Oblig 2 - Kryptering og kryptografi

IN2120 - 2023

I denne obligen skal vi jobbe som agentkontakt i Ganske Dårlig Hemmelig Tjeneste, ofte forkortet til GDHT. Fordi den bare er ganske dårlig, og ikke kjempedårlig, så er standard prosedyre å kryptere alle meldingene mellom agentene og agentkontaktene.

Formålet med obligen er å gi et innblikk i hvordan ulike typer kryptering brukes i praksis, slik som nøkkelutveksling med asymentrisk kryptering, obfuskering av passord med hash-funksjoner, og digital signatur for å bekrefte integriteten og avsenderen av for eksempel en melding eller et dokument.

Del 1 - Asymmetrisk kryptering

I asymmetrisk kryptering har vi nøkkelsett bestående av en privat nøkkel og en offentlig nøkkel. Vi kan bruke den private nøkkelen til å **dekryptere chiffertekster** og **generere signaturer** og den private nøkkelen til å **kryptere klartekst** og **verifisere signaturer**. I praksis kan asymmetrisk kryptering for eksempel brukes dersom vi vil sørge for at bare den som har den private nøkkelen kan lese chifferteksten, altså det som er kryptert. Det er altså kjempeviktig at den private nøkkelen holdes hemmelig. Den offentlige nøkkelen kan derimot alle få tak i. Det ble i sin tid foreslått at man kunne skrive de offentlige nøklene i en slags telefonbok, men vi har endt opp med noe som heter PKI (Public key infrastructure), som er et tema senere i kurset.

Men tilbake til jobb. Agenten Alice har sendt deg følgende melding:

```
cSliWYWIOYdFPl9PDCyZ/GU/dNuziGJH7UGV8s2E7i0Ow2mAxr6OrYHRk084Rry4fPEqo9co7UINwYL
zcrEXg+7AI5CZ7LaOvoR5WDTR2lyxh+urrx9wocH81YqYXhMq0x6zVueRmSkVo3FEC1DQNFpdvOndnI
kFdH/crWUu/kg=
```

Ifølge GDHT sin protokoll er denne meldingen kryptert med din offentlige nøkkel. I tillegg har den blitt formatert som Base64. Dette er en protokoll laget for å formatere data som tekst. Med Base64 gjør at man kan sende informasjon eller til og med for eksempel bilder og andre dataformat som tekst, for eksempel i en epost. Nedenfor står nøkkelparet ditt:

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCOZOldy1xaxZJgyEstfY2hbMmB
t7vRqvI2mFCT/QQg7kaiL7ulB9D4J4lr/hzr9zWmsBt9B2PCNLMjDUusEN7sxrGp
V5+INoIHESLK9zDrJveUXuRH47i73xmAXadnc7KgdA6V9Lznh2OiLMYnfUKPqtdp
9l+SIOAO6FiZNcY9FQIDAQAB
-----END PUBLIC KEY-----

