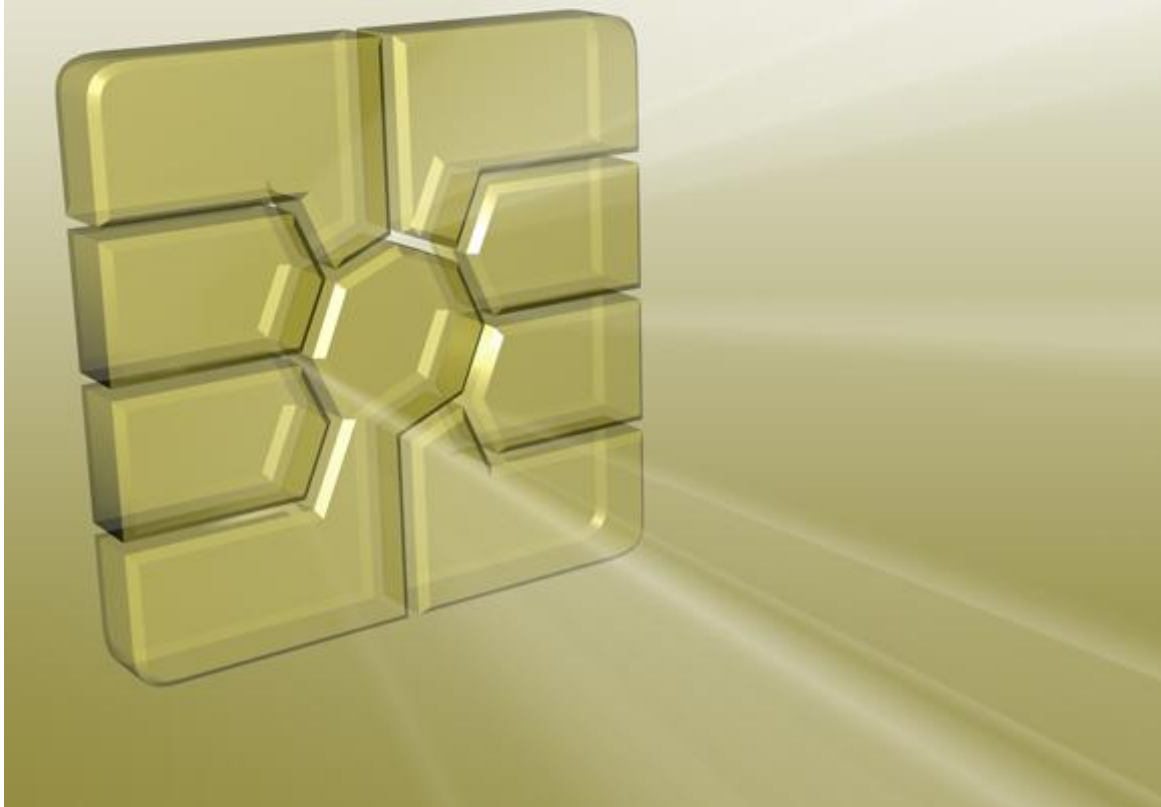


WINDOWS

# Aloaha Smartcard Connector



## **Aloaha SmartCard SDK DE**

© 2010 Wrocklage Intermedia GmbH

# Aloaha SmartCard SDK DE

© 2010 Wrocklage Intermedia GmbH

Copyright © 2010 Wrocklage Intermedia GmbH (im folgenden Wrocklage genannt). Alle Rechte vorbehalten. Der Inhalt dieses Dokumentes darf ohne vorherige schriftliche Genehmigung durch Wrocklage in keiner Form, weder ganz noch teilweise, vervielfältigt, weitergegeben, verbreitet oder gespeichert werden. Wrocklage entwickelt die Produkte im Rahmen eines kontinuierlichen Verbesserungsprozesses ständig weiter.

Wrocklage behält sich deshalb das Recht vor, ohne vorherige Ankündigung an jedem der in dieser beschriebenen Dokumentation beschriebenen Produkte Veränderungen bzw. Verbesserungen vorzunehmen. Wrocklage übernimmt keine Gewährleistung für technische oder redaktionelle Fehler oder Auslassungen in diesem Handbuch. Die Haftung für Schäden und Folgeschäden, welche auf einer leicht fahrlässigen Pflichtverletzung von Wrocklage eines gesetzlichen Vertreters und / oder Erfüllungsgehilfen von Wrocklage und auf der Verwendung dieses Dokumentes und der in ihm enthaltenen Informationen beruhen, ist ausgeschlossen, soweit keine Verletzung wesentlicher Vertragspflichten vorliegen. Wesentliche Vertragspflichten sind solche Pflichten, deren Erreichung die ordnungsgemäße Durchführung des Vertrages überhaupt erst ermöglichen und auf deren Einhaltung der Kunde regelmäßig vertrauen darf. Ansprüche aus dem Produkthaftungsgesetz bleiben hiervon unberührt. Der Inhalt dieses Dokumentes wird so dargelegt, wie er auch aktuell bekannt ist. Wrocklage übernimmt weder ausdrücklich noch stillschweigend irgendeine Gewährleistung für die Richtigkeit oder Vollständigkeit dieses Dokumentes.

Printed: Februar 2010

# Inhalt

	<b>Seite</b>
<b>1. Einleitung</b>	<b>5</b>
<b>2. Installation</b>	<b>7</b>
<b>3. Anwendung</b>	<b>10</b>
3.1 Sprachauswahl	12
3.2 Karten-Assistent	13
3.3 Digital Signieren	15
3.3.1 Konfiguration digitale Unterschrift	17
3.3.2 Signatureinstellungen	23
3.4 PKCS #7 Signatur erstellen	25
3.5 PIN Verwaltung	26
3.6 Zertifikate	28
3.7 CSP / Kartenleser	39
3.8 Zeitstempel	42
<b>4. Anwender Support</b>	<b>44</b>
4.1 MS Crypto API	45
4.1.1 Outlook Einstellungen	46
4.2 PKCS #11	51
4.2.1 Firefox Einstellungen	52
4.3 Language.ini	56
<b>5. Aloaha CSP API</b>	<b>57</b>
5.1 Laden der CSP API	58
5.2 Nützliche Hilfsfunktionen für Skriptsprachen	59
5.3 Digitale Signatur Funktionen	60
<b>6. APIs und Beispiele</b>	<b>61</b>
<b>7. Zertifikat Parser</b>	<b>61</b>
<b>8. PKCS#7 / S/Mime</b>	<b>62</b>
<b>9. PKCS#7 erzeugen / überprüfen</b>	<b>63</b>
<b>10. Smartkarten Zertifikate anzeigen</b>	<b>64</b>
<b>11. ADPU Tester</b>	<b>65</b>

<b>12. Smartkarten Tester</b>	<b>66</b>
<b>13. PKCS7 Signatur mit Zeitstempel</b>	<b>67</b>
<b>14. Allgemeine CSP Informationen</b>	<b>67</b>
<b>15. Zertifikat Management</b>	<b>68</b>
<b>16. Verwendung von Zertifikaten</b>	<b>70</b>
<b>17. RSA Datenverschlüsselung</b>	<b>71</b>
<b>18. FAQ</b>	<b>72</b>
<b>19. Hilfe</b>	<b>73</b>
<b>Index</b>	<b>75</b>

# 1. Einleitung

## Aloaha Smart Card SDK

Der Smart Card SDK von Aloaha, einschließlich eines von Microsoft abgenommenen CSP (Cryptographic Service Provider) und ein PKCS #11, stellen dem Betriebssystem von Microsoft Windows und den Anwendungen eine sicherheitserhöhende Plug´n´Play Lösung zur Verfügung.

Die Aloaha Smartcard Middleware unterstützt verschiedene SmartCards wie z.B. die Deutsche elektronische Gesundheitskarte (eGK), den Arztausweis (HBA), die Belgische e-ID (Belpic), die Schweizer GS1, die Italienische Infocamere, SagemOrga Micardo, CardOS, Sicrypt, Mifare 4k und weitere ...

**Die Systemintegration des "Aloaha Smart Card SDK" ist einfach und schnell. Das System ist nach einem 2-minutigen Installationsprozess einsetzbar.**

- Keine komplizierte Konfiguration
- Keine speziellen Personalisierungsprozesse.
- Keine Kopfschmerzen
- geringe notwendige Speicherkapazität

Sobald der CSP installiert ist, werden die durch Aloaha unterstützten SmartCards in Ihre Windows-Umgebung integriert und bieten eine sehr sichere, mobile Lösung.

Mit den Zertifikaten auf Ihrer SmartCard wird zu dem die Sicherheit Ihres Windows Betriebssystems erhöht.

### Beispiele:

- E-Mail-Schutz (Verschlüsseln und Signieren)
- Bürodokumentenschutz (Verschlüsseln und Signieren)
- Datei und Verzeichnisschutz (Zertifikatbasierte EFS / NTFS Verschlüsselung)
- kundenbeglaubigte Kommunikation (SSL/HTTPS)
- Formulare digital unterschreiben (PDF, Info Path, Share Point)
- VPN Beglaubigung wie OpenVPN
- Windows Logon (Aktive Directory)

### Aloaha Eigenschaften:

- Unterzeichnet und geprüft durch Microsoft
- Windows 2000/XP/Vista/Windows 7 kompatibel
- Windows 2000/2003/2008 Server kompatibel
- Berechtigte Signaturenkarten werden unterstützt
- PKCS #11 Modul integriert
- Plug & Play

**Unterstützte Standards:**

- ISO 7816
- PC/SC
- Sichere Pin-Eingabe über PC/SC II
- Mifare
- Microsoft CSP
- Remote CSP für gemeinsame Kartennutzung
- PKCS#11
- PKI

**Unterstützte Algorithmen:**

- RSA 1024 – 2048
- Elliptische Kurven (ECC)
- DES u. Triple-DES
- RC2 & RC4
- MD5, SHA-1 & SHA-2

## 2. Installation

Um die Software zu installieren, verwenden Sie nachfolgenden Link:  
<http://www.aloaha.com/download/Smart Card SDK.zip>

### Installations-Voraussetzungen

- Windows 2000/3/8
- Windows XP (Service Pack 3 empfohlen, aber nicht zwingend notwendig)
- Windows Vista
- Windows 7

Das Installationsprogramm enthält eine Datei "*Smart Card SDK.exe*", starten Sie diese durch Doppelklick.

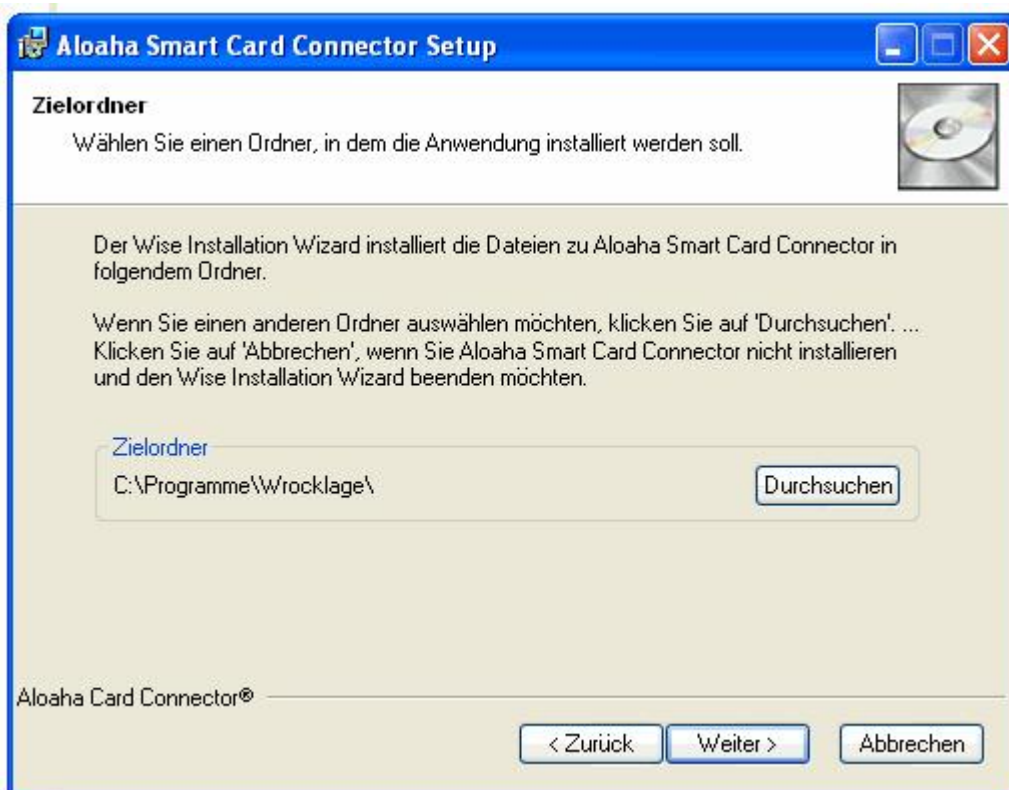
Nun startet die Windows Installation und fragt nach der gewünschten Installationssprache. Die Sprache kann später unter **HKCU\Software\Aloaha\pdf\language** geändert werden.



Nach der Sprachauswahl startet das Setup.




Setup fragt nach dem Installationspfad, bevor die Installation beginnt. Es ist ratsam, alle Aloaha Produkte im gleichen Zielordner zu installieren.





Klicken Sie auf "Fertigstellen", um das Setup zu beenden.



Nun sehen Sie ein kleines SmartCard Symbol  in Ihrer Systemleiste.

### 3. Anwendung

Der Aloaha Smart Card SDK beinhaltet einen von Microsoft anerkannten "Cryptographic Service Provider" (CSP) und eine PKCS #11 Bibliothek (aloaha\_pkcs11.dll in system32).

Der Vorteil eines CSP besteht darin, dass der CSP die Zertifikate bereitstellt, die auf der Karte gespeichert sind.

Wenn der CSP ein Zertifikat des angemeldeten Users speichert, wird das Zertifikat in entsprechendem Verzeichnis abgelegt.

Um ein Video hierzu anzusehen, verwenden Sie nachfolgenden Link:

<http://www.aloaha.com/movies/register.htm>

Sobald eine Karte in ein Lesegerät gesteckt wurde, registriert das Programm automatisch alle auf der Karte befindlichen Zertifikate. Die Zertifikate können jedoch auch, wie im Beispiel gezeigt, manuell registriert werden. Statt Autoregister klicken Sie stattdessen auf Register.

Zum ersten Mal sollte ein Zertifikat manuell registriert werden, da dann Aloaha eventuell fehlende Root Zertifikate vom Aloaha Server nachlädt und mit installiert.



