IT-sikkerhed

Hackere og modmidler

Hackerangrebet på Mærsk

Hackerangrebet på Mærsk

- I Mærsks afdeling i Ukraine installerede man systemet M.E.Doc på én computer
- M.E.Doc er et system til at beregne skat med – alle virksomheder i Ukraine bruger det
- M.E.Doc var inficeret med NotPetya – man mener, at den russiske hackergruppen Sandworm stod bag
- NotPetya var oprindelig udviklet af NSA!
- Via M.E.Doc kom NotPetya rundt til de vigtigste områder i Ukranie:
  - Energiforsyningen (strøm og gas)
  - Hospitaler
  - Lufthavne

Hackerangrebet på Mærsk

- Det kostede Mærsk 1,9 mia. kr.
- Mærsk var allerede inden angrebet klar over, at der var alvorlige sikkerhedsfejl
- Man havde godkendt og budgetteret med en sikkerhedsopdatering, men gennemførelsen af en sådanne blev ikke til en KEY PERFORMANCE INDICATOR (KPI). Dvs. ingen bonusser til it-chefer – derfor blev opdateringen ikke en realitet!
- Problemer:
  - Mange serverer kørte Windows 2000
  - Manglende netværkssegmentering

Øvelse: Hvem er hackerne? – dokumenteres i logbogen

- Video: Morfar forklarer: Ikke alle hackere er nogle "fuckhoveder"
- Læs artiklen nedenfor videoen og forklar kort og med egne ord hvad følgende hacker typer dækker over:

    - **Black** hats – "den kriminelle"
    - **White** hats – "den professionelle"
    - **Grey** hats – "den barmhjertige samaritaner"
    - **Script kiddies** – "den uvorne teenager"
    - **Red** hats – "selvtægtsmanden"
    - **Green** hats – "nybegynderen"

- Hvilken type hacker, er manden fra Viborg (som vi hørte om i "Frederik forklarer internettet?")

Øvelse: Hvorfor bliver man hacker?

- Tal med sidemanden om hvilke grunde der kan være til, at man vælger at blive hacker.

- Penge
- Politiske årsager
- Berømmelse
- Forskning og udvikling af antivirus-software (white hats)

HUSK CIA-modellen

[C] *fortrolighed*: Uvedkommende kan ikke opsnappe kommunikationen.

[I] *integritet*:

- *data-integritet*: Indholdet af en besked fra A til B kan ikke ændres under transporten over netværket.
- *autenticitet*: En uvedkommende aktør C kan ikke udgive sig for at være A eller B.
- *uafviselighed*: Hvis A sender B en besked, kan A ikke bagefter nægte det.

[A] *tilgængelighed*: Uvedkommende kan ikke afbryde A's og B's mulighed for at kommunikere.

Øvelse: Hvordan opnås CIA?

- Tal med sidemanden: Hvordan sikres elementerne i CIA-modellen:

[C] *fortrolighed*: Uvedkommende kan ikke opsnappe kommunikationen.

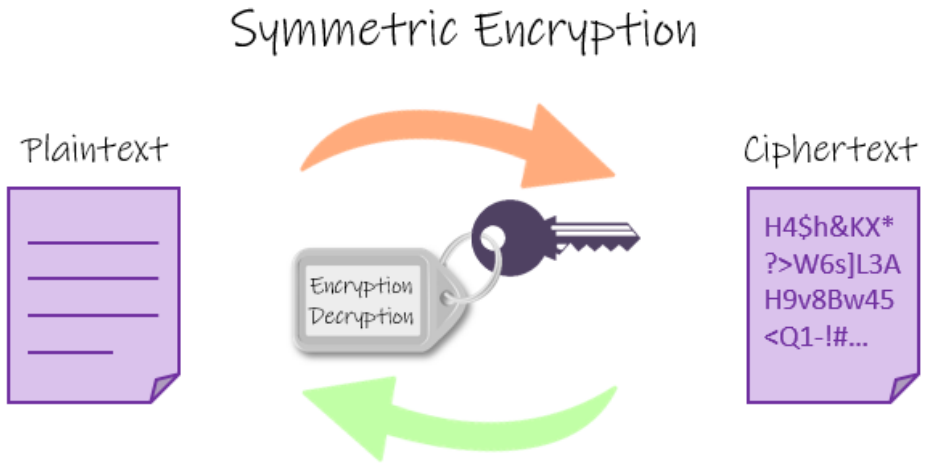
[I] *integritet*:

- *data-integritet*: Indholdet af en besked fra A til B kan ikke ændres under transporten over netværket.
- *autenticitet*: En uvedkommende aktør C kan ikke udgive sig for at være A eller B.
- *uafviselighed*: Hvis A sender B en besked, kan A ikke bagefter nægte det.

