Kartläggning av DNSSEC-kvalitet hos DNSSEC-säkrade se-domäner

Mapping the DNSSEC Quality of DNSSEC-secured .se Domains

Thomas Ido

Examensarbete inom
Elektroteknik
Grundnivå, 15 hp
Handledare på KTH: Maksims Kornevs
Examinator: Elias Said
TRITA-CBH-GRU-2023:083

KTH
Skolan för kemi, bioteknologi och hälsa
141 52 Huddinge, Sverige

Sammanfattning

Sveriges toppdomän, *.se*, drivs av Internetstiftelsen som också främjar användningen av DNSSEC, vilket är ett tillägg till protokollet DNS (Domännamnsystemet) som ger ett säkerhetsskydd till DNS-poster. Internetstiftelsen har god kunskap om antalet DNSSEC-säkrade domäner under *.se*, men ingen kunskap om dessa är uppsatta i enlighet med rekommendationerna för DNSSEC. I detta arbete görs en kartläggning av DNSSEC-kvaliteten hos underdomäner till *.se*-domänen, som är utpekade som DNSSEC-signerade. Slutsatsen i studien blev att majoriteten av de undersökta domänerna har implementerat DNSSEC i enlighet med DNS-standarder och rekommendationer. Däremot finns ett litet antal domäner som visar brister när det gäller DNSSEC. Denna kartläggning visar alltså att majoriteten av *.se*-domänerna har en god DNSSEC-kvalitet, men att det finns utrymme för förbättringar för att säkerställa en högre säkerhetsnivå för samtliga domäner.

Nyckelord
DNS, DNSSEC, toppdomän, DNSKEY, Internetstiftelsen.

Abstract

Sweden's top-level domain, *.se*, is operated by Internetstiftelsen (The Swedish Internet Foundation), which also promotes the use of DNSSEC, an extension to the DNS (Domain Name System) protocol that provides security protection to DNS records. Internetstiftelsen has good knowledge of the number of DNSSEC-secured domains, but no knowledge of whether they are set up in accordance with DNSSEC recommendations. This study maps the DNSSEC quality of subdomains to the *.se* domain, which are designated as DNSSEC-signed by the *.se* domain. The conclusion of the study is that the majority of the examined domains have implemented DNSSEC in accordance with DNS standards and recommendations. However, there are a small number of domains that show deficiencies in DNSSEC. This mapping thus shows that the majority of *.se* domains have good DNSSEC quality, but that there is room for improvement to ensure a higher level of security for all domains.

Keywords
DNS, DNSSEC, top-level domain, DNSKEY, Internetstiftelsen.

Förord

Denna rapport representerar kulmen på mina studier inom Elektroteknik vid Kungliga Tekniska Högskolan och är resultatet av ett examensarbete som jag har utfört under våren 2023. Arbetet har utförts i samarbete med Internetstiftelsen. Jag vill framföra mitt djupaste tack till min handledare från Internetstiftelsen, Mats Dufberg. Mats, din expertis, tålamod och ständiga beredskap att vägleda och ge råd har spelat en nyckelroll i detta examensarbete.

Jag vill även tacka min handledare från Kungliga Tekniska Högskolan, Maksims Kornevs. Under examensarbetets gång har Maksims aktivt hjälpt mig med att förbättra kvaliteten i rapporten.

Sist men störst av allt, vill jag tacka min Herre och Gud, Jesus Kristus. Ett bibelcitat som varit med mig under examensarbetet och under hela utbildningen är: *”Allt förmår jag i honom som ger mig kraft”* – Filipperbrevet 4:13.

Innehållsförteckning

1 Inledning

Internet har utvecklats och vuxit fram till att bli en viktig informationsinfrastruktur för sociala interaktioner mellan människor. Genom Internet är det möjligt för människor över hela världen att ansluta och kommunicera med varandra. För att bli tillgänglig på Internet måste varje dator ges en eller flera IP-adresser. Men eftersom IP-adresser är svåra för människor att komma ihåg, används domännamn istället. Domännamn är en läsbar text för människor som kopplas samman med en eller flera IP-adresser eller annan data för Internet. När en person skriver in ett domännamn på Internet så översätts domännamnet till en eller flera IP-adresser, vilket möjliggör en korrekt anslutning till rätt enhet.

DNS (Domännamnsystemet) har kommit att bli en effektiv distribuerad databas för hela internet. DNS är avgörande för Internetarkitekturen eftersom det möjliggör mappning mellan domännamn och IP-adresser. Denna mappning lagras i en hierarkisk trädstrukturerad distribuerad databas. Toppdomäner (eng. TLD) utgör nivån närmast rot i DNS-trädet, ett exempel på en toppdomän är *.com*. Varje land har en egen toppdomän, i Sverige är det *.se*. Eftersom domännamn börjar med den minst signifikanta delen innebär det att alla domännamn avslutas med en toppdomän, som till exempel *kth.se* [1].

1.1 Problemformulering

Toppdomänen *.se* är knuten till landet Sverige (men tillgänglig från hela Internet). *.se* drivs av Internetstiftelsen som främjar användningen av DNSSEC i domännamnen under *.se*, t.ex. *kth.se*. (beskrivning av DNSSEC kommer att ges kapitlet Teori och bakgrund). Internetstiftelsen har god kunskap om antalet DNSSEC-säkrade domäner, men ingen kunskap om dessa är uppsatta i enlighet med rekommendationerna för DNSSEC.

Examensarbetet ska kartlägga DNSSEC-kvaliteten hos alla DNSSEC-säkrade domäner under *.se*. Genom att kartlägga DNSSEC-kvaliteten hos *.se*-domäner kan man identifiera eventuella säkerhetsrisker, samt sårbarheter i implementationen av DNSSEC och även förhindra störningar, vilket kan bidra till att förbättra säkerheten och skydda mot attacker. Resultatet av denna undersökning bidrar till en förståelse av hur väl domänerna under *.se* möter kraven från DNSSEC-standarden.

1.2 Målsättning

Målsättningen med detta examensarbete är att undersöka alla DNSSEC-signerade domäner under toppdomänen *.se*. Examensarbetet ska kartlägga hur olika DNSSEC-parametrar hanteras genom att besvara följande frågeställningar för samtliga undersöka domäner under *.se*:

- Vilka algoritmer används för DNSKEY i varje undersökt domän (zon) och är dessa i enlighet med DNS-standarden?
- Om algoritmen är RSA, vilka nyckellängder används och är dessa i enlighet med DNS-standarden?
- När det gäller DNSKEY så kan en zon ha KSK-ZSK-par eller bara en CSK nyckel. Hur ser fördelningen ut inom materialet (de undersökta *se*-domänerna)?
- I vilken utsträckning finns det CDS-post resp. CDNSKEY-post i de undersökta domänerna under *.se*?
- Vilken signaturlivslängd används för RRSIG, och då gäller det de som används för SOA, NS, DNSKEY RRset, i de undersökta domänerna under *.se*?
- Är DNSSEC giltigt, och det ska värderas ifall SOA, NS och DNSKEY RRset kan valideras m.h.a. befintlig DS-post, i de undersökta domänerna under *.se*?

Detta är viktigt eftersom det ger en större förståelse om de undersökta domänerna under *.se*, vilket innebär att man kan identifiera eventuella sårbarheter enklare.

1.3 Avgränsningar

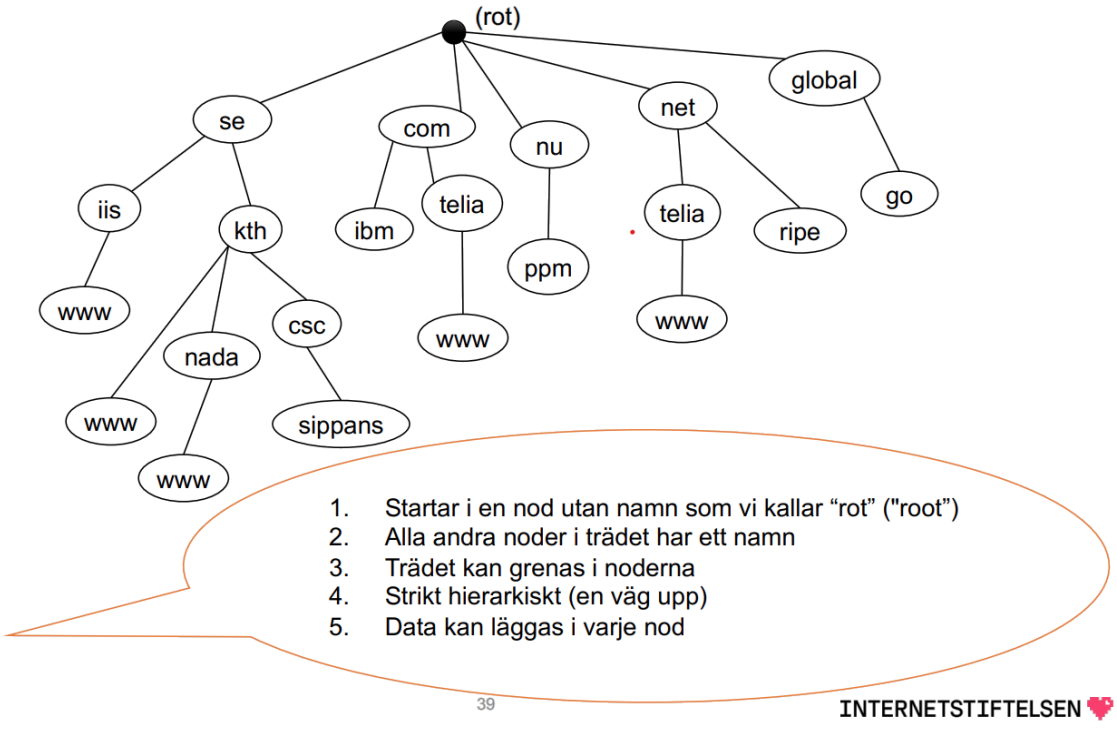
De avgränsningar som gäller för detta arbete inkluderar endast domäner som är registrerade under toppdomänen *.se* och är från *.se*-zonen utpekade som DNSSEC-signerade. Fortsättningsvis, kommer endast DNSSEC-parametrar som är associerade med DNSSEC-signerade domäner att undersökas. Andra säkerhetsaspekter som inte rör DNSSEC kommer inte beaktas i denna studie. Detta arbete kommer enbart fokusera på de undersökta domänerna under *.se*. Eventuella sårbarheter som upptäcks kommer inte att åtgärdas inom ramen för denna studie.

