

Oblig 3 - Nøkkelhåndtering og PKI

IN2120 - Høst 2023

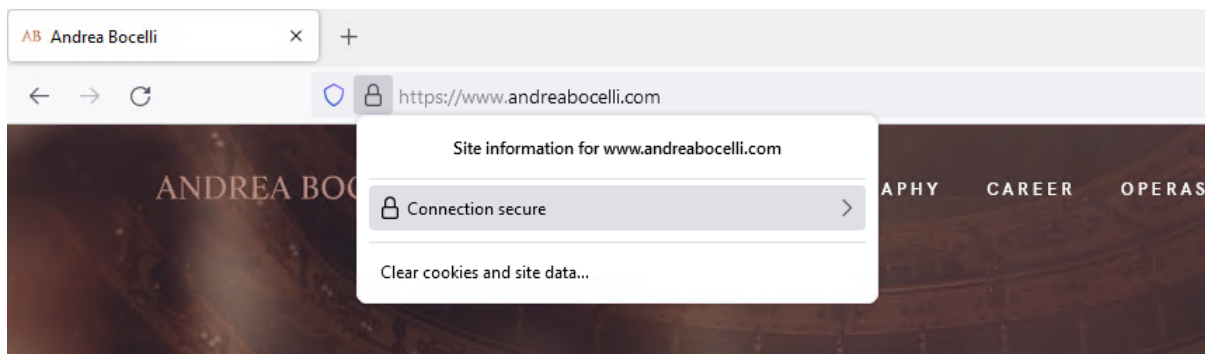
Introduksjon

Vi har tidligere lært at det er en vesentlig forskjell på symmetrisk og asymmetrisk kryptering. Ved symmetrisk kryptering benytter vi en nøkkel, den samme til både kryptering og dekryptering. Ved asymmetrisk kryptering jobber vi med nøkkelpar, en privat nøkkel og en offentlig nøkkel. Vi kan bruke den private nøkkelen til å **dekryptere chiffterekster** og **generere signaturer** og den offentlige nøkkelen til å **kryptere klartekst** og **verifisere signaturer**. Det er veldig essensielt at den private nøkkelen holdes privat, mens den offentlige nøkkelen er offentlig.

