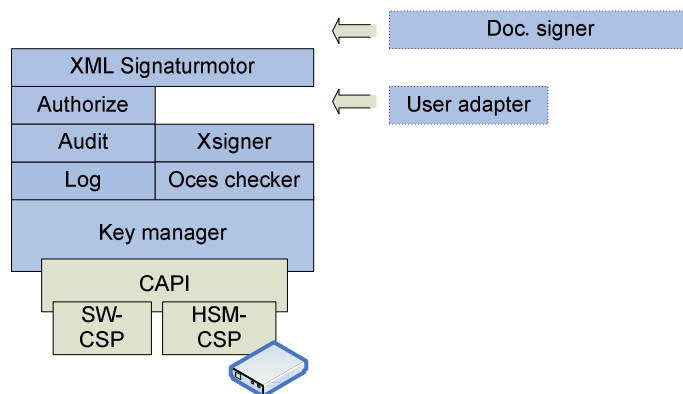


Whitepaper: Signaturgruppens e-tinglysnings motor

1	INTRODUKTION	2
2	MODULER I E-TINGLYSNINGS MOTOREN	2
2.1	KEY MANAGER	2
2.2	XSIGNER	3
2.3	OCES STATUS CHECKER	3
2.4	LOG	3
2.5	AUDIT	4
2.6	AUTHORIZE	4
2.7	XML SIGNATURMOTOR	5
3	EKSTRA MODULER TIL E-TINGLYSNINGS MOTOREN	5
3.1	DOC. SIGNER	6
3.2	USER ADAPTER	7
4	RÅDGIVNING	7

1 Introduktion

Dette dokument beskriver Signaturgruppens e-tinglysnings motor, som implementerer basal XML-DSig funktionalitet til brug for integration mod Realkreditnettets e-FPI samt Domstolsstyrelsen e-TL systemer til elektronisk tinglysning. Løsningen er målrettet imod e-tinglysnings anmelderordning med anvendelse af virksomhedssignaturer beskyttet af hardware security module (HSM).



Figur: Komponenter i e-tinglysnings motoren.

e-tinglysnings motoren leveres i en Java version og i en .Net 2.0 version. Nedenstående metodesignaturer følger Java syntaks og konventioner, .Net udgaven er stort set identisk.

2 Moduler i e-tinglysnings motoren

2.1 Key manager

Dette modul implementerer den direkte anvendelse af kryptografiske operationer. Desuden faciliteres udstedelse og fornyelse af TDC virksomheds- og medarbejder certifikater. Key manager komponenten implementerer ingen kryptografiske algoritmer selv, men anvender standard biblioteker fra HSM producenterne til dette.

Key manager stiller følgende funktionalitet til rådighed:

- `CredentialPair getCredentialPair(String serialNumber, String csp)`

Giver adgang til privat nøgle/certifikat par, på basis af et subject serial number. Er der adgang til flere certifikater med samme serial number, vælges det senest udstedte. Subject serial number bevares ved fornyelse af certifikatet.

- `void renewCredentialPair(String serialNumber, String csp)`

Danner nyt RSA nøgle par (i den konfigurerede CSP), og udsteder et nyt certifikat fra TDC via HTTPS webservice på basis af autentifikation med det gamle certifikat. Denne funktion kaldes f.eks. fra et kommandolinie-værktøj for at få fornyet et VOCES certifikat tæt på udløb.

- `String issueCredentialPair(String refNumber, String instCode, String csp)`

Initialiserer systemet til at få udstedt TDC VOCES certifikater. Der dannes nyt RSA nøgle par (i den konfigurerede CSP), og der skabes en certifikat request struktur (PKCS#10). Derpå kontaktes en

TDC HTTPS webservice for at få udstedt certifikat på baggrund af det referencenummer og den installationskode som udsendes fra TDC ved bestilling. Metoden returnerer subject serial number for det udstedte certifikat.

- `int getRemainingLifetime(String serialNumber, String csp)`

Returner tilbageværende levetid for seneste certifikat i dage. Kan f.eks. benyttes til overvågning for udløb.

Der leveres desuden et lille kommandolinie værktøj, hvorigennem der gives adgang til `renewCredentialPair`, `issueCredentialPair` samt `getRemainingLifetime`. Hermed kan der forholdsvis enkelt skabes et administrativt værktøj til overvågning og semi-automatiseret fornyelse af certifikater.