2 Teori och bakgrund

Detta kapitel beskriver relevant teori och bakgrund för arbetet och ger en teknisk bakgrund och förklaring till vad som tidigare har gjorts inom ämnesområdet.

2.1 Översikt av DNS

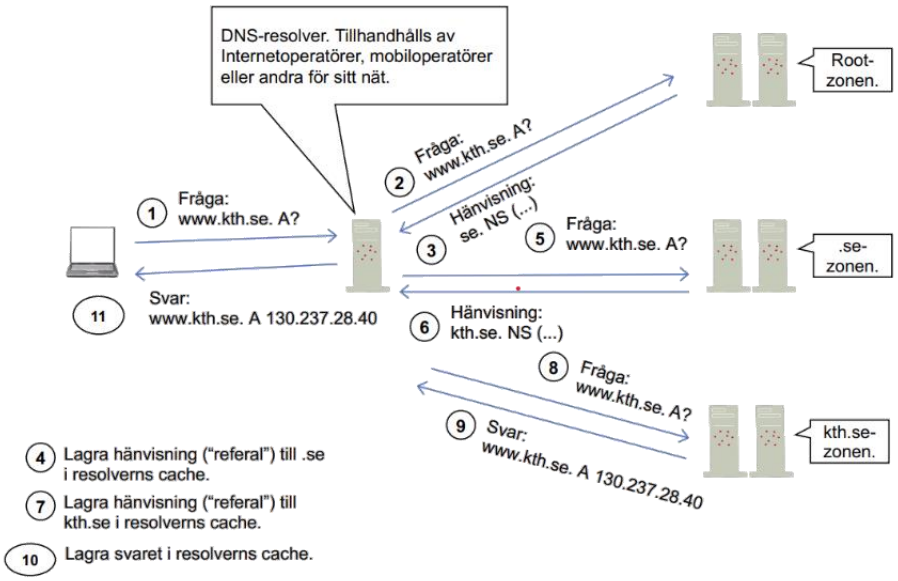
Varje domän i DNS är egentligen en sökväg i ett stort omvänt träd, kallat för DNS-trädet. Trädets hierarkiska struktur visas i figur 1, trädstrukturen i DNS är lik Unix-filsystemet [1]. Det finns särskilda servrar, kallade för auktoritativa namnservrar, som hanterar DNS-trädet och DNS-data. Alla domännamn är organiserade i en hierarkisk struktur, med en namnlös rot i toppen. Under rotnivån finns så kallade toppdomäner, som till exempel *.se* för Sverige, *.com* för kommersiella organisationer och *.org* för organisationer [2]. Under toppdomäner finns det domännamn för privatpersoner, myndigheter, företag och organisationer, till exempel *kth.se* för Kungliga Tekniska Högskolan i Sverige. Genom att använda DNS och domännamn, kan internetanvändare snabbt och enkelt hitta de tjänster de söker eftersom de bara behöver skriva in ett meningsfullt och lättförståeligt domännamn, i stället för att komma ihåg en lång och krånglig IP-adress. När en användare exempelvis försöker få åtkomst till en webbsida via ett domännamn, gör DNS en sökning i sitt register och returnerar IP-adressen eller IP-adresserna för den efterfrågade webbsidan. Dessutom används domännamn även som ett ankare för annan teknisk data, exempelvis kryptonycklar som kan läggas i DNS [3].



Figur 1. DNS-trädet [4].

2.1.1 Resolver

En central funktion i DNS är resolvern. En resolver är en tjänst i DNS som fungerar som en uppslagningsfunktion eftersom den hittar svaret på frågan vilken IP-adress ett domännamn motsvarar. Resolvrar realiseras som namnservrar med resolvningsfunktion [3]. För att IP-adressen för exempelvis www.*kth.se* ska erhållas, söker DNS-resolvern genom DNS-hierarkin i en viss ordning. Processen inleds med en förfrågan till rot, rot ger en hänvisning till toppdomänen *.se*, detta används för att göra en förfrågan till *.se*-zonen. På så sätt kan resolvern identifiera DNS-servern som ansvarar för *kth.se* genom att erhålla en hänvisning till namnservern för *kth.se* från *.se*. Nästa steg är att resolvern kan identifiera de ansvariga namnservrarna för *kth.se*-zonen. Slutligen skickar resolvern en förfrågan till DNS-servern som ansvarar för *kth.se*-zonen och ber om IP-adress för *www.kth.se*. DNS-servern söker sedan i sin databas efter IP-adressen och skickar tillbaka svaret till resolvern. Resolvern returnerar svaret till webbläsaren, vilket gör det möjligt för webbläsaren att ansluta till *www.kth.se*, se figur 2 [1] [4].



Figur 2. Uppslagning via resolver [4].

2.1.2 Delegering

Delegering innebär att en zon överlämnar kontrollen för en del av sin namnrymd till en eller flera namnservrar. Därmed uppstår en ny DNS-zon under den första DNS-zonen. För att kunna upprätthålla DNS över hela Internet har man utnyttjat att DNS är en distribuerad databas genom att dela upp DNS-trädet i mindre zoner och fördela administrationen. Delegering är en central teknik inom DNS, eftersom det gör det möjligt för organisationer att förvalta sina egna namnservrar och namnrymder. Detta resulterar att organisationer får en större flexibilitet och kontroll över sina domäner. Delegering går till genom att en överordnad zon pekar på namnservrar som ansvarar för den underliggande zonen. På så sätt är DNS hierarkiskt eftersom det

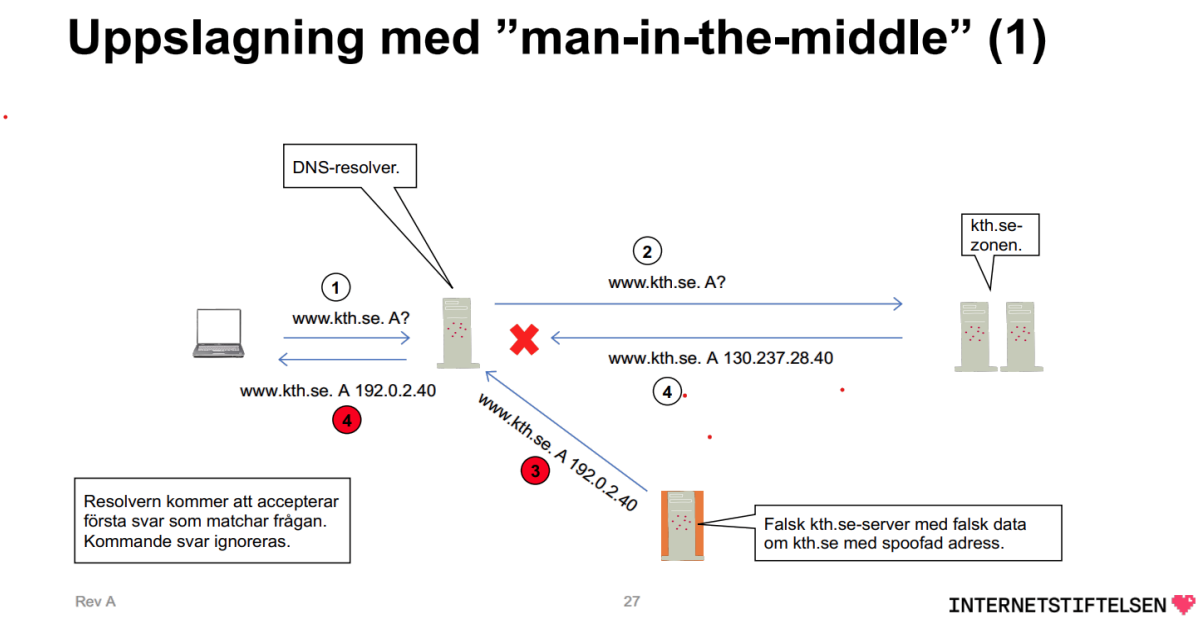
följer en strukturerad ordning för att hantera domännamn [5]. Utpekningen av namnservrar spelar en nyckelroll i delegeringen eftersom de bestämmer vilka servrar som ska hantera vilka domäner. Eftersom DNS är hierarkiskt utformat behöver inte toppdomänen *.se* lagra all information om *kth.se*, utan endast de namnservrar som ansvarar för *kth.se* (*kth.se*-zonen). Detta innebär att *.se*-zonen känner till var all information om *kth.se* och dess underdomäner finns, men har själv inte den informationen [6].