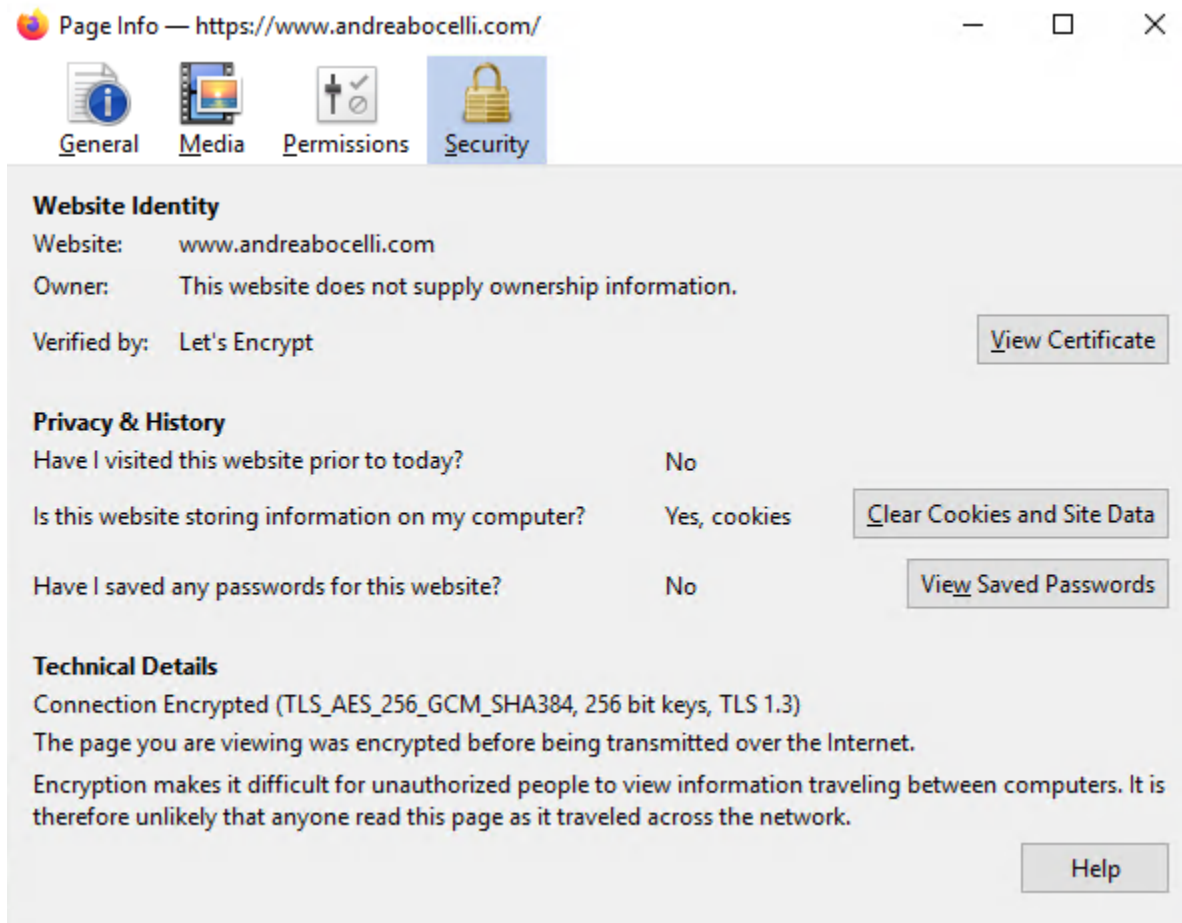
Som dere kanskje merket da dere jobbet med oppgavene rundt asymmetrisk kryptering så kan det være vanskelig å holde tunga rett i munn når vi jobber med private nøkler, offentlige nøkler, hvilken nøkkel skal brukes til å kryptere, hvilken skal brukes til signere, hvilket nøkkelpar tilhører denne nøkkelen osv. Dersom vi hadde skalert opp det eksempelet vi jobbet med fra 4 nøkkelpar til flere tusen nøkkelpar fordelt på individer, maskiner og applikasjoner så forstår dere at det er viktig å ha gode systemer som holder styr på dette. Derfor tar man i bruk PKI, Public Key Infrastructure, som et rammeverk bestående av diverse hardware, software, prosedyrer og regelverk, for å opprette, administrere, distribuere og oppbevare nøkler og sertifikater. Okei, les den siste setningen en eller to ganger til.

For å forstå mer hva PKI er, la oss starte med et eksempel på en veldig sentral del av PKI, nemlig digitale sertifikater. Digitale sertifikater binder offentlige nøkler til en gitt ID. Deretter kan disse digitale sertifikatene brukes for å verifisere identiteten til ulike parter som ønsker å kommunisere innenfor et økosystem. Du kan tenke på et digitalt sertifikat som et digitalt pass. Pass utstedes av Politiet, digitale sertifikater utstedes av CAs, Certificate Authorities. Dette er et system som er bygd på tillit, på samme måte som vi stoler på Politiet som en tredjepart utsteder pass i henhold til et gitt reglement (Lov om pass), stoler vi på CAs som en tredjepart til å utstede digitale sertifikater i henhold til et gitt reglement (X.509). Etter du har fått et pass av Politiet så kan du fly fra land til land fordi grensekontrollene i andre land stoler på at Politiet i Norge følger reglementet rundt passutsedelse. På samme måte kan du etter å ha fått tildelt et digitalt sertifikat, trygt kommunisere med andre parter innenfor et digitalt økosystem.