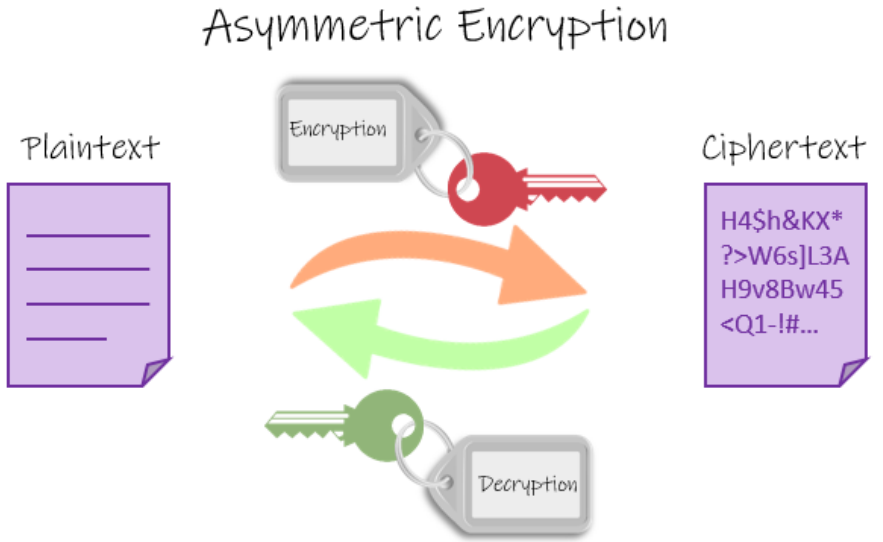
[A] *tilgængelighed*: Uvedkommende kan ikke afbryde A's og B's mulighed for at kommunikere.

