Universitetet i Oslo
IN2120 Informasjonssikkerhet
Høst 2023
Teoretiske oppgaver med løsningsforslag

Teori 4: (Del 4): Nøkkelhåndtering og PKI

Oppgave 1: Sertifikater

a. Hvorfor er det viktig å ha relativt begtrensede kryptoperioder for nøkler, og også hvorfor kryptoperioden for noen nøkler heller ikke bør være for kort ? Nevn fire grunner for å ha begrenset kryptoperiode, og en grunn for å ha lang kryptoperiode.
b. Hva menes med periodene for beskyttelse og prosessering ved bruk av kryptonøkler? Gå til nsm.no med din favorittnettleser. I de fleste nettlesere kan du klikke på hengelåsen på adresselinjen for å se serversertifikatet (X.509-sertifikat for offentlig nøkkel). (I Edge må du trykke på knappen med tre horisontale prikker øverst til høyre, deretter trykke på knappen (Settings/Innstillinger), deretter trykke på knappen (privacy, search and services) i venstre marg, og til slutt skrolle ned til «Security» og «Manage certificates». Finn sertifiseringskjeden og klikk på rotsertifikatet. Sjekk detaljer for rotsertifikatet for å se gyldighetsperioden, som tilsvarer kryptoperioden for den offentlige nøkkelen.
c. Hva er gyldighetsperioden (fra/til) for rotsertifikatet?
d. Den private signaturnøkkelen for rotsertifikatet ble brukt til å signere det mellomliggende CA-sertifikatet. Sjekk datoen for utstedelsen av det mellomliggende CA-sertifikatet. Vil du si at gyldighetsperioden for den private signaturnøkkelen i rotsertifikatet følger anbefalingene for beskyttelsesperioden (OUP) for private signaturnøkler i henhold til NIST SP800-57?
e. Hvis man antar at kraftige kvantekomputere vil være praktisk tilgjengelig rundt 2030, er gyldighetsperioden av rotsertifikatet fornuftig? Hva måtte gjøres med rotsertifikatet hvis kraftige kvantecomputere ble tilgjengelig rundt 2030?
f. Bruk tjenesten på https://www.ssllabs.com/ssltest/ til å teste kvalitet på sertifikat og TLS-tjener på nettstedene **nsm.no** og **universitetsforlaget.no**. Sammenlign resultatene.

Svarforslag

a. Det er flere gode grunner til å begrense kryptoperioden:
   - Begrenser mengden data tilgjengelig for kryptanalyse av en enkelt nøkkel.
   - Begrenser skaden hvis en enkelt nøkkel er knekt/kompromittert.
   - Begrenser bruken av en bestemt algoritme til dens estimerte effektive levetid.
   - Begrenser tiden som er tilgjengelig for beregningsintensive kryptanalytiske angrep, f.eks. uttømmende søk gjennom (deler av) nøkkelrommet.
   - Begrenser tiden en angriper kan foreta fysiske, prosedyremessige og logiske angrep mot sikkerhetsmekanismer som skal beskytte nøkkelen mot uautorisert tilgang.
   - Begrenser perioden der informasjon kan bli kompromittert ved utilsiktet utlevering av nøkkelmateriale til uautoriserte aktører.

En grunn til å ha lange kryptoperioder er for å unngå kompleksiteten og tidsbruk som bytte av nøkler medfører.

b. Perioden der en nøkkel brukes for beskyttelse er når nøkkelen brukes for kryptering og for digital digital signatur. NIST kaller denne perioden «Originator Usage Period» (OUP). Perioden der en nøkkel brukes for prosessering er når nøkkelen brukes for dekryptering og for å validere digitale signaturer. NIST kaller denne perioden «Recipient Usage Period» (RUP).
c. GTS Root R1 (fra Google Trust Services LLC) rotsertifikat er gyldig fra 22.06.2016 til 22.06.2036.
d. GTS CA 1P5 (fra Google Trust Services LLC) sertifikatet er gyldig fra 13.08.2020 til 30.09.2027, dvs.at det ble utstedt 4 år etter at rotsertifikatet ble utstedt. Anbefalingen fra SP800-57 for beskyttelsesperioden for private signaturnøkler (tabell 1, s.45) er 3 år. Derfor har den private signaturnøkkelen for rotsertifikatet blitt brukt lenger enn NIST-anbefalingen. Det er vanlig praksis i PKI-bransjen at den private nøkkelen for rotsertifikater brukes over lang tid, fordi det forenkler distribusjon av rotsertifikater.
e. Gyldighetsperioden for GTS Root R1 (fra Google Trust Services LLC) rotsertifikat til år 2036 er for lang hvis man antar kraftig QC (kvantecomputere) vil være tilgjengelig rundt 2030. Det bør være en bufferperiode på mange år mellom tiden sertifiatet går ut på dato til kraftige kvantecomputere blir tilgjengelig. Imidlertid er det svært usannsynlig at kraftige kvantecomputere vil være tilgjengelig i 2030.
f. Resultatet vil endres ettersom webtjenerens konfigurering endres. I juni 2023 var resultatet for beegge nettstedene «A» eller «A+», som er en svært god skåring. Begge nettsteder støtter TLS 1.3 som gir fremoverhemmelighold.

