IN1020 – Nettverkskommunikasjon og kryptografi

Løsningsforslag gruppetime 8 (uke 42)

Oppgave 1: Lagdeling i datakommunikasjon

a) Applikasjonslaget og Transportlaget.

b) *En adresse som identifiserer ett nettverks interface (NIC) som bruker overføringsteknologier som f.eks. Ethernet eller WiFi. Logisk sett opererer en MAC-adresse på linklaget, selv om den ofte er bygget inn i maskinvaren som gjør overføringen. En MAC-adresse gjør at man kan adressere andre maskiner på samme lokale nettverk (LAN).*

c)
Applikasjonslaget – SSL
Transportlaget – TCPCRYPT (Lite brukt)
Nettverkslaget – VPN (IPSEC)
Linklaget – Spesielt hvis man bruker broadcast-nettverk, som WiFi – WPA2/WPA3

Oppgave 2: Nettverkslaget

a) IP'en til en maskin er gitt i CIDR-notasjon ved: 192.168.0.165/28
   a. Hva er IP-adressen med hver oktett notert binært?
      Svar: 11000000.10101000.00000000.10100101
   b. Hva blir nettverksmasken med hver oktett notert binært?
      Svar: 11111111.11111111.11111111.11110000

b) For å finne subnettadressen til maskinen må du gjøre en bitvis AND operasjon mellom IP-adressen og nettverksmasken.
   a. Hva blir subnettet til maskinen over i CIDR-notasjon?
      Svar: 192.168.0.160/28
   b. Hvor mange adresser er det i vertsdelen av dette subnettet?
      Svar: 14 (husk å trekke fra subnett og kringkasting*)*

c) For å finne kringkastingsadressen til et subnett, må du gjøre en bitvis OR-operasjon mellom maskinens IP-adresse og bit komplement (bitvis invers) av nettverksmasken.
   a. Hva er kringkastingsadressen til subnettet i oppgave 2 a)?
      Svar: 192.168.0.175

Oppgave 3: NAT og IPv6

a) NAT står for «Network Address Translation», og er en metode for å skrive om en IP-adressetil en annen når den passerer en enhet i nettverket. Den benyttes for det meste for å gjøre at man kan gjenbruke IP-adresser i forskjellige lokale nettverk ved å skrive dem om, slik at alle forbindelser med opprinnelse innenfor et NAT-et nettverk ser ut som det kommer fra den samme offentlige IP-adressen sett fra utsiden. Dette gjøres ved at hver forbindelse kobles opp mot et portnummer tilhørende den eksterne IP-adressen. NAT-enheten holder oversikt over hvilke enheter/forbindelse på innsiden av LANet som korresponderer med de forskjellige portnumrene som benyttes utad.

b) Man er i ferd med å slippe opp for ledige IP-adresser i Internett med IPv4, selv med den utstrakte bruken av NAT. Dette vil gjøre det mer å mer vanskelig å få muligheten til å benytte offentlig synlige IP-adresser. IPv6 har veldig mange flere adresser (128 bits adresse mot 32 bit i IPv4).

c) Alle enhetene som trenger å behandle IP-protokollen på veien mellom avsender og mottaker må ha støtte for IPv6 om ikke leveringen skal mislykkes. Det vil si at for å sette det i global drift er det svært mange enheter som må støtte dette om det ikke skal oppstå mange soner som ikke får levert data. Dette tar lang tid og er krevende.

Oppgave 4: DHCP

a) DHCP står for «Dynamic Host Configuration Protocol» og benyttes for å automatisk tildele IP-adresser til enheter som kobler seg på et lokalt nettverk (LAN). Den brukes fordi det er en omstendelig prosess å skulle manuelt tildele (og formidle) IP-adresser til alle som vil koble seg opp mot nettverket. Automatisering av denne prosessen gjør at de færreste trenger å tenke over dette når de kobler seg opp.

b) Nettverk med mange enheter som kobler seg av og på over tid vil da kunne gjenbruke IP-adressene til enheter som ikke har vært aktive på en stund. Dette gjør at man ikke trenger et veldig stort subnett med mange IP-adresser som holdes opptatt av enheter som ikke svarer eller som har forlatt nettverket.

