Elementær Kryptografi

(Appendix A, Cryptography Basics, Building Secure Software)

Michael Mortensen
michaelm@ii.uib.no

10/10/05

INF329 –Utvikling av sikre applikasjoner

Elementær kryptografi

- Hva er kryptografi
  - Kryptografi er vitenskapen som omhandler språklige og matematiske teknikker for å sikre informasjon, spesielt gjelder dette innenfor kommunikasjonsteknologien.
  - Målene med kryptografi er å tilby:
    - Konfidensialitet
    - Autentisering
    - Integritet
    - Ikke- fornektbarhey (nonrepudation)

Konfidensialitet

- Kontrollere tilgangen til data slik at bare autoriserte parter kan forstå den
- Uvedkommende kan ofte få tak i dataen, men klarer ikke å få noe meningsfylt ut av den
- Autoriserte parter deler en hemmelighet som gjør det mulig å dekryptere teksten
- Som oftest er hemmeligheten en kryptografisknøkkel

Autentisering

- Det skal være mulig for en mottaker å bekrefte opphavet til en melding
- Hindrer at uvedkommende kan forfalske identideten sin
- For identifisering brukes gjerne en nøkkel, som et passord.

Integritet

- Gjør det mulig for en mottaker å sjekke at meldingen ikke har blitt modifisert eller skadet under overføring
- Hinder en angriper i å bytte en gyldig melding med en falsk/ ugyldig melding

Ikkefornektbarhet (nonrepudation)

- Handler om bekreftelse
- Skal være mulig for en sender å kunne bekrefte at mottaker har mottatt meldinger
- Skal være mulig for en mottaker å bekrefte at senderen har sendt meldingen

Kryptografi

- I en perfekt verden ville kryptografi tilby en solid uknekkelig løsning på de nevnte målene.
- Verken kryptografi eller verden er perfekt og derfor må vi analysere våre algoritmer for å gjøre de så sikker som mulig. Dette kalles kryptoanalyse.

Kryptoanalyse

- Kryptoanalyse er vitenskapen som studerer metoder for å kunne forstå meningen av den krypterte teksten.
- Kort sagt er det ofte metoder for å finne den hemmelige nøkkelen.
- Kryptoanalyse er også analysen av metoder for å unngå eller overgå sikkerheten til kryptografiske-algoritmer og protokoller.

Brute force angrep

- Brute force angrep er det enkleste angrep mot et kryptosystem.
- Handler om å prøve alle mulige nøkler helt til man finner riktig nøkkel
- Meningsløst på kryptosystemer med store nok nøkkler

Brute force angrep

En vanlig kryptografisknøkkel er i dag på 128 bits. I et bruteforceangrep betyr det at vi må prøve 2^{128} forskjellige nøkler. Hvis vi antar at vi vil klare å finne riktig nøkkel etter vi har prøvd halvparten av alle nøklene må vi fortsatt prøve 2^{127} ulike nøkler.

Har vi en maskin som klarer 16777216 nøkler per sekund vil det fortsatt ta 3215754947206518014958666 år å finne riktig nøkkel!

Angrepstyper

- Når man angriper et kryptosystem har man et utgangspunkt, et angrep går som oftest under et eller flere av disse utgangspunktene:
  - Kjent siffertekstangrep
  - Kjent klartekstangrep
  - Valgt klartekstangrep
  - Valgt siffertekstangrep
  - Relaterte nøkler angrep
  - Sidekanalsangrep
  - Gummislangeangrep

Kjent siffertekstangrep

- Tilgang til en eller flere meldinger i kryptert form
- Vi vet ikke hva slags informasjon er kryptert så vet ikke hva vi forventer å se etter vi har dekryptert
- Å identifisere den riktige klarteksten kan kreve svært komplekse analyser
- Et typisk bruteforce angrep

Kjent klartekstangrep

- Tilgang til både sifferteksten og hele eller deler av dens klartekst versjon
- Mulige å teste nøkler på sifferteksten for å se om de dekrypterer til den kjente klarteksten.
- Altså enklere å sjekke at vi har funnet riktig nøkkel
- Linær kryptoanalyse er en form for kjent klartekstangrep.