[C] fortrolighed

- Ønske: data skal holdes hemmelig for uvedkommende
- Løsning: kryptering



Samme nøgle bruges til enkryptering og til dekryptering

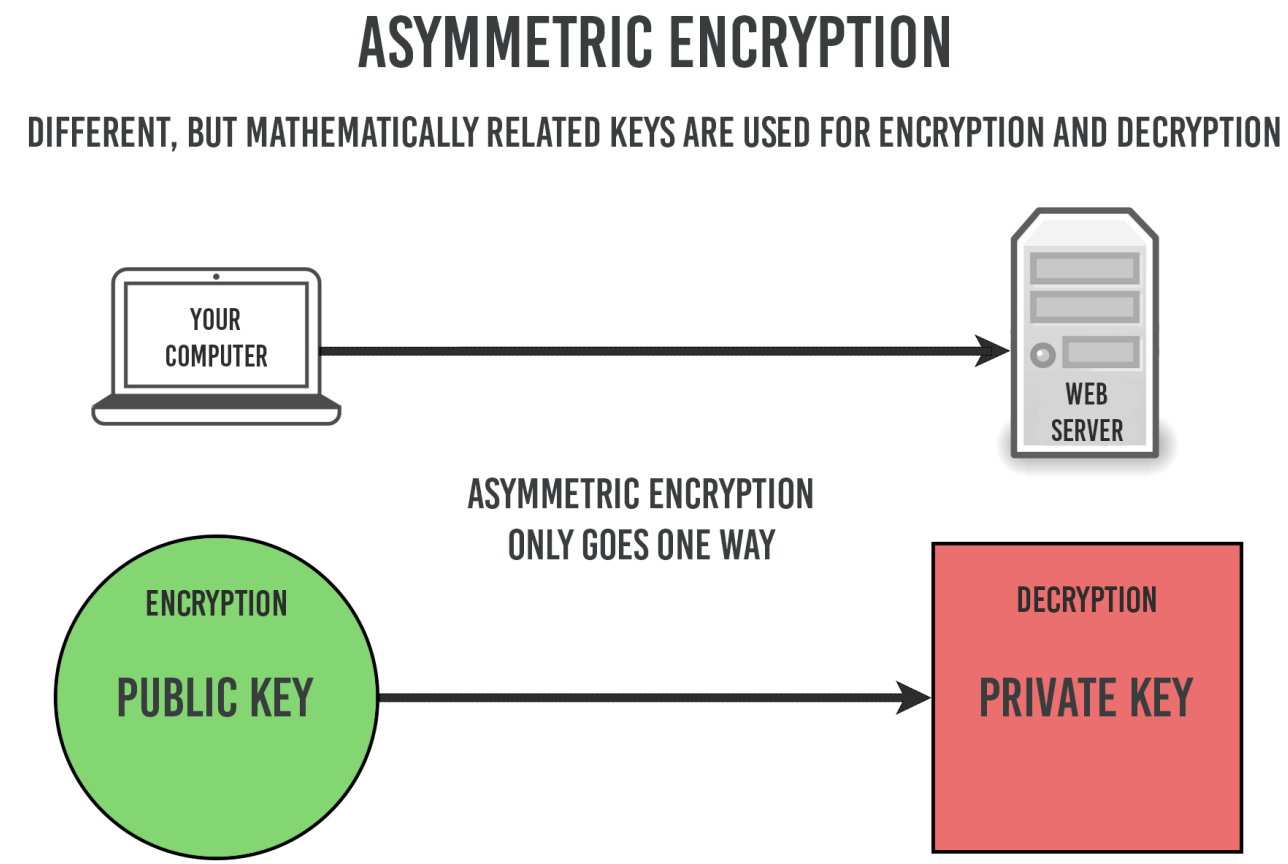


Offentlig nøgle bruges til enkryptering og privat nøgle bruges til dekryptering

[C] fortrolighed

- På et åbent internet: det er svært at udveksle nøgler → brug asymmetrisk kryptering
  (til at udveksle en hemmelig nøgle, som kan bruges til symmetrisk kryptering efterfølgende)

- Anvendes i protokollen HTTPS



En sidebemærkning om symmetrisk kryptering

Eksempler på symmetriske krypteringsalgoritmer:

- DES (Data Encryption Standard) fra 1977:
  - 56-bits nøgle (muligheder $2^{56} = 7{,}21 \cdot 10^{16}$)
  - Kryptoanalysehold brød DES på under 4 måneder i 1997 (i 1999: 22 timer)

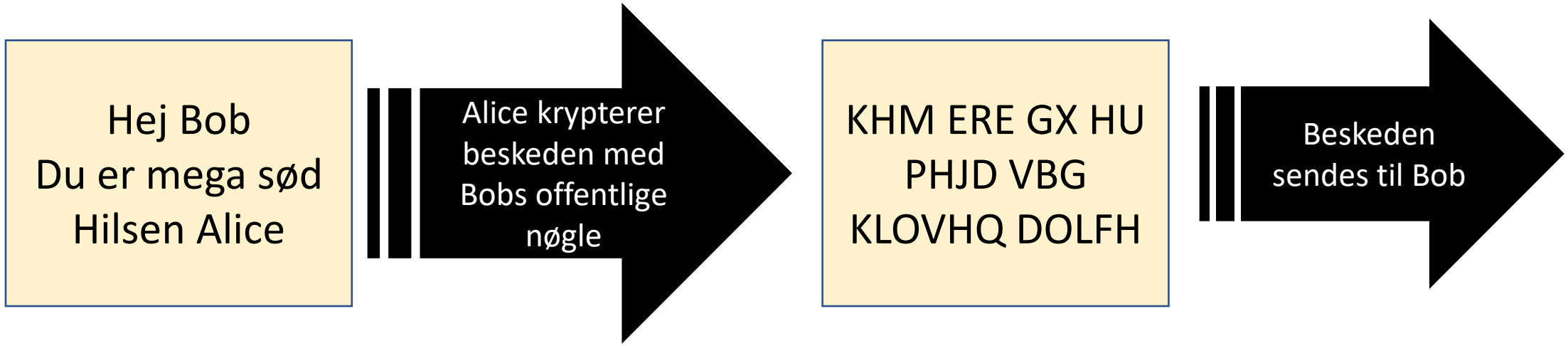
En sidebemærkning om symmetrisk kryptering

Eksempler på symmetriske krypteringsalgoritmer:

- AES (Advanced Encryption Standard) fra 2001:
  - Nøglelængder på: 128, 192 eller 256 bits
  - Brute force attack: Afprøv alle muligheder…
  - 256-bits nøgle (muligheder $2^{256} = 1{,}16 \cdot 10^{77}$)
  - En maskine, der kan knække en DES-krypteret besked på 1 sekund, skal bruge omtrent 150 billioner år på at knække en besked, der er AES-krypteret med en 128-bits nøgle.
- AES bruges af:
  - Den Amerikanske regering
  - NSA når Top Secret dokumenter skal hemmeligholdes

[I] integritet (dataintegritet)

Ønske: Alice vil gerne sende en hemmelig besked til Bob



[I] integritet (dataintegritet)