Oppgave 5: Transportlaget

a) UDP tilbyr multiplexing over en IP-adresse (porter) og kontroll av nyttelasten (checksum).

b) Samme som UDP, men i tillegg pålitelighet (garantert levering), levering i rekkefølge, flytkontroll (for ikke å sende fortere enn. mottakeren kan ta imot), metningskontroll (for å dele nettverksressursene med andre datastrømmer.

c) Det krever mindre ekstra data for overføring (mindre overhead). Det kan hende man ønsker å bygge sin egen versjon av flytkontroll, mentningskontroll e.l. i applikasjonen. Det at sendetidspunktet ikke styres av metningskontroll, gjør at applikasjonen som sender pakken kan bestemme sendetidspunktet selv (nyttig i f.eks. telekonferanse og videostreaming).

Oppgave 6: Praktisk anvendelse av kryptografi

a) Sikre trygg lagring av data på utrygge lagringsenheter, sikre trygg overføring av data i f.eks åpne datanettverk.

b) Kryptografi kan som nevnt i a) bidra til å sikre konfidensialitet. I tillegg kan integritet til data sjekkes ved å benytte sjekksumalgoritmer (hash-funksjoner). Bruk av symmetrisk kryptering kan brukes for å sikre autentisitet på data/meldinger sendt (når du vet hvem som kjenner den hemmelige nøkkelen), mens bruk av asymmetrisk kryptering for *digital signering* også kan sikre *uavviselighet* av informasjon.

c) Er disk på PC kryptert? Hva med mobiltelefon? Nettverkskommunikasjon, inkludert over wifi til trådløse aksesspunkter? Sender du kryptert epost, evt. krypterte dokumenter i epost? Kommunikasjon til nettsider? Mobilapper? Signerer du noe digitalt? Lagring av passord? Sjekker du eller datamaskinen din integritet til data/filer/nedlastet programvare/meldinger over nett?

Oppgave 7: Nøkkel og nøkkeldistribusjon

a) En kryptografisk nøkkel er rett og slett et stykke informasjon, som brukes som *parameter* til krypteringsalgoritmen (chifferen) man benytter, og som er med på å bestemme resultatet (altså cipherteksten) når klartekst krypteres med en kryptografisk algoritme. Algoritmene må antas å være kjent, mens kryptografiske nøkler (men unntak av offentlige nøkler) skal holdes hemmelige for å unngå at uvedkommende får tak i dem og klarer å dektyptere ciphertekst de ikke skal ha tilgang til.

b) **Typer av nøkler**: Symmetrisk hemmelig nøkkel, asymmetrisk privat nøkkel og asymmetrisk offentlig nøkkel.
   - **Symmetriske hemmelige nøkler** og **asymmetriske private nøkler** må alltid holdes konfidensielle (dvs. beskyttet mot uautorisert eksponering) og må ha integritet/autentisitet (dvs. beskyttet mot forfalskning/uautorisert modifisering).
   - **Asymmetriske offentlige nøkler** må ha integritet/autentisitet (dvs. beskyttet mot forfalskning/uautorisert modifisering), men trenger ikke konfidensialitet.

c) Formålet med *offentlig-nøkkel sertifikater* og *offentlig-nøkkel infrastruktur* (PKI) er å legge til rette for autentisering av entiteter på internett, samt kryptert overføring av informasjon mellom enheter. PKI skal sikre autentiske offentlige nøkler. PKI er et rammeverk for å legge til rette for å binde en offentlig nøkkel til en navngitt enhet, og denne bindingen kan bekreftes av en betrodd myndighet som utsteder et *sertifikat*. Et sertifikat forteller/bekrefter følgende: "Offentlig nøkkel K eies av X".

Oppgave 8: Symmetrisk kryptering

a) Alle kan lytte til radiosignaler så hvis nøkkelen blir lekket eller koden knekkes kan det være svært alvorlig.

b) Ikke alle har maskiner som kan spille av DVD-er eller kjøre CD-er. De må også transporteres på en måte. Posten er ikke 100% sikkert og dersom du transporterer den selv er det jo nesten litt unødvendig å kryptere.

c) USB-er er små og kan lett mistes. De kan også holde på mye data, som gjør det ekstra ille å miste dem.

d) Som oftest kan man ikke kryptere selve e-posten. Så mottaker kan se avsender, evt. andre mottakere, emnefelt, all tekst som er inne i selve e-posten og navnene på filer man sender med. Det er kun vedlegg (filer) som er kryptert. Dersom du sender e-posten feil eller noen klarer å få tilgang til e-posten er det altså mye de kan lese ut av denne infoen.

Å overlevere kryptografiske nøkler på en trygg og sikker måte er heller ikke uproblematisk, men det bør alltid skje over en annen kanal enn den man bruker for å overlevere de krypterte dataene.

