Dokumendi informatsioon	
Loomise kuupäev	1. aprill 2010
Teema	ID-kaardi baastarkvara
Viide	
Kellele	Veebisüsteemidesse digitaalallkirjastamise integreerijad
Koostajad	Ahto Jaago, Urmo Keskel, Kristi Uukkivi
Klienditeegi versioon	0.15
Muudatuste ajalugu	8. oktoober 2010. Lisatud uue ID-kaardi baastarkvara tugi, tehtud muid vormilisi korrektuure
	30.12.2010. API meetod getCertificates() asendus meetodiga getCertificate(). Täpsustatud peatükis 5.5 objekti Certificate väljade kirjeldusi.
	14.04.2011. Parandatud meetodi sign() tagastusväärtuse kirjeldust (punkt 5.3.2) ning täiendatud elemendi kasutamise infot (punkt 5.1)
	30.08.2011. Lisatud peatükk 4, mis räägib javascripti teegi sisemisest loogikast, mis järjekorras allkirjastamispluginaid laadida üritatakse. Lisatud peatükis 6 uued veakoodid.
	16.03.2012 Parandused dokumenti seoses Java Appleti toe kadumisega.
	04.03.2014. Muutunud on meetodite getCertificate(), sign() ja getVersion() API ja tööpõhimõte, meetodeid tuleb kasutada asünkroonselt. Võrreldes eelmise versiooniga on teegi kasutamiseks vaja viia sisse peatükis 5.3 kirjeldatud meetodite muudatused.

Sisukord

1 Sissejuhatus 3

2 Veebis allkirjastamise protsessi kirjeldus 3

3 Javascripti teegi pakutav funktsionaalsus 4

4 idCard.js-is allkirjastamise brauseriplugina valimise loogika 5

5 Teegi API –meetodid ja objektid 6

5.1 Meetod loadSigningPlugin(lang, pluginToLoad) 6

5.2 Meetod getLoadedPlugin() 6

5.3 Objekt IdCardPluginHandler(lang) 6

5.3.1 Meetod getCertificate(successCallback, failureCallback) 7

5.3.2 Meetod sign(id, hash, successCallback, failureCallback) 8

5.3.3 Meetod getVersion(successCallback, failureCallback) 9

5.4 Objekt IdCardException 10

5.5 Objekt Certificate 10

6 Veakoodid 11

1 Sissejuhatus

Käesolev dokument annab ülevaate veebis allkirjastamise protsessist ning kirjeldab JavaScripti teeki (idCard.js), mis on loodud lihtsustamaks veebis allkirjastamise toe realiseerimist veebirakendustesse. Javascripti teek on mõeldud kasutamiseks ID-kaardi ja Digi-ID kaartidega ning eeldab, et kasutaja arvutisse on paigaldatud Eesti ID-kaardi baastarkvara.

Teek on mõeldud kasutamiseks veebilehtedel ID-kaardiga, Digi-ID kaardiga ja teiste PKI võimeliste kiipkaartidega signeerimise toimingu teostamiseks.

NB! alates idCard.js teegi versioonist 0.15 on muutunud objekti IdCardPluginHandler meetodite API ja kasutamise põhimõte. Teegi versiooni 0.14 kasutajatel tuleb versiooni 0.15 kasutamiseks viia sisse järgmised muudatused:

- tuleb realiseerida meetodid, mille kaudu on võimalik **asünkroonselt** getCertificate(), sign() ja getVersion() meetodite tulemused kätte saada;
- getCertificate(), sign() ja getVersion() meetoditele tuleb lisada vastavad sisendparameetrid.