Oppgave 2: Tillitsmodell

a. Beskriv tillitsmodellen for PKI'en som brukes av nettlesere.
b. Nevn fordeler og ulemper ved denne modellen.
c. Man sier at nøkkelsertifikater og PKI skaper tillit. Hva menes med dette?

Svarforslag

a. Tillitsmodellen i nettleseres PKI er basert på et relativt stort antall (≈ 100) forhåndsinstallerte rotsertifikater som brukes som tillitsanker for validering av mottatte sertifikater. Hvert rotsertifikat definerer en isolert hierarkisk PKI. Brukeren har implisitt tillit til leverandøren bak nettleseren som leverte de installerte rotsertifikatene, og dermed indirekte de rundt 100 rot-CA'ene og deres rotsertifikater. Nettleseren validerer automatisk mottatte serversertifikater som kryptografisk kan lenkes tilbake til et forhåndsinstallert rotsertifikat.
b. Relevante elementer er:
   - Fordeler: brukervennlighet, automatisert validering av tjenersertifikater og programvaresertifikater
   - Ulemper:
     - I nettleseren har rootsertifikater en relativt svak beskyttet mot angrep
     - Når det presenteres en pop-up om at et sertifikat ikke kan valideres, har brukere en tendens til å alltid godkjenne sertifikatet uansett, så formålet med å spørre brukeren om godkjenningen er tvilsom.
     - Enhver kompromittert CA kan true hele infrastrukturen (svakeste ledd).

c. PKI og sertifikater skaper tillit til kryptert og autentisk kommunikasjon, men ikke til pålitelighet eller rettskaffenhet. Med andre ord, hvis en kriminell organisasjon har en nettside med et nøkkelsertifikat kan vi som brukere ha tillit til at kommunikasjonen med nettsiden er kryptert og at domenenavnet for nettsiden er autentisk. Likefult kan nettsiden inneholde skadevare eller forsøke å villede brukere på en kriminell måte.

Oppgave 3: Sikkerhet for kryptonøkler

a. Nevn faktorer som bestemmer styrken på kryptografiske sikkerhetsløsninger.
b. Hvorfor er nøkkelhåndtering viktig for kryptografiske sikkerhetsløsninger?
c. Tre viktige nøkkelkategorier er: i) symmetriske hemmelige nøkler, II) asymmetriske offentlige nøkler og III) asymmetriske privatnøkler. Forklar hvilken type sikkerhet tjenester/beskyttelse (dvs. konfidensialitet, integritet og autentisitet) som kreves for hver nøkkelkategori.
d. Beskriv sikkerhetstiltak som kan brukes til å beskytte kryptografiske nøkler.