Oppgave 9: Asymmetrisk kryptering

a) Du bruker en offentlig nøkkel for å kryptere. Kollegaen bruker den tilhørende private nøkkelen for å dekryptere. Dette sikrer konfidensialitet for dokumentet mellom deg og kollegaen din, fordi ingen andre enn den som kjenner til den private nøkkelen kan dekryptere dokumentet.

b) Du bruker en privat nøkkel for å kryptere og kollegaen din bruker den tilhørende offentlige nøkkelen for å dekryptere (også kalt å *verifisere* avsender). Dette sikrer autentisitet og uavviselighet så lenge det kun er en aktør (du) som kjenner den private nøkkelen, og dermed har man en visshet om at det er du som har signert dokumentet.

Oppgave 10: Hash-algoritmer (sjekksumalgoritmer)

- **Lett å beregne:** Enkelt å regne ut sjekksum for en gitt melding
- **Enveis**: Ikke mulig (vanskelig) å finne en melding for en gitt sjekksum
- **Kollisjonsresistent**: Ikke mulig (vanskelig) å finne to ulike meldinger med samme sjekksum, og Ikke mulig (vanskelig) å endre en melding uten at sjekksummen også blir en annen.

Oppgave 11: Kommunikasjon i datanettverk

a) Med http som protokoll går all kommunikasjon mellom Bruker og Server i klartekst, og hvem som helst (som har teknisk innsikt) kan lytte til kommunikasjonen.
b) Med sikker forbindelse mellom Bruker og WiFI-ruter forhindres Lytter1 i å fange opp kommunikasjonen. Alle de andre aktørene kan fremdeles se data i klartekst.
c) Kun Bruker og Server ser data de sender seg imellom i klartekst.
d) Kun Bruker og Server ser data de sender seg imellom i klartekst.
e) Hvis man bruker et usikret trådløst nettverk, kan Lytter1 (og alle de andre) se headere i datapakkene som sendes, f.eks. at Bruker har en «dialog» med serveren devilry.ifi.uio.no. Men kryptert trådløsforbindelse krypteres også headerne mellom Bruker og WiFi-ruter, og Lytter1 forhindres i å fange opp noe som helst fornuftig.

Oppgave 12: Case-oppgave (Hash-kryptering/sjekksumalgoritme)

a) Dersom to brukere har samme passord, har de den samme hashen. Det finnes databaser som inneholder hasher til vanlige passord, så hvis brukerne deres benytter dårlige passord kan trusselaktører finne disses passord-hashene i slike databaser, og dermed finne en match. Teamet ditt burde kanskje ikke prøve å lagre brukeres passord selv? En mulighet er å benytte en eller annen tilbyder av autentiseringstjenester isteden (Google, Facebook, Feide, BankID?).
b) Her ivaretas faktisk konfidensialitet ved at passord ikke lagres i klartekst, men kun som en hash-verdi, og dermed ikke kan leses av uvedkommende.
c) Hash algoritmer “fordøyer” data ned til en streng som har fast lengde (ofte 16, 32 eller 64 bit og i heksadesimal).

Praktiske oppgaver:

Oppgave 13: Test av sjekksumalgoritmer – praktisk oppgave

a) Nei, sjekksummen blir en annen når fila endres. Dette er en grunnleggende egenskap ved sjekksumalgoritmer. Men observer gjerne at sjekksummen blir den samme når du bruker programmet sha256sum på samme fil flere ganger.

b) Man kan sammenligne sjekksum av en datafil opp mot en kjent sjekksum av denne fila. Er den lik? Isåfall er fila ikke endret mellom første og andregangskjøring av sjekksum.

c) Sjekksumalgoritmer brukes ofte ved deling av programvare (f.eks. nedlasting fra nett). Tilbyderen av programvare genererer en sjekksum av orignalen (ofte et filarkiv, f.eks. .zip-fil), som publiseres sammen med programvaren. Brukere som laster ned programvare kan enkelt selv kjøre en sjekksumalgoritme etter nedlasting, og kan med det finne ut om programvaren er det den utgir seg for å være, eller er den kanskje endret etter at sjekksummen ble generert? Vil også kunne avdekke hvorvidt en nedlasting har gått greit eller ikke (pakketap?). Helt konkret eksempel: https://www.postgresql.org/ftp/source/v17.0/, nedlasting av kildekode PostgreSQL databaseserver. Her finner du både programvaren og tilhørende sjekksummer for både MD5 og SHA256.

Oppgave 14: Å bruke SSL Labs – praktisk oppgave

Praktisk oppgave uten løsningsforslag.