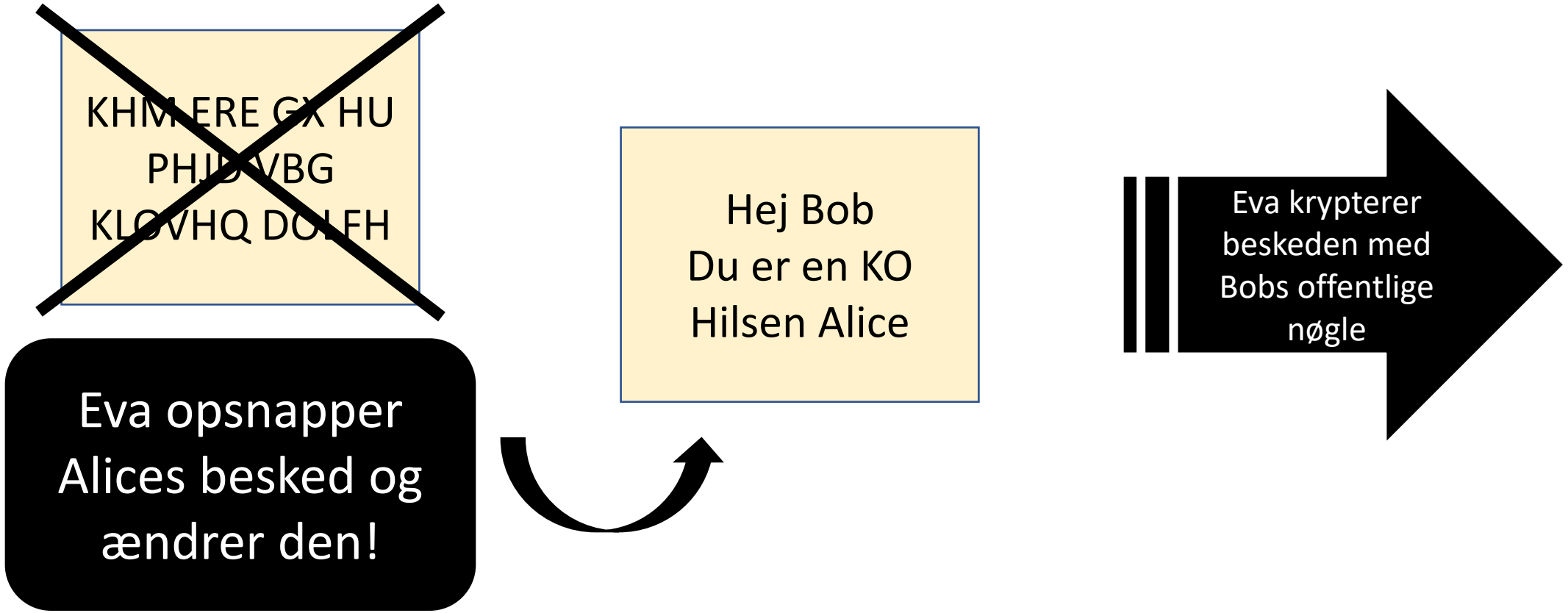
Bob modtager nu Alices besked

KHM ERE GX HU
PHJD VBG
KLOVHQ DOLFH

Hej Bob
Du er mega sød
Hilsen Alice

[I] integritet (dataintegritet)

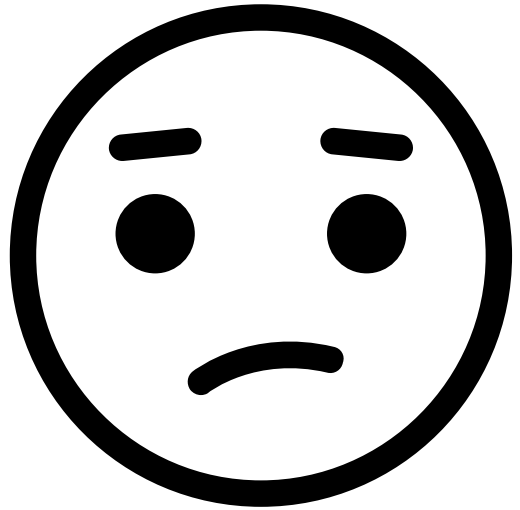
Men Alice har en rival: Eva (evil Eve)



[I] integritet (dataintegritet)