2.2 Säkerhetsproblem i DNS

Traditionellt, har DNS varit utan kryptografiskt skydd och potentiellt sårbart för förändringar av data i ont syfte. Detta har gjort det möjligt för angripare att manipulera data i DNS-system, vilket har resulterat till allvarliga säkerhetsproblem såsom cacheförgiftning (eng. cache poisoning) som innebär att man lyckas få in skadlig och felaktig data i en resolvers cache, som sedan överförs till de som frågar efter den. Det kan exempelvis handla om en felaktig A- eller AAAA-post med en IP-adress som styr offret till en falsk webbplats [4].

För att skydda sig mot DNS-cache-förgiftning är det viktigt att kunna säkerställa att svaret kommer från den ursprungliga källan. Ett sätt att uppnå detta är genom att använda ID-nummer i DNS-svaret, ett svar måste ha samma ID som frågan för att det ska kunna accepteras. Eftersom ID är ett heltal med 16 bitar ger det 65.536 olika möjliga värden. Säkerhetsexperten Dan Kaminsky upptäckte att om en angripare kan förutse ID-numret kan denne enkelt injicera falska svar i cacheminnet hos DNS-servrarna och genomföra en attack [6] [4].

I traditionell DNS har mottagaren av data från en DNS-namnserver ingen möjlighet att autentisera dess ursprung eller validera dess data. Resolvern kan endast autentisera ursprunget av DNS-svarsdatapaket med hjälp av IP-adressen för DNS-servern, destination- och källportnummer och DNS-transaktions-ID. En angripare kan enkelt skapa ett DNS-svarspaket för att uppfylla dessa parametrar och klienten har inget annat val än att lita på den data som tillhandahålls av en angripare som pålitlig, se figur 3 [6].



Figur 3. Man-in-the-middle attack [4].

Sedan drygt 15 år finns en utökning av protokollet DNS som heter DNSSEC, som tillför säkerhet till DNS-protokollet genom att ge ursprungssäkerhet och dataintegritet. Varje DNS-post i en DNSSEC-zon är digitalt signerat. Genom att göra en kontroll av signaturen kan man säkerställa att informationen härstammar från den ursprungliga källan. Med DNSSEC skyddas man från man-in-the-middle (MITM)-angripare eftersom man kan utnyttja digitala signaturer i svaren [7].

Vad som är speciellt för utökningen DNSSEC är att DNSSEC bygger på en hierarkisk tillitsmodell. Detta betyder när tillit har etablerats för zonen genom digitala signaturer, är det möjligt att överföra denna tillit till underliggande zoner. Detta kallas för tillitskedja (eng. chain of trust) [7].

Orsaken till att resolvern kan lita på rots data är för att rot är DNSSEC-signerad och resolvern har rots publika DNSSEC-nyckel som tillitsankare. Tillitsankaret är alltså en publik nyckel som placeras i en validerande resolver av den anledningen att validerande resolver ska kunna validera resultaten för en given förfrågan tillbaka till en känd eller betrodd offentlig nyckel [8]. Ur en icke-teknisk synvinkel är det en förutsättning för DNSSEC-modellens funktion att man har förtroende för ICANN:s avancerade processer som ansvarar för hanteringen av DNSSEC för rotzonen [9].

2.3 Internetstiftelsen och toppdomänen *.se*

Som konstaterats ovan drivs toppdomänen *.se* av Internetstiftelsen, en organisation som främjar användningen av DNSSEC i domännamnen under *.se* [3]. Internetstiftelsen har gjort stora ansträngningar för att öka antalet DNSSEC-säkrade domäner under *.se*. Idag har drygt 950 000 domännamn under *.se* skydd genom DNSSEC, av en totalt ungefär 1,5 miljoner domäner [10]. Detta visar på en stark trend för användningen av DNSSEC i Sverige och ett ökat fokus på cybersäkerhet.

Som redan konstaterats i inledningen är det viktigt att notera att Internetstiftelsen har god kunskap om antalet DNSSEC-säkrade domäner under *.se*, men ingen kunskap om dessa är uppsatta i enlighet med rekommendationerna för DNSSEC. Det är upp till domännamnsinnehavarna att säkerställa att deras DNSSEC-säkring är i enlighet med de relevanta standarderna och rekommendationerna. Sammantaget är DNSSEC en viktig säkerhetsfunktion som främjar en säker användning av Internet, och Internetstiftelsen spelar en central roll i att främja DNSSEC i Sverige genom sitt arbete med *.se*-toppdomänen.

2.4 DNSSEC

DNSSEC adderar ett lager av säkerhet för DNS-uppslagningar. Det sker genom att man lägger till kryptografiska signaturer till befintliga DNS-poster. Äktheten säkerställs genom en hierarkisk nyckelhantering, vilket innebär att varje nivå i domännamnsystemet är beroende av den högre nivån. Detta betyder att en zon kan endast verifieras genom sin egen överordnad zon, och denna överordnade zon kan endast verifieras av sin egen överordnade zon. Detta fortsätter ända vägen upp till rot och därmed etableras en hierarki av tillit [11].

2.4.1 Kryptografisk signatur

För att skydda DNS-data från att bli förvanskat och manipulerat använder man i DNSSEC kryptografiska hashfunktioner. Hashen i den kryptografiska signaturen är krypterat med den privata nyckeln och kan verifieras med den publika nyckeln (DNSKEY). En hashfunktion är en matematisk funktion, som implementerar matematiska regler för att omvandla data till ett kortare nummer. En hashfunktion tar en mängd data som inmatning och genererar en kort sträng av tecken som representerar denna data. Hashing verifierar DNS-datat genom att jämföra ursprungliga hashvärdet med det hashvärde som genereras av resolvern på en senare tidpunkt. Om de två hashvärderna stämmer överens med varandra betyder det att DNS-datat inte har manipulerats. Om dataändringar skulle inträffa i hashen, kommer hashvärdet att ändras. Mottagaren kan alltså verifiera mottagen DNS-data [4] [12].

2.4.2 SOA-post

SOA-posten (Start of Authority) är en viktig resurspost i DNS som finns i varje zon och som anger auktoriteten för en zon. SOA-posten innehåller inställningsparametrar för zonen och dess sekundärservrar [7].

2.4.3 DNSKEY

För att skapa digitala signaturer på DNS-data används kryptografiska algoritmer, t.ex. RSA. Algoritmerna som används är asymmetriska och består av ett nyckelpar, en publik nyckel och en privat nyckel. DNSKEY är en posttyp i DNSSEC som innehåller den publika nyckeln. Resolvers använder DNSKEY-posten för att verifiera digitala signaturer som har signerats med den privata nyckeln. Den privata nyckeln är placerad utanför DNS, t.ex. i en HSM [13].

2.4.4 DS

DS-posten är en posttyp som innehåller en hash av DNSKEY. DS-posten är placerad i moderzonen, medan DNSKEY finns i dotterzonen. Detta innebär att när en resolver gör en uppslagning mot ett domännamn i en zon med DNSSEC, kommer resolvern

att hasha dotterzonens DNSKEY och jämföra hashvärdet med innehållet i DS-posten. Om de är identiska är DNS-data från uppslagningen autentisk [14].

2.4.5 CDNSKEY och CDS

Att byta nycklar för att signera domäner med DNSSEC har blivit mycket enklare med CDS och CDNSKEY. CDNSKEY är en DNS-post som likt DNSKEY också innehåller en publik nyckel som används för att verifiera en domän [15]. CDNSKEY placeras i dotterzonen och används av moderzonen för att skapa och uppdatera motsvarande DS-post. Med CDNSKEY är det möjligt för domänägaren att kontrollera DNSSEC-statusen utan att behöva kontakta registrar eller registry [16] [17].

När en DNSSEC-zon lägger till eller ändrar sina DNSSEC-nycklar, måste DS-posten i överliggande zon uppdateras för att det ska gå att verifiera den nya eller ändrade nyckeln, och därmed bevara tillitskedjan. För att undvika risken för fel med manuell hantering, kan man genom CDS-posttypen automatisera nyckelhanteringen och signaleringen i DNSSEC. Genom CDS- och CDNSKEY-posten kan man alltså automatisera nyckelhanteringen och signaleringen i DNSSEC istället för att sköta det manuellt och riskera fel [17].

2.4.6 RRset

Första steget mot att säkra upp zonen med DNSSEC är att gruppera alla poster med samma ”owner name” och posttyp i en Resource Record set (RRset). Om en zon har fem AAAA-poster med samma domännamn, exempelvis *kth.se*, skulle dessa bli ett enda AAAA RRset [18].

2.4.7 RRSIG

En RRSIG-post innehåller en kryptografisk signatur för ett RRset (se beskrivning av kryptografisk signatur ovan). En zon signerar sina auktoritativa RRset genom den privata nyckeln, och därmed skapas RRSIG-posterna. Motsvarande publika nyckel lagras i DNSKEY RR (eng. Resource Records). Resolvern kan då använda publika nyckeln för att validera digitala signeringen på RRSIG-poster. RRSIG-posten innehåller referens till vilket RRset som den är signatur för och information om algoritmen som användes för att generera signaturen, giltighetstiden och publika nyckeln som tillämpades för att validera den digitala signaturen [19].

2.4.8 ZSK och KSK

Varje zon som är signerad med DNSSEC har en zonsigneringsnyckel, eller ZSK (eng. ”Zone Signing Key). Denna nyckel består av både en publik nyckel (publicerad i DNS) och en privat nyckel (lagrad utanför DNS). Den privata nyckeln signerar varje RRset i zonen med undantag DNSKEY RRset, vilket resulterar i skapandet av RRSIG-poster för alla RRset. Den publika nyckeln används för att verifiera signaturerna hos RRSIG-posterna. Detta innebär när en DNS-resolver gör en förfrågan om en specifik posttyp, kan namnservern svara med motsvarande RRSIG-post. Resolvern kan sedan validera RRSIG-posten genom DNSKEY-posten, som innehåller ZSK:s publika nyckel [18].