Key manager komponenten benyttes af delkomponenterne OCES services adapter og Xsigner.

2.2 Xsigner

Dette modul faciliterer håndtag til XML signering. Modulet adresserer de af CSC anbefalede valg, og generer XML i overensstemmelse med disse, og bibringer således et stærkt forenklet XML DSig API.

- `SignatureDocument sign(Document xml, List idsToSign, String signerSN, String csp)`

Skaber en digital signatur af et xml dokument. Der skabes en samlet signatur over de id'er, der medsendes i `idsToSign`. `SignatureDocument` indeholder et XML `ds:Signature` element. Denne metodesignatur afspejler, at CSC forventes at specificere brug af detached XML signatures i eTL.

- `ValidationResult validate(Document xml, List<CertificationAuthority> trustedAuthorities)`

Validerer et xml dokument. Validering af de certifikater, der er brugt til signering delegeres til OCES status checker. `ValidationResult` indeholder information om alle signere, status af hver enkelt signatur, samt det overordnede resultat af valideringen. Se under OCES status checker, for beskrivelse af `trustedAuthorities` argumentet.

2.3 OCES status checker

Dette modul stiller funktionalitet til rådighed, der tillader caller at validere et OCES certifikat (POCES, MOCES eller VOCES). I konfigurationen indstilles hvilken metode, der ønskes benyttet til status checket; Fuld spærreliste, Partiel spærreliste eller OCSP (online certificate status protocol), samt hvilken/hvilke CA, der ønskes understøttet.

- `Status checkStatus(OCESCertificate certificate, List<CertificationAuthority> trustedAuthorities)`

Returnerer status, enten VALID, EXPIRED, NOTYETVALID, REVOKED eller UNTRUSTEDISSUER. `TrustedAuthorities` indeholder en liste over udstedere af certifikater, der stoles på. `CertificationAuthority` repræsenterer en CA, og eksponerer bl.a. funktionalitet til at foretages spæringscheck af certifikater.

`OCESCertificate` er en convenience klasse, der enkapsulerer det rå certifikat, og stiller metoder som `isPOCES`, `isMOCES`, `isVOCES`, `getCVR`, `getEmail` m.m til rådighed, og således forenkler udtræk af OCES specifikke oplysninger af certifikatet.

2.4 Log

Dette modul implementerer logning. Alle komponenter logger til log modulet. Log modulet vedligeholder

følgende forskellige logs:

- Event log – System log for tekniske informationer og detaljeret logning.
- Audit log – Revisions log for sikkerhedsrelaterede hændelser

Log interfacet kan substitueres med brugerens egen implementation, hvis f.eks. et særligt format ønskes. En default implementation der skriver direkte til filer medfølger. I .Net udgaven medfølger desuden en implementation der logger til MS eventlog.

I .Net versionen anvendes log4net. I Java versionen anvendes Apache commons logging.

Audit log er sikret imod uautoriseret ændring ved anvendelse af hashed message authentication code (HMAC) integritets beskyttelse. Nøgle til HMAC opbevares i HSM.

2.5 *Audit*

Dette modul implementerer en grænseflade til audit loggen og understøtter udtrækning af forskellige foruddefinerede rapporter over sikkerhedsmæssige hændelser i et givent tidsrum. Audit modulet kan skrive rapport direkte til fil eller kan aflevere data til anden forretningslogik via bruger-suppleret interface for sammenbygning af audit information fra flere systemer.

Der logges på tre niveauer, LOG, EVENT og ALARM. Audit loggens integritet sikres med en Hashed-based Message Authentication Code (HMAC), hvor den private HMAC nøgle holdes i hardware.

Logging modulet består af to interfaces, AuditLogger, samt AuditLogPersister. AuditLogger interfacet implementeres i audit modulet, og stiller funktionalitet til rådighed for at skabe nye audit log entries samt for udtræk af bestemte hændelser indenfor givne tidsrum.

AuditLogPersister interfacet definerer funktionalitet, der skal forefindes for at kunne persistere audit log. Bruger kan vælge selv at implementere dette interface. Der medfølger en implementation til persistering af audit log til filer, samt en implementation af en simpel database-persistering.