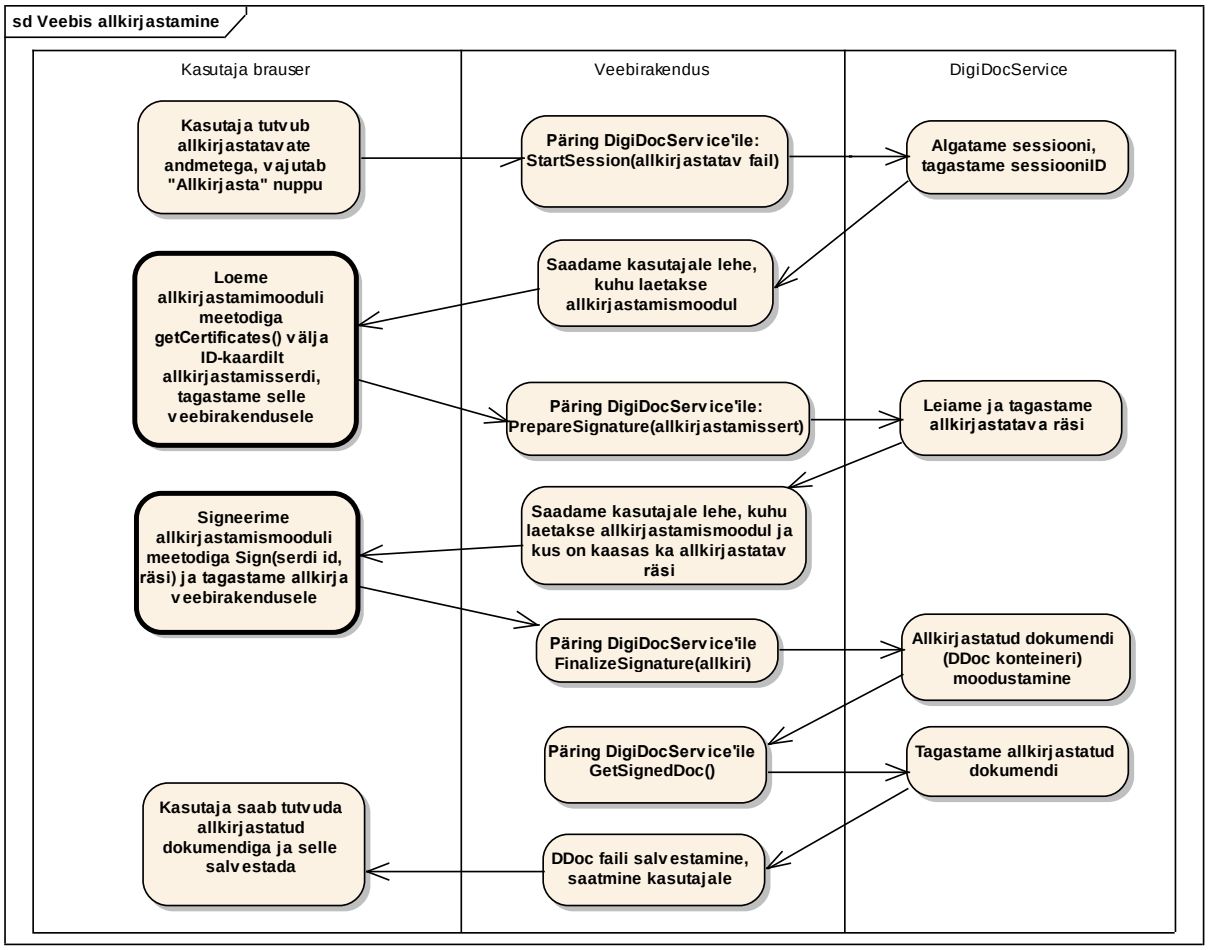
Täpsemad muudatuste kirjeldused on toodud peatükis 5.3.