Namnservrar som använder DNSSEC har utöver ZSK även en KSK, ”Key Signing Key”. KSK validerar DNSKEY RRset på samma sätt som ZSK säkrade upp andra

RRset. Mer specifikt, signerar KSK hela DNSKEY RRset, vilket inkluderar den publika delen av både ZSK och KSK själv. Detta resulterar i skapandet av en RRSIG för DNSKEY RRset. För att validera DNSSEC-skyddade zondata som har signerats med KSK:s privata nyckel, krävs KSK:s publika nyckel. Detta innebär att KSK skapar digitala signaturer för DNSSEC-RRset. KSK fungerar som en form av DNSKEY som används för att signera andra DNSSEC-nycklar, vilket skapar ett så kallat tillitsankare. Detta tillitsankare, som är en KSK-nyckel används sedan av DNS-resolvern för att verifiera DNSSEC-nycklarna längs DNS-hierarkin [18].

2.4.9 Tillitskedja

Tilliten till KSK grundar sig på DS-posten (Delegation Signer) som publiceras i den överordnade zonen. DS-posten verifierar KSK i den underliggande zonen, vilket skapar en tillitskedja [15]. Tillitskedjan bygger på principen att en DNSSEC signerad zon är kryptografiskt skyddad ända upp till rot. Tillitskedjan etableras när tillit finns inom zonen och sen genom DS-posten tillåts överföring av tillit till underliggande zon. DS-posten är avgörande för trovärdigheten eftersom den utgör länken mellan zonerna. Anledningen till att man kan lita på DS-posten är eftersom DS-posten också är signerad och har sin RRSIG-post i sin överordnad zon, och detta fortsätter upp till rot [18].

2.5 Tidigare arbeten

Artikeln från ICANN:s Office of the Chief Technology Officer (OCTO) undersöker användningen av algoritmer och nyckelstorlekar i DNSKEY-poster. Studien baseras på data från både generiska toppnivådomäner (gTLD) och landskodade toppnivådomäner (ccTLD). Generiska toppnivådomäner (t.ex. *.com*) är inte kopplade till en geografisk plats och i undersökningen används data från samtliga gTLD-zoner. Dessutom inkluderas data från landskodade toppnivådomäner, såsom *.se* och *.nu* (tillhandahållna av Internetstiftelsen), samt .ch och .li (tillhandahållna av SWITCH). Resultaten från undersökningen visar att de mest använda algoritmerna i DNSKEY-poster är RSA med 65,2% och ECDSA med 30,5%. När det gäller RSA-algoritmer, är de mest förekommande nyckelstorlekarna 1024 bitar (45,1%), 2048 bitar (51,4%) och 4096 bitar (0,7%) [21].

Enligt en annan artikel [22] från 2018 använder över 90 procent av alla toppdomäner (TLDs) DNSSEC. Detta ligger i linje med förväntningarna, eftersom det är ett krav att implementera DNSSEC för alla TLDs som har ett avtal med ICANN [12]. Däremot är det endast 3 procent av underdomänerna till toppdomänerna världen över som har implementerat DNSSEC. För att öka användningen av DNSSEC hos underdomäner finns det ekonomiska incitament i form av årlig rabatt för varje undertecknad domän. Däremot är det oklart om dessa incitament främjar säkerheten och tillämpningen av DNSSEC på ett bra sätt. Eftersom om det existerar fasta kostnader för att implementera DNSSEC kan ekonomiska incitament som bara gäller per domän leda till att stora operatörer väljer att implementera osäkra varianter av DNSSEC. Forskare har studerat hur säker DNSSEC är i de högst nivå-domänerna som erbjuder ekonomiska incitament och hittat att säkerheten i DNSSEC i de undersökta domänerna inte följer standardrekommendationer. Dessutom har det visat sig att stora operatörer har haft svagare DNSSEC-säkerhet än små operatörer, som då

antyder att nuvarande incitament inte är hållbart för att öka säkerheten och avskräcka osäkra implementeringar. Artikeln menar att ekonomiska incitament har bidragit till ökat antal DNSSEC-implementeringar, däremot visar studien att ekonomiska incitament inte ger samma effekt på kvaliteten på DNSSEC-implementeringen. Därför förespråkar studien att ändra fokus för ekonomiska incitament från kvantitet till kvalitet. Artikeln tar upp att dessa argument har framkommit i en privat diskussion med det svenska registret Internetstiftelsen. Internetstiftelsen bekräftade då att deras ursprungliga avsikt var att påskynda kvantiteten av implementeringen och att Internetstiftelsen ser främjandet av kvalitet som ett långsiktigt mål [22].

I [20] utfördes det ett arbete i försök att förklara varför DNSSEC distributionen är låg trots att DNSSEC har funnits i drygt 15 år. Artikeln hävdar att en majoritet av alla toppdomäner använder DNSSEC, däremot är det en mycket liten skara av underdomänerna som har DNSSEC implementerat. Författaren fann att stödet bland registrarer, som är återförsäljare av domännamn, var mycket ojämnt fördelat. De visades sig att ett fåtal av de stora registrerarna hade stöd för DNSSEC som DNS-operatör. Detta innebär att om en domänägare vill implementera DNSSEC för sin domän, kommer det inte att finnas stöd för detta. Artikeln menar att orsaken till att användningen av DNSSEC är låg beror på bristande stöd bland registrarer, dessutom att implementeringen av DNSSEC kan vara komplex med risker för fel. En åtgärd för den låga distributionen är att underlätta DS-postens hantering, och automatisera processen med hjälp av CDS- och CDNSKEY-post, istället för en manuell hantering.

3 Metoder och resultat

Detta kapitel beskriver den valda metodiken, lösningsmetoder och verktygen för examensarbetet.

3.1 Datainsamling

Som beskrivits i kapitel 1.2 delas detta arbete upp i sex frågeställningar som var och en representerar ett steg i arbetsprocessen. Alla sex steg har inkluderat verktyget ”dig”, version ”DiG 9.18.1-1ubuntu1.2-Ubuntu”. Verktyget ”dig” används för att fråga efter specifika DNS-poster. Verktyget utför en DNS-uppslagning och visar svaren från en namnserver. Det är ett känt verktyg i DNS-branschen och används frekvent av DNS-administratörer eftersom det är smidigt att hämta information om domäner [1]. Arbetet utfördes genom en WSL (Windows Subsystem Linux), med ett installerat Ubuntu operativsystem där ”dig” var förinstallerat.

För att kunna besvara frågeställningarna, samlades först all DNS-data för toppdomänen *.se*. Genom att skriva kommandot: ”dig @zonedata.iis.se se AXFR > se.zone.txt” kunde all DNS-data för *.se*-zonen erhållas i en textfil. Därefter skrevs ett program i programmeringsspråket C som extraherade ut alla från *.se*-zonen utpekade domäner som var DNSSEC-signerade, och placerade dessa i en ny textfil som döptes till ”filtrerade_domains.txt”.

Textfilen "filtrerade_domains.txt" innehöll 953 444 DNSSEC-domäner. Med detta som utgångspunkt togs beslutet att göra ett statistiskt urval där 100 000 domäner slumpmässigt valdes ut från filen "filtrerade_domains.txt".

3.1.1 Metod för identifiering av algoritmer för DNSKEY

När urvalet på 100 000 domäner hade erhållits var nästa steg enligt första frågeställningen i arbetet att undersöka vilka algoritmer som används för DNSKEY. För att veta vilka algoritmer som används för DNSKEY, skrevs ett Python-skript som läste in textfilen med 100 000 domäner och gjorde en uppslagning mot varje domän med ”dig”. Därigenom extraherades alla algoritmer från varje undersökt domän till en separat CSV-fil ”output_algorithm.csv”.

Dessutom visade det sig att en del domäner inte fanns eller var trasiga. Skriptet tog hänsyn till att placera alla fungerande domäners DNSKEY i CSV-filen ”output_algorithm.csv” och alla trasiga domäner placerades i en annan CSV-fil som kallades ”emptyDomains.csv”. Bland dessa gavs en del domäner statusen NXDOMAIN, vilket betyder att de inte längre existerar. Anledningen till att de inte längre existerar beror på ett tidsfönster, som innebär när zonfilen hämtades existerade domänerna, men när uppslagningen ägde rum hann domänerna försvinna, och får därför status NXDOMAIN. En annan del av domänerna hade status SERVFAIL, som betydde en av två saker, antingen att domänen hade ett DNSSEC-relaterat problem eller att domänen inte gick att nå för att den var trasig. Därefter skrevs två program i Python, en kod som räknade ut antalet domäner med status NXDOMAIN och en annan kod som räknade ut hur många av domänerna som hade status SERVFAIL som berodde på ett DNSSEC-relaterat problem. I denna studie var det totalt 6091 domäner som

inte togs med av de undersökta 100 000 domänerna, eftersom 3640 hade status NXDOMAIN och 2451 hade status SERVFAIL. Bland SERVFAIL-domänerna var 513 relaterade till DNSSEC-problem, medan 1938 klassades som icke nåbara. När detta hade fastslagits, skrevs en ny Python-kod som rensade alla trasiga domäner (NXDOMAINS och SERVFAIL) från textfilen med 100 000 domäner. Fortsättningsvis beaktar studien endast de domäner som gick att validera DNSKEY på, vilket var 93 909 domäner.