Valgt klartekstangrep

- Angriper velger klartekst som skal krypteres.
- Får den krypterte versjonen av sin klartekst kryptert med en hemmelig nøkkel
- Hvis alle klartekstene velges på forhånd er det *gruppevis valgt klartekst angrep* hvis angriper har mulighet til å velge klartekst etterhvert, og f.eks bruke informasjon fra tidligere krypteringer, er det adaptiv valgt klartekstangrep
- Differntiell kryptoanalyse er et valgt klartekstangrep

Valgt siffertekstangrep

- Angriper velger siffertekst og ser hva klartekst versjonen blir dekrypter med en hemmelig nøkkel
- Mange kryptosystemer er knekkelig med valgt siffertekst angrep, som ElGamal.
- Finnes både adaptiv og gruppevis valgt siffertekstangrep

Relaterte nøklerangrep

- Observere hvordan den kryptografiskealgoritmen bruker nøklene
- Finne/ bruke sammenhengen mellom nøklene.
- WEP er et eksempel på et siffer som ikke er sikker mot relaterte nøkler angrep .

Sidekanalsangrep

- Angriper *ikke* algoritmen, men implementasjonen til algoritmen.
- Finner visse egenskaper implementasjonen har som kan lekke informasjon som kan brukes i angrepet.
  - Tidtaking
  - Strømforbruk
  - Radio- / lyd- signaturer

Gummislangeangrep

- Et angrep som omgår både algoritmen og dets implementasjon.
- Ofte det mest effektive angrepet, bruker metoder som:
  - Social Engineering
  - Bestikkelser
  - Trusler

Symmetrisk kryptografi

- Brukes primært for å kryptere data
- En hemmelig nøkkel deles mellom partene som skal kommunisere
- Nøkkelen må utvekles på en sikker måte:
  - Direkte
  - Nøkkelutvekslingsprotokoll (Diffie Hellman)
- Kryptert data kan overføres på en usikker kanal

Symmetrisk kryptografi

K = Hemmelig nøkkel

M = Klartekstmeldingen

S = Sifferteksten

Krypterings/ Dekrypteringsfunksjonene er ofte like hverandre (f.eks DES), men er ikke nødvendigvis det (f.eks AES)

Symmetrisk kryptografi

- Vi har to typer symmetrisk kryptografi
  - Strømsifferer
  - Blokksifferer

Strømsiffer

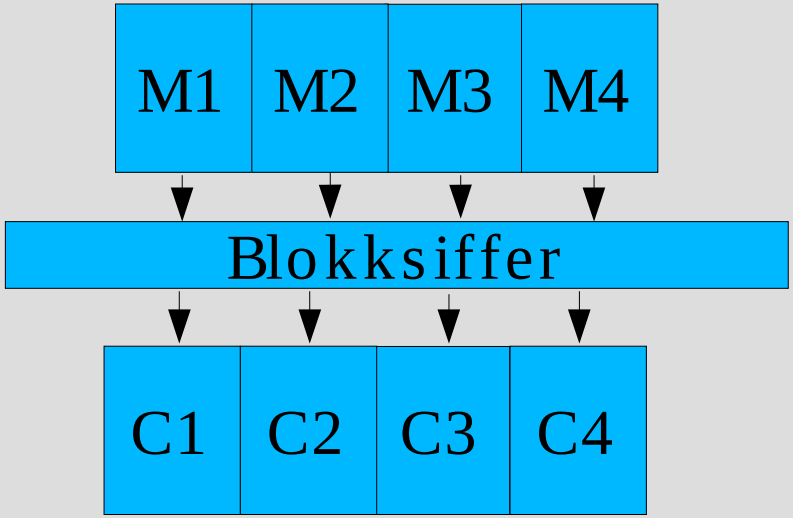
- Et og et tegn krypteres om gangen.
- Raskere enn blokksifferer
- Lettere/ mindre komplekst enn blokksifferer, noe som er viktig når det skal lages hardwaresystemer. Kompleksitet koster mer.
- Samme starttilstand må aldri brukes mer enn en gang!