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCOZOldy1xaxZJgyEstfY2hbMmBt7vRqvI2mFCT/QQg7kaiL7ul
B9D4J4lr/hzr9zWmsBt9B2PCNLMjDUusEN7sxrGpV5+INoIHESLK9zDrJveUXuRH
47i73xmAXadnc7KgdA6V9Lznh2OiLMYnfUKPqtdp9l+SIOAO6FiZNcY9FQIDAQAB
AoGABErPbAfew0+0T12KQpdHxydn1ZSrwTC1d2bU+vkEAFgeYfItZlabend6oXo5
9fTwNZOWaIjp/WcoRTS35LpnddEto691SBDYX9TsneDBgRwgdmbkYX7bs6qSUzbH
AW+Hp+FbNYcwb1dZ9xJptboveFqNtmYfyV7QG9FzmmvfDCsCQQDEUdN3/wCIseQF
rjIEJ3VYzyuYJg1Dn403sAGPWH77cK/cdl4LUgpNQWLpgOlvPh8m48Et0MB+wYlB
zqsnMMIvAkEAua5yqWYl0vKVj7TE1MslYRnDHhz41Cq8+GfGENKrwOEmTfozlXMW
yq+cvlePlC33m3adE7HO6IfFRSn8o4x3+wJAFOl2cjDKmMUCOgytLT4axOQEi8d5
4DstqsHn2KIfnJ8LTJubNu99Q29OntTpiL0ASn82Q7AZ4SayfT0LhDgwZwJBAKsy
raOEe/FQGxehmy17lB5AC4eTPNUaBpudyp+uLg+rR47z1mW/9E6ytflFcCs2aqqb
JV6IPXZl4DvR546/r8sCQHCr3Vf3/0M2nKSI17+tJ7RLLSUGAGHQfVMPREyiQXv0
IZbk1zYpVJ9TOaaOJZQrYXfhBySx/vzhxXDX6zH7xpc=
-----END RSA PRIVATE KEY-----
```

For å dekryptere meldingen må du

- Gå til cyberchef.io (GDHT kjører selvsagt ikke sitt eget krypteringsverktøy).
- Importere "From Base64" og "RSA Decrypt". RSA er krypteringsalgoritmen som er brukt.

Spørsmål 1

Hva står det i den dekrypterte meldingen som ble sendt?

Del 2 - Symmetrisk kryptering

Symmetrisk kryptering går ut på å bruke samme nøkkel til både kryptering og dekryptering. Alice har nettopp sendt oss en slik delt hemmelig nøkkel ved å bruke asymmetrisk kryptering. Nå ønsker vi å ha en samtale med Alice ved å bruke denne felles nøkkelen som bare vi skal ha. Både vi og Alice vil nå kryptere og dekryptere meldingene til hverandre med denne.

Grunnen til at vi heller vil bruke symmetrisk kryptering kan for eksempel være fordi det er mye raskere og mer effektivt enn asymentrisk kryptering. Det er fint om vi for eksempel skal sende mye data. Idag skal vi ikke sende veldig mye data, men GDHT bruker likevel denne fremgangsmåten til generell kommunikasjon.

IV står for initialiseringsvektor, og er det som gir krypteringsalgoritmen sin initielle tilstand. Den bør egnetlig være tilfeldig eller uforutsigbar, litt avhengig av hvilken algoritme som er i bruk. Den skal nemlig sørge for at samme melding kryptert to ganger i praksis aldri skal være like hverandre. Det er derfor svært tvilsomt av GDHT å bruke initialieringsvektoren som de gjør her.

- **Algoritme brukt:** AES
- **Modus:** CFB
- **IV:** 00000000000000000000000000000000

- **Hemmelig melding:**

```
e64655e95d680272623ee1aae13185cb3d435eaed9a58808d28b45ef586d07a
52f38f5fc0bd48472e9e458cab8af6c7f4c07d64e238b5d9c63d1eda4bbe6ca
09e59925998caca552ac1bbdf19d9a4005efad031557db9201db124d3ad4f41
1b55fc518a258fb9e0aacc26f1d2b5eedce0395ed3a263fe60c85a0133a22fe
21d032aa8930684eac261cf3c6d12d6057bc59c1fbb90440af4012e56584bdb
de4cbc9f624b749243014e5d089f645dede743197c47b8a14e21afcf60e5bd6
563c4ea08a5567a695e634c6b8e9c0db2136e7b1551773d90a57eb17a53afe8
588456147716b6b73d2d7fb14735a119e4c4e1dc48915ed0514a63084826612
49f943a4143694de3572aac0f67c0cd60064dc26c221a1904e55b85a6048d0e
408b9bf0c5d7222827f4840aa4b7ac1c980c33e78f91bd29120b1f93e9f5d
```

Spørsmål 2

Ved å bruke Cyberchef igjen, finn ut: Hva heter disponenten som Alice spionerer på?

Del 3 - Hashfunksjoner

Alice har altså fått tak i en passordhash. Å beregne hash-verdien til passord med en hashfunksjon er en vanlig måte å skjule eller obfuskere dem på når man lagrer dem. Formålet med slike hashfunksjoner er at det skal være praktisk umulig å reversere de. Altså kan vi finne hashen med et passord, for å sjekke at det er riktig. Men vi skal ikke kunne finne passordet dersom vi bare har hashen.

Dersom disponenten hadde hatt et kort passord kunne vi brute-forcet det. Det vil si å prøve alle kombinasjoner av tall, store og små bokstaver og symboler og se hvem som matchet med hashen. Passord på opptil ni tegn ville da er ansett for å være mulig å knekke på under tre dager. Dessverre er ryktet av passordet er langt. Det blir raskt vanskeligere. Forskjellen er så stor, at hvis passordet bare er på 12 tegn, vil det kunne ta oss to tusen år å brute-force.

Men, kanskje vi har flaks. Kanskje disponenten har misforstått passordsikkerhet og i god tro valgt et langt passord som samtidig er et ord?

Gå til:
crackstation.net

(GDHT kjører ikke sitt eget passordcrackings-verktøy.)

Dette er en nettside som har hasher av alle ord på hele wikipedia, og som lynraskt kan sjekke en hash mot denne databasen. Sjekk om disponenten har gjort en kardinalfeil og valgt et passord som finnes i databasen.

Spørsmål 3

Hva er passordet til disponenten?

Del 4 - Digital signatur

Nå sier Alice at hun ønsker å møte kontakten sin. Litt stress, men la gå, tenker du. Men så slår det deg: Hvordan kan du forsikre deg om at du ikke går rett i en felle, slik som på film? Du innser at du faktisk ikke kan være sikker på at det faktisk er Alice du har kommunisert med hele arbeidsdagen.

Alice brukte jo din offentlige nøkkel for å opprette kontakt og utveklse en hemmelig nøkkel, men det kan jo hvem som helst ha gjort, denne nøkkelen er jo offentlig. For Alice er saken anneledes, hun kan jo være sikker på at du er deg, siden du kunne lese den første meldingen hennes. Det kan jo bare innehaveren av den private nøkkelen din kunne gjøre.

Lettet over at du kom på dette i tide selv om du jobber i Ganske Dårlig Hemmelig Tjeneste ber du Alice sende tidspunkt og sted dere skal møtes, men du ber henne også signere meldingen med digital signatur. Digital signatur er en metode for å bekrefte innholdet OG avsenderen av en melding eller av data. Det fungerer slik at vi sender meldingen/dataen og signaturen separat. Signaturen er den beregnede hash-verdien av meldingen, kryptert med Alice sin private nøkkel. Fordi nøkkelpar med en offentlig og en privat nøkkel fungerer begge veier, kan nå i teorien alle dekryptere denne meldingen. Poenget er at viss vi kan dekryptere meldingens hashverdi med Alice sin offentlige nøkkel, er det KUN Alice med sin private nøkkel som kan ha kryptert den. I tillegg vil den beregnede hash-verdien av meldingen være unik for den orginale meldingen, så vi kan være sikker på at Alice har signert AKKURAT denne meldingen, og ikke en hvilken som helst annen.

Som du ser nedefor har Alice signert en forholdsvis kort melding. Signaturen er faktisk lenger enn selve meldinga, som er uvanlig. En stor del av poenget med digital signatur er at vi kan beregne unike hash-verdier for små og store mengder data slik at vi for eksempel kan signere hele dataprogrammer eller store og små datafiler på samme måte.

Alice sin public key:

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3uw8JQ0LvIlWUjWg/LwdEPo7+
d8EZ3hzN5b0PbN/zR/wYsAlMGHN3HE4KwhM+wNlmFtmIccKmjpdSQg/4Sk7FOrkY
SB1NBPcQFHHssA92m2dcu6YQQDrMZEy/UowgOhEogI1AhvbVV0TX+04Sn9Q1kNF2
cTl3k9HcKFQ+oLhGeQIDAQAB
-----END PUBLIC KEY-----
```

