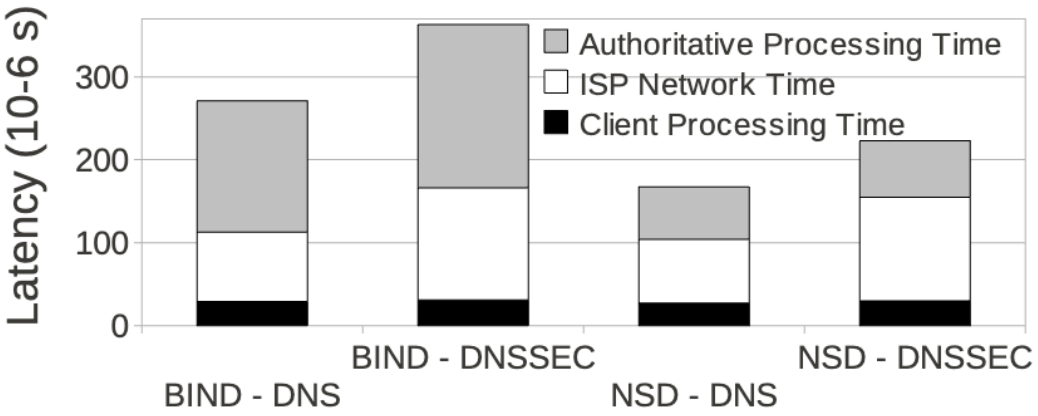
علیرغم این که DNSSEC مسائل بسیاری را حل می‌کند ولی خود DNSSEC نیز چالش های جدیدی را پدید می‌آورد.

یک سری از این چالش ها ناشی از سربار تازه ایست که به سیستم اضافه شده

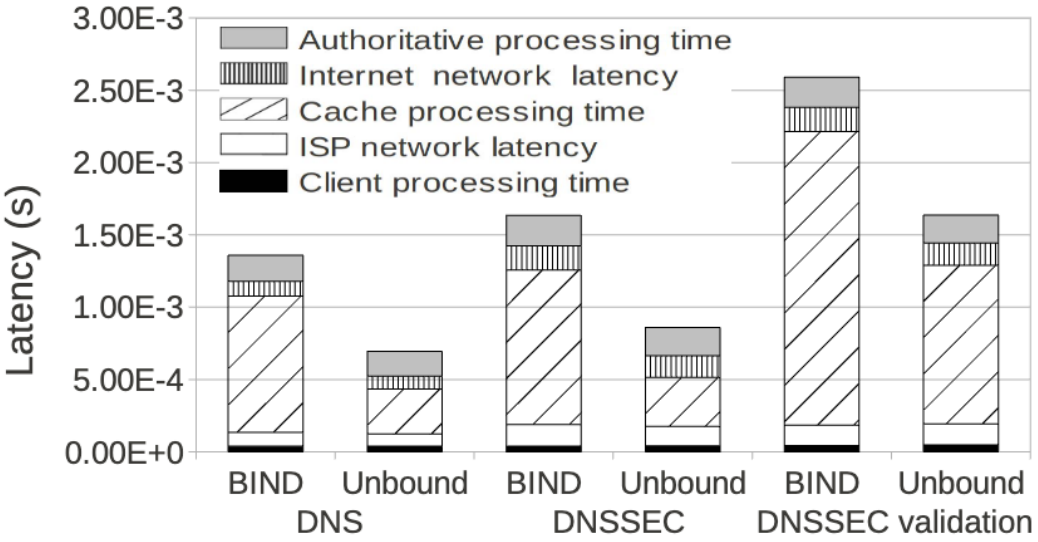
بخش بیشتر این چالش ها ناشی از پیچیدگی DNSSEC است که خود را در نرخ همه گیری [۵] پایین، Deploy های سخت و ... نشان می‌دهد.

١.٤ چالش های Performance

DNSSEC در کمترین حالت مجبور میکند تا سرور ۲ رکورد اضافه تر را پیدا کند. همچنین سایز جواب نیز زیاد شده است. علیرغم اینکه مقدار سربار اضافه شده به نوع بار سیستم وابسته است، در تحقیقی [۵] که انجام شده، تقریبا ۴۲۵ درصد سربار پردازشی به سیستم وارد شده است که هم به علت داده بشتر و هم به علت محاسبه امضا است. از همچنین با توجه به نرخ رشد اینترنت این موضوع واضح است که لود همواره در حال بیشتر شدن است.