2 Veebis allkirjastamise protsessi kirjeldus

Veebis ID-kaardiga allkirjastamine sisaldab:

1. Veebirakenduse ja kasutaja ID-kaardi vahelist suhtlust (veebilehele laetud allkirjastamismooduli kaudu), et ID-kaardilt kätte saada allkirjastamissertifikaat ja et teostada allkirjastamistoiming ID-kaardi kiibil.
2. Veebirakenduse ja DigidocService'i vahelist suhtlust – allkirjastatavate andmefailide saatmine DigiDocService-ile, allkirjastatava räsi tagasisaamine ning peale allkirjastamisoperatsiooni ID-kaardil allkirja saatmine DigiDocService-ile ning allkirjastatud konteineri (.ddoc faili) tagasisaamine. (Märkus: DigiDocService'i asemel võib kasutada ka DigiDoc C või Java teeki)

Skeemina näeb ID-kaardiga allkirjastamine veebis välja järgmine:



3 Javascripti teegi pakutav funktsionaalsus

Allkirjastamise javascript-teek võimaldab allkirjastamismooduli laadimist veebilehele ja suhtlust mooduliga. Kasutatavad objektid ja meetodid on kirjeldatud peatükis 5.Teegiga on kaasas ka näidisrakendus (sign.html), kus on realiseeritud veebis allkirjastamise protsessi need etapid, kus pöördutakse allkirjastamismooduli poole:

1. Allkirjastamismooduli laadimine lehele
2. Kasutaja ID-kaardilt allkirjastamissertifikaadi väljalugemine
3. Andmete signeerimine ID-kaardiga

Antud teegi eesmärk on pakkuda universaalset API-it nimetatud toimingute teostamiseks ning ID-kaardi baastarkvara eri versioonide ja komponentide iseärasused ära peita. Seoses erinevate ID-kaardi baastarkvara erinevate versioonidega on allkirjastamise mooduleid mitmeid: universaalse API-ga mudel 2010 aastal välja tulnud uue ID-kaardi baastarkvara jaoks, varasema tarkvaraga koos levitatavad erineva API-ga activeX'id Windowsi operatsioonisüsteemil Internet Exploreri, Windowsi Mozilla Firefoxi plugin, Mac'i plugin (Firefox-i ja Safari jaoks) – neist mitme komponendi API erineb omavahel. Antud teek tuvastab

kasutaja arvuti konfiguratsiooni ning laeb ise automaatselt sobiliku veebis signeerimise komponendi ning võimaldab seda läbi universaalse API pruukida.

Teegi kasutamine eeldab, et kasutaja arvutis on installeeritud ID-kaardi tarkvara ametlikult levitatav versioon aadressilt https://installer.id.ee või ID-kaardi baastarkvara beetaversiooni aadressilt http://id.eesti.ee.

4 idCard.js-is allkirjastamise brauseriplugina valimise loogika

Alljärgnev tabel kirjeldab, kuidas ehk missuguses prioriteetsuse järjekorras toimub idCard.js sees kasutaja arvutist sobiva allkirjastamiseks mõeldud brauseriplugina valik sõltuvalt operatsioonisüsteemist ja brauserist. Näiteks Windowsis IE brauseri puhul püütakse esmalt laadida uue ID-kaardi baastarkvara koosseisus olevat allkirjastamispluginat (digiDocPlugin). Kui selle laadimine ebaõnnestub, siis püütakse laadida vana ID-kaardi baastarkvaraga levinud activeX komponenti veebis allkirjastamiseks. Kui ka selle laadimine ebaõnnestub, siis tagastatakse veakood 100 (veakoodide kohta vaata alt eraldi peatükist).

<table>
<tr><td>OpSüs\

Brauser</td><td>IE</td><td>Firefox</td><td>Safari</td><td>Teised</td></tr>
<tr><td>Windows</td><td>digidocPlugin → activeX → viga 100</td><td>digidocPlugin → winMozPlugin → viga 100</td><td>digidocPlugin → viga 100</td><td rowspan="3">digidocPlugin → viga 100</td></tr>
<tr><td>Mac</td><td>-</td><td>digidocPlugin → macPlugin</td><td>digidocPlugin → macPlugin</td></tr>
<tr><td>Teised</td><td colspan="3">digidocPlugin → viga 100</td></tr>
</table>