Blokksiffer

- De fleste kjente og brukte symmetriske kryptosystemer er blokksiffer (DES, 3DES, AES, IDEA, BlowFish)
- Deler meldingen opp i like store deler (blokker)
- Blokkene er som oftest 64 eller 128 bits
- Det finnes flere moduser for blokksifferet, bl.a.
  - ECB (Electronic Code Book mode)
  - CBC (Cipher Block Chaining mode)
  - CFB (Cipher FeedBack mode)
  - CTR (Counter Mode)

Electronic Code Book Mode

- Hver blokk blir kryptert uavhengig av de andre

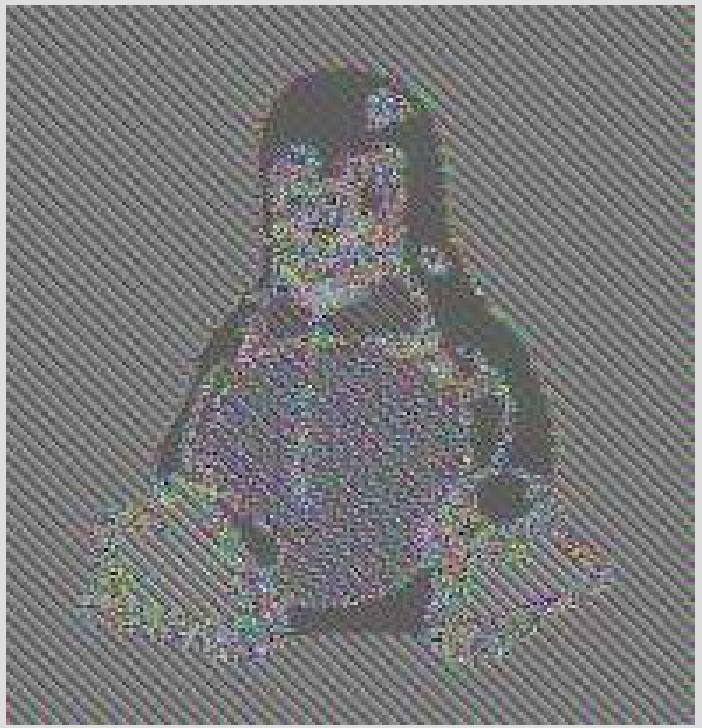


- Like klartekstblokker gir like siffertekstblokker, dermed kan en angriper finne og misbruke gjentakende mønstre.