(a) Authoritative



(b) Cache

شکل ۴: نرخ تاخیر در حالت های مختلف

[۵] Adaption

از طرف دیگر بخشی از راه حل های این مشکل صرفا به پخش بار پرداخته اند و ایده ای برای کم تر کردن سربار پردازشی ارائه نداده اند.

۱.۱.٤ Prefetch

یکی از راه هایی که بیان شده استفاده از معماری Prefetch است که با استفاده از یک سیستم کش چندلایه و توزیع شده سعی در حل این موضوع دارد. [۶]

راه‌حل اصلی: تقسیم‌بندی هوشمند ترافیک معماری PREFETCH_X با دو مکانیسم اصلی کار می‌کند:

- **پیش‌بارگذاری (Prefetching)**:
  - X دامنه پرترافیک (HEAD_X) در یک کش اختصاصی ذخیره می‌شوند
  - این کش می‌تواند روی کارت‌های شتاب‌دهنده سخت‌افزاری شبکه (NHAC) پیاده‌سازی شود
  - برای $X = 2000$، حدود ۶۸٪ ترافیک به این روش پاسخ داده می‌شود
- **مدیریت دامنه‌های کم‌ترافیک (TAIL_X)**:
  - با استفاده از ساختار Distributed Hash Table
  - مبتنی بر پروتکل Pastry برای توزیع یکنواخت بار پردازشی
  - پنج مدل مختلف برای مدیریت کش ارائه شده است

مزایای کلیدی معماری

- **کاهش منابع مورد نیاز**:
  - تا ۴ برابر کاهش تعداد سرورها نسبت به معماری سنتی IP_{XOR}
  - کاهش ۵۵٪ منابع برای DNS معمولی و ۸۰٪ برای DNSSEC
- **توزیع یکنواخت بار پردازشی**:
  - با انتخاب $X = 2000$، تغییرات بار پردازشی کمتر از ۱۰٪ می‌شود
  - توزیع یکنواخت کوئری‌ها و عملیات resolve بین سرورها
- **سازگاری با زیرساخت موجود**:
  - نیاز به تغییرات اساسی در زیرساخت شبکه فعلی ندارد
  - امکان پیاده‌سازی تدریجی

مدل‌های مختلف مدیریت $\mathbf{TAIL}_X$ مقاله پنج مدل مختلف برای مدیریت دامنه‌های کم‌ترافیک بررسی کرده است:

- Pastry (پایه): بدون مکانیسم کش‌گذاری خاص
- Pastry-SF (Stateless Forwarding): ارسال پاسخ مستقیماً به کاربر نهایی
- Pastry-PC (Passive Caching): کش‌گذاری غیرفعال پاسخ‌ها
- Pastry-R (Replication): تکثیر پاسخ‌ها روی k همسایه
- Pastry-AC (Active Caching): کش‌گذاری فعال برای دامنه‌های پرترافیک

نتایج تجربی و اعتبارسنجی

- پیاده‌سازی آزمایشی با FreePastry انجام شده است
- همبستگی ۹۹۹۱.۰ بین مدل نظری و نتایج عملی
- برای $X = 2000$ و ۱۸ سرور:
  - کاهش ۳۰٪ بار پردازشی نسبت به IP_{XOR}
  - توزیع یکنواخت ترافیک بین سرورها

چالش‌ها و محدودیت‌ها

- نیاز به تحلیل ترافیک برای تعیین مقدار بهینه X
- پیچیدگی مدیریت همزمان دو لایه کش (HEAD_X و TAIL_X)
- هزینه اولیه پیاده‌سازی سخت‌افزارهای شتاب‌دهنده

۲.٤ چالش‌های همه‌گیری

با وجود مزایای امنیتی قابل توجه DNSSEC، نرخ پذیرش [۶] آن در سطح جهانی همچنان پایین است. این مسئله ناشی از چندین چالش کلیدی است:

- **پیچیدگی پیاده‌سازی**: فرآیند تولید، ذخیره‌سازی و مدیریت کلیدهای ZSK و KSK، امضای رکوردها، و نگهداری زنجیره اعتماد برای بسیاری از سازمان‌ها پیچیده و زمان‌بر است.
- **عدم آگاهی عمومی**: بسیاری از مدیران سیستم و توسعه‌دهندگان از نحوه عملکرد و اهمیت DNSSEC اطلاع کافی ندارند یا تصور می‌کنند که تهدیدات مرتبط با DNS برای آن‌ها جدی نیست.
- **مشکلات سازگاری**: برخی سرویس‌ها و زیرساخت‌های قدیمی با رکوردهای DNSSEC سازگاری کامل ندارند، یا نیاز به پیکربندی‌های خاصی دارند که مانع از پذیرش آن می‌شود.