Melding:

```
La oss møtes på trikkestoppet bak torget klokka 12.
```

Signatur (SHA-256):

```
ReZF29RwQ8/WKXqQDiNm23nVo+vWXV2h8Nalxj/AZZX9geyZr6SV2ta4lPa0Hcfd5wdND8aTETp2
vTm2IbQohQlcGxxOTtKTzwrs6qnB/wzg1zlfgJNm2TdHcP3abYlLVce9ovody4Ean8wEkG8VztH6
xxo6ZpZveWBTYX7ssAk=
```

Bruk modulen RSA Verify på cyberchef.io til å sjekke om det faktisk er Alice som har sendt meldingen. Legg merke til at signaturen er Base64-formatert!

Spørsmål 4

Hva står det i output-feltet om det viser seg at det er Alice som har signert og sendt meldingen?
Spoiler alert: Det er Alice som har signert den.

Del 5 - Liten oppsummering

Dagen etter, har du fått følgende to meldinger fra Alice:

```
DPrrSPlvdVrqYAGB/4XKzpoSUTa9k2UnuwewOVbngrqjaiKQ3UCgYUUCcGid7DPYvOh+OAs
q00WbkOtnEauBSnK5km88RmEdxCaASNnF1aOPaYoptgHEAwkrSgOp5Hx+kpqt7n6C9vuszk
yx7D7ULjD4nb3XmdVJ//mXbuUH7+M=
```

```
6add596f75799b4dc618808a4bef3ffd2ab72dc4d59bf60f96e77c0af351fea512883e826
e3cf0d5a6e8538950c5e38bdb5c72ede60542f4a14af7b53eddad1f256dbfb183e2e66129
c8db5fae0d9bc07fca3c8d6f78c3efdb38976d0acd547688b0c7567dca29a4fe4d67609ec
d4922bafbff2ff869b93a9e05c39f09e2aa7183bde1b7cf914c334c5a9314aec2240e904d
ec963f74a64418a0aa8f5c9ba1ccb1cf6daa42668e20c302767e9e8db8e564
```

Spørsmål 5

Hva er etternavnet som den andre meldinga er signert med?