Electronic Code Book Mode

Illustrerende eksempel

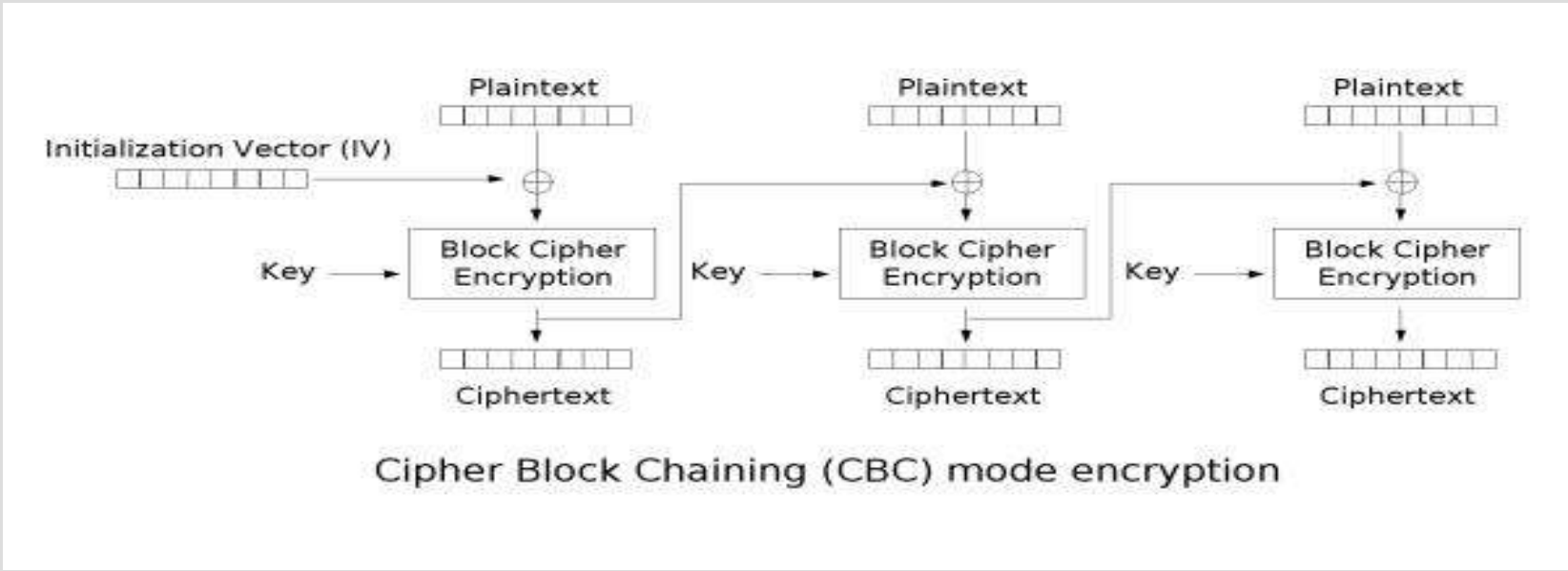
Ukryptert Tux



Tux kryptert med ECB

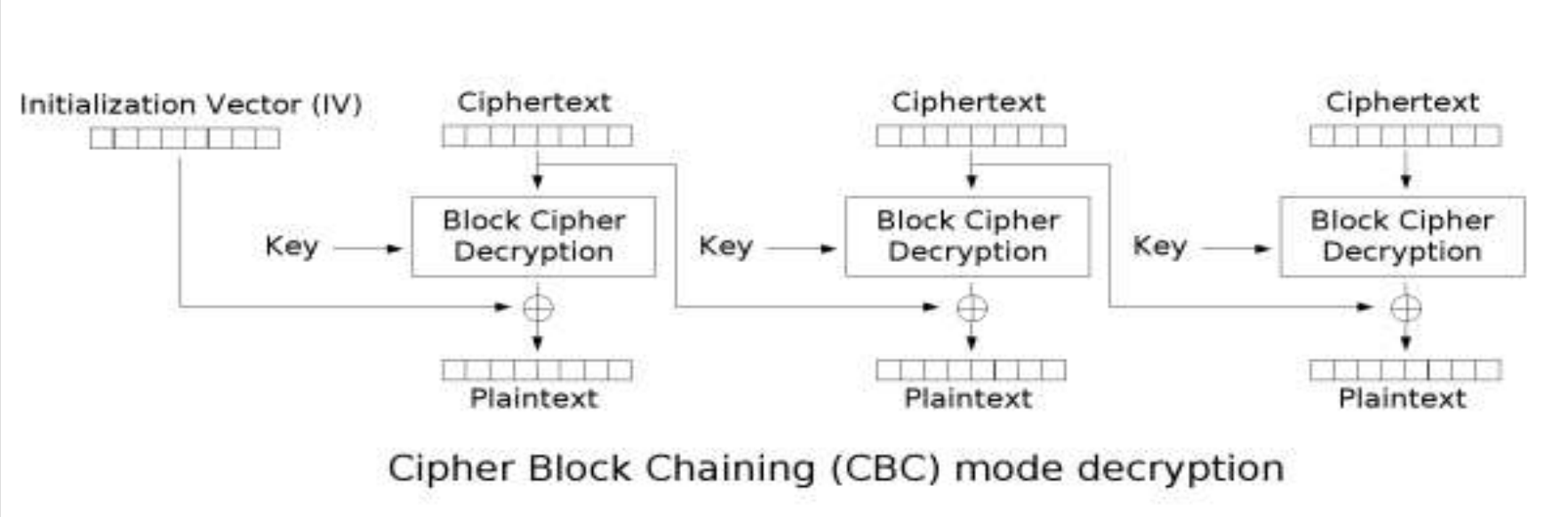
Cipher Block Chaining Mode

- Starter med en IV (Initialization vector)som er like stor som blokkstørrelsen
- Hver blokk XORes med forrige blokk, første blokk XORes med Iven.