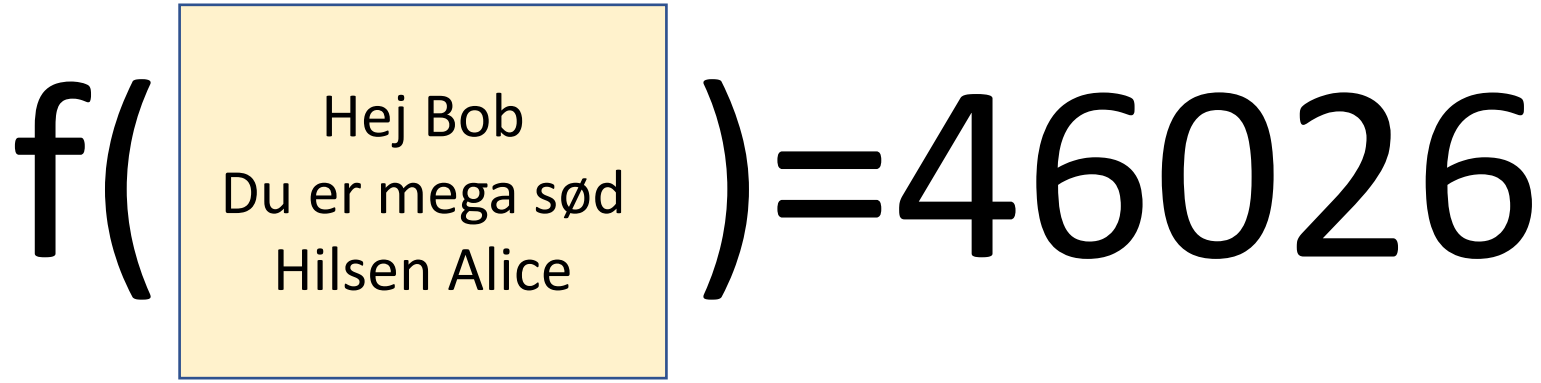
Det vil sige…

- Fortroligheden er sikret (Eva kunne jo ikke læse Alices hemmelige besked til Bob)
- Men vi har ikke sikret dataintegritet (beskeden er blevet ændret!!)



[I] integritet (dataintegritet)

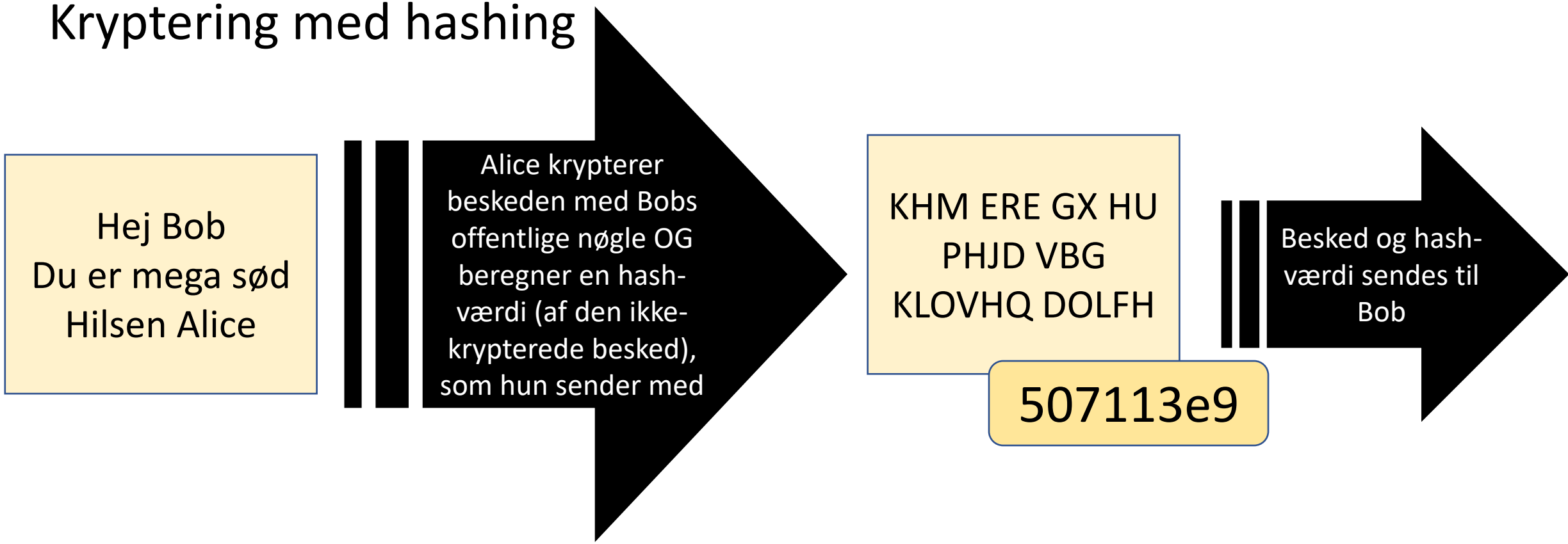
- Løsning: anvend en hash-funktion!



- Hash-funktionen tager en besked som input og beregner et output (hash-værdien)
- Samme besked giver altid samme hash-værdi
- To forskellige beskeder giver (næsten) aldrig samme hash-værdi (man siger at hash-funtionen er kollisionsfri)
- Givet en hash-værdi er det umuligt at komme tilbage til den oprindelige besked