För att kunna presentera innehållet i CSV-filen procentuellt, skrevs ett program i den integrerade utvecklingsmiljön IDE PyCharm i programmeringsspråket Python. Python-koden inkluderade mjukvarubiblioteket pandas för att kunna manipulera och analysera datat i CSV-filen.

3.1.2 Resultat för identifiering av algoritmer för DNSKEY

I den första frågeställningen undersöktes användningen av algoritmer för DNSKEY i respektive domän (zon) och hur väl de följer DNS-standarden.

Resultaten från studien visas i tabell 1, där fördelningen av algoritmer för underdomäner till *se*-zonen presenteras. Tabellen belyser inte enbart de olika algoritmerna, utan även de olika kombinationer som förekommer i zonerna. Enligt tabellen är de två mest populära algoritmerna 13 och 8, samt deras kombination. Tillsammans står dessa algoritmer för mer än 99 procent av underdomänerna i se-zonen.

Tabell 1. Fördelning av algoritmer för underdomäner till se-zonen (separata och kombinerade algoritmer). Algoritmer eller kombinationer av algoritmer som förekommer mindre än 15 gånger är listade under övriga.

Kryptoalgoritm	Antal	Procent
13	56452	61,505
8	33264	36,242
13, 15	1276	1,390
15	452	0,493
13, 8	206	0,224
7	65	0,071
13, 5	27	0,029
Övriga	42	0,0458

Algoritm 8, känd som RSA (Rivest-Shamir-Adleman), är en traditionell kryptografisk algoritm som skapar digitala signaturer och grundar sig på principen om stora primtal och deras multiplikativa inverser [23]. Algoritm 13, som är ECDSA (Elliptic Curve Digital Signature Algorithm), är en jämförelsevis säkrare metod än RSA vid en mindre nyckelstorlek eftersom den bygger på elliptiska kurvor [24]. ECDSA är därför snabbare och mer effektiv än många andra kryptografiska algoritmer. Algoritmerna 13 och 8 är rekommenderade och följer DNS-standarden.

Däremot är algoritmerna 10, 7 och 5 inte rekommenderade att användas. I tabell 1 visas förekomsten av algoritmer med mindre antal än 15 gånger under kategorin ”Övriga”, och algoritm 10 återfinns i den kategorin. Det är dock värt att notera att en icke-rekommenderad algoritm inte nödvändigtvis betyder att den är kryptografiskt osäker. Exempelvis avråds användningen av algoritm 10 för DNSSEC-signering eftersom den saknar bred användning på Internet, vilket gör att den inte bör användas, men den har likvärdig kryptografisk styrka som algoritm 8 [26].

- Algoritm 5, har mnemonik RSASHA1.
- Algoritm 7, har mnemonik RSASHA1-NSEC3-SHA1.
- Algoritm 8, har RSASHA256.
- Algoritm 10, har mnemonik RSASHA512.
- Algoritm 13, ECDSAP256SHA256.
- Algoritm 15, har mnemonik ED25519.

Algoritmerna 5, 7, 10 och 15 används inte lika stor utsträckning som de traditionella algoritmerna 8 och 13 och utgör därför en mindre andel av den totala användningen. En nyare algoritm är algoritm 15 som i jämförelse med algoritm 8 erbjuder förbättrad säkerhet och prestanda [26].

Tabell 1 visar även kombinationer av rekommenderade algoritmer och icke rekommenderade algoritmer, trots att detta avråds. Dessa kombinationer förekommer ändå, om än i begränsad omfattning, exempelvis kombineras algoritm 13 och 5.

3.2.1 Metod för RSA-nyckellängder

Nästa frågeställning som skulle undersökas var ifall algoritmen visade sig vara RSA, och i sådana fall vilka nyckellängder det används för RSA. Med utgångspunkt i frågeställning ett kunde algoritmen för DNSKEY erhållas. Nästa steg var att genom Python-kod konvertera algoritmen som var i kodad i base64 till en binär representation. Den binära representationen bestod av exponentlängd, exponent och modulus. Dessa tre utgör strukturen för RSA-nycklar [27]. Exponentlängden, som anger hur lång exponenten är, kan vara antingen en eller tre oktetter. Exponentlängden bestämmer längden på exponenten som är en komponent i nyckeln som används för att kryptera data. Den sista komponenten i strukturen modulus tar upp resterande utrymme efter exponenten. Enligt RFC ska exponenten vara 65337 [28], vilket är vad

detta arbete utgått från att exponenten är. Eftersom studier har visat att 2048 bitar är en standardiserad nyckellängd [29], kan slutsatsen dras att av den binära representation som är 2080 bitar, är 2048 bitar enbart nyckellängd och resterande bitar är exponent och exponentlängd.

3.2.2 Resultat för RSA-nyckellängder

Resultatet presenterar fördelningen av nyckellängder för RSA-algoritmer. Nyckellängder för RSA-algoritmer anses vara säkra och i enlighet med DNS-standarden inom intervallet 512 till 4096 bitar [25]. Däremot rekommenderas en minsta nyckellängd på minst 2048 bitar [31].

Som beskrivits tidigare är primtal en fundamental faktor i RSA-algoritmen [23]. Ett primtal är ett heltal som är större än ett och som enbart delas med sig själv eller ett. Ett exempel på tal som är primtal 3, 5, 7 och 11. Antalet primtal kan bli oändligt, och nyckellängder med 2048 bitar kan vara upp till 600 siffror långa. Detta innebär att längre nyckellängder resulterar att även dekrypteringen blir mycket långsammare. För att optimera balansen mellan snabbhet och säkerhet är det viktigt at välja en lämplig nyckellängd [23]. I detta fall enligt tabell 3 är det tydligt att den mest dominerande nyckellängden är 2048 bitar.

Tabell 2. RSA-nyckellängder. Nyckellängder med mindre antal än 10 är listade under övriga.

Binär representation	Nyckellängd	Procent	Antal
4128	4096	0,092	90
2080	2048	98,281	95762
1568	1536	0,133	130
1312	1280	0,14	136
1056	1024	1,343	1309
Övriga	1304 till 3072	0,010	10

Tabellen 2 visar att 98,28 procent av RSA-nyckellängder i de undersökta domänerna använder en nyckellängd på 2048 bitar. Men andra nyckellängder förekommer också som är i enlighet med DNS-standarden, men som inte rekommenderas.

3.3.1 Metod för ZSK/KSK och CSK

Frågetällning tre i processen var att undersöka fördelningen av nycklar mellan ZSK/KSK (beskrivning av ZSK/KSK hittas i delkapitel 2.5.9 och 2.5.10) och CSK inom olika zoner för de undersökta domänerna. CSK, ”Combined Signing Key”, är en

kombinerad signeringsnyckel som då betyder att nyckeln är en KSK, men även fungerar som en ZSK. Så en CSK-nyckel används istället för två separata nycklar som i fallet med ZSK och KSK [26]. Det är möjligt att en zon har en ZSK-nyckel och en KSK-nyckel, men implementeringen av zonen är av typen CSK, alltså att ZSK-nyckel är helt oanvänd i zonen. Fältet ”flags” som innehåller flaggorna 256 och 257 är en decimalrepresentation av ett bit-signifikant fält. Det decimala talet 256 innebär att bit 7 i 16-bitars fältet indikerar en ZSK. Det decimala talet 257 representerar både bit 7 (ZSK) och bit 15. Att bit 15 är satt innebär att SEP ”Secure Entry Point-bit” är satt, vilket betyder att nyckeln är utpekad av DS i överordnad zon [1].

Efter att steg ett och två hade fastställts, genomfördes implementeringen av detta steg smidigt. Genom att använda ett Python-skript kunde man enkelt läsa in etableringen av fungerande domäner och sedan tillämpa algoritmerna som definierade ZSK/KSK- och CSK-nycklar. Genom ett Python-skript kontrollerades om det fanns två nycklar med flaggorna 257 och 256, som signerade DNSKEY RRset respektive SOA, eller om det endast fanns en nyckel i zonen som signerade båda.

3.3.2 Resultat för ZSK/KSK och CSK

I frågeställning tre framkommer fördelningen mellan ZSK/KSK- och CSK-nycklar i de undersökta domänerna genom en kvantitativ analys. Tabell 4 presenterar denna fördelning där 83,36 procent av domänerna använder ZSK/KSK-nycklar, 16,21 procent använder CSK-nycklar och 0,43 procent av domänerna använder andra modeller.

Domän som uppfyllde följande kriterier räknas använda ZSK/KSK:

1. DNSKEY RRset ska innehålla minst två DNSKEY-poster varav minst en med flaggfält med värde 257 och minst en med flaggfält med värde 256.
2. DNSKEY RRset signerad av DNSKEY med flaggfält med värdet 257.
3. SOA RRset signerad av DNSKEY med flaggfält med värdet 256.

Domän som uppfyllde följande kriterier räknas använda CSK:

1. DNSKEY RRset ska innehålla minst en DNSKEY-poster varav minst en med flaggfält med värde 257.
2. DNSKEY RRset signerad av DNSKEY med flaggfält med värdet 257.
3. SOA RRset signerad av DNSKEY med flaggfält med värdet 257.

De som ej uppfyllde definitionerna enligt ovan räknades som obestämda.

Tabell 3. Fördelning mellan ZSK/KSK- och CSK-nycklar i se-zonen.