Die erste Zahl zeigt die Anzahl der Kartenlesegeräte an. Der Screenshot zeigt die Zertifikate der Karten in angeschlossenen Kartenlesern.

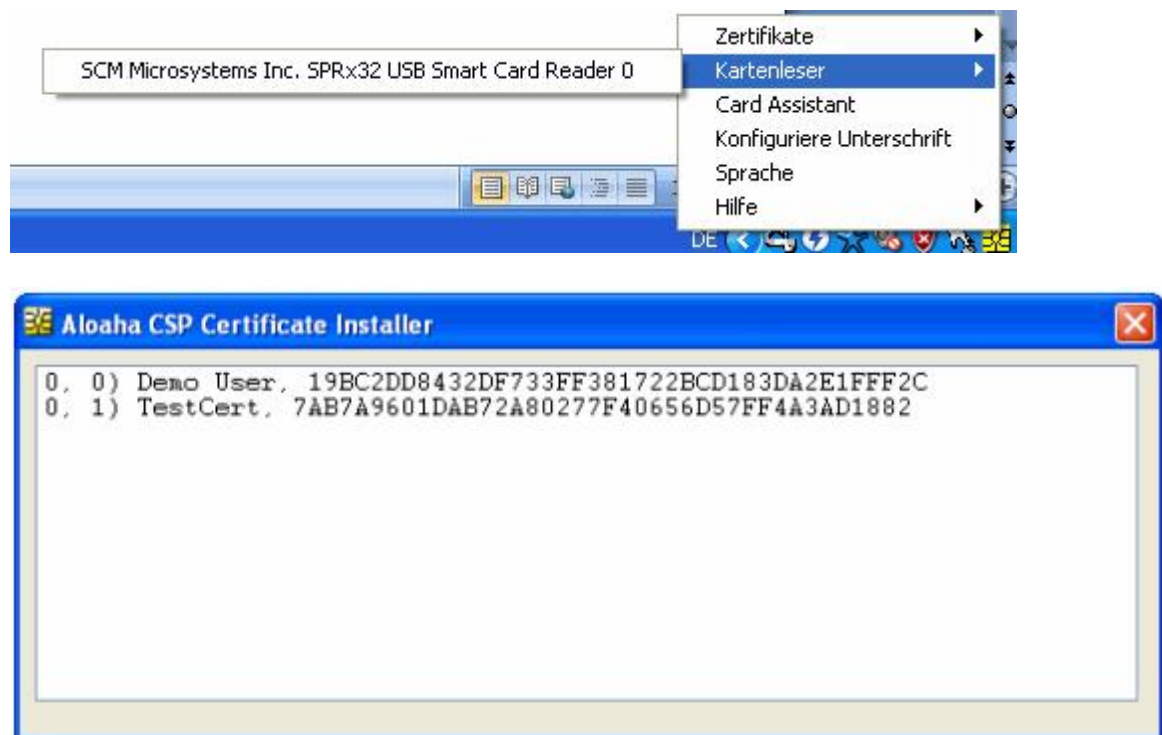
Die Zahl nach dem Komma zeigt den Zertifikat-Typ an.

Typ 0 = Unterschriftszertifikat,  
Typ 1 = Authentifizierungszertifikat,  
Typ 2 = Verschlüsselungszertifikat.

Enthält eine Karte nur ein Zertifikat enthält, wird dieses als Typ 1 angezeigt.

Um alle registrierten Zertifikate zu entfernen, klicken Sie auf "alle entfernen". Ist Autoentfernen aktiviert, werden alle registrierten Zertifikate gelöscht, sobald sämtliche Karten aus den Kartenlesegeräten entfernt wurden.

In einigen Fällen gibt es mehrere mit einem System verbundene Kartenlesegeräte. Die Zertifikate aller Lesegeräte aufzuzählen, nimmt Zeit in Anspruch. In diesem Fall können Sie das Kartenlesegerät direkt (wie gezeigt) auswählen. Aloaha liest dann nur die Zertifikate der Karte im gewählten Leser aus.



Sie können das Zertifikat nun anklicken, um es anzeigen zu lassen oder es per Doppelklick im aktuellen Verzeichnis zu speichern.

#### Manuelle Registrierung hat Vorteile:

1. Wenn das Ausgabezertifikat im System nicht verfügbar ist, wird Aloaha versuchen, es von der Aloaha Website herunterzuladen.
2. Das eingetragene Zertifikat wird automatisch als Standardzertifikat konfiguriert.

### 3.1 Sprachauswahl

So ändern Sie die Anwendersprache unabhängig von der Systemsprache des Betriebssystems. Die Anwendersprache kann über das Windows Startmenü oder das Traymenü geändert werden.



Zur Auswahl stehen die angezeigten Sprachen.

Nachdem Sie die Sprache geändert haben, werden Sie aufgefordert, das Programm neu zu starten.

Sollten Sie das Programm nicht neu starten, bleibt die bis dahin eingestellte Anwendersprache erhalten.



## 3.2 Karten-Assistent

Der "Aloaha Card Assistant" kann über das Windows Startmenü oder das Schnellstartmenü gestartet werden.



Durch Aufklappen des Auswahlménüs können die im System verfügbaren Kartenlesegeräte ausgewählt werden



Abhängig vom Kartentyp können PINs geändert oder PINs mit PUK entsperrt bzw. zurückgesetzt werden.

## 3.3 Digital Signieren

### Die digitale Signatur

Eine digitale Signatur im Sinne des Gesetzes ist „ein mit einem privaten Signaturschlüssel erzeugtes Siegel zu digitalen Daten, das mit Hilfe eines zugehörigen öffentlichen Schlüssels, der mit einem Signaturschlüssel-Zertifikat einer Zertifizierungsstelle versehen ist, den Inhaber des Signaturschlüssels und die Unverfälschtheit der Daten erkennen lässt“ (SigG).

Mit der Entwicklung der digitalen Signatur wurde das Ziel verfolgt, eine der persönlichen Unterschrift äquivalente Signierungsmethode zu entwickeln, mit der auf elektronischem Weg Daten unterzeichnet werden können.

Denn das Hauptproblem bei der Übermittlung elektronischer Daten ist die leichte Manipulierbarkeit. Erst durch die elektronische Signatur kann dieses Problem behoben werden, da eine unbemerkte Datenmanipulation nicht mehr möglich ist.

Voraussetzung hierfür ist, dass die elektronische Signatur wie eine handschriftliche untrennbar mit dem jeweiligen Dokument verbunden ist. Sie kann von jedem eingesehen, aber nur vom Unterzeichner selbst geändert werden. Der Unterzeichner kann somit eindeutig identifiziert werden und die Signatur macht jede eventuelle Manipulation, wie das nachträgliche Streichen oder Ändern von Textpassagen eines Dokuments, sofort erkennbar.

Durch die Zertifikatsprüfung kann zudem bewiesen werden, dass die Signatur nicht gefälscht wurde, der Zertifikatsinhaber also echt ist. Dabei werden außer seinem Namen keine persönlichen Daten preisgegeben.

### Gesetzliche Regelungen

Definitionen der unterschiedlichen Arten der digitalen Signatur finden sich im Signaturgesetz (SigG) und in der Verordnung zum Signaturgesetz (SigV). Außerdem werden darin Anforderungen an die elektronischen Unterschriften dargestellt sowie Zertifizierungsdiensteanbieter (ZDA) definiert.

Es wird unterschieden in einfache, fortgeschrittene und qualifizierte digitale Signaturen. Jede Signatur steht für eine bestimmte Qualitätsstufe. Je höherwertiger die Signatur, desto mehr Bedeutung hat sie für den Rechtsverkehr, und desto größer ist ihre Funktionalität. Nur qualifizierte Signaturen erfüllen die Anforderungen in Bezug auf elektronische Daten genauso wie die handschriftliche Unterschrift Anforderungen in Bezug auf Daten in Papierform erfüllt. Sie sind sogar vor Gericht als Beweismittel zugelassen.

Die für qualifizierte elektronische Signaturen zugelassenen kryptografischen Algorithmen werden von der Bundesnetzagentur genehmigt und veröffentlicht. Unter [www.bundesnetzagentur.de](http://www.bundesnetzagentur.de) finden Sie zudem eine Liste aller akkreditierten Zertifizierungsdiensteanbieter (Trustcenter). Dort sind auch die für eine qualifizierte elektronische Signatur zugelassenen Produkte aufgelistet.

Die Voraussetzungen für eine qualifizierte Signatur sind dann gegeben, wenn sie ausschließlich dem Unterzeichner zugeordnet werden kann, die eindeutige Identifizierung des Unterzeichners zulässt, mit Mitteln erstellt wird, die nur der Unterzeichner kontrolliert, jede nachträgliche Änderung der signierten Daten ersichtlich macht und auf einem qualifizierten Zertifikat beruht.

Ein qualifiziertes Zertifikat kann nur von einem akkreditierten Zertifizierungsdiensteanbieter ausgestellt werden. Dabei gelten ganz besonders strenge Anforderungen hinsichtlich der Sicherheit der Schlüsselerstellung und der Organisation des Trustcenters. Die Einhaltung der gesetzlichen Vorschriften durch die Trustcenter wird in Deutschland ebenfalls von der Bundesnetzagentur kontrolliert.

## Public Key Verfahren

Digitale Signaturen basieren auf asymmetrischen Kryptosystemen und verwenden ein Schlüsselpaar, das aus einem privaten (geheimen) und einem öffentlichen (nicht geheimen) Signaturschlüssel besteht und sich gegenseitig ergänzt.

Daten, die mit dem einen Schlüssel geschlossen wurden, können nur mit dem anderen wieder geöffnet werden. Beim Signieren wird der private Schlüssel verwendet. Dieser befindet sich auf dem Chip der Karte und lässt sich nicht auslesen. Die zu verarbeitenden Daten werden auf den Chip geladen, dort ver- oder entschlüsselt und wieder in den Computer übertragen.

Um den privaten Schlüssel zu benutzen, wird die richtige PIN benötigt, die zusätzliche Sicherheit gewährleistet. Die Signatur kann also nur vom Karteninhaber sein, denn nur er ist in Besitz von Karte und PIN. Der öffentliche Schlüssel ist in ein Zertifikat integriert und steht jedermann in Verzeichnisdiensten im Internet zur Verfügung oder kann per E-Mail versandt werden. Um zu gewährleisten, dass dieses Zertifikat und somit der Schlüssel nicht gefälscht wurde, lässt sich die Signatur des Herausgebers prüfen.

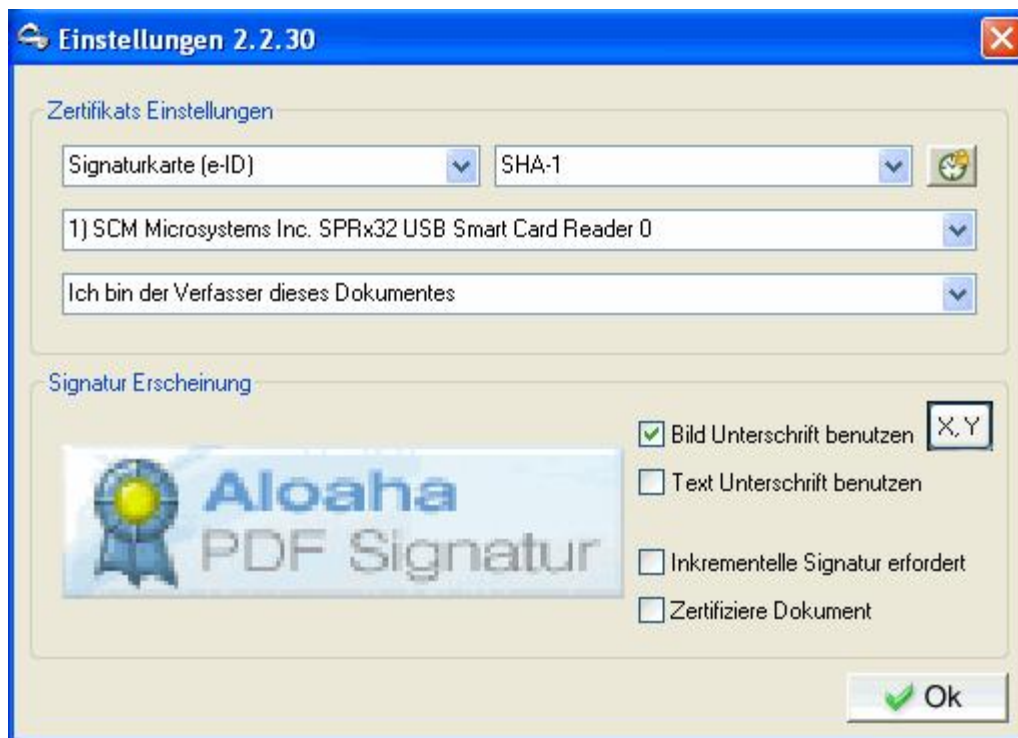
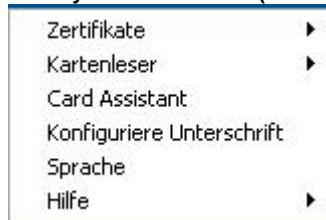
Beim Prüfen der Signatur wird der öffentliche Schlüssel des Empfängers verwendet, so dass nur dieser die Daten mit seinem privaten Schlüssel wieder entschlüsseln kann. Beim Signieren einer Datei wird ein Hashwert gebildet, der mit einem Fingerabdruck vergleichbar ist. Zwei verschiedene Dokumente können so nie denselben Hashwert haben. Der Hashwert wird nach dem RSA Verfahren unter Verwendung eines Schlüssels mit einer Länge von mindestens 1024 Bit (abhängig von der verwendeten Karte) verschlüsselt.

Die Verschlüsselung des Hashwerts findet auf dem Chipkartenprozessor statt, welcher kleinere Datenmengen verarbeiten kann. So wird sichergestellt, dass der private Schlüssel die Karte nicht verlässt. Die verschlüsselten Daten werden anschließend wieder in den Computer zurückgeschickt. Vorher muss der private Schlüssel durch die richtige PIN (Personal Identification Number) freigegeben werden.



### 3.3.1 Konfiguration digitale Unterschrift

#### CSP Symbol Menüleiste (rechte Maustaste)



#### 1. Zertifikatquelle

Hier können Sie zwischen verschiedenen Arten von Zertifikaten wählen, die Sie zum Signieren Ihrer PDF-Dateien verwenden möchten.