- digiDocPlugin – 2010ndal aastal uue ID-kaardi baastarkvara arendamise käigus loodud uus veebis allkirjastamise brauseriplugin, kõigist senistest veebis allkirjastamise komponentidest kõige stabiilsem ja põhjalikumalt testitud
- activeX – vana ID-kaardi baastarkvaraga levitatud allkirjastamisplugin IE brauseri jaoks
- winMozPlugin – vana ID-kaardi baastarkvaraga levitatud Mozilla Firefox-i plugin Windowsile
- macPlugin – uue ID-kaardi baastarkvara eelselt levinud plugin Mac operatsioonisüsteemis kasutamiseks

5 Teegi API –meetodid ja objektid

5.1 Meetod loadSigningPlugin(lang, pluginToLoad)

Meetod laeb veebilehele allkirjastamismooduli. Eeldatakse, et lehe html-is on olemas koht (täpsemalt html element id'ga **pluginLocation**), kuhu moodul laadida, näiteks:

NB! Tuleks vältida elemendi kasutamist alljärgneval kujul, sest sellisel juhul ebaõnnestub plugina laadimine näiteks Firefox ja Chrome brauserite korral:

~~ ~~

Parameetrid:

- **lang** – variandid: 'est', 'eng'; näitab, mis keelsetena tulevad veateated.
- **pluginToLoad** – mittekohustuslik parameeter; kui on määramata, siis laetakse sobiv moodul automaatselt sõltuvalt kasutatavast ID-kaardi baastarkvara versioonist, operatsioonisüsteemist ja brauserist. Tavaolukorras kasutatav väärtus on ''. Parameetri võimalikud väärtused lisaks tühjale stringile: 'activeX', 'winMozPlugin', 'macPlugin', 'digidocPlugin'

5.2 Meetod getLoadedPlugin()

Tagastab stringina, mis moodul laaditi – variandid: 'activeX', 'winMozPlugin', 'macPlugin', ' digidocPlugin'.

5.3 Objekt IdCardPluginHandler(lang)

Objekt, mille meetoditega saab ID-kaardilt sertifikaate lugeda ja allkirjastamisoperatsiooni teostada. Sisendparameetrist **lang** (variandid 'est', 'eng') sõltub, mis keelsetena tulevad veateated.

NB! alates idCard.js teegi versioonist 0.15 on muutunud objekti **IdCardPluginHandler** meetodite API ja kasutamise põhimõte, kasutamine toimub **asünkroonselt**.

Teegi versiooni 0.14 kasutajatel tuleb versiooni 0.15 kasutamiseks viia sisse järgmised muudatused:

- tuleb realiseerida meetodid, mille kaudu on võimalik **asünkroonselt** getCertificate(), sign() ja getVersion() meetodite tulemused kätte saada;
- getCertificate(), sign() ja getVersion() meetoditele tuleb lisada vastavad sisendparameetrid.

Täpsemad kirjeldused getCertificate(), sign() ja getVersion() meetodite asünkroonseks kasutamiseks on antud järgmistes alampeatükkides.

5.3.1 Meetod getCertificate(successCallback, failureCallback)

Meetod tagastab asünkroonselt leitud/valitud allkirjastamissertifikaadi. Vea korral tagastatakse asünkroonselt IdCardException.

Sisendparameetrid:

Nimetus	Andmetüüp	Kirjeldus
successCallback	String	Antud parameetri väärtuseks tuleb panna meetodi nimetus, mille poole teek asünkroonselt getCertificate() meetodi **positiivse stsenaariumi** korral pöördub ja allkirjastamise sertifikaadi väärtuse edastab (Certificate objekt).
failureCallback	String	Antud parameetri väärtuseks tuleb panna meetodi nimetus, mille poole teek asünkroonselt getCertificate() meetodi **negatiivse stsenaariumi** korral pöördub ja vastava veateate ja veakoodi edastab (IdCardException objekt).