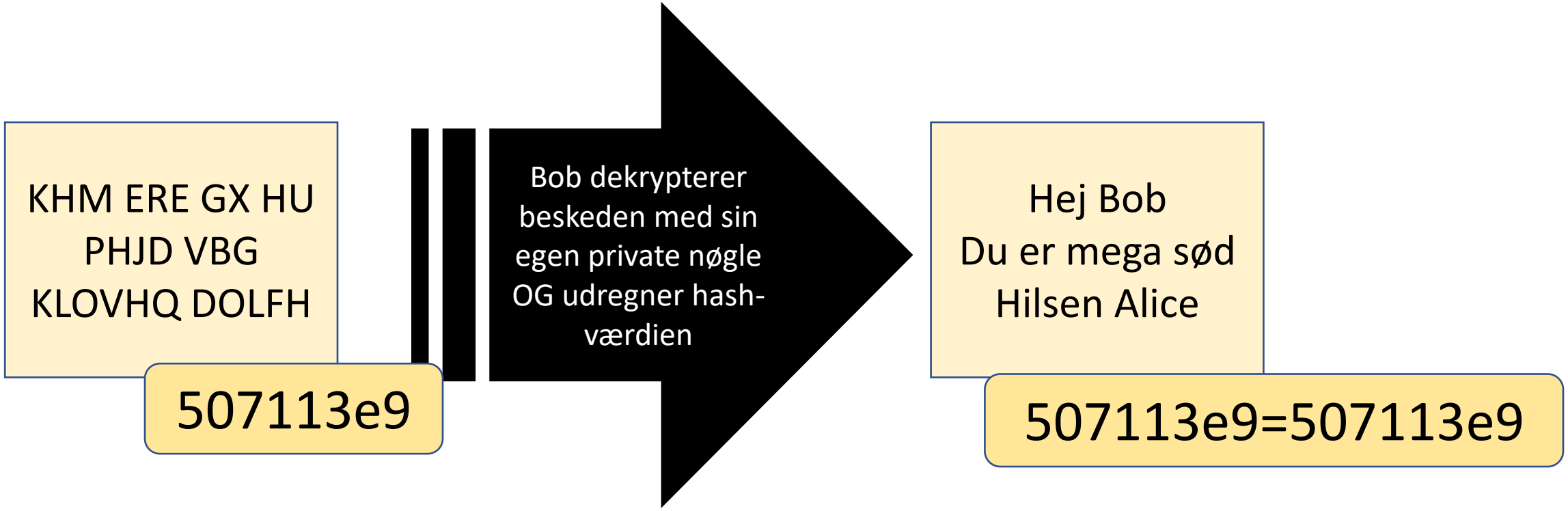
[I] integritet (dataintegritet)

Kryptering med hashing



[I] integritet (dataintegritet)

Bob modtager nu Alices besked incl. hash-værdi



De to hash-værdier stemmer overens og Bob ved nu, at der ikke er blevet ændret i beskeden! HURRA!

Øvelse med sidemanden: kryptering og hashing

- Skriv en kort besked og kryptér den vha. one-digit encryption. Brug gerne denne side:
  https://www.ramskovsressourcer.dk/undervisning/monoalfakryptologi.html

- Brug app'en fra sidst (kortlink.dk/2r7qc) og udregn hash-værdien for din besked (inden den blev krypteret).

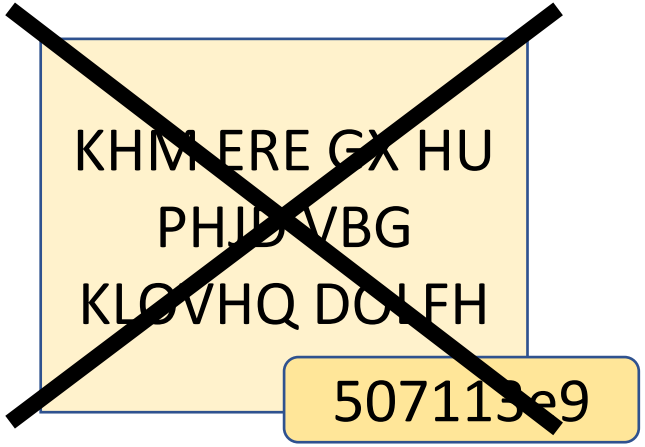
- Angiv din nøgle sammen med din krypterede besked og hash-værdien (send til din sidemand via Lectio)

Opsamling

- Hvilke beskeder modtog I? Stemte hashværdierne overens?
- Diskuter med sidemanden: Er der stadig problemer med denne metode?

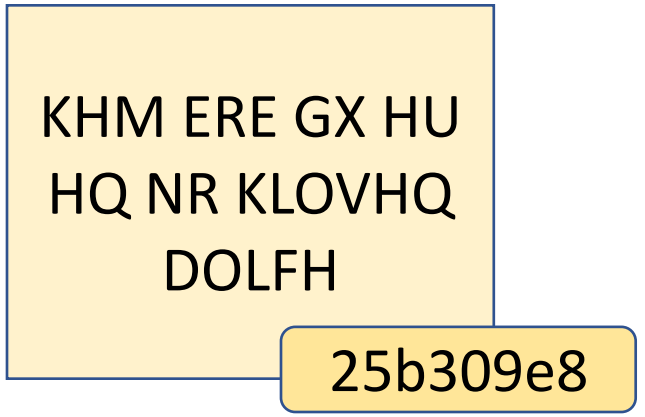
[I] integritet (dataintegritet) – problem…

Okay, det vil sige at hashing IKKE er nok til at sikre dataintegritet, fordi…



Eva opsnapper Alices besked og hash-værdi!