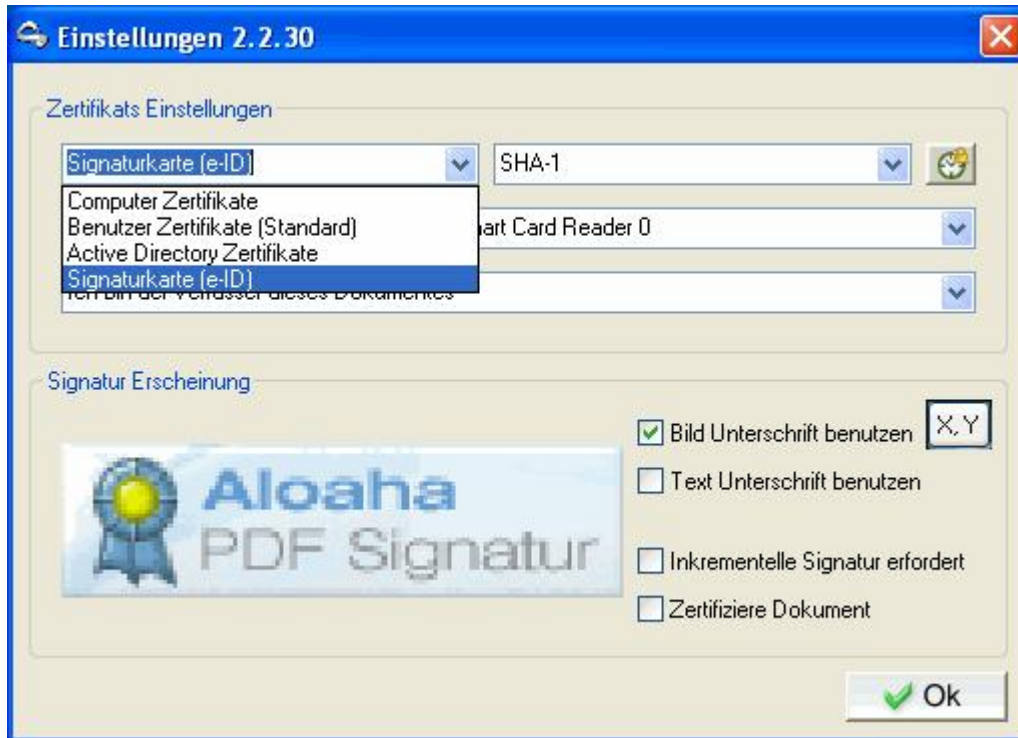
Zur Auswahl stehen:

- **Computer Zertifikate**  
Es werden in der Zertifikats-Auswahlliste alle Zertifikate angezeigt, die dem Computer zugeordnet sind.
- **Benutzer Zertifikate (Standard)**  
Es werden in der Zertifikats-Auswahlliste alle Zertifikate angezeigt, die dem aktuellen Benutzer zugeordnet sind.
- **Active Directory Zertifikate**  
Es werden in der Zertifikats-Auswahlliste alle Zertifikate angezeigt, die im Active Directory zur Verfügung stehen.
- **SmartCard (e-ID)**  
Es werden in der Zertifikats-Auswahlliste alle angeschlossenen Kartenleser angezeigt.

## 2. Art des Zertifikats

Hier können Sie die Zertifikatsliste der angezeigten Zertifikate nach besonderen Zertifikat-Attributen filtern.

Wenn als Zertifikatsquelle "Smartcard" ausgewählt wird, können Sie zwischen SHA-1 und SHA-256 als Signatur-Algorithmus auswählen. SHA-256 ist sicherer und länger gültig, jedoch können nicht alle Chipkarten diesen Algorithmus bedienen.



## 3. Zertifikat auswählen

Dieses Menü hängt von der Zertifikatsquelle ab. Wählen Sie "Benutzerzertifikat", erhalten Sie in diesem Feld eine Auflistung aller Benutzerzertifikate auf Ihrem PC und können das entsprechende Zertifikat auswählen.

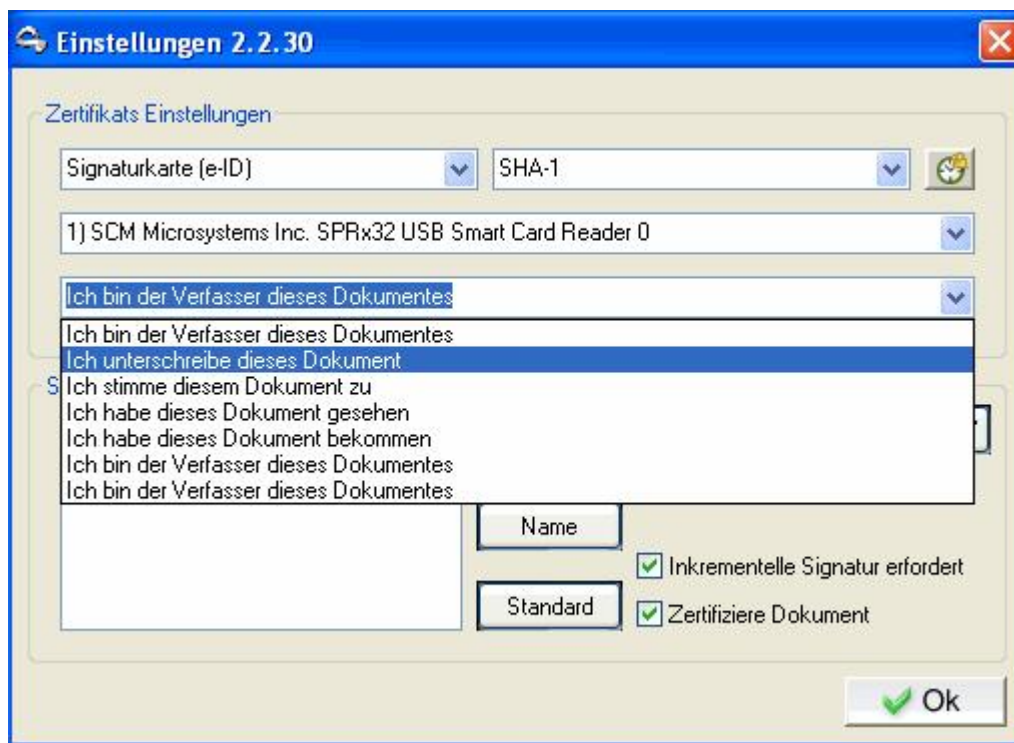
Wählen Sie als Zertifikat die SmartCard (e-ID), erscheint eine Auflistung aller zur Zeit installierten SmartCard-Lesegeräte auf Ihrem Rechner. Der Aloaha Smart Card SDK erkennt selbstständig die im Kartenleser eingelegte Smart-Card und kann die Zertifikate von unterstützten Karten lesen.

#### 4. Zweck der Signatur

Folgende Möglichkeiten stehen zur Auswahl:

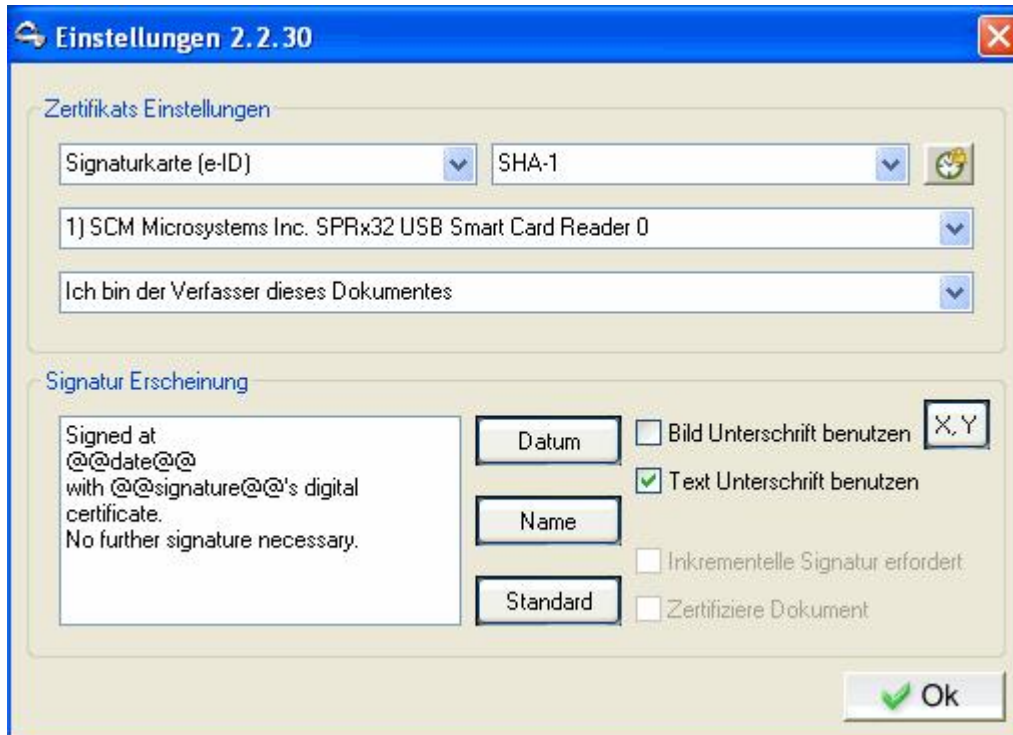
- Ich bin der Verfasser dieses Dokumentes
- Ich unterschreibe dieses Dokument
- Ich stimme diesem Dokument zu
- Ich habe dieses Dokument gesehen
- Ich habe dieses Dokument bekommen

Man kann auch frei einen Text eingeben



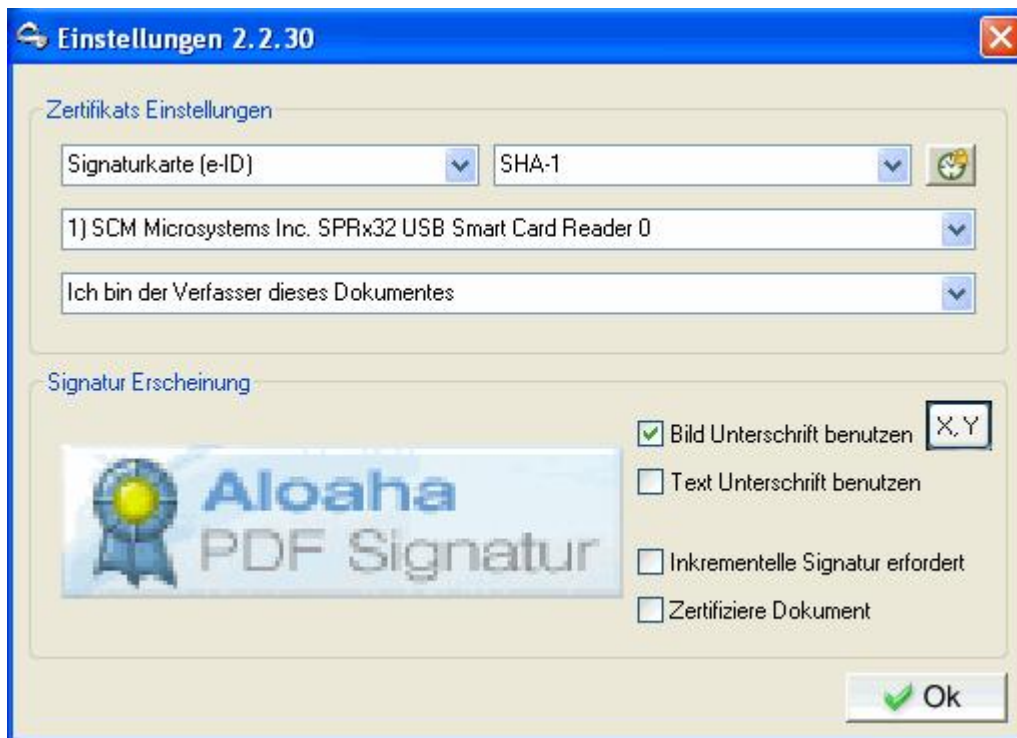
## Text Unterschrift

Ist die Option "Text Unterschrift benutzen" gewählt, wird der in dem erscheinenden Feld eingegebene Text in das PDF eingesetzt. Sie haben die Möglichkeit, an der aktuellen Cursorposition durch Klick auf "Datum" und "Name" einen Platzhalter für Datum und Namen einzufügen. Im Signaturvorgang wird dieser Platzhalter durch das aktuelle Datum und der Name des Zertifikatinhabers ersetzt.



## Bild Unterschrift

Sie haben natürlich auch die Möglichkeit, das Dokument durch eine Bildunterschrift zu signieren. Näheres zu den Einstellungen finden Sie unter "Signatureinstellungen"



## 5. Einstellungen für den Zeitstempel



Wenn Sie auf das Uhren-Symbol neben der Auswahl für die Art des Zertifikates klicken öffnet sich ein weiteres Fenster:

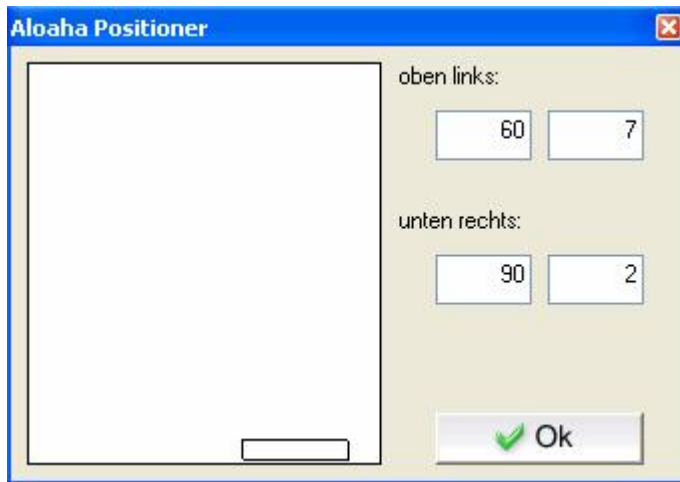
Näheres zu den Einstellungen des Zeitstempels finden Sie unter "[Zeitstempel](#)"

## 6. Position der Signatur



In den vier Feldern geben Sie die Position der Signatur vor. Hierbei wird immer in % der Seitengröße gerechnet. Das Koordinatensystem startet mit 0% links unten auf dem PDF. Unter "oben links" konfigurieren Sie die linke obere Ecke des Signaturfeldes, angefangen in der X-Achse. Unter "unten rechts" stellen Sie die Position der unteren rechten Ecke des Signaturfeldes ein. Wenn also in allen Feldern 45 eingetragen wird, erscheint das Feld in der Mitte des Blattes.

Alternativ können Sie die Position auch mit der Maus bestimmen. Klicken Sie mit der rechten Maustaste um die bisherige Wahl zu löschen. Nun fahren Sie mit der Maus die gewünschte obere linke Ecke der gewünschten Position an und klicken einmal mit der linken Maustaste. Danach fahren Sie die rechte untere gewünschte Position an und klicken noch einmal mit der linken Maustaste. So haben Sie die Position dann festgelegt.