Procentandel ZSK/KSK	83,36%
Procentandel CSK	16,21%
Procentandel för obestämda modeller	0,43%

Hur ZSK/KSK-modellen fungerar är beskrivet i delkapitel 2.5.9 och 2.5.10. ZSK/KSK-modellen är enklare att tillämpa än CSK-modellen eftersom det är enklare att byta ut ZSK-nyckeln och bibehålla KSK-nyckeln, än att göra ändringar på CSK-nyckeln som fungerar som ZSK och KSK.

Andelen på 0,43 procent som inte kunde fastställas i studien kan bero på olika orsaker. Det kan vara att en del domäner använder alternativa säkerhetsmodeller eller inte har korrekt implementerat DNSSEC. Det kan även vara möjligt att vissa av dessa domäner var under konfiguration vid tidpunkten för studien och därmed kunde inte kategoriseras. Ytterligare forskning krävs för att få en djupare förståelse av dessa fall. Det är dock viktigt att notera att denna lilla andel inte förändrar bilden av ZSK/KSK som den dominerande modellen.

3.4.1 Metod för CDS-och CDNSKEY-poster

I frågeställning fyra undersöktes det i vilken utsträckning det finns CDS-poster respektive CDNSKEY-poster i de undersökta domänerna under .*se*-toppdomänen (se delkapitel 2.5.5 och 2.5.6 för en utförlig beskrivning av CDS- och CDNSKEY-poster). För att implementera denna analys utvecklades ett Python-skript som följer en flerstegsprocess.

Python-skriptet läste in textfilen med de fungerande undersökta domänerna som hade fastslagits i steg ett. Därefter använde skriptet DNS-verktyget ”dig” för att göra en uppslagning för varje domän i textfilen, med fokus på att söka efter CDS- och CDNSKEY-poster. Efter varje uppslagning så extraherade skriptet relevant information om CDS- och CDNSKEY-poster och lagrade den insamlade informationen i en CSV-fil. För att analysera den insamlade informationen utvecklades en ytterligare Python-kod, skriven i den integrerade utvecklingsmiljön (IDE) PyCharm. Denna kod använde biblioteket pandas för att läsa in CSV-filen och utföra statistiska analyser samt dataframställning.

Sammantaget användes en kombination av Python-skript, DNS-uppslagningar med ”dig” och pandas-baserad dataanalys för att besvara frågeställning fyra, vilket möjliggjorde undersökningen av förekomsten av CDS- och CDSNKEY-poster i de undersökta .*se*-domänerna.

3.4.2 Resultat för CDS-och CDNSKEY-poster

Resultatet presenterar först de domäner som saknar eller har en av posttyperna CDS- eller CDNSKEY. En överväldigande majoritet visar enligt tabell 4 att det saknas CDS- och CDNSKEY-poster för de undersökta domänerna.

Tabell 4. Fördelning av domäner som saknar eller har en av posttyperna CDS- eller CDNSKEY.

Procent av domäner som saknar både CDS och CDNSKEY	98,60%
Andelen domäner som har antingen CDS, CDNSKEY eller både CDS och CDNSKEY.	1,40%

I tabell 5 presenteras fördelningen av CDS- och CDNSKEY-poster bland de undersökta domänerna. Tabellen illustrerar att 68,68 procent endast har CDS-poster, medan endast 1,02 procent av domänerna har enbart CDNSKEY-poster. För resterande 30,31 procent av domänerna finns det både CDS- och CDNSKEY-poster.

Tabell 5. Fördelning av CDS och CDNSKEY inom domäner som har CDS eller CDNSKEY.

Procent av domäner som endast har CDS	68,68%
Procent av domäner som endast har CDNSKEY	1,02%
Procent av domäner som har både CDS och CDNSKEY	30,31%

3.5.1 Metod för signaturlivslängd för RRSIG

För att besvara frågeställning fem som syftade till att undersöka signaturlivslängden för RRSIG-poster, i synnerhet för SOA, NS och DNSKEY RRset, utvecklades separata Python-skript för var och en av dessa posttyper. Alla skript använde likt tidigare steg DNS-verktyget ”dig” för att göra uppslagningar och hämta ut information om de respektive posttyperna, med syftet att extrahera giltighetstiden för RRSIG-posterna. SOA, NS och DNSKEY RRset har valts eftersom de alltid finns med i alla DNSSEC-signerade zoner.

De tre skripten som skrevs för SOA, NS och DNSKEY RRset inleddes med att läsa in textfilen med fungerande domäner som etablerades i frågeställning ett. Sedan tillämpades ”dig” för att göra uppslagningar på respektive posttyp. Genom uppslagningarna kunde man extrahera lämplig data om RRSIG-posterna och då i synnerhet intervallet för giltighetstiden. Efteråt placerades denna data i en CSV-fil, som därefter analyserades genom en annan Python-kod skriven på IDE PyCharm.

3.5.2 Resultat för signaturlivslängd för RRSIG

En analys av giltighetstiden för SOA- och NS-poster, avslöjar att den vanligaste giltighetstiden är 21 dagar, vilket omfattar 84,798 procent av fallen, se tabell 6. Även om det finns flera andra giltighetstider, är de betydligt mindre frekventa än 21 dagar.

Tabell 6. Giltighetstid för SOA- och NS-poster.

Intervall (Dagar)	Procent
0-13	1,030
14-20	6,106
21	84,798
22-35	7,308
36-60	0,720
61-1825	0,035

Anledningen till att SOA och NS delar samma tabell beror på den marginella skillnaden i giltighetstid mellan dem. Däremot i DNSKEY-poster finns en tydligare skillnad, exempelvis så har den dominerande dagen 21, en procentandel på ca 79 procent, se tabell 7.

Tabell 7. Giltighetstid för DNSKEY-poster.

Intervall (Dagar)	Procent
0-13	0,770
14-20	5,719
21	78,853
22-35	14,037
36-60	0,510
61-1825	0,269

Sammanfattningsvis visar resultaten att den mest förekommande giltighetstiden för SOA-, NS- och DNSKEY-poster är 21 dagar.

3.6.1 Metod för giltigheten för DNSSEC

Slutligen undersöktes den sista frågeställningen som berörde huruvida DNSSEC var giltigt för de undersökta domänerna under *.se*-toppdomänen. För att avgöra giltigheten av DNSSEC validerades posttyperna SOA, NS och DNSKEY, men DNSKEY var redan validerad genom frågeställning ett, eftersom de zoner som inte validerade DNSKEY blev uteslutna.

För att validera posttyperna SOA och NS utvecklades ett Python-skript som kontrollerade ifall följande villkor uppfylldes: att DNS-uppslagningens status var ”NoError”, att ”ad”-flaggan var satt och att ”answer” var större än noll. ”NoError” innebär att DNS-uppslagningen inte gav felmeddelanden, vilket är en grundförutsättning för DNSSEC. ”Ad”-flaggan (Authentic Data) indikerar att DNS-data har validerats. ”Answer” indikerar antalet svarsposter i DNS-svaret, där ett värde större än noll betyder att det finns minst ett svar på uppslagningen.

I två separata Python-skript, ett för SOA och ett för NS, analyserades de undersökta domänerna som etablerades i frågeställning ett. Varje domän kontrollerades mot villkoren nämnda ovan i dessa skript. Om en domän inte uppfyllde samtliga villkor ansågs den inte ha giltig DNSSEC för SOA- eller NS-posterna. Resultatet från denna valideringsprocess kunde därefter analyseras och presenteras genom ett annat Python-skript som använde pandas, med hjälp av PyCharm.

3.6.2 Resultat för giltigheten för DNSSEC

Resultatet visar att DNSSEC är giltigt för både SOA och NS. Både tabell 8 och tabell 9 presenterar liknande resultat. Antalet korrekt signerade domäner är 99,40 procent för både SOA- och NS-poster. Antalet icke korrekt signerade domäner är 0,60 procent för både SOA- och NS-poster.

För SOA-poster är antalet korrekt signerade domäner 99,40 procent, vilket motsvarar 93346 domäner. Samtidigt är antalet icke korrekt signerade domäner 0,60 procent, vilket motsvarar 562 domäner, se tabell 8.

Tabell 8. DNSSEC validering för SOA.

Antal korrekt signerade domäner	93346 (99,40%)
Antal icke korrekt signerade domäner	562 (0,60%)

För NS-poster uppvisar resultaten samma fördelning. Antalet korrekt signerade domäner är 99,40 procent, vilket motsvarar 93 346 domäner, och antalet icke korrekt signerade domäner är 0,60 procent som motsvarar 562 domäner, se tabell 9.

Tabell 9. DNSSEC validering för NS.

Antal korrekt signerade domäner	93346 (99,40%)
Antal icke korrekt signerade domäner	562 (0,60%)

Resultatet visar tydligt att majoriteten av de undersökta domänerna har implementerat DNSSEC på ett korrekt sätt för både SOA- och NS-poster. Däremot är det viktigt att notera att den lilla andelen på 0,60 procent av domänerna inte har korrekt signerade poster, vilket kan bero på att domänerna blev avregistrerade fram tills att uppslagningen inträffade från att zonfilen med domänerna laddades ned.

4 Analys och diskussion

4.1 Resultatanalys

I den första frågeställningen, som handlar om vilka algoritmer som används för DNSKEY i varje undersökt domän, visar resultaten att ungefär 99,9 procent av domänerna använder rekommenderade algoritmer. Detta tyder på att majoriteten av domänerna följer godkända och rekommenderade algoritmer, vilket bidrar till en robust och säker DNS-infrastruktur. En väldigt liten andel av domänerna använder icke rekommenderade algoritmer, men dessa är så få att i det stora hela så gör dessa ingen skillnad. Förbättringar kan alltid göras, men i detta fall visar Internetstiftelsen att de har arbetat strävsamt med att använda rekommenderade algoritmer.