2.6 *Authorize*

Dette modul fungerer som adgangskontrol inden virksomhedssignatur anvendes til en signering. Authorize kan konfigureres til at understøtte et antal "authorizers" som er eksterne autorisationskilder, som skal afgøre om en bruger må gennemføre en digital signatur med den private nøgle hørende til et virksomhedscertifikat.

For hver authorizer, skal kunden implementere et simpelt Java- eller .Net interface authorize, som specificeret herunder:

- `String getName()`
Returnerer navnet på denne authorizer.
- `Status authorize(String userId, Operation operation, String vocesSN, Element xml)`
Returnerer AUTHORIZED, REJECTED, USERUNKNOWN.
UserId: Streng som genkendes som en bruger nøgle i den anvendte authorizer, f.eks. RACF.
Operation: Kan være enten Sign eller Decrypt (hvis dette modul er tilvalgt) og angiver den ønskede operation af nøglen hørende til det pågældende certifikat.
VocesSN: Subject serial number for det virksomhedscertifikat, der ønskes benyttet til operationen.
Xml: Det XML element, operationen ønskes udført på.

- `Flag getFlag()`

Returnerer flag, der indikerer, hvordan den overordnede autorisationsprocess fungerer, hvis flere authorizers er konfigureret. Kan være REQUIRED enten SUFFICIENT.

Authorize modulet eksponerer følgende metode:

- `boolean isAuthorized(String userId, Operation operation, String vocesSerialNumber, Element xml)`

Metoden returnerer den overordnede autorisationsstatus afhængigt af svarene fra de konfigurerede authorizers.

2.7 XML Signatormotor

Dette modul implementerer en høj niveau tilgang til XML-DSig funktioner til elektronisk tinglysning. Anvendelse af denne komponent frem for direkte at anvende funktionaliteten på komponent-niveau sikrer korrekt autorisation, auditerbar logning af denne samt korrekt audit-spor af alle valideringer.

- `SignatureEngine(User user, Configuration configuration)`

Skaber en ny SignatureEngine instans.

User repræsenterer den bruger, der benytter systemet, og benyttes til autentifikation samt audit logging af efterfølgende aktivitet.

Configuration objektet definerer konfigurationsmæssige detaljer såsom understøttede certifikat-udstedere (f.eks. TDC OCES, TDC OCES Systemtest), ønsket mekanisme til certifikat status-check, foretrukken signaturnøgle, implementationer af SecurityWorld interfacet osv.

Desuden følgende forenkede metoder til XML-Dsig håndtering:

- `SignatureDocument sign(Document xml, List idsToSign)`

Signerer et xml dokument.

- `ValidationResult validate(Document xml)`

validerer et xml dokument.

3 Ekstra moduler til e-tinglysnings motoren

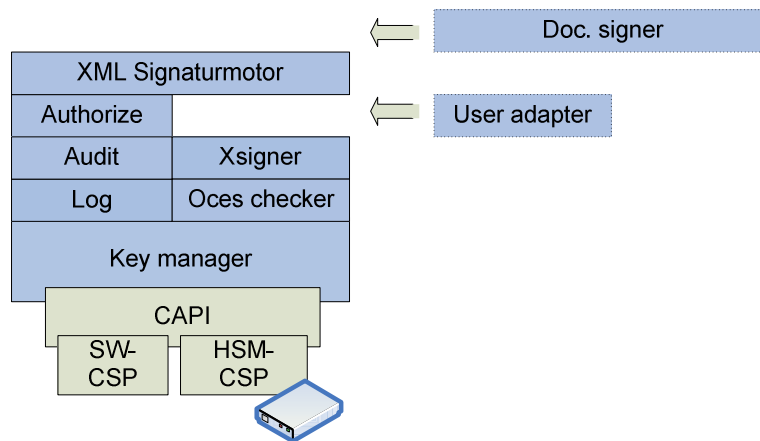


Fig: Tilvalgsmoduler til e-tinglysnings motoren

3.1 Doc. Signer

Dette modul implementerer en høj niveau adgang til at håndtere visse signerede dokumenter som ikke følger XML-DSig formateringen. Modulet kan anvendes generelt til at implementere erstatning af papirbaserede arbejdsgange med elektroniske arbejdsgange med OCES digital signatur. Tamper resistent Audit log understøttes for dannelse og validering af signaturer. Der understøttes PKCS #7 formaterede signerede dokumenter samt signerede dokumenter genereret vha. værktøjet OpenSign¹.