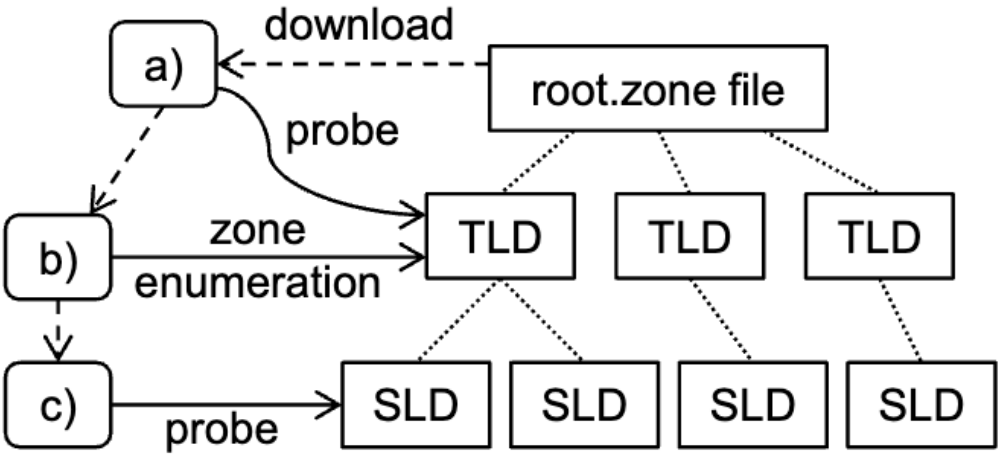
[۶] Adaption

◇ **محدودیت در ابزارها و مستندسازی:** به دلیل نسبتاً جدید بودن DNSSEC، ابزارهای مدیریتی کمتری برای آن وجود دارد و مستندسازی رسمی نیز نسبت به سایر فناوری‌ها کمتر است.

این چالش‌ها باعث شده‌اند که با وجود تهدیدات جدی علیه DNS، بسیاری از دامنه‌ها همچنان از DNSSEC استفاده نکنند. در آمارهایی که توسط سازمان‌های ثبت دامنه منتشر شده‌اند، مشاهده می‌شود که نرخ فعال‌سازی DNSSEC برای دامنه‌های سطح بالا (مانند .com، .org) هنوز به درصد مطلوبی نرسیده است.

افزایش نرخ پذیرش DNSSEC نیازمند تلاش مشترک شرکت‌های ثبت دامنه، ارائه‌دهندگان خدمات DNS، توسعه‌دهندگان زیرساخت، و سازمان‌های استانداردگذاری است.

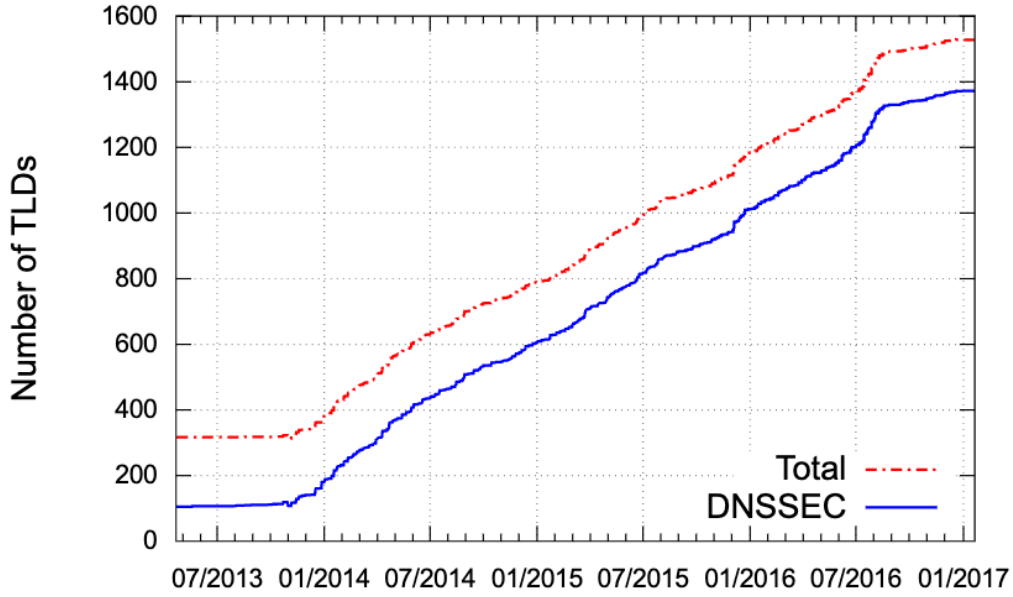
در یک تحقیق که در سال ۲۰۱۷ انجام شد [۱۰] با استفاده از گرفتن لیست zone های ریشه از IANA و سپس انجام Zone enumeration بر روی همه TLD [۷] ها می‌توان دید که تنها حدود ۱ درصد از دامنه ها از DNSSEC استفاده می‌کنند. البته که این موضوع بسته به جایگاه و نوع رکورد DNS دارد.