Andra frågeställningen undersökte vilka nyckellängder som används för RSA-algoritmer, och resultaten visar att säkerhetsbehovet kräver ständigt ökade nyckellängder. Tidigare ansågs [25] nyckellängder under 512 bitar vara tillräckligt säkra, men dagens säkerhetskrav kräver längre nyckellängder [31]. DNS-standarden kräver för närvarande nyckellängder mellan 512 och 4096 bitar, men en minsta nyckellängd på 2048 bitar rekommenderas för att säkerställa tillräcklig skyddsnivå. Även om det för närvarande inte är ett krav enligt standarden, varnar experter för att RSA-nycklar med mindre än 2048 bitar kan vara osäkra, och detta kan komma att inkluderas som standarden längre fram i tiden. Det är viktigt att observera att användning av mycket långa nyckellängder kan leda till prestandaproblem och göra systemet långsamt.

Paul Hoffmans undersökning [21], som omfattade toppdomänen *.se* tillsammans med andra toppdomäner, visade att de mest förekommande nyckelstorlekarna för RSA-algoritmer var 1024 bitar (45,1%), 2048 bitar (51,4%) och 4096 bitar (0,7%). Hoffmans resultat visar en jämnare procentuell fördelning, medan vår studie tyder på att RSA-algoritmerna använder den rekommenderade nyckellängden på 2048 bitar i ungefär 98 procent av fallen, vilket visar att underdomänerna till *.se*-zonen har en högre DNSSEC-kvalitet än andra toppdomäner. Teoretiskt kan nivåerna ha förbättrats efter Hoffmans undersökning, men sannolikheten är förmodligen liten.

För närvarande anses en nyckelstorlek på 2048 bitar vara säker [31]. Men med den snabba teknikutvecklingen kan det hända att en 2048-bitars nyckellängd inte längre anses som säker om några år. Det är viktigt att notera att en nyckel på 2048 bitar kan ha upp till 600 tecken [23], vilket betyder att längre nyckellängder kommer att ha ännu fler tecken och ta längre tid att dekryptera. En lösning till detta problem kan vara att överväga att standardisera användningen av ECDSA-algoritmen (Elliptic Curve Digital Signature Algorithm). ECDSA erbjuder en liknande säkerhetsnivå som RSA, men med kortare nyckellängder. Detta innebär att ECDSA kan erbjuda samma skyddsnivå mot säkerhetshot samtidigt som det kräver färre bitar i sin nyckellängd. Denna fördelaktiga egenskap innebär kortare dekrypteringstider och minskade prestandaproblem jämfört med RSA-algoritmen, utan att kompromissa med säkerheten [30].

För frågeställning tre analyserades andelen ZSK/KSK- och CSK-nycklar i de undersökta domänerna. Däremot en andel om 0,43 procent som inte kunde fastställas i resultatet kan ha flera tänkbara orsaker. Eventuellt kan mätmetoden som implementerats under arbetet ha missat något. Det kan vara så att vissa domäner använder alternativa säkerhetsmodeller eller att de inte har implementerat DNSSEC på ett korrekt sätt. Det är också möjligt att några av dessa domäner var under konfiguration vid tidpunkten för studien, vilket gjorde det svårt att kategorisera dem. För att få en djupare förståelse av dessa fall och eventuellt kunna dra mer precisa slutsatser, skulle ytterligare forskning vara nödvändig. Detta skulle kunna inkludera att undersöka de specifika orsakerna bakom dessa fall och analysera hur de kan påverka den övergripande bilden av DNSSEC-implementering inom de undersökta domänerna.

I frågeställning fyra undersöktes fördelningen av domäner som har både CDS- och CDNSKEY-poster. Initialt analyserades fördelningen av hur många domäner som saknar båda posttyperna CDS och CDNSKEY och hur många domäner som hade nån av posttyperna. Som presenterats i avsnitt 3.4.2 var avsaknaden av CDS och CDNSKEY totalt 98,60 procent och andelen domäner som hade CDS- eller CDNSKEY var 1,40 procent. En orsak till att en enorm majoritet av domänerna saknar CDS- och CDNSKEY är eftersom dessa används främst vid byte av KSK eller CSK inom DNSSEC-systemet. Detta byte sker normalt sett inte särskilt ofta. Rekommendationen idag är att dessa byten inte ska göras regelbundet utan endast när det är nödvändigt, till exempel om en nyckel har blivit äventyrad eller om det finns ett behov av en bättre nyckel. Därefter analyserades fördelningen inom materialet (domäner) som hade CDS eller CDNSKEY. Resultatet visar att 68,68 procent av domänerna endast har CDS-poster, 1,02 procent endast har CDNSKEY-poster och 30,31 procent har både CDS- och CDNSKEY-poster. Som standard [12] genererar namnserverprogrammet Bind både CDS- och CDNSKEY-poster, men anledningen till att CDS-poster används i större utsträckning än CDNSKEY-poster är för att toppdomänen *.se* har valt en policy som endast tar hänsyn till CDS-poster, vilket innebär att de ignorerar CDNSKEY-poster.

För frågeställning fem visar resultaten att den mest förekommande giltighetstiden för SOA-, NS- och DNSKEY-poster är 21 dagar. Detta tyder på att det möjligen finns en viss standardisering när det gäller giltighetstid för dessa poster, vilket kan bidra till en mer stabil och säker DNS-miljö. Däremot i ett dokument från Internetstiftelsen [32] rekommenderas giltighetstid på 32 dagar, vilket uppenbarligen inte följs av dessa domäner. Jag har inte hittat andra rekommendationer och har inte undersökt default-värden i namnserverprogram som Bind.

Standardiseringen av giltighetstider kan underlätta för administratörer att upprätthålla och övervaka DNS-system, samtidigt som det minskar risken för oavsiktliga avbrott i tjänsten på grund av utgångna poster. Emellertid kan variationen i giltighetstider för vissa poster vara ett tecken på att det fortfarande finns utrymme för förbättringar när det gäller att optimera DNS-konfigurationer. Det hittades extremvärden både uppåt och nedåt, alltså med för långa- och korta tider. Dessa variationer kan bero på olika faktorer, såsom felkonfigurationer, organisationens policy eller specifika säkerhetsbehov. För att ytterligare förbättra säkerheten och stabiliteten i

DNS-miljön kan det vara värdefullt att undersöka orsakerna bakom dessa variationer och identifiera eventuella bästa praxis för att fastställa optimala giltighetstider för DNS-poster.

I diskussionen om frågeställning sex visar resultaten att 99,40% av de undersökta domänerna har korrekt signerade SOA- och NS-poster med DNSSEC, vilket påvisar en god DNS-infrastruktur. Däremot kan man fundera på varför resultatet inte visade 100 procent? En hypotes är att en del domäner hann avregistreras från att zonfilen laddades ned tills uppslagningen ägde rum, så kallad tidsfönster. Min åsikt är att alla domäner som togs med är korrekta, men på grund av tidsfönstret fås en 0,60 procent.

4.2 Hållbar utveckling

Utifrån resultaten och diskussionen av de olika frågeställningarna kan man reflektera över hållbar utveckling inom DNS-säkerhet och hur det påverkar det digitala ekosystemet. Hållbar utveckling inom detta område handlar inte bara om tekniska aspekter, utan även om samhällets och miljöns välbefinnande. Eftersom digitaliseringen och användningen av internet ständigt ökar, blir det allt viktigare att säkerställa att DNS-infrastrukturen är både säker och stabil för att upprätthålla tillförlitliga och tillgängliga digitala tjänster. Att använda DNSSEC och överväga alternativa algoritmer som ECDSA bidrar till att stärka säkerheten utan att kompromissa med prestanda, vilket kan främja en mer effektiv och resurseffektiv infrastruktur. Dessutom kan ökad medvetenhet om fördelarna med DNSSEC och implementering av CDS- och CDNSKEY-poster bidra till en mer inkluderande och säker digital miljö för alla användare. På detta sätt bidrar studien till en förståelse av hur tekniska lösningar och standarder kan främja en hållbar utveckling inom det digitala landskapet och därmed säkerställa en långsiktigt trygg och tillgänglig internetmiljö för framtida generationer.

5 Slutsatser

Denna studie har undersökt underdomäner till toppdomänen *.se,* i syfte att kartlägga underdomänernas DNSSEC-kvalitet. Studiens slutsats är som följer:

- De dominerande algoritmerna för DNSKEY i undersökta *.se*-domäner är RSA och ECDSA, vilka är i enlighet med DNS-standarden.

- För RSA-algoritmen dominerar nyckellängden på 2048 bitar med 98 procent, vilket är i enlighet med DNS-standarden.

- Bland de undersökta *.se*-domänerna fördelades nycklarna enligt följande: 83,36 procent ZSK/KSK, 16,21 procent CSK och 0,43 procent andra modeller.

- Fördelningen av domäner utan CDS- och CDNSKEY-poster eller med en av posttyperna är enligt följande:

  - 98,60 procent av domänerna saknar både CDS och CDNSKEY.

  - 1,40 procent av domänerna har antingen CDS eller CDNSKEY, eller både CDS och CDNSKEY.