NB! getCertificate() sisendparameetrite nimetused tuleb teegi kasutajal ise määrata ja määratud nimetusega meetodid ise realiseerida (vt. ka allolev näide ja Web Sign Demo rakenduse lähtekood (https://www.openxades.org/web_sign_demo/async/sign_async.html)):

- Sertifikaadi väärtuse kättesaamiseks tuleb teegi kasutajal realiseerida „**successCallback**“ parameetri väärtusele vastava nimega meetod ning määrata sellele Certificate tüüpi sisendparameeter. Teek pöördub selle meetodi poole, kui kaardilt õnnestub kasutaja sertifikaat lugeda.
- Veaolukorra puhul pöördub teek „**failureCallback**“ parameetri nimetusega määratud meetodi poole, mis tuleb samuti teegi kasutajal realiseerida määrates sellele IdCardException tüüpi sisendparameetri.

Näide:

```
<!-- kutsu välja getCertificate() meetod -->
new IdCardPluginHandler('est').getCertificate(handleCertificate, handleError);

<!-- kirjuta meetodid, mille poole teek tulemuste edastamiseks pöördub -->

function handleCertificate(cert) {
	<!-- realiseeri kood sertifikaadi edasiseks kasutamiseks -->
}

function handleError(ex) {
	<!-- realiseeri kood veaolukorra lahendamiseks -->
```

```
}
```

5.3.2 Meetod sign(id, hash, successCallback, failureCallback)

Meetod räsi (hash) signeerimiseks soovitud sertifikaadiga. Meetodi välja kutsumise eelselt on vajalik teada saada sertifikaadi identifikaator (id) kasutades getCertificate() meetodit.

Sisendparameetrid:

Nimetus	Andmetüüp	Kirjeldus
id	String	Signeerimisel kasutatava sertifikaadi identifikaator, Meetod getCertificate(**successCallback, failureCallback)** meetodi tulemuses allkirjastamisel kasutatava sertifikaadi id atribuudi väärtus.
hash	String	Signeeritav räsi HEX kujul, nt. „FAFA0101FAFA0101FAFA0101FAFA0101FAFA0101“
successCallback	String	Antud parameetri väärtuseks tuleb panna meetodi nimetus, mille poole teek asünkroonselt sign() meetodi **positiivse stsenaariumi** korral pöördub ja signatuuri väärtuse edastab (hex stringina).
failureCallback	String	Antud parameetri väärtuseks tuleb panna meetodi nimetus, mille poole teek asünkroonselt sign() meetodi **negatiivse stsenaariumi** korral pöördub ja vastava veateate ja veakoodi edastab (IdCardException objekt).

NB! sign() meetodi „successCallback“ ja „failureCallback“ sisendparameetrite nimetused tuleb teegi kasutajal ise määrata ja määratud nimetusega meetodid ise realiseerida (vt. ka allolev näide ja Web Sign Demo rakenduse lähtekood (https://www.openxades.org/web_sign_demo/async/sign_async.html)):

- Signatuuri väärtuse kättesaamiseks tuleb teegi kasutajal realiseerida „**successCallback**“ parameetri väärtusele vastava nimega meetod ning määrata sellele String tüüpi sisendparameeter. Teek pöördub selle meetodi poole, kui signeerimine õnnestub.
- Veaolukorra puhul pöördub teek „**failureCallback**“ parameetri nimetusega määratud meetodi poole, mis tuleb samuti teegi kasutajal realiseerida määrates sellele IdCardException tüüpi sisendparameetri.

Näide:

```
<!-- kutsu välja sign() meetod -->
new IdCardPluginHandler('est').sign(id, hash, handleSignature, handleError);

<!-- kirjuta meetodid, mille poole teek tulemuste edastamiseks pöördub -->
function handleSignature(signature) {
	<!-- realiseeri kood signatuuri edasiseks kasutamiseks -->
}

function handleError(ex) {
	<!-- realiseeri kood veaolukorra lahendamiseks -->
}
```

5.3.3 Meetod getVersion(successCallback, failureCallback)

Tagastab veebis allkirjastamise plugin'i versiooninumbri asünkroonselt.