### 3.3.2 Signatureinstellungen

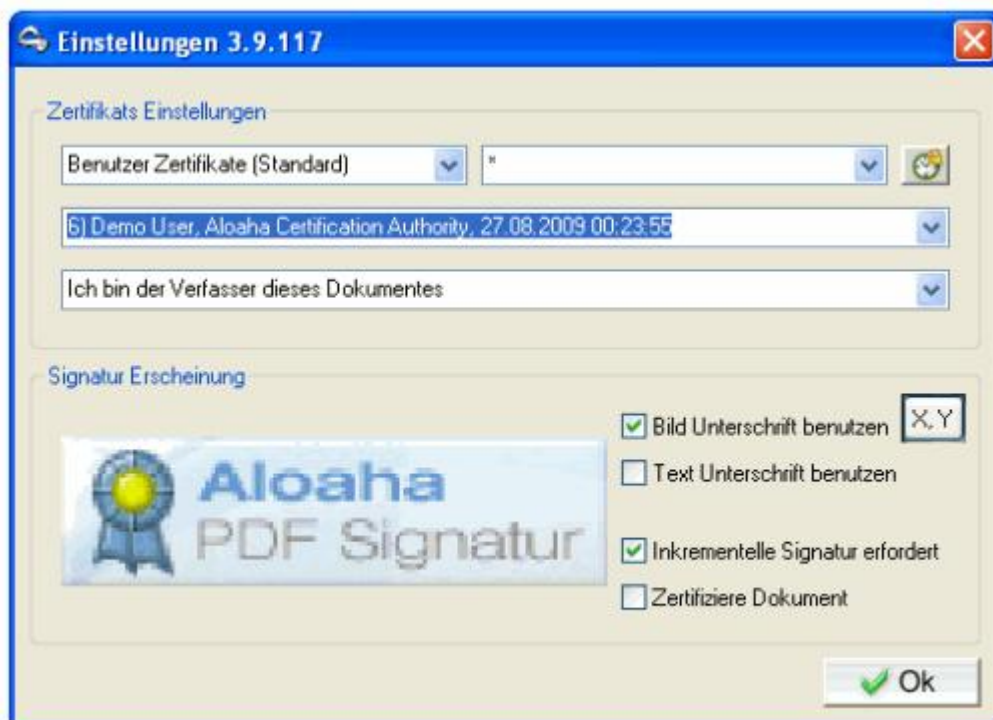
Wenn Sie in der Menüleiste mit der rechten Maustaste auf das Symbol klicken, gelangen Sie zu den Signatur-Einstellungen

In den Zertifikats-Einstellungen kann die Art des Zertifikates gewählt werden, mit der ein Dokument signiert werden soll.

Das zu signierende Dokument kann im Explorer mit Klick auf die rechte Maustaste gewählt werden.

Für den Fall, dass Sie ein PDF Dokument signieren möchten, ist es möglich, zwischen unterschiedlichen Signaturen zu wählen.

Um PDF Dokumente signieren zu können, ist es erforderlich, dass der Aloaha PDF Signator oder die PDF Suite installiert ist.



Anstatt eines sich im System befindlichen Zertifikates kann ggf. auch ein Kartenlesegerät als Signatur-Datenquelle gewählt werden.

Dies kann bei Verwendung mehrerer Signatur-Karten hilfreich sein. Das Karten-Lesegerät ist in den Grundeinstellungen als Datenquelle bereits definiert.

In diesem Fall nutzt Aloaha die Signatur der Karte des konfigurierten Lesegerätes.

Der Vorteil besteht darin, dass der Anwender die Unterschriftseinstellungen bei Nutzung weiterer Karten nicht erneut konfigurieren muss.



Um die Signatur zu ändern, klicken Sie auf die Grafik. Anschließend können Sie das Bild austauschen.

#### **Bild Unterschrift benutzen**

Ist dieses Feld aktiviert, wird ein Bild in das PDF eingesetzt, so wie es die Vorschau in diesem Dialog zeigt. Durch Klick auf die Anzeige des aktuellen Unterschriftsbildes können Sie eine eigene Bild-Datei von Ihrer Festplatte laden. Dieses Bild muß im 24 Bit JPG Format sein und wird dann als Bild in das PDF gesetzt.

#### **Text Unterschrift benutzen**

Ist diese Option aktiviert, wird der in dem darüber erscheinenden Feld eingegebene Text in das PDF eingesetzt. Sie haben die Möglichkeit, an der aktuellen Cursorposition durch Klick auf "Datum" und "Name" einen Platzhalter für Datum und Namen einzufügen. Hier wird dann im Signaturvorgang dieser Platzhalter durch das aktuelle Datum und der Name des Zertifikatinhabers ersetzt.

#### **Inkrementelle Signatur erfordert**

Aloaha wird das Dokument inkrementell signieren. Dabei wird die Signatur so an das Dokument angehängt das sich jederzeit das Originaldokument wiederherstellen lässt!

Mit diesem Programm ist es möglich, die Signatur mit einem Zeitstempel zu versehen. Um den Zeitstempel zu konfigurieren, klicken Sie auf das Uhrensymbol.

Der <https://tsa.aloaha.com> ist ein unabhängiger Zeitstempel-Server, der von Aloaha bereitgestellt wird.

Falls Sie <http://AloahaTimestamper> gewählt haben, verwendet das Programm die lokale Systemzeit, um die Signatur mit einem Zeitstempel zu versehen.

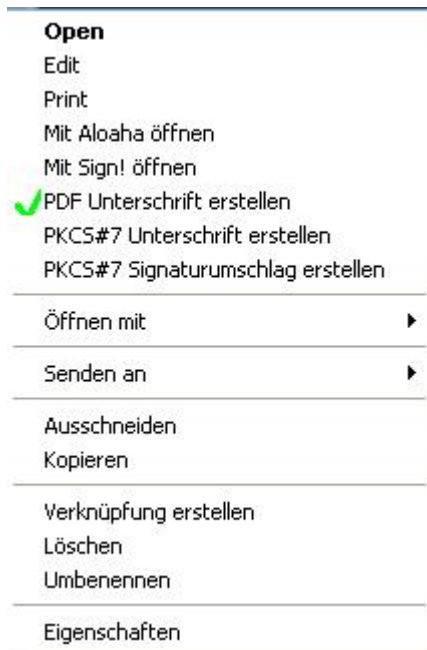
**Hinweis:** Viele Zeitstempel-Berechtigungen sind nicht RFC 3161 kompatibel, daher erteilt Aloaha dem Anwender keine Befugnis, weitere Berechtigungen zu vergeben.

Für weitere Berechtigungen, wenden Sie sich schriftlich an [aloaha@wrocklage.de](mailto:aloaha@wrocklage.de)



### 3.4 PKCS #7 Signatur erstellen

Der Aloaha Smart Card SDK wird während der Installation in Systemumgebung und damit in den Windows Explorer automatisch eingebunden.

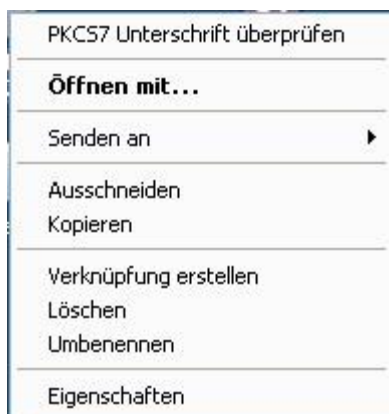


Mit einem Klick auf die rechte Maustaste öffnet sich im Windows Explorer ein Kontextmenü. Die Aloaha Shell Extension erlaubt es dem Anwender, eine PKCS #7 Signatur sowie einen PKCS #7 Signaturumschlag zu erstellen.

Aloaha verwendet dabei das über den Unterschriften-Einstellungsdialog konfigurierte Zertifikat.

Im Falle dass der Aloaha PDF Signator oder die Aloaha PDF Suite installiert sind und auf derselben Maschine lizenziert wurden, ist ein dritter Eintrag zur Erstellung einer PDF Unterschrift verfügbar. Durch Klicken darauf wird eine PDF Unterschrift erstellt.

#### PKCS #7 Bestätigung



Die Aloaha Shell Extension ermöglicht weiterhin PKCS#7 Signaturen auszulesen. Mit einem Klick auf die **rechte Maustaste**>**PKCS7 Unterschrift überprüfen** wird die Signatur angezeigt. Sollte die Signatur eine kuvertierte Signatur sein, entpackt Aloaha die Original Signatur und speichert diese.

**Signatur Gültigkeitsprüfung ist ein Freeware-Tool von Aloaha.**

## 3.5 PIN Verwaltung

Über das Auswahlm Menü **PIN Verwaltung** können Sie folgende PIN's verwalten:

Signatur-PIN  
Karten-PIN  
PIN Home

Hier können PIN's geändert oder zurückgesetzt werden.



Nachdem Sie mit der Maus auf den Button **PIN ändern** geklickt haben, erscheint folgendes Bild



Geben Sie zunächst die alte PIN über das Kartenlesegerät ein und bestätigen Sie die Eingabe mit der grünen Taste.

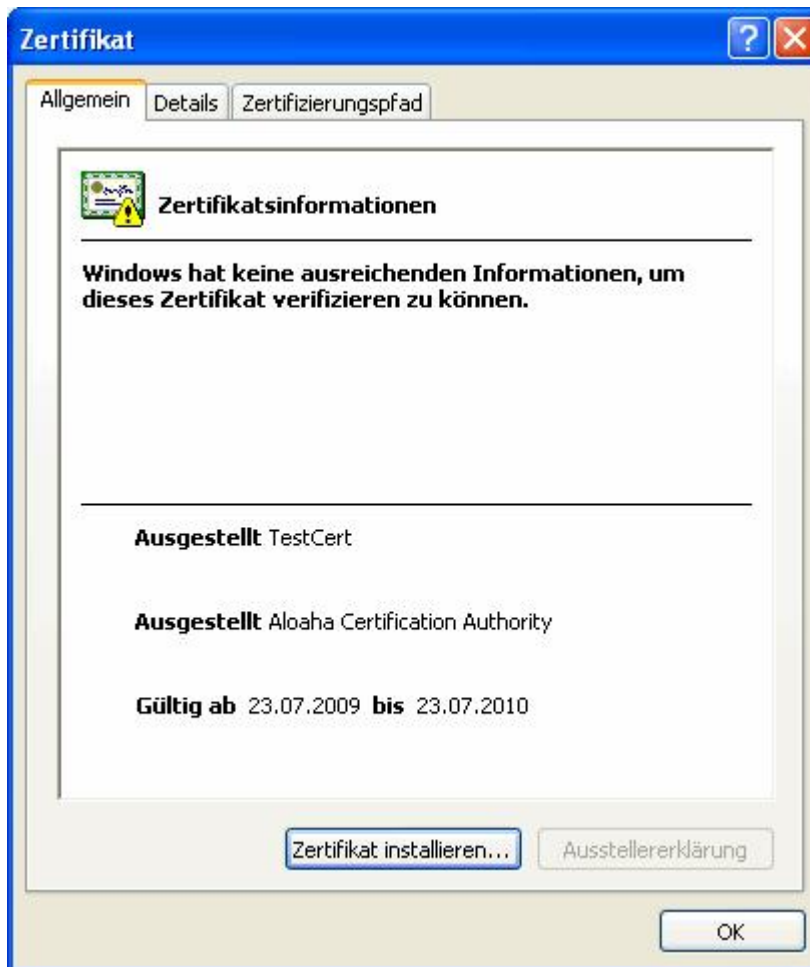
Anschließend geben Sie die neue PIN ein. Sie werden aufgefordert, diese durch nochmalige Eingabe zu bestätigen.

Danach ist der Vorgang abgeschlossen.

## 3.6 Zertifikate

### Allgemeine Zertifikatsinformationen

Um Informationen zu den Zertifikaten zu erhalten, rufen Sie den Card Assistant auf. Klicken Sie auf eines der im Card Assistant enthaltenen Zertifikate und Sie erhalten im Reiter "**Allgemein**" die Informationen zum ausgewählten Zertifikat. Hier wird angezeigt, wie lange das Zertifikat noch gültig ist, ob es ggf. abgelaufen ist, wer das Zertifikat ausgestellt hat und welchen Namen das Zertifikat trägt.



Im Reiter "Details" finden Sie weitere Informationen zum jeweiligen Zertifikat, wie z.B.:

Version

Seriennummer

Signaturalgorithmus

Aussteller

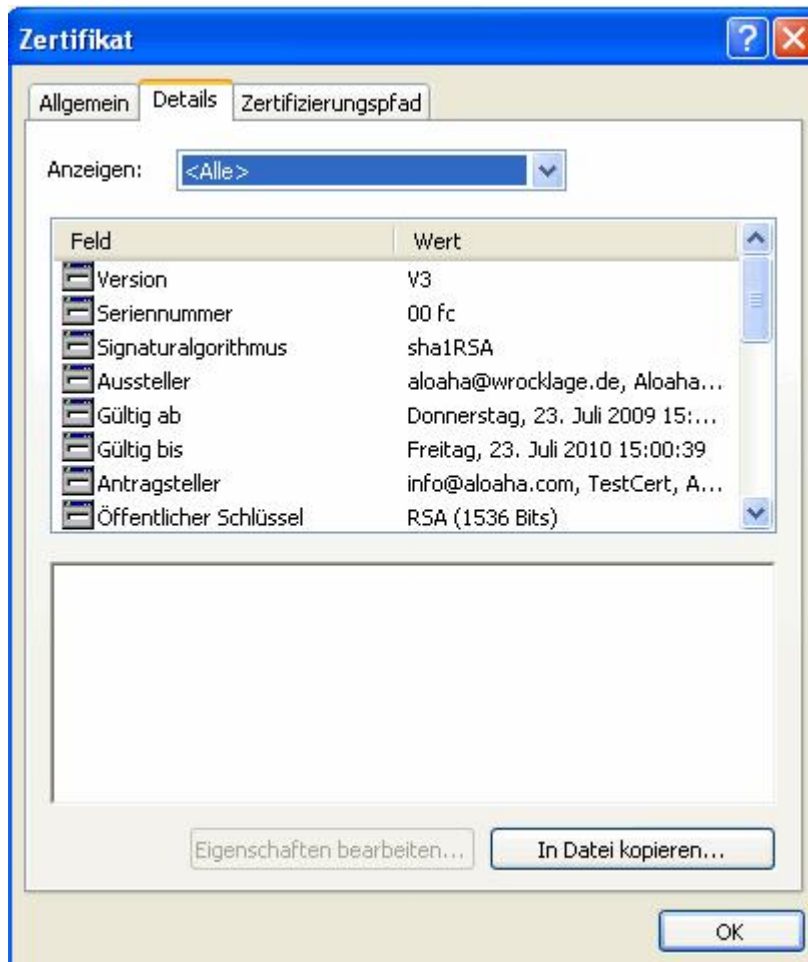
gültig ab

gültig bis

Antragsteller

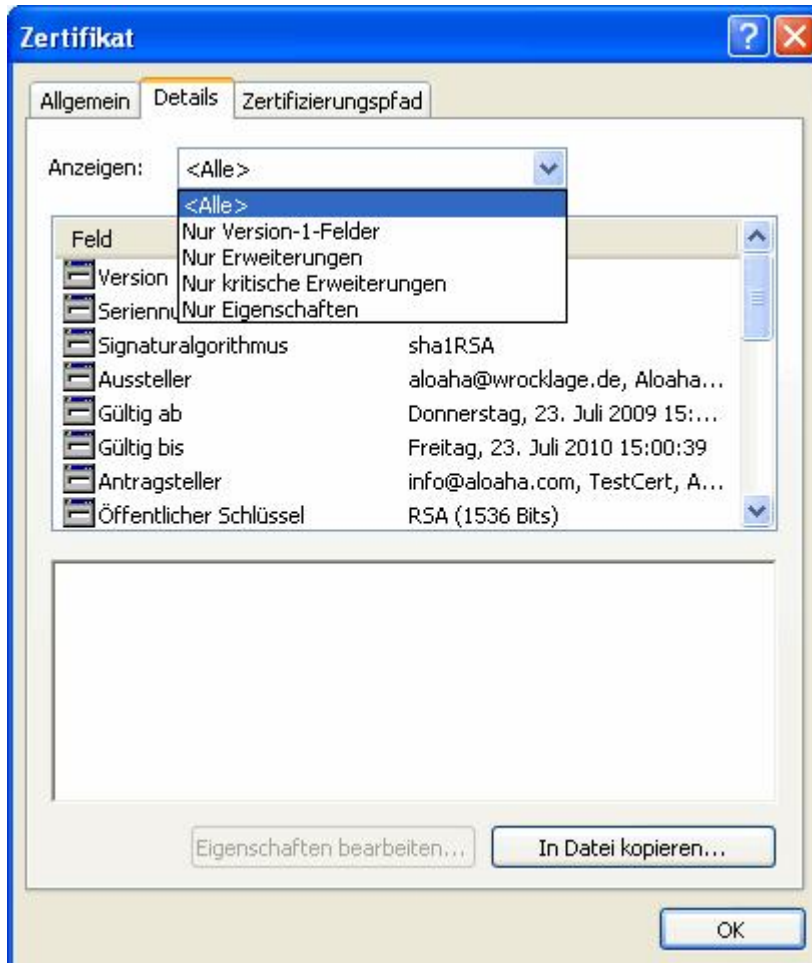
öffentlicher Schlüssel

...



Sie können sich durch das Auswahlménú folgende Informationen zum jeweiligen Zertifikat anzeigen lassen:

- Alle
- Nur Version-1-Felder
- Nur Erweiterungen
- Nur kritische Erweiterungen
- Nur Eigenschaften



Zum Zertifikatsexport-Assistent gelangen Sie, wenn Sie in vorher gezeigtem Bild "In Datei kopieren" wählen.



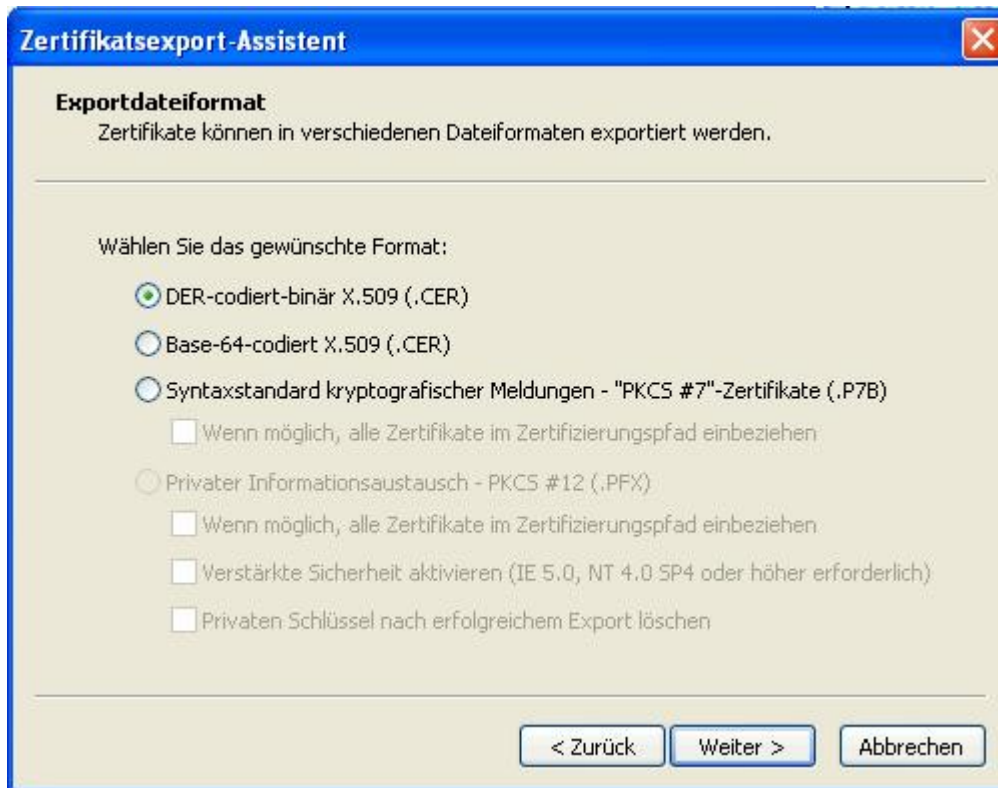
Nachdem Sie mit "Weiter" bestätigt haben, öffnet sich ein Fenster, wobei Sie dann das Exportdateiformat auswählen können.

Es stehen drei mögliche Exportformate zu Verfügung.

DER-codiert-binär X.509 (.CER)

Base-64-codiert X.509 (.CER)

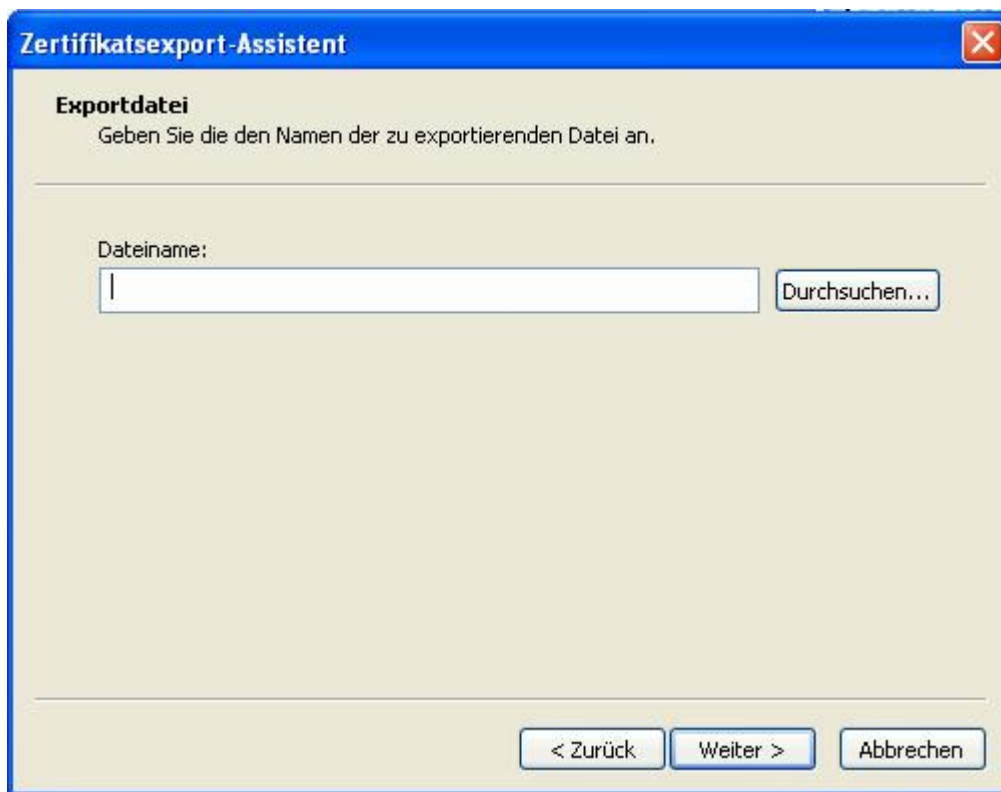
Syntaxstandard kryptografischer Meldungen - PKCS #7-Zertifikate (.P7B)



Nachdem Sie das Dateiformat ausgewählt haben, bestätigen Sie mit "Weiter" und gelangen zu folgender Anzeige.

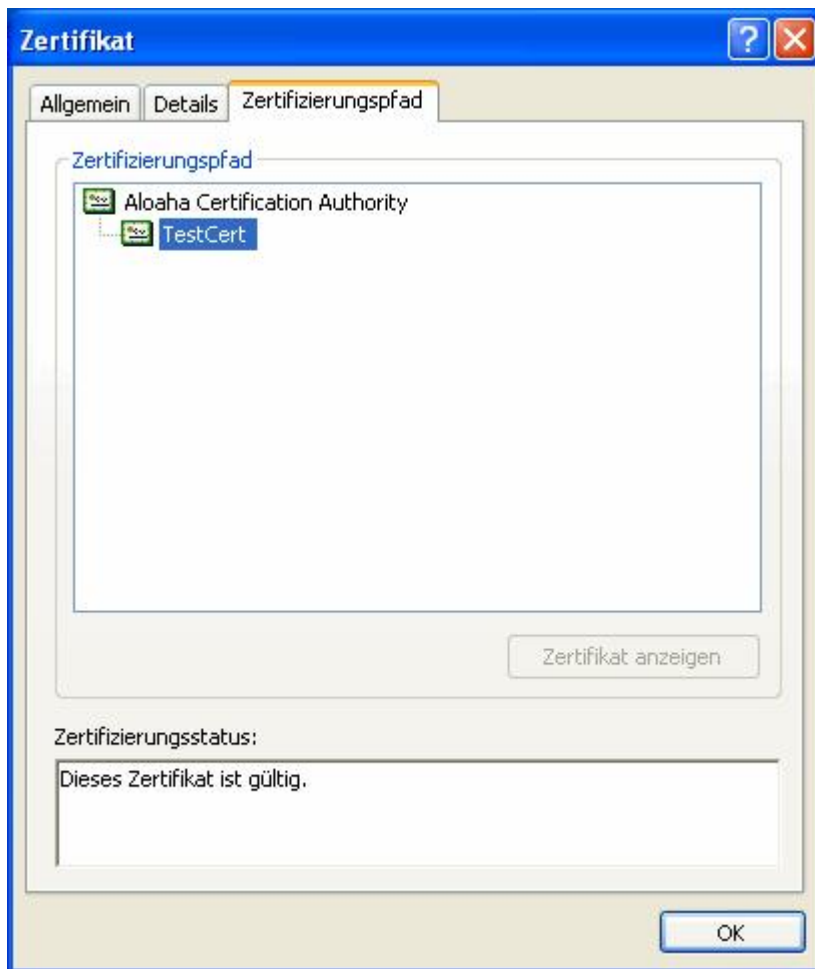


Hier müssen Sie der Datei einen Namen geben und in einem Verzeichnis Ihrer Wahl speichern.



The image shows a Windows-style dialog box titled "Zertifikatsexport-Assistent". The main heading is "Exportdatei" with the instruction "Geben Sie die den Namen der zu exportierenden Datei an." Below this is a text input field labeled "Dateiname:" which is currently empty. To the right of the input field is a button labeled "Durchsuchen...". At the bottom of the dialog, there are three buttons: "< Zurück", "Weiter >", and "Abbrechen".

Der Zertifizierungspfad gibt an, wer Ersteller des Dokumentes ist.

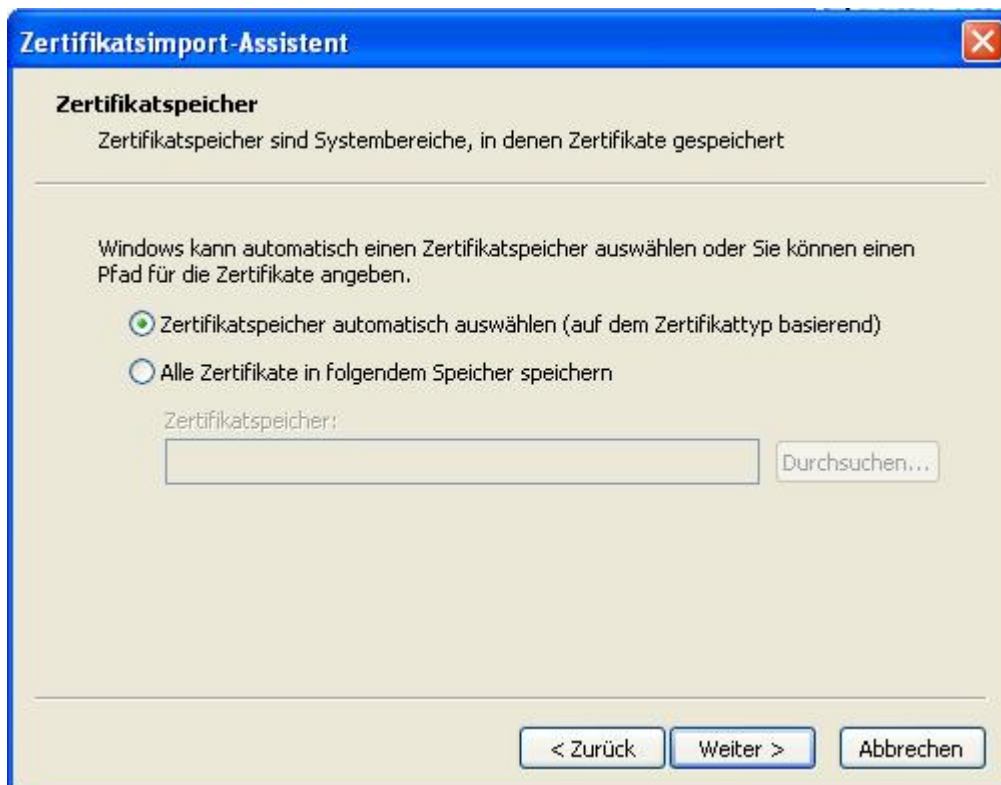


Zum Zertifikatsimport-Assistent gelangen Sie, indem Sie im Reiter "Allgemein" auf Zertifikat installieren klicken.