شکل ۵: شیوه های یافتن دامنه های با پشتیبانی از DNSSEC

◇ TLD ها: TLD ها به نسبت نرخ Adaption خوبی دارند. در سال ۲۰۱۴ حدود یک سوم آن ها از DNSSEC استفاده می کردند و این آمار همواره رو به رشد بوده است.

[۷] Top level domain



شکل ۶: تعداد TLD ها در گذر زمان

۱.۲.٤ پیچیدگی

این مورد درواقع ریشه بسیاری از چالش های DNSSEC است. به طوری که نرخ پذیرش پایین و یا حتی نیاز به توان پردازشی بیشتر نیز ناشی از این هستند. البته که این موضوع چندان بدیهی هم نیست زیرا DNSSEC اولین سیستم رمزنگاری کاملا توزیع شده است و مشابه هر سیستم توزیع شده دیگری، برای غلبه بر چالش های موجود به سمت پیچیده شدن رفته است.

نظارت هرچند همچنان میتوان کارهایی کرد که درزمینه های مختلف اثر این پیچیدگی کمتر شود

یکی از راه هایی که برای کم تر کردن اثر این مشکل پیشنهاد شده،استفاده از یک سیستم نظارت بر DNSSEC است که به صاحبان سرور وضعیت هم دامنه خود و هم دامنه هایی که کاربرانشان مشاهده می‌کنند را اطلاع دهد. از جمله این سیستم ها می‌توان به SecSpider اشاره کرد. [۷]

یکی از فایده های این سیستم آگاهی دادن به افراد است درصورتی که DNSSEC درست Deploy نشده باشد. جالب است که این آمار کم نیست و حدود ۴۵.۰ درصد سایت هایی که DNSSEC را فعال کرده بودند، درست نبودند.

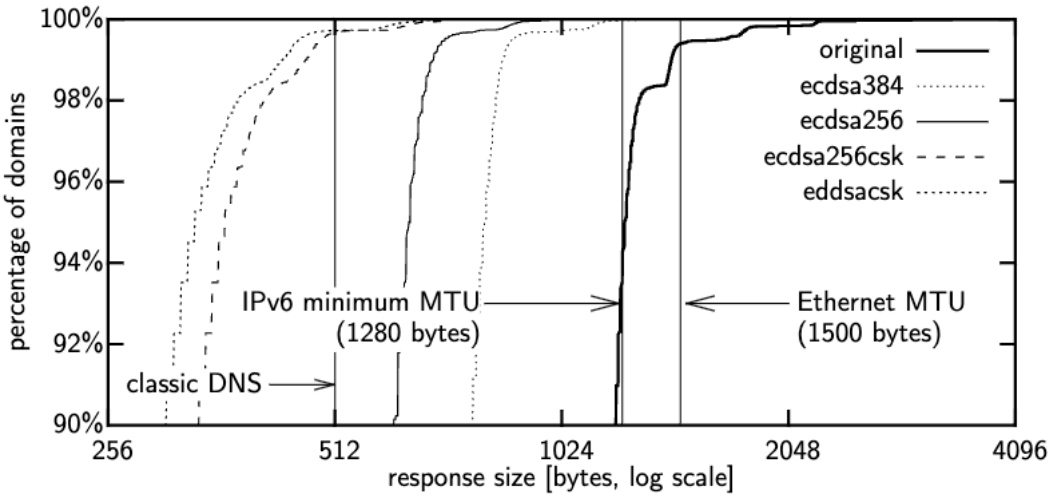
ابزاری مثل secspider باید سایت های با پشتیبانی از DNSSEC را از جاهای مختلف رصد کند زیرا ممکن است در یک جا امضا سالم به شمار بیاید و در جایی دیگر خیر.