Cipher Block Chaining (CBC) mode encryption

Cipher Block Chaining Mode

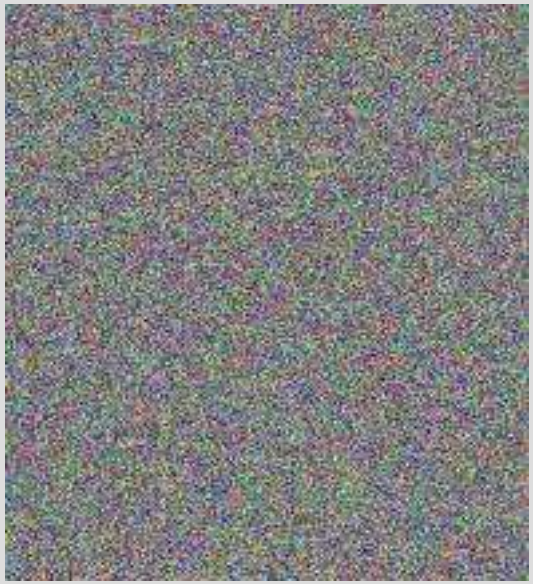


Cipher Block Chaining (CBC) mode decryption

- Må utføres sekvensielt, utelukker parallelle krypteringer
- Følgefeil
- Er den mest vanlige modusen for kryptering

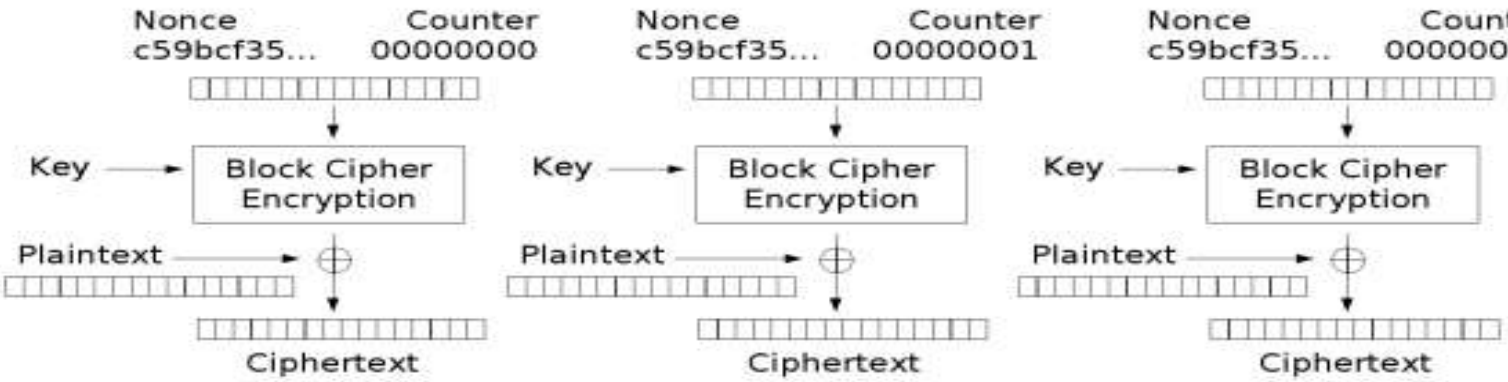
Cipher Block Chaining Mode

Tux med CBC



Counter Mode

- En modus som får et blokksiffer til å fungere som et strømsiffer
- En teller krypteres med den hemmelige nøkkelen, resultatet av dette XORes med klarteksten for å danne sifferteksten
- Telleren må være stor nok til at den ikke gjentar seg ofte!