Sisendparameetrid:

Nimetus	Andmetüüp	Kirjeldus
successCallback	String	Antud parameetri väärtuseks tuleb panna meetodi nimetus, mille poole teek asünkroonselt getVersion() meetodi **positiivse stsenaariumi** korral pöördub ja versiooni väärtuse edastab (stringina).
failureCallback	String	Antud parameetri väärtuseks tuleb panna meetodi nimetus, mille poole teek asünkroonselt getVersion() meetodi **negatiivse stsenaariumi** korral pöördub ja vastava veateate ja veakoodi edastab (IdCardException objekt).

NB! getVersion() meetodi „successCallback" ja „failureCallback" sisendparameetrite nimetused tuleb teegi kasutajal ise määrata ja määratud nimetusega meetodid ise realiseerida (vt. ka allolev näide ja Web Sign Demo rakenduse lähtekood (https://www.openxades.org/web_sign_demo/async/sign_async.html)):

- Versiooni väärtuse kättesaamiseks tuleb teegi kasutajal realiseerida „**successCallback**" parameetri väärtusele vastava nimega meetod ning määrata sellele String tüüpi sisendparameeter. Teek pöördub selle meetodi poole, kui versiooni pärimine õnnestub.
- Veaolukorra puhul pöördub teek „**failureCallback**" parameetri nimetusega määratud meetodi poole, mis tuleb samuti teegi kasutajal realiseerida määrates sellele IdCardException tüüpi sisendparameetri.

Näide:

```
<!-- kutsu välja getVersion() meetod -->
new IdCardPluginHandler('est').getVersion(handleVersion, handleError);

<!-- kirjuta meetodid, mille poole teek tulemuste edastamiseks pöördub -->
function handleVersion(version) {
	<!-- realiseeri kood versiooni edasiseks kasutamiseks -->
}

function handleError(ex) {
	<!-- realiseeri kood veaolukorra lahendamiseks -->
}
```

5.4 Objekt IdCardException

Vea puhul tagastatav objekt väljadega **returnCode** ja **message** (vt jaotist Veakoodid) ning meetoditega **isError()** ja **isCancelled()**. **isCancelled()** tagastab true juhul, kui returnCode on 1, **isError()** kõigi muude nullist erinevate veakoodide korral.

5.5 Objekt Certificate

Objekti **IdCardPluginHandler** meetodi **getCertificate()** poolt asünkroonselt tagastatava objekti struktuur. Struktuuril on järgmised väljad:

Nimetus	Andmetüüp	Kirjeldus
id	String	Sertifikaadi identifikaator
cert	String	Sertifikaat HEX kujul
CN	String	Sertifikaadi nimi (CommonName)
issuerCN	String	Sertifikaadi väljaandja nimi
validFrom	String	Sertifikaadi kehtivuse algusaeg Zulu ajavööndis kujul „dd.mm.yyyy hh:mm:ss”
validTo	String	Sertifikaadi kehtivuse lõppemiseaeg Zulu ajavööndis kujul „dd.mm.yyyy hh:mm:ss”

6 Veakoodid

Veakood	Semantika
1	Allkirjastamine katkestati
2	Sertifikaate ei leitud
9	Vale allkirjastamise PIN
12	ID-kaardi lugemine ebaõnnestus
14	Tehniline viga
15	Vajalik tarkvara on puudu
16	Vigane sertifikaadi identifikaator
17	Vigane räsi
19	Veebis allkirjastamise käivitamine on võimalik vaid https aadressilt
100	Teie arvutist puudub allkirjastamistarkvara või ei ole Teie operatsioonisüsteemi ja brauseri korral veebis allkirjastamine toetatud. Allkirjastamistarkvara saate aadressilt https://installer.id.ee