سیستم رمزنگاری در پاسخ به پیچیدگی و همچنین Resource intensive بودن فرآیند DNSSEC پیشنهادی که مطرح شد این است که به جای استفاده از رمزنگاری پیش‌فرض RSA از رمزنگاری های مبتنی بر خم های بیضوی [۸] استفاده شود.

این کار چندین فایده مهم دارد:

- **کاهش احتمال IP Fragmentaion** کلید های ECC در مقایسه با RSA سایز کوچکتری دارند و این امر منجر به این می‌شود که احتمال IP Fragmentaion کمتر بشود.

[۸] Elliptic Curve

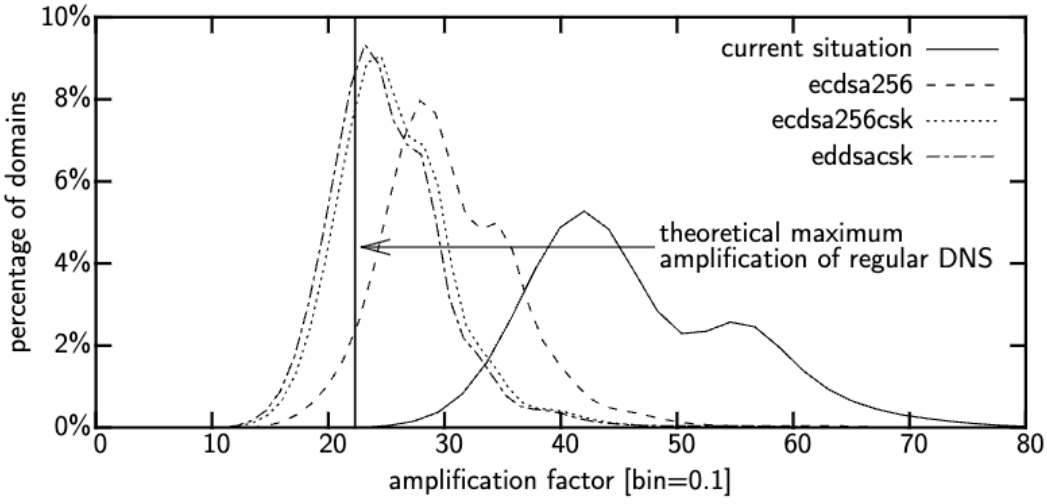


شکل ۷: در اینجا واضح است که اگر از RSA استفاده شود. روی Ethernet درصدی از دامنه ها و روی IPV6 تقریبا همه دامنه ها سایز جواب از MTU بیشتر می‌شود و Fragmentation رخ می‌دهد ولی اگر از رمزنگاری های مبتنی بر ECC استفاده شود، می‌توان دید که سایز جواب بسیار کمتر است و احتمال Fragmentation کمتر است.

- **کاهش ریسک Amplification Attack** همانطور که بیان شد، DNSSEC باعث افزایش لود سیستم می‌شود و از این رو هدف حمله های از نوع DDoS [۹] است.

یکی از معروف ترین نوع این حملات حملات تشدیدی [۱۰] است. از آنجایی که در DNSSEC برای دریافت یک رکورد به مراتب کارهای بیشتری انجام می‌شود پس نسبت به DNS معمولی به این حمله آسیب پذیر تر است. ولی از آنجایی که ECC به علت سایز کلید کوچک تر، به طور خاص در رمزگشایی رمزنگاری سبک تری است، انتظار می‌رود که اثر کمتری بپذیرد.

طی تحقیقی که انجام شد، این کار می‌تواند بین ۴۰ تا ۵۰ درصد اثر حملات تشدیدی را کاهش دهد.



شکل ۸: می‌توان دید که در ECC میزان ضریب تشدید از حالت معمول بسیار کمتر است.
همچنین می‌توان دید که با این حال DNS معمولی از همه ضریب کمتری دارد.

