## SSL-handskakning

**Förutsättningar**

Klienten har en keystore med sin egen nyckel för att kunna autentisera sig. De har också en truststore med serverns certifikat.

Servern har en keystore med sin egen nyckel för att kunna autentisera sig. Den har också en truststore med alla klienters certifikat.

Klienten tar kontakt med servern  
Servern autentiserar sig för klienten  
Servern ber klienten autentisera sig  
Klienten kontrollerar serverns certifikat  
Klienten skickar sitt certifikat till servern  
Klienten skapar(?) en symmetrisk nyckel som ska användas under informationsöverföringen och krypterar den med serverns publika nyckel.  
Servern kontrollerar att klientens certifikat finns i truststore:n och noterar vilka accessrättigheter klienten har(?)   
Servern tar emot nyckeln, dekrypterar den med sin privata nyckel och byter till krypterat läge.  
Klienten byter till krypterat läge  
Krypterad data skickas mellan parterna

## Attackmetoder

**MITM**Ett farligt scenario är om data som skickas från klient till server kan stoppas på vägen av illvillig part och använder denna information för att ”ta över” uppkopplingen mot servern. Förutsatt att detta lyckas styr MITM all kommunikation mellan server och klient och får således tillgång till all sekretessbelagd information som skickas.   
Vårt system avhjälper detta genom att klient/server att först autentiserar sig mot varandra. Med hjälp av certifikaten i respektive truststore blir aldrig MITM ett problem då denne inte kan mer än vidarebefordra dessa meddelanden. Han sitter inte inne på klientens privata nyckel och kan därför inte skapa en egen session mot servern. Detta innebär också att han inte heller har möjlighet att få tag i den symmetriska nyckeln som krypterar överföringen av patientjournalerna.

Med andra ord, när överförandet av den känsliga informationen börjar har MITM ingen möjlighet att ta del av den, även om han får tag i alla paket. Han har inte heller möjlighet att göra egna uttag från registret eftersom han saknar rätt nyckel för att kunna kommunicera med servern.

**Spoofing**  
Systemet är känsligt för spoofing-attacker, i och med att alla keystores integritet är vital för att stänga ute obehöriga. En falsk klient som samlar in information om en användares keystore skulle få tillgång till nyckelpar & certifikat, vilket skulle medföra att känslig information skulle hamna i orätta händer. Detta kan motverkas genom att utbilda användarna i vad de bör kontrollera för att verifiera klientens äkthet, innan de börjar använda klienten.

Att begränsa nedladdnings-/installationsmöjligheterna på samtliga datorer där klienten finns installerad förhindra också att någon obehörig får tillgång till journaldatabasen. Detta kan dessvärre vara svårt att implementera eftersom det kan finnas verksamhet som gör att man inte kan genomföra den typen av restriktioner.

**Avlyssnad kommunikation**  
Det finns alltid en risk för att paket innanhållande journalinformation snappas upp av någon som avlyssnar kommunikationslinjen mellan klient och server. Eftersom dessa paket är krypterade med en svårforcerad symetrisk kryptering är det rimligt att anta att man med dagens datorkraft inte kan forcera krypteringsskyddet inom rimlig tid.  
Det går däremot inte att skydda paket mot offline-attacker i all evighet, vilket innebär att någon med datorkraft och tid nog förr eller senare kommer åt journalinformationen oavsett hur svårforcerad krypteringsalgoritmen är. Även de starkaste symmetriska krypteringsalgoritmerna som inkluderas i mjukvaran, så som AES, kommer med stor säkerhet gå att forcera i framtiden. Detta eftersom beräkningskapaciteten för den här typen av operationer ökar i samband med att ny datorteknik görs tillgänglig.

**Falska certifikat**  
En icke-auktoriserad användare skulle kunna försöka skapa ett förfalskat certifikat och med detta försöka få tillgång till materialet på servern. Emellertid skulle detta inte vara signerat av TTP-certifikatet och således kunna uppdagas som falskt. Systemet utestänger alltså alla som inte fått ett certifikat utfärdat, vilket gör att man inte kan penetrera systemet utan ett giltigt certifikat.

Enda möjligheten att lyckas med ovan nämnda attack är om TTP skulle hackas och utfärda certifikat till obehöriga.

**Lösenordspolicy**  
En strikt lösenordspolicy är nödvändig för att kunna undgå vanliga typer av attacker för att forcera lösenord. Lösenordet bör vara minst 12 tecken långt, innehålla varierande teckentyper och inte bestå av ord. En sådan policy gör det förvisso svårare för användarna att minnas sina lösenord men har å andra sidan säkrat systemet mot exempelvis Brute force och Dictionary-attacker.

Användarna måste även byta lösenord på regelbunden basis och systemet bör logga misslyckade inloggningsförsök.  
  
**Mänskliga faktorer**  
Innan systemet tas i bruk måste samtliga användare informeras om vikten av att inte lämna oavslutade sessioner igång då detta kan leda till att obehöriga får tag i/ändrar journalinformation.  
Utöver detta måste det finnas restriktioner mot att på spara och hantera journalinformation digitalt utanför klienten, eftersom okrypterade lokala kopior av informationen utgör en säkerhetsrisk.

Se till att vanliga användare inte kan ändra på säkerhetspolicy/lägga till certifikat. Ha minimumkrav för egenskapade lösenord.

**MITB?  
Attack mot journaldatabasen?  
Ladda ned och sprida journaler utanför den säkra uppkopplingen?  
Elektromagnetisk avläsning av knapptryckningar?  
Attacker mot en enskild mot keystore för att skapa nytt certifikat?  
Överbelasta servern, DDOS eller motsvarande?  
Möjlighet att förneka att man utfört en viss handling i en journal?**

Frågeställningar  
Vem får skapa certifikat/ändra accessnivåer, kan detta missbrukas?  
Journal som tas bort av misstag eller av illvilja, kan den återställas?   
Finns det någon möjlighet till SQL-injektion?