Okei nå trenger vi et hands-on eksempel her. Andrea Bocelli er en legendarisk operasanger. Gå inn på andreabocelli.com, trykk på hengelåsen, trykk deretter på "Connection secure" og "More information". Du trenger ikke å gjøre dette i din Windows-VM, slik det er demonstrert under. Dersom du bruker Mac vil du få en mye mer intuitiv oversikt ved å bruke safari (Trykk på hengelås og "Show Certificate").



Under *Website Identity* ser vi at nettsiden er verifisert av *Let's Encrypt*. Vi ser forøvrig også under *Technical Details* at den algoritmen vi jobbet med i starten av forrige oblig, AES-256, er en del av protokollen for å sikre forbindelsen vår. Trykk deg videre til *View Certificate*.



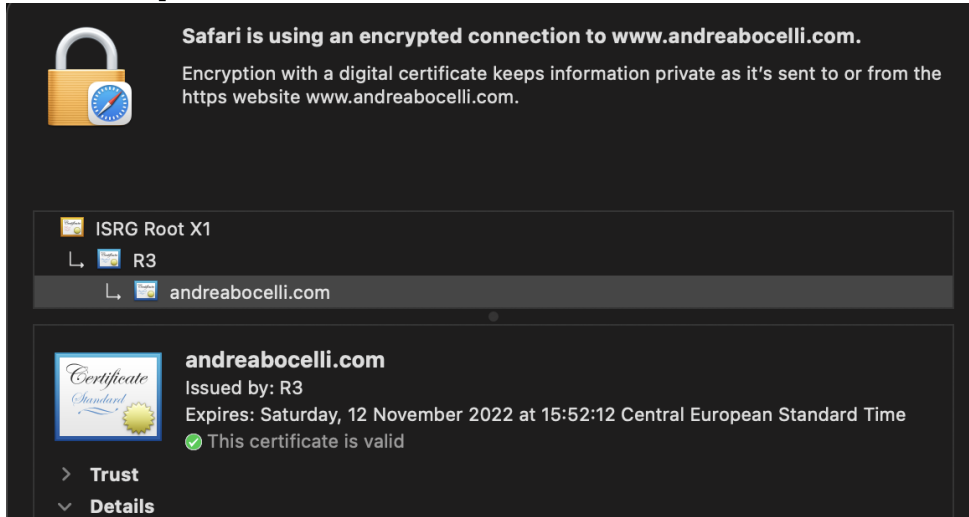
Her ser vi (i bildet under) at R3 (Let's Encrypt), er CA til andreabocelli.com.

Spørsmål 1

Finn ut hvem som er CA (Certificate Authority) for findtheinvisiblecow.com. Skriv inn svaret i svar.in2120.uiocloud.no.

Det er altså R3 som har sagt "vi går god for andreabocelli.com". Men hvem går god for R3? R3 er en CA, så hvem går god CAs?. På Windows ser vi at det står ISRG Root X1 ved siden av R3. Dette er en Root CA, en CA som verifiserer andre CAs. Dersom du bruker Safari på Mac kommer denne strukturen tydelig frem.

Slik det ut på Mac:



Slik det ut på Windows:
Certificate

andreabocelli.com	R3	ISRG Root X1
Subject Name		
Common Name	andreabocelli.com	
Issuer Name		
Country	US	
Organization	Let's Encrypt	
Common Name	R3	
Validity		
Not Before	Sun, 14 Aug 2022 14:52:13 GMT	
Not After	Sat, 12 Nov 2022 14:52:12 GMT	
Subject Alt Names		
DNS Name	andreabocelli.com	
DNS Name	www.andreabocelli.com	
Public Key Info		
Algorithm	RSA	
Key Size	2048	

Spørsmål 2

Finn ut hvem som er root CA til nba.com. I hvilket år utløper deres sertifikat?

Vi kan også bruke kommandolinjeverktøyet til OpenSSL for å innhente slik informasjon. Åpne en Cygwin-terminal.