- **سرعت** از آنجایی که ECC سرعت رمزگشایی از RSA کمتر است، این مورد یک چالش است.

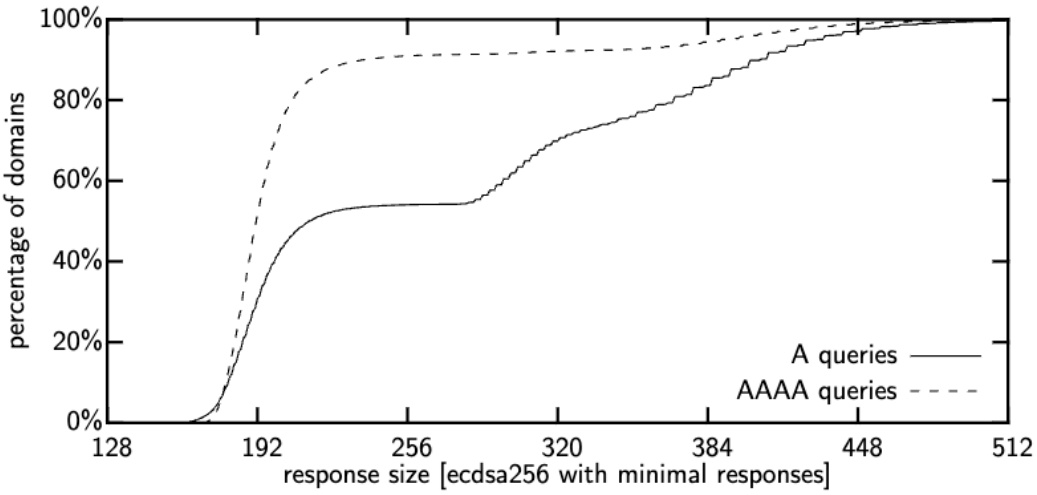
[۹] Distributed Denial of Service
[۱۰] Amplification Attack

رمزنگاری های مختلف ECC می‌توانند تا چندین برابر کند تر از RSA باشد.

	RSA 1024	RSA 2048	RSA 3072
ECDSA P-256	26.3 (6.6)	13.4 (3.4)	7.6
ECDSA P-384	79.1	40.4	22.8
EdDSA	5.6	2.9	1.6

شکل ۹: هر خانه می گوید رمزنگاری متناظر ECC چند برابر رمزنگاری متناظر RSA دور کلاک نیاز دارد.

- **حجم** به طور کلی حجم مرسوم رکورد های DNS حداکثر ۵۱۲ بایت بوده (این در RFC نیامده ولی به طور مرسوم اینقدر بوده است.) از آنجایی که حجم ECC کمتر است میتوان انتظار داشت ردپای کمتری روی حجم جواب ها به نسبت RSA داشته باشد که همین طور هم است.



شکل ۱۰: میتوان دید که اکثر جواب ها زیر ۵۱۲ byte فضا اشغال می‌کنند.

- Adaption طی تحقیقی که در سال ۲۰۱۷ انجام شد، اکثریت دامنه ها همچنان از RSA استفاده می‌کنند و درصد کمی از آن ها از ECDSA استفاده می‌کنند.

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

شکل ۱۱: میزان استفاده از الگوریتم های رمزنگاری در Second level domain ها (ژانویه ۲۰۱۷)

۲.۲.٤ Zone enumeration

همانطور که بیان شد وجود رکورد های NSEC به فرد این امکان را می‌دهد تا همه دامنه های یک zone را دربیاورد.

البته اگر از NSEC3 استفاده شود به علت اینکه از هش دامنه استفاده می‌کند این کار سخت تر می‌شود. ولی در تحقیقی که برای یافتن همه گیری DNSSEC انجام شده [۱۰] ، با استفاده از ۷ عدد GPU و شیوه Brute force حدود ۷۸ درصد هش های رکورد های NSEC3 ظرف ۲ هفته شکسته شده اند.

البته در همین تحقیق شکستن هشت رکورد NSEC3 دو مورد از TLD ها با شکست مواجه شد. یک مورد به علت خطای تنظیم NSEC از سوی TLD بوده ولی مورد دیگر دربار TLD مربوط به py بوده که salt هش را هر ده دقیقه یکبار عوض میکرده و شکستن هش را سخت و عملا غیر ممکن می‌سازد.