Løsningsforslag

a. Faktorer er f.eks.: 1) Nøkkelstørrelse, 2) algoritmens styrke mot kryptanalyse, 3) korrekthet av implementeringen i HW/SW, 4) Fysisk skjerming av utstyr, 5) nøkkelhåndteringen.
b. Det svakeste ledd bestemmer sikkerheten i en kryptoløsning. Sikkerheten av kryptografiske løsninger er dermed begrenset av kvaliteten i håndtering og beskyttelsen av kryptonøklene, selvsagt i tillegg til kryptografisk styrke og sikker/feilfri implementering av algoritmene, og sikker fysisk design og skjerming av utstyr.
c. Nødvendig beskyttelse er:
   i) symmetriske hemmelige nøkler og II) asymmetriske private nøkler må alltid holdes konfidensielle (dvs. beskyttet mot uautorisert eksponering) og må ha integritet/autentisitet (dvs. beskyttet mot forfalskning/uautorisert modifisering).
   III) asymmetriske offentlige nøkler må ha integritet/autentisitet (dvs. beskyttet mot forfalskning/uautorisert modifisering), men trenger ikke konfidensialitet.
d. Fysisk sikringen og tilgangskontroll kan brukes å beskytte kryptonøkler, og er faktisk den eneste måten å beskytte rotnøkler. Manipuleringsresistente (tamper resistant) enheter kan brukes til å generere, lagre og arkivere nøkler. Kryptografiske teknikker kan brukes for å beskytte alle andre kryptografiske nøkler, f.eks. 1) sertifikater for å beskytte offentlige nøkler mot endringer og 2) kryptering for å gi konfidensialitet beskyttelse av underordnede nøkler eller sesjon/øktsnøkler under distribuering.

Oppgave 4: Nøkkeltyper

Se på de 19 nøkkeltypene beskrevet i Tabell 1, s.45,46,47 i NIST SP800-57
https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final

a. Hva er en nøkkeltransportnøkkel (key-transport key) (for nøkkelinnkapsling) ?
b. Nøkkeltypene 10 og 11 kalles henholdsvis privat nøkkel-transportnøkkel og offentlig nøkkel-transportnøkkel. I hvilken type kryptosystem (illustrert i figur 4.10 i læreboka) brukes disse nøkkeltypene?
c. Hva er fremoverhemmelighold, og kan kryptosystemet i b) gi «fremoverhemmelighold»?

Diagrammet for nøkkeltilstander og overganger mellom tilstander i figur 3, s.80, NIST SP800-57 (også vist i figur 5.2 i læreboka) indikerer at når en nøkkel er i aktiv tilstand, kan den angis til bare beskytte, bare prosessere eller begge deler.

d. Hvilke nøkkeltyper er kun ment for å beskytte?
e. Hvilke nøkkeltyper er kun ment for å prosessere?
f. Hvilke nøkkeltyper er ment for både å beskytte og prosessere?

Svarforslag

a. En nøkkel-transportnøkkel (key-transport key»kel) brukes til å kryptere / dekryptere (innkapsle) andre kryptografiske nøkler eller kryptomateriale slik at det kan sendes sikkert over Internett.
b. Private og offentlige nøkkel-transportnøkler brukes i hybride kryptosystemer beskrevet på s.45 i L02 om kryptografi.
c. Foroversikkerhet betyr at mulig fremtidig kompromittering av en (privat) langtidsnøkkel ikke skal gjøre det mulig for en angriper å avdekke øktnøkler brukt i tidligere økter. Et hybrid kryptosystem der øktnøkler krypteres med en offentlig nøkkel gir ikke foroversikkerhet. Det er fordi fremtidig kompromittering av den private nøkkelen gjør en angriper i stand til å avdekke tidligere øktnøkler hvis angriperen har lagret meldinger fra tidligere økter. Angriperen kan da bruke den private nøkkelen til å dekryptere tidligere øktnøkler.

Kryptonøkler i aktiv tilstand:
d. Bare beskytte: Alle private nøkler for signaturer, og alle offentlige nøkler for kryptering (f.eks. Offentlig nøkkel-transportnøkkel).
e. Bare prosessere: Alle offentlige nøkler for validering av signaturer og alle private nøkler for dekryptering (f.eks. Privat nøkkel-transportnøkkel)
f. Beskytte og prosessere: Alle symmetriske nøkler som brukes til symmetrisk kryptering / dekryptering og autentisering med MAC.

Oppgave 5: CA-autorisering

a. Hvilket problem er løst med CA-autorisering (CAA) og sertifikat-transparens?
b. Hvor lagres CA'ens autoriseringspolicy?
c. Hva er OSCP-sertifikater, og hva menes med «must staple»?

Svarforslag

a. CA-autorisering forhindrer at kompromitterte CA'er utsteder falske sertifikater.
b. CA autoriseringspolicyen er lagret i DNS-recorden for domenet.
c. OCSP (Online Certificate Status Protocol) er en metode for å informere nettlesere om status for sertifikater. Dette gjøres ved at et tjenersertifikat alltid skal sendes sammen med et OCSP-sertifikat som er en signert og relativt fersk statusrapport for et tjenersertifikat. Kravet om at OCSP-sertifikatet skal sendes sammen med subjektsertifikatet kalles «must-staple» (må stiftes på) fordi det ikke skal være nettleserens oppgave å skaffe OCSP-sertifikater for tjenersertifikater som mottas fra webtjenere.