Eva ændrer beskeden og udregner en hash-værdi for den nye besked!



Eva sender den nye besked og den nye hash-værdi til Bob

[I] integritet (dataintegritet) – problem…

Bob modtager nu beskeden

KHM ERE GX HU
HQ NR KLOVHQ
DOLFH

25b309e8

Bob dekrypterer beskeden og udregner hash-værdien

Hej Bob
Du er en KO
Hilsen Alice

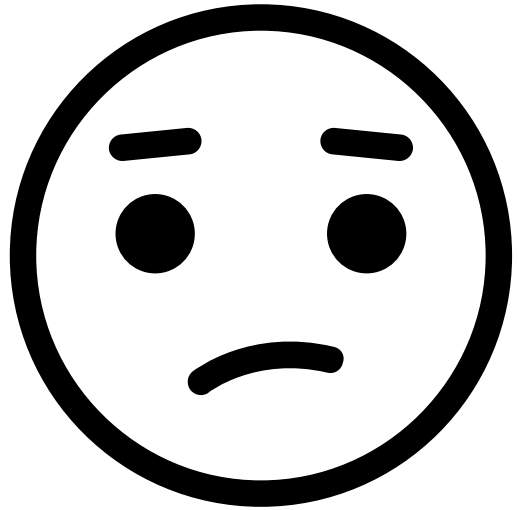
25b309e8=25b309e8

Bob konstaterer, at den beregnede hash-værdi er identisk med den medsendte hash-værdi og tror derfor fejlagtigt, at der ikke er blevet ændret i beskeden!!

[I] integritet (dataintegritet)

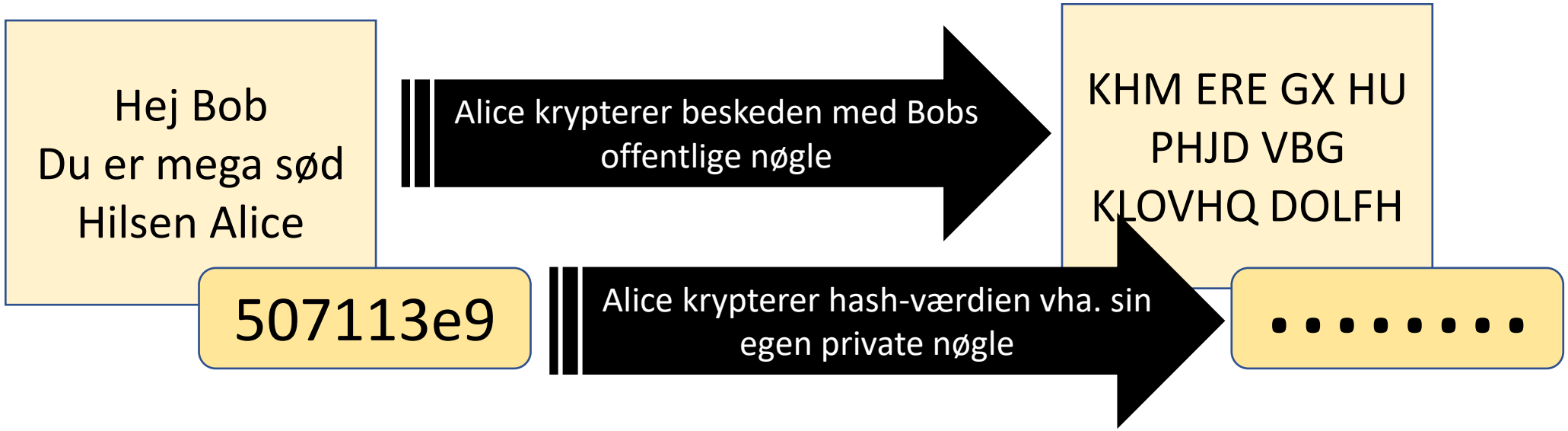
Det vil sige…

- Hashing alene er altså ikke nok til at sikre dataintegritet (beskeden er stadig blevet ændret, uden at Bob har opdaget det!!)