NSEC5 برای حل مشکل Zone Enumeration یکی از راه حل هایی که برای این مشکل مطرح شد،ایده NSEC5 بود [۲] . اساس این ایده بر این پایه است که مشکل NSEC3 استفاده از الگوریتم های هش بدون کلید است و اگر از یک هش با کلید استفاده کنیم،(عملا امضا را به جای هش محاسبه کنیم) از آنجایی که فقط سرور قابلیت ایجاد هش را دارد ولی همه کاربر ها قابلیت تایید هش را دارند و می‌توانند ببینند هش گرفته شده از سرور آیا مربوط به دامنه مدنظرشان است یا خیر.

این ایده با اینکه در ژورنال معروفی چاپ نشده ولی نسخه پیش چاپ آن در آرشیو iacr [۱۱] بیش از ۸۰ مرتبه ارجاع خورده است.

۵ جمع‌بندی

مسائلی که DNSSEC قصد دارد آن ها را حل کند، مسائل درستی هستند ولی به علت پیچیدگی بیش از حد این افزونه و اثر آن در Performance و راحتی استقرار باعث شده که اقبال عمومی به سمت آن کم باشد. کما اینکه در TLD ها که کار اصلی آن ها این است میبینیم DNSSEC توانسته Adaption خوبی داشته باشد.

[۱۱] https://eprint.iacr.org

مهم ترین جایی که DNSSEC نتوانسته به نرخ پذیرش خوبی برسد، در SLD [۱۲] ها است. هرچند با توجه به اینکه برای چالش های مختلف آن مثل مانیتورینگ، Performance ، حجم درخواست ها، Fragmentaion و ... راه حل هایی ارائه شده. به نظر می‌آید که مشکل جدی ای برای DNSSEC نباشد و با رفع این مشکلات بتواند Adaption بالایی داشته باشد.

[۱۲] Second level domain

٦ مراجع

[1] Nikolaos Alexiou et al. “Formal Analysis of the Kaminsky DNS Cache-Poisoning Attack Using Probabilistic Model Checking”. In: *IEEE* (). DOI: https://doi.org/10.1109/HASE.2010.25.

[2] Sharon Goldberg et al. *NSEC5: Provably Preventing DNSSEC Zone Enumeration*. Cryptology ePrint Archive, Paper 2014/582. 2014. URL: https://eprint.iacr.org/2014/582.

[3] *How does DNSSEC work?* Cloudflare. URL: https://www.cloudflare.com/learning/dns/dnssec/how-dnssec-works/.

[4] Christopher Makarem. *How DNSSEC Works*. URL: https://medium.com/iocscan/how-dnssec-works-9c652257be0.

[5] Daniel Migault, Cédric Girard, and Maryline Laurent. “A performance view on DNSSEC migration”. In: *2010 International Conference on Network and Service Management*. 2010, pp. 469–474. DOI: 10.1109/CNSM.2010.5691275.

[6] Daniel Migault et al. “PREFETCHing to overcome DNSSEC Performance Issue on large Resolving Platform”. In: July 2013, pp. 694–703. DOI: 10.1109/TrustCom.2013.84.

[7] Eric Osterweil, Dan Massey, and Lixia Zhang. “Deploying and Monitoring DNS Security (DNSSEC)”. In: *2009 Annual Computer Security Applications Conference*. 2009, pp. 429–438. DOI: 10.1109/ACSAC.2009.47.

[8] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. “Making the Case for Elliptic Curves in DNSSEC”. In: *SIGCOMM Comput. Commun. Rev.* 45.5 (Sept. 2015), pp. 13–19. ISSN: 0146-4833. DOI: 10.1145/2831347.2831350. URL: https://doi.org/10.1145/2831347.2831350.

[9] Jonathan Spring and Leigh B. Metcalf. *Probable Cache Poisoning of Mail Handling Domains*. Carnegie Mellon University, Software Engineering Institute. URL: https://insights.sei.cmu.edu/blog/probable-cache-poisoning-of-mail-handling-domains/.

[10] Matthäus Wander. “Measurement survey of server-side DNSSEC adoption”. In: *2017 Network Traffic Measurement and Analysis Conference (TMA)*. 2017, pp. 1–9. DOI: 10.23919/TMA.2017.8002913.