Vi kan kjøre følgende kommando for å observere sertifikat-kjeden til vg.no (Husk at port 443 er standard port for HTTPS):

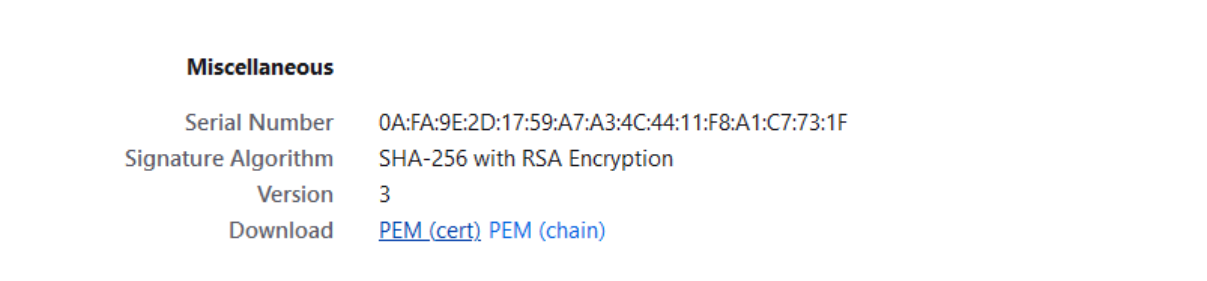
```
openssl s_client -showcerts -servername www.vg.no -connect www.vg.no:443
```

Scroll deg gjennom dataen du får opp.

Spørsmål 3

Hvor mange bits er serveren sin public key?

Bruk FireFox på labmaskinen din. Last ned sertifikatet til både NRK.no og VG.no (som du finner ved å navigere deg til sertifikat-siden som vi gjorde tidligere)

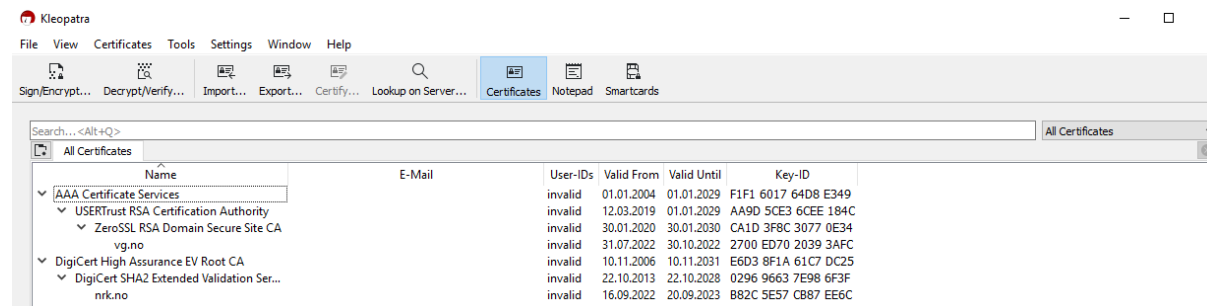


The screenshot shows the 'Miscellaneous' tab in the Firefox Certificate Manager. It displays the following information:

- Serial Number:** 0A:FA:9E:2D:17:59:A7:A3:4C:44:11:F8:A1:C7:73:1F
- Signature Algorithm:** SHA-256 with RSA Encryption
- Version:** 3
- Download:** [PEM \(cert\)](#) [PEM \(chain\)](#)

Når du åpner disse filene vil maskinen spørre om du ønsker å åpne dem i Kleopatra, det ønsker du.