Doc. Signer modulet stiller *ikke* arkiveringsfunktionalitet til rådighed. De signerede dokumenter skal opbevares i en separat database med tilhørende forretningslogik for understøttelse af workflows mod disse dokumenter.

- `SignatureEngine(User user, Configuration configuration)`

Skaber en ny SignatureEngine instans.

User repræsenterer den bruger, der benytter systemet, og benyttes til autentifikation samt audit logning af efterfølgende aktivitet.

Configuration objektet definerer konfigurationsmæssige detaljer såsom understøttede certifikat-udstedere (f.eks. TDC OCES, TDC OCES Systemtest), ønsket mekanisme til certifikat status-check, foretrukken signaturnøgle, implementationer af authorize interfacet osv.

Desuden følgende forenkede metoder til signatur håndtering:

- `SignatureDocument sign(Document xml, List idsToSign)`

Signerer et dokument.

- `ValidationResult validate(Document xml)`

validerer et signeret dokument.

¹ Se www.openoces.org

3.2 *User adapter*

Dette modul stiller adgang til TDC's tjenester: PID2CPR, isLRA, RID2CPR til rådighed. Bemærk, at der særskilt skal træffes aftale med TDC om adgang til disse tjenester. Lignende tjenester forventes udbudt af andre OCES nøglecenter operatører.

Bemærk at user adapter *ikke* stiller værktøjer til rådighed for generering af slutkunders signatur. Til dette formål henvises brugeren til standard Applet eller Active-X komponenter, f.eks. OpenSign fra projektet openoces.org.

- `String getCPR(OCESCertificate certificate, String vocesSN)`

Returnerer certifikatindehavers CPR nummer. Voces serial number angiver serienummer på det VOCES certifikat, der skal bruges til autentifikation overfor TDC's tjeneste. Kun for offentlige.

- `boolean matchesCPR(OCESCertificate certificate, String cpr, String vocesSN)`

Returnerer true, hvis borgeren, der indehaver det medsendte certifikat har CPR cpr.

- `boolean isLRA(MocesCertificate certificate, String vocesSN)`

Returnerer true, hvis indehaver af medsendte certifikat er LRA (local registration authority) for sin virksomhed/arbejdsgiver.

- `String getEmployeeCPR(MocesCertificate certificate, String vocesSN)`

Returnerer certifikat indehavers CPR nummer, hvis dette er registreret. Bemærk, at CPR numre for medarbejdere registreres af medarbejderens lokale administrator, og således ikke er underkastet nogen form for valideringen (udover mod11 check). Der kan således ikke påregnes samme kvalitet af CPR-numre, der fås vha. `getEmployeeCPR`, som for CPR numre fra `getCPR`.

4 Rådgivning

Signaturgruppens e-tingslysnings motor kan akkompagneres af rådgivningsydelser fra signatur specialisterne fra Signaturgruppen. Eksempler på relevante rådgivningsydelser gives her:

- Workshop for arkitekter og opfølgende løsningsdesign. Herunder gives eksempler på emner som kan dækkes under workshop:
 - OCES
 - XML DSig
 - XML kryptering
 - Kommunikations standarder: SOAP, WS-Security, MQ/JMS
 - Præcisering af ovenstående standarder i e-TL
 - Signering med medarbejder/person signatur i webapplikation (Opensign/openlys)
 - Administration og mobilitet af medarbejder signatur
 - Andre muligheder med PKI/OCES
 - Web authentication
 - SMIME
 - Ekstern adgang til intranet
- Gennemgang af løsningsdesign med kundens sikkerhed og revision.
 - Hardware sikkerheds modul

- Signatur lifecycle
 - Log / audit
 - Design af deployment strategi og test miljøer
- Udarbejdelse af key management drejebog for HSM: Driftsmanual for initialisering, nøglegenerering, certifikat udstedelse, backup, restore og arkivering af nøgler på HSM.