Counter (CTR) mode encryption

Offentlig nøkkel kryptografi (assymetrisk kryptografi)

- Løser det største problemet med symmetrisk kryptografi, nøkkelutveksling.
- Hver person har en offentlig og en privat nøkkel.
- Det finnes ingen **delte** hemmelige nøkler i assymetrisk kryptografi.
- Den offentlige nøkkelen kan sendes over usikre kanaler eller distribueres på en annen måte. Den private nøkkel må forbli hemmelig.
- Bygger på ideen om 'trapdoor one- way functions'
- De mest kjente offentlig nøkkel algoritmene er
  - Diffie- Helman
  - RSA
  - El Gamal

Offentlig nøkkel kryptografi (eksempel)

- Bob vil sende en melding til Alice
- Bob får Alice sin offentlige nøkkel fra Alice via epost, telefon eller et annet medium.
- Bob krypterer meldingen M med Alice sin offentlige nøkkel og sender den krypterte meldingen til Alice.
- Alice, med sin private nøkkel, er den eneste som klarer å dekryptere meldingen.

Offentlig nøkkel kryptografi

- Er veldig komplekst og dermed tungvindt å bruke på store meldinger
- Det sies at DES er 100 ganger raskere enn RSA!
- Hybridløsninger er ofte best
  - En symmetrisk nøkkel utveksles via det assymetriske kryptosystemet
  - Etter nøkkelutveksling har funnet sted gjøres all kryptering på den raske symmetriske metoden.

Offentlig nøkkel kryptografi

- Sikkerheten til et assymetrisk kryptosystem er antatt til å være like vanskelig å løse som enveis-funksjonen den bygger på.
- I RSA, hvor den offentlige nøkkelen består bl.a. av produktet N til to store primtall er sikkerheten antatt til å være lik vanskeligheten av å finne en av faktorene til N.
- Siden det er ikke bevist om det finnes en faktoriseringsalgoritme som kjører i polynomisk tid kan heller ikke sikkerheten til RSA fastslås.
- Rekorden på faktorisering hittil, med standard faktoriseringsmetoder, er 663 bits. En typisk RSA-nøkkel er >1024 bits. I dag anbefales det minst 2048 bits.

Offentlig nøkkel kryptologi

- Sårbart for Man- In- the- Middle angrep. Eksempel:
- Bob vil sende en melding til Alice
- Bob *tror* han får Alice sin nøkkel via epost, men egentlig har Eve snappet opp Alice sin nøkkel på veien og sendt Bob sin egen.
- Bob krypterer meldingen med Eve sin nøkkel og sender den til Alice.
- Igjen snapper Eve meldingen på veien, men nå kan Eve lese meldingen siden den er kryptert med hennes nøkkel.
- Folk må bekrefte gyldigheten til nøklene! Dette kan gjøres med PKI, Ring of trust, sikker utveksling, osv.

Kryptografisk Hashing

- Enveisfunksjoner som tar en tekst som input og genererer en streng av konstant lengde (ofte kjent som et digitalt fingeravtrykk)
- Skal oppføre seg tilsynelatende tilfeldig, men skal fortsatt være deterministisk og helst rask.
- En kryptografisk hash må oppfylle disse kravene for å antas sikker:
  - Gitt en hash *h* skal det være ugjennomførlig å finne *h = hash(m)*
  - Gitt en melding *m* skal det være ugjennomførlig å finne en annen melding *x* (*x != m)* der *hash(m) = hash(x)*.
  - Det skal være ugjennomførlig å finne to forskjellige meldinger *m1* og *m2* slik at *hash(m1) = hash(m2)*

Kryptografisk hashing

- Hashing kan brukes for å sjekke om en melding er identisk med en annen. To like meldinger vil gi samme hash resultat.
- Hashing alene kan sjekke om feil har oppstått under overføringen, men kan ikke hindre misbruk fra en angriper ettersom han kan alltids endre meldingen og regne ut en ny hash.
- MAC (Message Authentication Code) kombinerer hashing med en kryptografisknøkkel for rask sjekking av autentisitet og integritet. Ideen bak en MAC er å ta en hash av meldingen slå den sammen med nøkkelen og litt annen data og så ta en hash av den totale strengen. Ikke skriv dine egne MACs, bruke kjente sikre metoder, som HMAC.