Oppgave 6: Sertifikater og tillit

a. Forklar hvordan offentlig-nøkkel sertifikater kan garantere autentisitet av offentlige nøkler.
b. Hvilke betingelser bør man sette for å ha tillit til et digitalt sertifikat? Begrunn ditt svar.
c. Rotsertifikater er selvsignert. Gir det tillit, eller hvordan skal man få tillit til rotsertifikater?

Svarforslag

a. Et offentlig-nøkkel sertifikat etablerer en logisk binding mellom en identitet og dens offentlige nøkkel gjennom den CA-genererte digitale signaturen. Validering av signaturen er bevis for at den offentlige nøkkelen faktisk eies av entiteten med den spesifikke identiteten.
b. Du kan ha tillit til informasjonen i sertifikatet (bindingen mellom identitet og offentig nøkkel) hvis du stoler på at utsteder-CA (sertifiseringsautoritet) har identifisert sertifikat-eieren på en korrekt måte, hvis du stoler på autentisiteten til CA'ens offentlige nøkkel, og hvis du kan validere sertifikatet kryptografisk med CA'ens offentlige nøkkel.

- Nei, selvsignering gir ingen tillit, det er kun en metode for å gjøre rotsertifikater syntaktisk komplett med en digital signatur. Semantisk gir det ingen mening. Tillit til rotsertifikater avhenger av sikkerheten i distribusjonskanalen. Rotsertifikater for nettlesere lastes ned, og oppdateres, sammen med nettleseren, som ikke gir høy sikkerhet. Et annet eksempel er rotsertifikater for pass, der representanter for alle lands myndigheter må reise personlig til hovedkvarteret for ICAO (International Civil Aviation Authority) i Montreal, Canada for å utveksle rotsertifikater, noe som gir høy sikkerhet.

Oppgave 7: Sertifikater i nettlesere

Tilgang til lagrede rotsertifikater i nettleseren din(e) er via nettlesermenyene. Se gjennom rotsertifikater som er installert i webleseren for å se deres utløpsdatoer.
Metode for å inspisere rotsertifikater

Chrome: Select «Update»  in the top right corner. → Settings → Privacy and Security → Security → Manage Certificates → Trusted Root Certification.

Microsoft Edge: Se fremgangsmåte i oppgave 1.

Firefox: Select «Menu button»  in top right corner. → Settings → Security & Privacy → Scroll down to Certificates → View Certificates → Authorities → View

a. Hvor mange rotsertifikater er installert? Variere for eksempel mellom Firefox og Chrome?
b. Hvilke sertifikater har kort levetid?
c. Kan du finne sertifikater med utløpsdato i overkant av ti år fra nå?
d. Kan du finne sertifikater som allerede har utløpt? Hva skjer når du inspiserer dem?

Svarforslag

Detaljene er ikke viktig og variere mellom ulike installasjoner. Vanligvis er sertifikater utstedt med levetid fra ett til ti år. CA-sertifikater er vanligvis selv-sertifisert og har levetid på 10 til 50 år. Du kan sannsynligvis finne CA-sertifikater som allerede har utløpt. Både IE og Firefox gir noen form for advarsel når du åpner en utløpt sertifikat.

Oppgave 8: Sertifikatrevokering

a. Hvorfor er det nødvendig å tillate revokering / tilbakekallelse av sertifikater?
b. Hvilke problem er løste med «must-staple protokollen»?

Svarforslag

a. Sertifikatrevokering er nødvendig for eksempel hvis eieren av et sertifikat har mistanke om at den private nøkkelen er kompromittert.
b. Den tidligere brukte OCSP-protokollen (online certificate status protocol) fører til brudd på brukerens personopplysningsvern. Den nye «must-staple protokollen» sjekker om sertifikater fremdeles er gyldige (ikke revokert) og sender vedlagt (stapled) et status sertificat som viser om certifikatet fremdeles er gyldig. Dette ghør at brukerens nettleser ikke trenger å bruke OCSP for å undersøke om server-sertifikatet er gyldig, slik at surfing ikke medfører brudd på personopplysningsvern.