Anschließend öffnet sich folgendes Fenster. Mit "Weiter" setzen Sie den Vorgang fort.



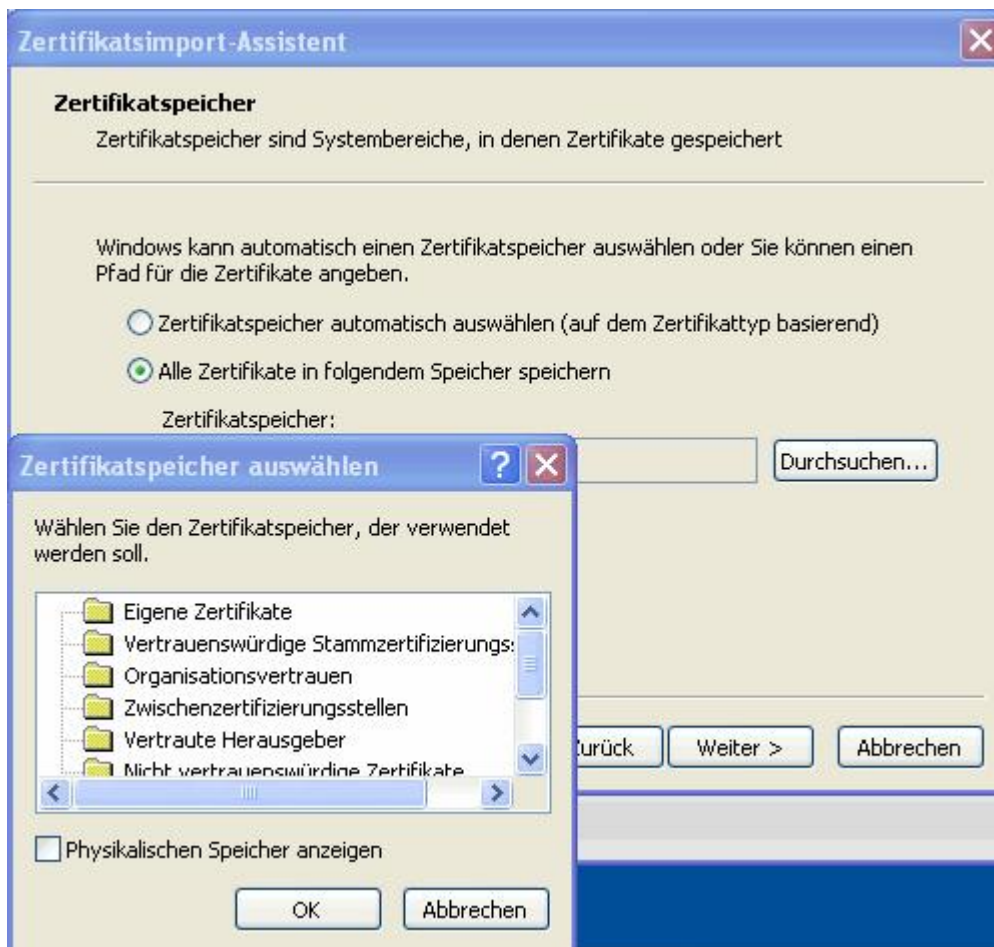
Hier wählen Sie den Zertifikatspeicher entweder basierend auf dem Zertifikattyp automatisch aus oder Sie legen fest, an welchem von Ihnen vorgegebenen Speicherort Zertifikate gespeichert werden sollen.



In diesem Fall wurde der automatische Zertifikatspeicher gewählt. Sie sehen, die von Ihnen vorgenommenen Einstellungen bezogen auf das Zertifikat. Mit "Fertigstellen" schließen Sie den Vorgang ab.



Sie haben nicht nur die Möglichkeit, Zertifikate automatisch durch den Assistenten zu speichern. Falls Sie sich dazu entscheiden, den Zertifikatspeicher selbst auszuwählen, aktivieren Sie das entsprechende Feld und wählen den Speicherort selbst aus der dann erscheinenden Liste aus. Anschließend bestätigen Sie die Auswahl mit OK.



## 3.7 CSP / Kartenleser

### Unterstützte Kartenleser

Aloaha unterstützt derzeit ca. 45 Kartenleser der Sicherheitsklasse 2 und 3. Sie wurden nach dem deutschen Signaturgesetz bestätigt und dürfen zur Erzeugung qualifizierter elektronischer Signaturen eingesetzt werden.

Hier einige Beispiele:

Chipkartenleser OmniKey Cardman 3621 Trust

Chipkartenleser Omnikey CardMan® 3821 USB

Chipkartenleser SCM CHIPDRIVE® pinpad pro

CHERRY® Smart Terminal ST-2000UC-R

CHERRY® Tastatur G83-6744 LUADE-2 USB DE

Reiner SCT cyberjack

Reiner SCT cyberJack® e-com

alle PC/SC konformen Leser

Wenn Sie im Traymenü mit der rechten Maustaste auf das Icon klicken, gelangen Sie zu den Einstellungen des Karten-Lesegeräts.

Wenn mehrere Karten-Lesegeräte verfügbar sind, werden alle angezeigt und Sie können das gewünschte auswählen.



Sobald eine Karte in ein Lesegerät gesteckt wurde, registriert das Programm automatisch alle auf der Karte befindlichen Zertifikate. Die Zertifikate können jedoch auch manuell registriert werden. Statt Autoregister klicken Sie stattdessen auf Register.

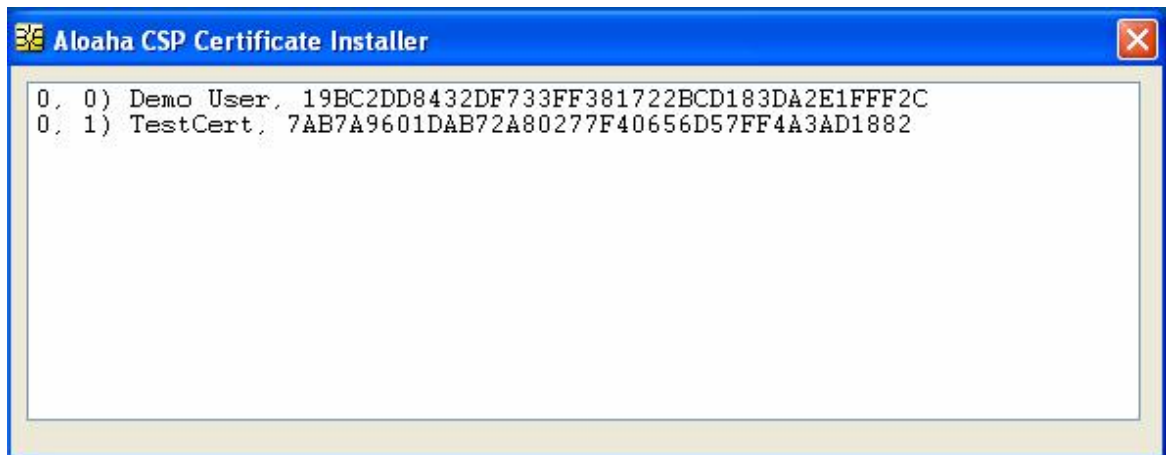
Die erste Zahl zeigt die Anzahl der Kartenlesegeräte an. Der nachfolgende Screenshot zeigt die Zertifikate der Karte(n) in angeschlossenen Kartenlesern. Die Zahl nach dem Komma zeigt den Zertifikat-Typ an.

Typ 0 = Unterschriftszertifikat,  
Typ 1 = Authentifizierungszertifikat,  
Typ 2 = Verschlüsselungszertifikat.

Enthält eine Karte nur ein Zertifikat enthält, wird dieses als Typ 1 angezeigt.

Um alle registrierten Zertifikate zu entfernen, klicken Sie auf "alle entfernen". Ist Autoentfernen aktiviert, werden alle registrierten Zertifikate gelöscht, sobald sämtliche Karten aus den Kartenlesegeräten entfernt wurden.

## Kartenleser



In einigen Fällen gibt es mehrere mit einem System verbundene Kartenlesegeräte. Die Zertifikate aller Lesegeräte aufzuzählen, nimmt Zeit in Anspruch. In diesem Fall können Sie das Kartenlesegerät direkt auswählen. Aloaha liest dann nur die Zertifikate der Karte im gewählten Leser aus.

Sie können das Zertifikat nun anklicken, um es anzeigen zu lassen oder es per Doppelklick im aktuellen Verzeichnis zu registrieren.



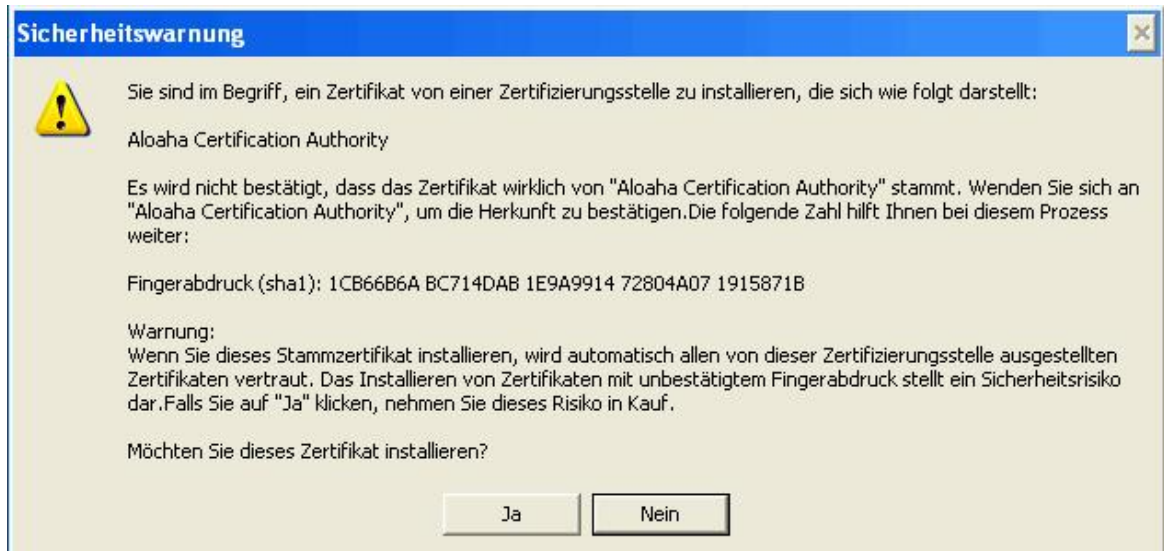
### Manuelle Registrierung hat Vorteile:

1. Wenn das Ausgabezertifikat im System nicht verfügbar ist, wird Aloaha versuchen, es von der Aloaha Website herunterzuladen.
2. Das eingetragene Zertifikat wird automatisch als Standardzertifikat konfiguriert.



### Sicherheitswarnung bei Zertifikat anzeigen / registrieren

Wenn Sie ein Zertifikat registrieren, erhalten Sie folgende Sicherheitswarnung. Lesen Sie sich den Inhalt durch und entscheiden anschließend, ob Sie das Zertifikat registrieren / installieren möchten oder nicht. Der Dialog erscheint NUR wenn **ERSTMALIG** ein neues Root Zertifikat eingepflegt wird!



## 3.8 Zeitstempel

### Einstellungen für den Zeitstempel

Wenn Sie auf das Uhren-Symbol  im Signatur-Konfigurationsmenü klicken öffnet sich nachfolgend gezeigtes Fenster für die Zeitstempелеinstellungen:

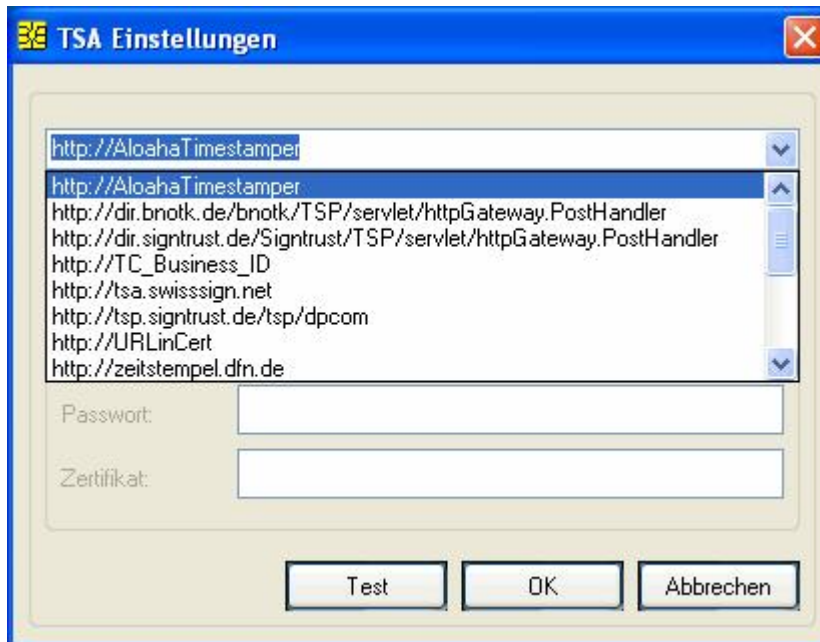


The screenshot shows a dialog box titled "TSA Einstellungen". At the top left is a logo and the title. Below the title bar is a dropdown menu containing the text "http://AloahaTimestamper". Underneath the dropdown is a checkbox labeled "Aktiviere Zeitstempel" and a button labeled "Lade TSA Liste vom Server". Below these are three input fields: "Benutzername:", "Passwort:", and "Zertifikat:". At the bottom of the dialog are three buttons: "Test", "OK", and "Abbrechen".

Hier können Sie die Einstellungen für den integrierten RFC 3161 kompatiblen Zeitstempel Client anpassen.

Im oberen Feld wählen Sie einen verfügbaren Zeitstempelservers aus. Ist die Liste leer, können Sie die Liste der möglichen Zeitstempelservers durch Klick auf den Button "Lade TSA Liste von Server" von der Aloaha Webseite herunterladen.

Wenn Sie `http://AloahaTimestamper` auswählen, wird der TimeStamp-Server benutzt. Hierbei wird die Lokale Systemzeit als Grundlage für den Zeitstempel genommen.  
Unter Benutzerdaten konfigurieren Sie Ihre Zugangsdaten zum jeweiligen Zeitstempeldienst.



## 4. Anwender Support

Der Aloaha Smart Card SDK bindet transparent die Zertifikate und das Schlüsselmateriale in das Betriebssystem ein. Dadurch ist es möglich, dass Windows Anwendungen von den Smartcard Zertifikaten Gebrauch machen können, als ob sie Softwarezertifikate wären. Dank der Aloaha-Software registrieren die Windows Anwendungen nicht, dass Zertifikate und Schlüssel auf einer Smartcard verwaltet werden.