Kleopatra er en certificate manager som vi kan bruke for å opprette, lagre, fornye og opprette både private og offentlige X.509 sertifikater og nøkler. Det vil se slik ut dersom du har importert sertifikatene riktig:



The screenshot shows the Kleopatra application window. The 'Certificates' tab is active, displaying a list of certificates. The list has columns for Name, E-Mail, User-IDs, Valid From, Valid Until, and Key-ID. The certificates listed are:

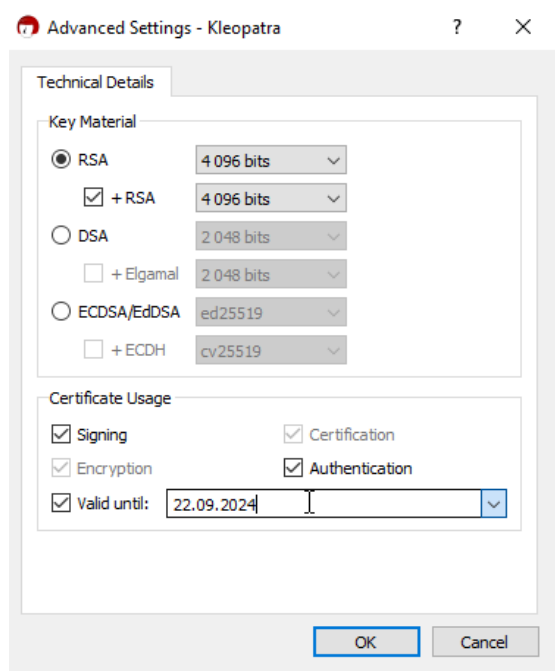
Name	E-Mail	User-IDs	Valid From	Valid Until	Key-ID
AAA Certificate Services		invalid	01.01.2004	01.01.2029	F1F1 6017 64D8 E349
USERTrust RSA Certification Authority		invalid	12.03.2019	01.01.2029	AA9D 5CE3 6CEE 184C
ZeroSSL RSA Domain Secure Site CA		invalid	30.01.2020	30.01.2030	CA1D 3F8C 3077 0E34
vg.no		invalid	31.07.2022	30.10.2022	2700 ED70 2039 3AFC
DigiCert High Assurance EV Root CA		invalid	10.11.2006	10.11.2031	E6D3 8F1A 61C7 DC25
DigiCert SHA2 Extended Validation Ser...		invalid	22.10.2013	22.10.2028	0296 9663 7E98 6F3F
nrk.no		invalid	16.09.2022	20.09.2023	B82C 5E57 CB87 EE6C

Spørsmål 4

Venstreklikk på begge Root-CA'ene og velg "Trust Root Certificate". Hva står det nå under User-IDs?

Nå skal vi bruke Kleopatra til å signere og verifisere sertifikat.

Start med å opprette 2 nye nøkkelpar, et av gangen. Trykk på File - New Key Pair. Vi ønsker et "Personal OpenPGP Key Pair". Skriv inn Alice. Deretter - Advanced settings. Velg 4096 bits RSA og huk av for Authentication::



Huk av for passord og velg et passord. Når du oppretter dette nøkkelparet kan det ta litt tid. Dersom du taster på tastaturet, beveger musa, klikker rundt så går det fortore fordi den bruker og trenger tilfeldig generert input. Når du er ferdig, lag et nytt nøkkelpar med Bob som navn slik at du har 2 nøkkelpar.

Lag en ny tekstfil som vi kan signere og kryptere, og kall den for eksempel testfil.txt. Trykk deretter på Sign/Encrypt øverst til venstre. Velg testfil.txt. Under "Prove authenticity" ønsker du å "sign as Alice". Under "Encrypt" ønsker du å kryptere for både Alice og for Bob. Skriv inn passordet til Alice og lagre den signerte filen. Trykk nå på Decrypt/Verify i verktøylinjen øverst. Velg den filen som Alice nettopp signerte (Den skal hete testfil.txt.gpg dersom du ikke endret noe). Dersom du har gjort alt riktig vil du nå få opp en grønn melding om at file2.txt.gpg inneholder en gyldig signatur fra Alice!

Spørsmål 5

Mens du har denne gyldige signaturen oppe, trykk på "Show Audit Log" øverst til høyre i den grønne boksen. Helt på slutten av den nederste linjen står det noe inne i [klammer], hva står det inne i klammene?