Hashing (eksempler på misbruk)

Uten kryptografi

- Bob sender 'Overfør 1000kr til Alice' til banken hashet med SHA-1
- Alice har i det siste blitt litt grådig, så hun snapper opp meldingen og endrer den til 'Overfør 100000kr til Alice', regner ut en ny hash med SHA-1 og sender så meldingen til banken.
- Banken ser at meldingen passer til hashen, så den er gyldig, og utbetaler 100000kr.

Med kryptografi

- Bob sender 'Overfør 1000kr til Alice' til banken hashet med SHA-1 og krypterer hashen.
- Alice er fortsatt grådig, så hun snapper opp meldingen og sender den samme meldingen til banken 100 ganger.
- Siden banken ser på dette som 100 uavhengige, men gyldige, transaksjoner blir pengene utbetalt.

Hashing (eksempel)

- Bob sender 'Overfør 1000kr til Alice' til banken, men inkluderer denne gangen en teller med i meldingen eller i MACen. MACen bruker en nøkkel som deles mellom Bob og banken,
- Banken mottar meldingen, sammen med nøkkelen K. Hvis den stemmer sjekker de at telleren er større enn telleren de har lagret og hvis den også stemmer inkrementer banken sin teller og gjennomfører transaksjon.
- Hvis Alice snapper opp meldingen er det lite hun kan gjøre. Hun kan ikke endre meldingen siden hun ikke kjenner til nøkkelen, og replay- angrepet fungerer heller ikke lengre pga. telleren.

Hashing

- Kryptografisk sterke hashalgoritmer er meget vanskelig å få til riktig. Du bør derfor, som det meste innenfor kryptografi, holde deg til gode utprøvde rutiner. Boken anbefaler SHA-1, men siden den ble publisert har det blitt funnet flere svakheter i SHA-1, du bør derfor vurdere andre hashalgoritmer som SHA-256, SHA-512 eller RIPEMD-160.

Digitale Signaturer

- Målet med digitale signaturer er:
  - Signaturen skal være et bevis på autentisitet
  - Den skal være umulig å forfalske og som en følge av dette skal det ikke være mulig for han/ hun som står signert å fornekte signaturens gyldighet.
  - Etter at et dokument er signert skal det være umulig å endre dokumentet uten å underkjenne signaturen
  - Det skal ikke være mulig å flytte signaturen fra et dokument til et annet (ulikt) dokument
- Bruker en kombinasjon av hashing og assymetrisk kryptografi

Digitale Signaturer

Sending

- Først hashes dokumentet til å produsere en kryptografisk sterk hash.
- Hashen krypteres med brukerens **private** nøkkel
- Videre sendes dokumentet og signaturen til mottaker

Mottak

- Dokumentet og signaturen mottas. Mottakeren prøver først å dekryptere signaturen med senderens offentlige nøkkel. Dette resulterer i en hash. Mottaker regner så ut hashen av dokumentet og sammenliker den med den utregnte hashen. Hvis de to er like er signaturen gyldig.

Digitale Signaturer

- Digitale signaturer kan angripes fra flere hold.
- Man kan angripe hashen, f.eks ved å finne en annen melding som har samme hash- verdi som den orginale.
- Akkurat som assymetrisk kryptografi kan man ikke alltid være sikker på opphavet til den offentlige nøkkelen.

Konklusjon

- Kryptografi kan løse mange problemer for oss, men hvis ikke det anvendes på en gjennomtenkt måte blir det ofte meningsløst
- Man bør bruke kjente sikre algoritmer ettersom de har stått imot flere års angrep og analyse.
- Det finnes én kryptografiskalgoritme som er 100%sikker, det er one-time pads. Den er veldig upraktisk og er derfor ikke tatt med her

Spørsmål?

Spørsmål?