  Avseende fördelningen mellan dem domäner som har CDS- och CDNSKEY-poster i de undersökta *.se*-domänerna fördelades resultaten enligt följande:

  - 68,68 procent av domänerna hade endast CDS

  - 1,02 procent av domänerna hade endast CDNSKEY

  - 30,31 procent av domänerna hade både CDS och CDNSKEY

- I de undersökta *.se*-domänerna var den dominerande signaturlivslängden för RRSIG, gällande SOA, NS och DNSKEY RRset 21 dagar.

- DNSSEC visade sig vara giltigt i de undersökta *.se*-domänerna, med 99,40 procent korrekt konfigurerade domäner för både SOA och NS. Det bekräftades att alla undersökta domäner hade en giltig DNSKEY.

Sammanfattningsvis kan det sägas att majoriteten av de undersökta domänerna har implementerat DNSSEC i enlighet med DNS-standarder och rekommendationer, medan det finns ett litet antal som fortfarande visar brister, i både konfiguration och som går emot DNS-standarder. Denna kartläggning visar alltså att majoriteten av *.se*-domänerna har en god DNSSEC-kvalitet, men det finns fortfarande utrymme för förbättringar för att säkerställa en högre säkerhetsnivå för samtliga domäner.

Rekommendationer

Ett framtida forskningsområde kan vara att analysera och utvärdera användningen av ECDSA-algoritmen som en potentiell standard. Med tanke på att nyckellängder blir allt längre och RSA-dekryptering kräver mer prestanda och tid, kan det vara värt att överväga ECDSA som en alternativ lösning. ECDSA kan eventuellt erbjuda snabbbare och mer energieffektiva dekrypteringsprocesser, vilket kan förbättra energin och stabiliteten ur ett hållbart perspektiv för företag som använder dessa domäner.

Källförteckning

[1] Liu, C., & Albitz, P. DNS & Bind. Sebastopol, CA: O'Reilly Media, Inc.;2006

[2] Root Zone Database [Internet]. Iana.org. 2023 [citerad 2023 May 16]. Från: https://www.iana.org/domains/root/db

[3] Vad är DNS? | Internetstiftelsen [Internet]. Internetstiftelsen. 2021 [citerad 2023 Mar 23]. Från: https://internetstiftelsen.se/guide/dns-internets-vagvisare/sa-fungerar-dns/

[4] Dufberg, Mats. Med tillstånd från Internetstiftelsen. Undervisningsmaterial för kursen HI1037 "Internets domännamnssystem" på Kungliga Tekniska Högskolan, vårterminen 2023. [citerad 2023 Mar 24].

[5] greg-lindsay. Översikt över Azure DNS-delegering [Internet]. Microsoft.com. 2022 [citerad 2023 Mar 24]. Från: https://learn.microsoft.com/sv-se/azure/dns/dns-domain-delegation

[6] Telling H, Gunnarsson A. DNSSEC en säkerhetsförbättring av DNS : en studie om Svenska kommuners syn på DNSSEC [Internet]. DIVA. 2023 [citerad 2023 Mar 30]. Från http://lnu.diva-portal.org/smash/record.jsf?pid=diva2%3A424641&dswid=-8369

[7] Ariyapperuma S, Mitchell CJ. Security vulnerabilities in DNS and DNSSEC. The Second International Conference on Availability, Reliability and Security (ARES'07) [Internet]. 2007 [citerad 2023 Mar 24]; Från: https://ieeexplore.ieee.org/abstract/document/4159821

[8] DNSSEC Guide : Trust Anchors | The DNS Institute [Internet]. Dnsinstitute.com. 2023 [citerad 2023 Mar 25]. Från: https://dnsinstitute.com/documentation/dnssec-guide/ch03s04.html

[9] DNSSEC Practice Statement [Internet]. Iana.org. 2020 [citerad 2023 May 16]. Från: https://www.iana.org/dnssec/procedures

[10] Domain Count Statistics for TLDs - DomainTools [Internet]. Domaintools.com. 2023 [citerad 2023 feb 2]. Från: https://research.domaintools.com/statistics/tld-counts/

[11] Perfect. DNSSEC - Webb.se [Internet]. Webb.se. 2023 [citerad 2023 Apr 24]. Från: https://www.webb.se/dnssec/

[12] Mats Dufberg. Internetstiftelsen. Muntlig kommunikation, [2023-05-16].

[13] Taylor R. What is DNSSEC and how does it work? – BlueCat Networks [Internet]. BlueCat Networks. 2019 [citerad 2023 Apr 22]. Från: https://bluecatnetworks.com/blog/breaking-down-dnssec-how-does-it-work/

[14] DNS DNSKEY and DS records [Internet]. Cloudflare. 2023 [citerad 2023 Apr 23]. Från: https://www.cloudflare.com/learning/dns/dns-records/dnskey-ds-records/

[15] Automatiserad DNSSEC | Internetstiftelsen [Internet]. Internetstiftelsen. 2021 [citerad 2023 Apr 1]. Från: https://internetstiftelsen.se/domaner/domannamnsbranschen/teknik/automatiserad-dnssec/

[16] Mens JP. DNSSEC provisioning automation with CDS/CDNSKEY in the real world | APNIC Blog [Internet]. APNIC Blog. 2021 [citerad 2023 Apr 1]. Från: https://blog.apnic.net/2021/11/02/dnssec-provisioning-automation-with-cds-cdnskey-in-the-real-world/

[17] Gudmundsson O, Wouters P. Managing DS Records from the Parent via CDS/CDNSKEY. 2017 Mar [citerad 2023 Apr 11]; Från: https://www.rfc-editor.org/rfc/rfc8078

[18] How DNSSEC Works [Internet]. Cloudflare. 2014 [citerad 2023 Apr 6]. Från: https://www.cloudflare.com/dns/dnssec/how-dnssec-works/

[19] Rfc-editor.org. 2023 [citerad 2023 Apr 12]. Från: https://www.rfc-editor.org/rfc/rfc4034.txt

[20] Chung T, van Rijswijk-Deij R, Choffnes D, Levin D, Maggs BM, Mislove A, et al. Understanding the role of registrars in DNSSEC deployment. Proceedings of the 2017 Internet Measurement Conference. 2017 Nov;

[21] Hoffman P. Algorithm Use in 2022 | OCTO-033 [Internet]. 2022 [citerad 2023 May 3]. Från: https://www.icann.org/en/system/files/files/octo-033-04apr22-en.pdf

[22] Le T, van Rijswijk-Deij R, Allodi L, Zannone N. Economic incentives on DNSSEC deployment: Time to move from quantity to quality. NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium [Internet]. 2018 Apr [citerad 2023 Apr 15]; Från: https://ieeexplore.ieee.org/abstract/document/8406223

[23] Ehsas N. Introduktion till krypteringsmetoderna RSA och Merkle-Hellman [Internet]. DIVA. 2023 [citerad 2023 May 1]. Från: https://www.diva-portal.org/smash/record.jsf?dswid=3311&pid=diva2%3A419432&c=1&searchType=SIMPLE&language=en&query=Introduktion+till+krypteringsmetoderna++RSA+och+Merkle-Hellman+&af=%5B%5D&aq=%5B%5B%5D%5D&aq2=%5B%5B%5D%5D&aqe=%5B%5D&noOfRows=50&sortOrder=author_sort_asc&sortOrder2=title_sort_asc&onlyFullText=false&sf=all

[24] Dhanashree Toradmalle, Rohan Bir Singh, Shastri H, Naik N, Vishal Panchidi. Prominence Of ECDSA Over RSA Digital Signature Algorithm. 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on [Internet]. 2018 Aug 1 [citerad 2023 Apr 26]; Från: https://ieeexplore.ieee.org/abstract/document/8653689

[25] Jansen J. RFC ft-ietf-dnsext-dnssec-rsasha256: Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC [Internet]. IETF Datatracker. 2009 [citerad 2023 May 5]. Från: https://datatracker.ietf.org/doc/html/rfc5702#section-2

[26] Wouters P, Sury O. Algorithm Implementation Requirements and Usage Guidance for DNSSEC. RFC [Internet]. 2019 Jun 1 [citerad 2023 May 1]; Från: https://www.rfc-editor.org/rfc/rfc8624.html#section-3.1

[27] [dns-operations] .nz DNSKEY encoding [Internet]. Dns-oarc.net. 2023 [citerad 2023 Apr 23]. Från: https://lists.dns-oarc.net/pipermail/dns-operations/2012-January/007970.html

[28] Allman E, Callas J, Delany M, Libbey M, Fenton J, Thomas M. DomainKeys Identified Mail (DKIM) Signatures. 2007 May 1 [citerad 2023 Apr 23]; Från: https://www.rfc-editor.org/rfc/rfc4871#section-3.3.1

[29] Comparing TLD DNSSEC Practices with RFCs [Internet]. 2012. Från: http://iepg.org/iepg/2012-03-ietf83/IETF83IEPGLewis.pdf

[30] Dhanashree Toradmalle, Rohan Bir Singh, Shastri H, Naik N, Vishal Panchidi. Prominence Of ECDSA Over RSA Digital Signature Algorithm. 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on [Internet]. 2018 Aug 1 [citerad 2023 Apr 26]; Från: https://ieeexplore.ieee.org/abstract/document/8653689

[31] Barker E. Recommendation for Key Management Part 1: General. 2016 Jan; Från: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

[32] DNSSEC Rekommendationer, Version 2013R2 [Internet]. Stockholm: Kirei AB; 2013 [citerad 17 maj 2023]. Från: https://www.kirei.se/xfiles/dnssec-rek-2013r2-sv.pdf