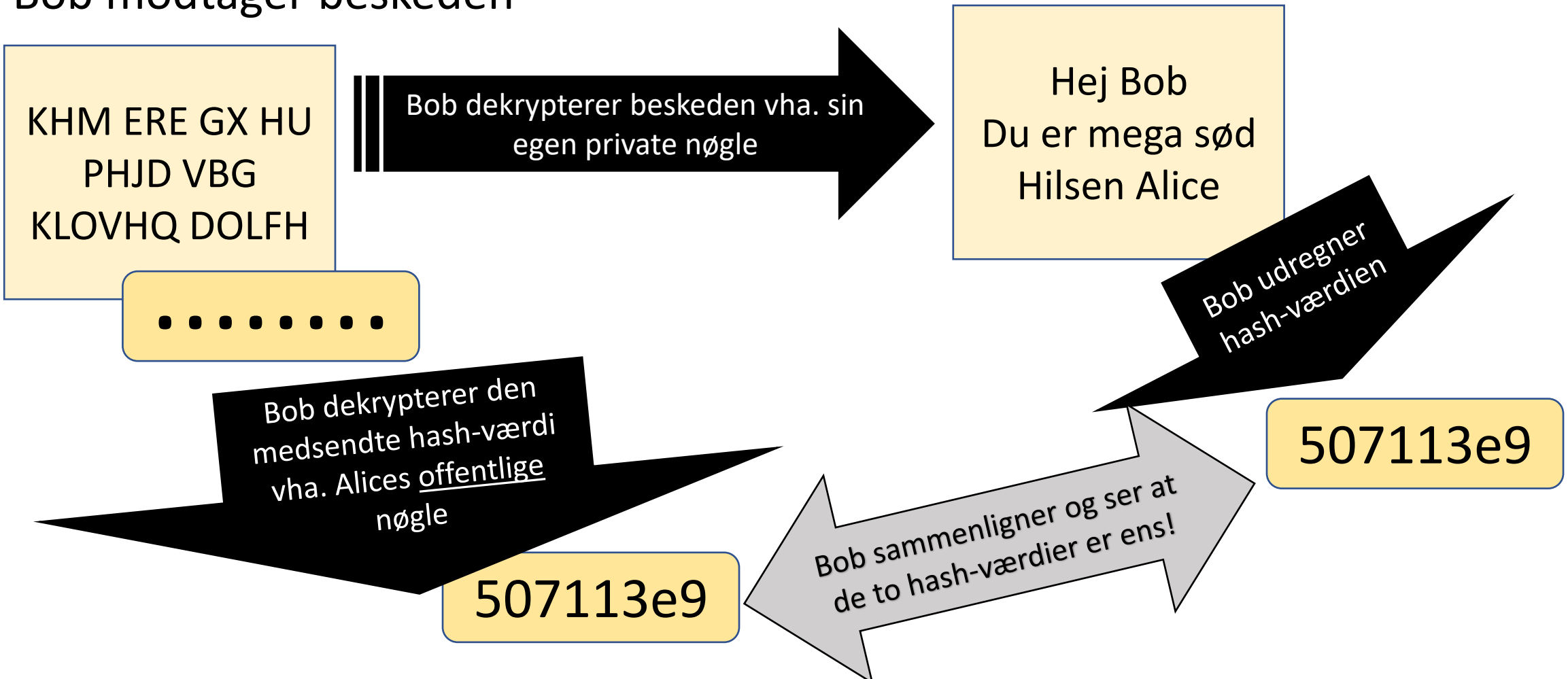
[I] integritet (autenticitet) – digital signatur

- Problemet er, at vi ikke har sikret *autenticitet* – Eva har udgivet sig for at være en anden (nemlig Alice)!
- Løsning: Alice skal kryptere sin hash-værdi vha. sin egen <u>private</u> nøgle! Det kaldes for **DIGITAL SIGNATUR**



[I] integritet (autenticitet) – digital signatur

- Bob modtager beskeden



[I] integritet (autenticitet, dataintegritet, uafhviselighed) – digital signatur

- Fordi det kun er Alices offentlige nøgle, som kan dekryptere hash-værdien, så ved Bob at beskeden kommer fra Alice (AUTENTICITET)
- Samtidig kan Bob konstaterer, at hash-værdierne er ens – derfor ved han at beskeden ikke er blevet ændret (DATAINTEGRITET)
- Alice kan nu heller ikke påstå, at beskeden kommer fra Eva (UAFVISELIGHED)

Øvelse – udfyld skemaet (sæt X)

	[C] – fortrolighed	[I] – dataintegritet	[I] – Autenticitet	[I] – Uafviselighed
Kryptering sikrere…				
Hashing sikrer…				
Digital signatur sikrer…				
Alle tre sikrer…				

Opsamling

	[C] – fortrolighed	[I] – dataintegritet	[I] – Autenticitet	[I] – Uafviselighed
Kryptering sikrere…	X			
Hashing sikrer…		X		
Digital signatur sikrer…		X	X	X
Alle tre sikrer…	X	X	X	X

Logbog

- Forklar med egne ord hvordan fortrolighed og integritet opnås vha. kryptering, hash-funktioner og digital signatur.

[C] *fortrolighed*: Uvedkommende kan ikke opsnappe kommunikationen.

[I] *integritet*:

- *data-integritet*: Indholdet af en besked fra A til B kan ikke ændres under transporten over netværket.
- *autenticitet*: En uvedkommende aktør C kan ikke udgive sig for at være A eller B.
- *uafviselighed*: Hvis A sender B en besked, kan A ikke bagefter nægte det.

[A] *tilgængelighed*: Uvedkommende kan ikke afbryde A's og B's mulighed for at kommunikere.