Die meisten Open Source Anwendungen machen keinen Gebrauch vom Windows Cryptographic System, nutzen dafür aber den PKCS #11 Standard. Für jene Anwendungen installiert Aloaha seine PKCS #11 Bibliothek im Windows-Verzeichnis "System32". Der Name der Aloaha Bibliothek lautet aloaha\_pkcs11.dll.

**Flashbasierte Anwendungsbeispiele finden Sie im Internet unter:**  
<http://www.aloaha.com/wi-software-en/csp-usage.php>

## 4.1 MS Crypto API

Das Cryptographic Application Programming Interface (auch bekannt unter Crypto API, Microsoft Cryptography API oder einfach nur CAPI), nachfolgend API oder CAPI genannt, ist ein API welches im Microsoft Windows Betriebssystem eingebunden ist, um Entwicklern den sicheren Gebrauch von Kryptographie-Modulen zu ermöglichen.

Crypto-API unterstützt sowohl symmetrische als auch allgemeine Kryptographie. Es schließt sowohl Funktionalitäten zum Ver- und Entschlüsseln als auch Beglaubigungen für Digitalzertifikate ein.

Crypto API wird durch das auf dem System installierten Aloaha CSP ergänzt. CSPs sind Module, welche die Verschlüsselung und Entzifferung von Daten durch kryptografische Funktionen ermöglichen. Sie sind weiterhin für die Kommunikation zwischen Smartcards und dem Windows Betriebssystem verantwortlich.

## 4.1.1 Outlook Einstellungen

### Signatur Einstellungen in Microsoft Outlook

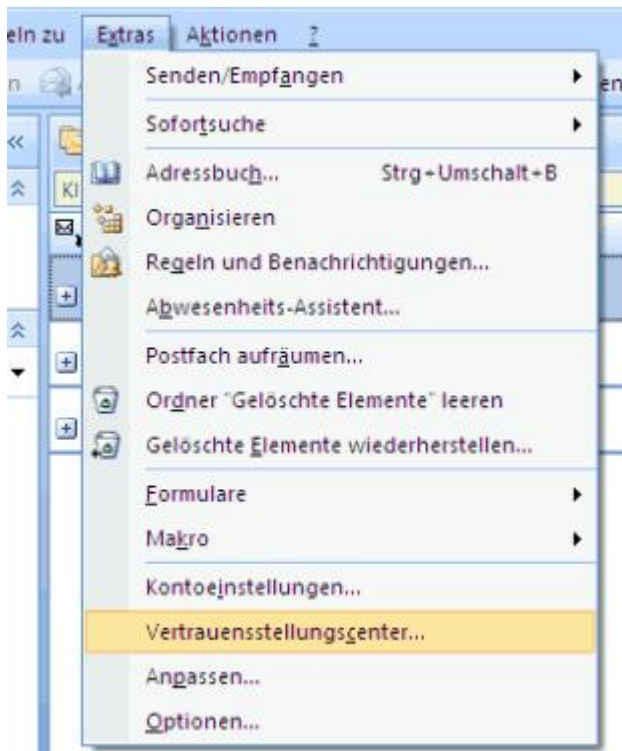
In Microsoft Outlook ist es sehr einfach eine Signatur oder ein Verschlüsselungs-Zertifikat zu erstellen.

Um Anwendungsbeispiele anzusehen, verwenden Sie nachfolgenden Link:

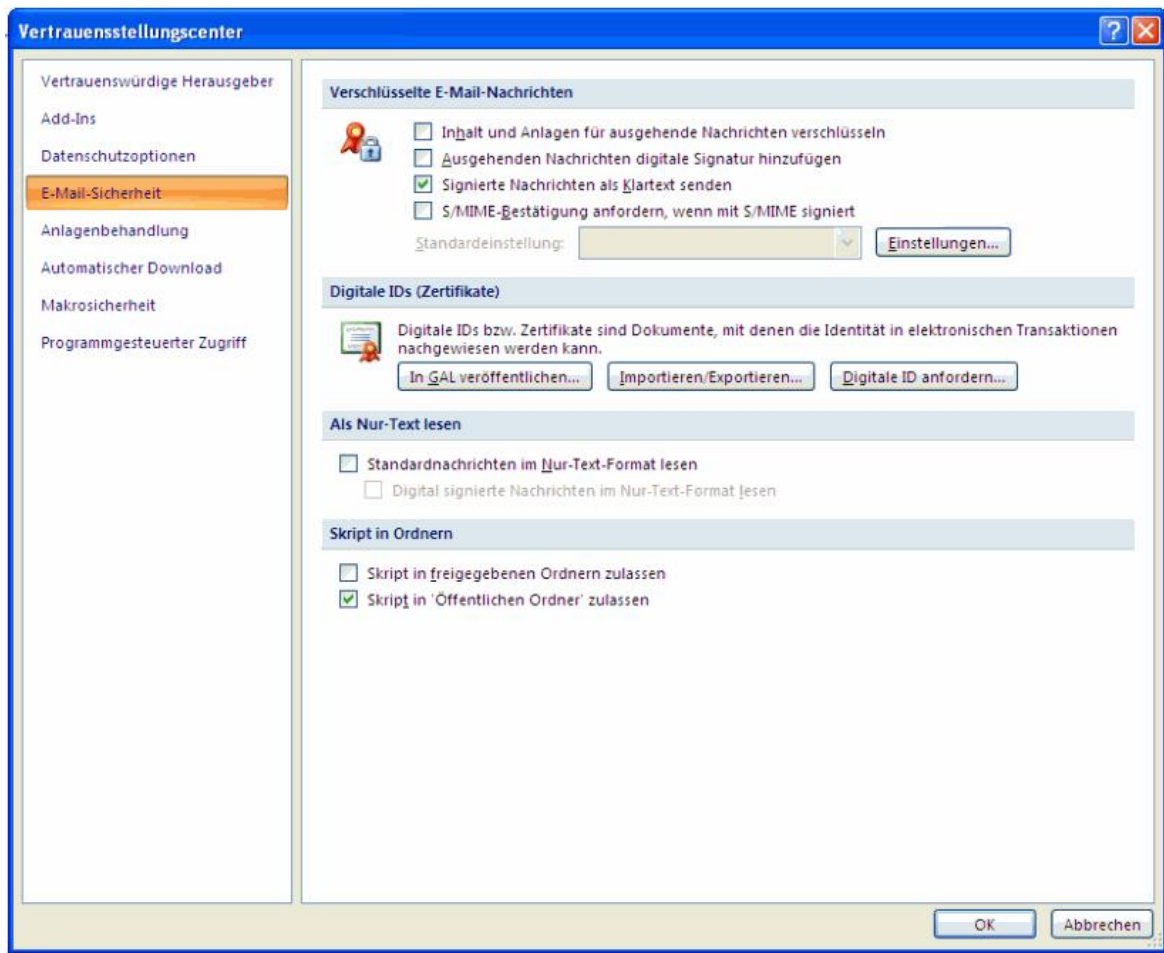
<http://www.aloaha.com/movies/outlook.htm>

Um eine Signatur oder ein Verschlüsselungs-Zertifikat zu erstellen klicken Sie auf das Startmenü von Windows:

**Start>Programme>Microsoft Office>Outlook>Extras>Vertrauensstellungscenter>E-Mail-Sicherheit**

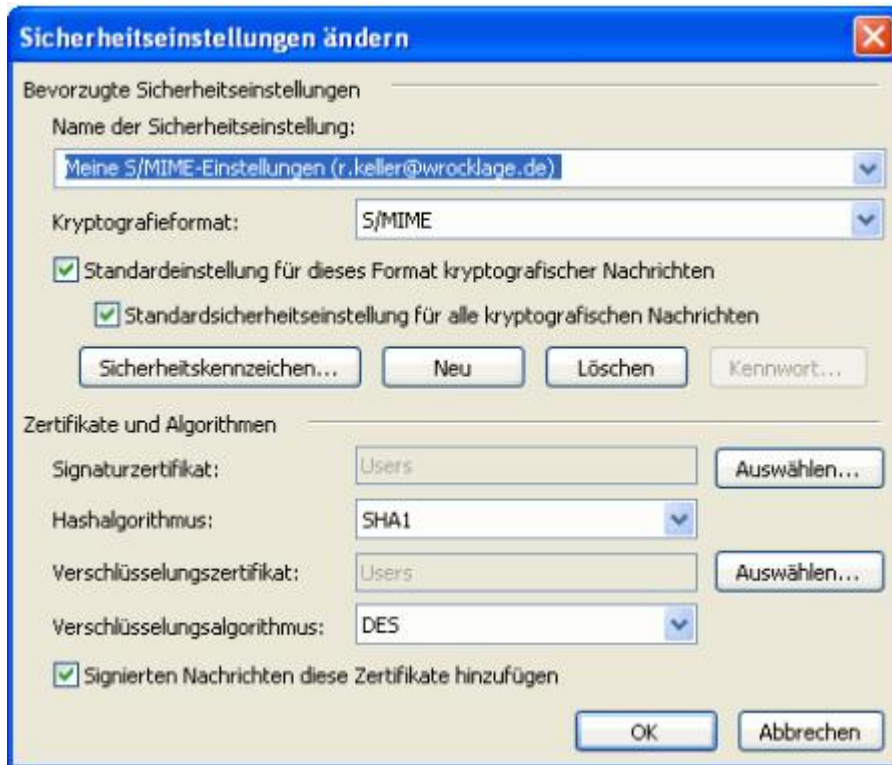


Im Falle dass Outlook standardmäßig jede ausgehende Nachricht signieren soll, aktivieren Sie das Kästchen "Ausgehenden Nachrichten digitale Signatur hinzufügen". Wenn Sie die Signatur lediglich vorkonfigurieren möchten, lassen Sie das Kästchen deaktiviert. Um Zertifikate und den verwendeten Algorithmus zu wählen klicken Sie auf den Knopf Einstellungen.



Hier können Sie das Signatur- oder das Verschlüsselungs-Zertifikat auswählen. Stellen Sie sicher, dass die Zertifikate vorher registriert wurden! Im Falle, dass ihr Zertifikat nicht angezeigt wird, kann es sein, dass ein wichtiges Attribut fehlt, das Zertifikat ungültig oder abgelaufen ist.

Einige E-Mail-Programme verlangen, dass die konfigurierte E-Mail-Adresse die E-Mail-Adresse im Zertifikat vergleicht! In diesem Fall haben Sie keine Möglichkeit das Zertifikat zu anzuwählen.



Mit OK bestätigen Sie die vorgenommenen Einstellungen.

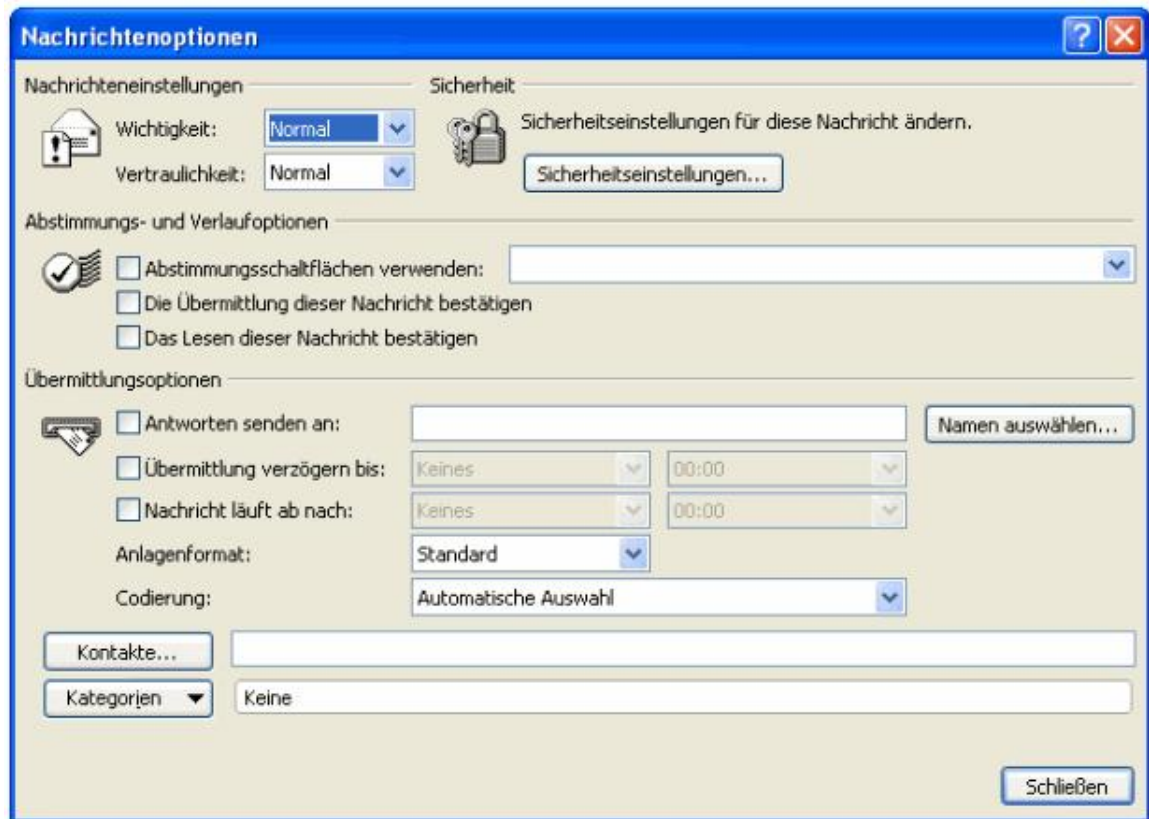


## Versenden digital signierter E-Mails

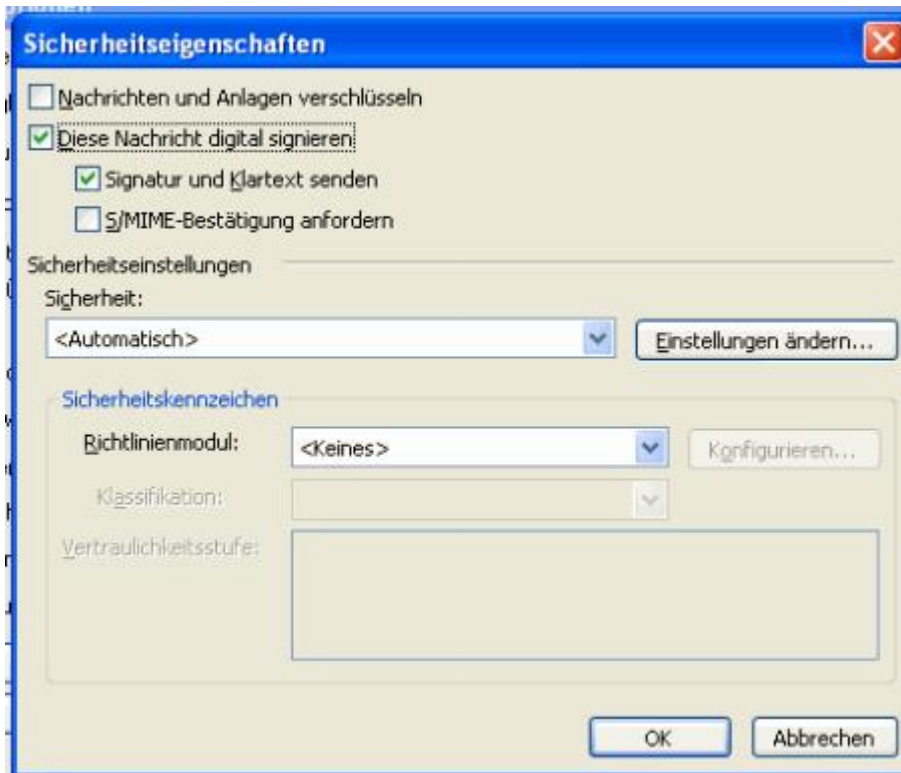
Um ein Mail digital zu signieren, klicken Sie auf das Startmenü von Windows:

**Start>Programme>Microsoft Office>Outlook>Extras>Vertrauensstellungscenar>E-Mail Sicherheit>Einstellungen**

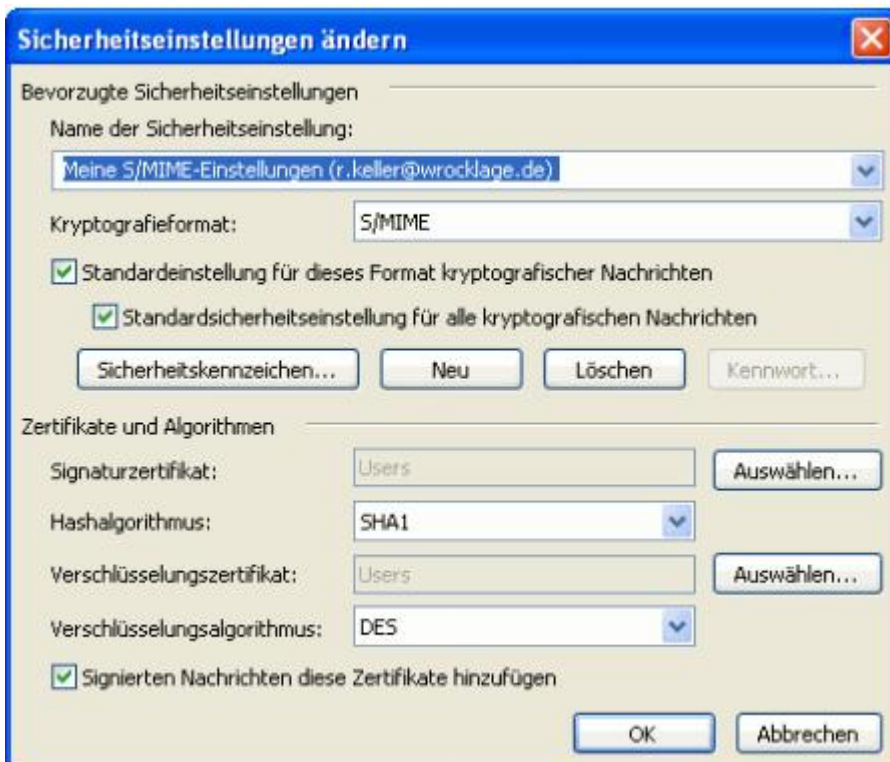
In einem neuen Dialog klicken Sie auf Sicherheits-Einstellungen.



Aktivieren Sie **Diese Nachricht digital signieren** und bestätigen Sie anschließend mit **OK**.



Ändern Sie die Zertifikate und bestätigen Sie anschließend mit **OK**.



## 4.2 PKCS #11

In der Kryptografie ist PKCS #11 eine der Standardfamilien, die Public-Key Cryptography Standards (PKCS) genannt werden, welche durch die RSA Labors veröffentlicht werden. Es definiert eine von der plattformunabhängige Zugriffsschnittstelle zu kryptografischen Kürzeln, wie Hardware-Sicherheitsmodule und Smart-Cards.

Da es keinen richtigen Standard für kryptografische Kürzel gibt, wurde diese API entwickelt, um eine Abstraktionsstufe für allgemeine kryptografischen Kürzel darzustellen. Die PKCS#11 API definiert verwendete kryptografische Objektarten (RSA Schlüssel, X.509 Zertifikate, DES/Triple DES Schlüssel, usw.), die in der Lage sein müssen Funktionen zu verwenden, erzeugen, modifizieren und Objekte zu löschen.

PKCS#11 wird größtenteils verwendet, um auf SmartCards zuzugreifen. Plattformübergreifende Software wie Mozilla Firefox und OpenSSL welche SmartCards benötigt, nutzt PKCS#11.

### 4.2.1 Firefox Einstellungen

Firefox nutzt statt der Microsoft Crypto API die PKCS #11 Programmbibliotheken. Nachfolgend wird beschrieben, wie Sie vorgehen müssen, wenn Sie die Aloaha PKCS#11 in Firefox registrieren möchten.

Anwendungsbeispiele finden Sie unter nachfolgenden Link:

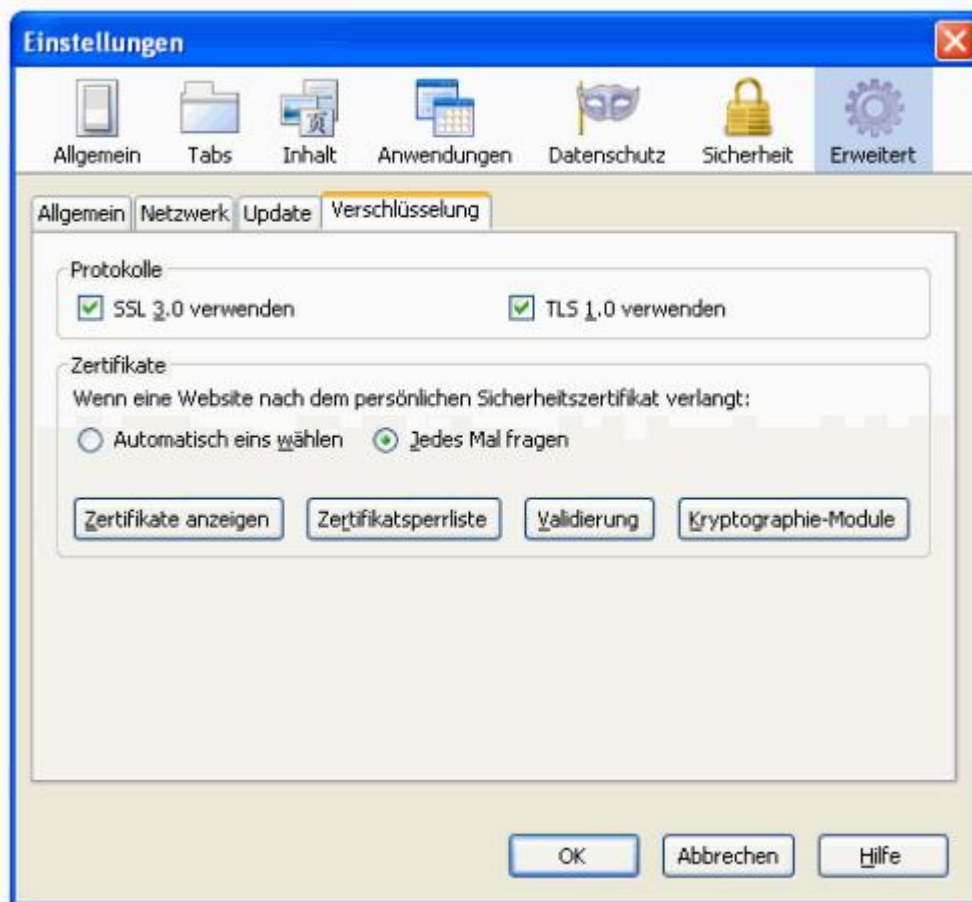
<http://www.aloaha.com/movies/firefox.htm>

Um Aloaha Module in Firefox zu registrieren klicken Sie auf das Startmenü von Windows:

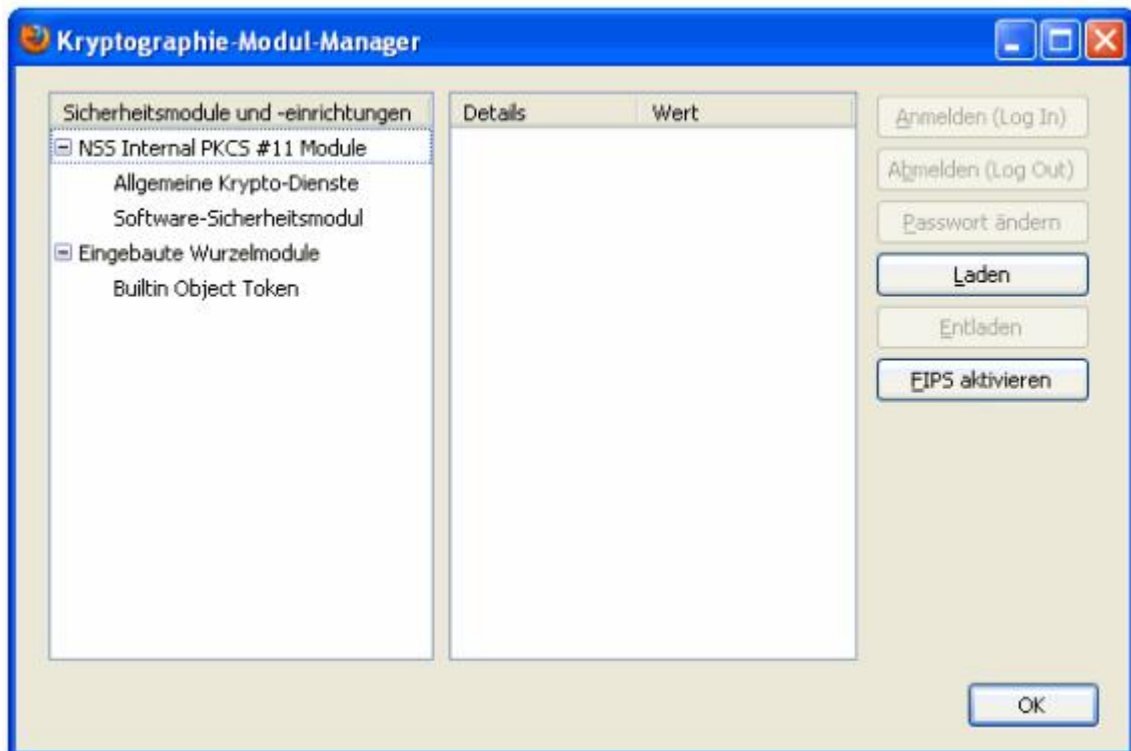
**Start>Programme>Mozilla Firefox>Extras>Einstellungen>Erweitert>Verschlüsselung**



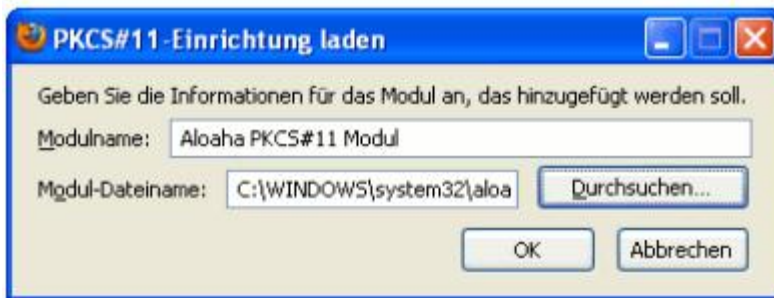
Wählen Sie jetzt Kryptographie-Module.



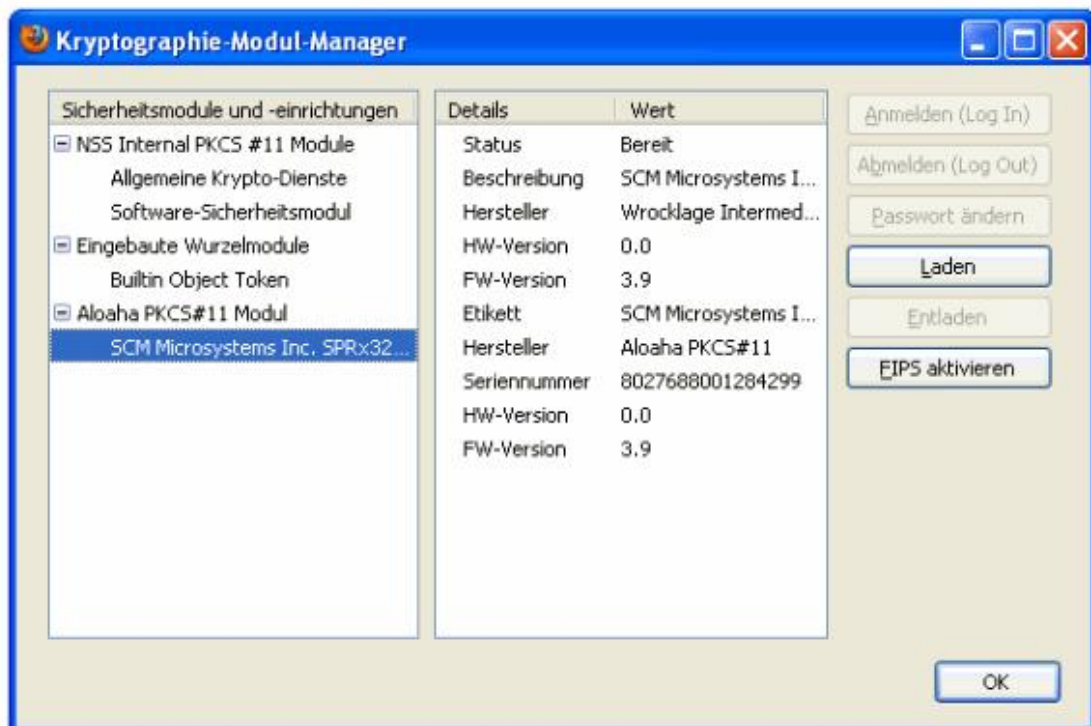
Klicken Sie nun auf Kryptographie-Module, um den Pfad für die Aloaha PKCS#11 Module zu definieren.



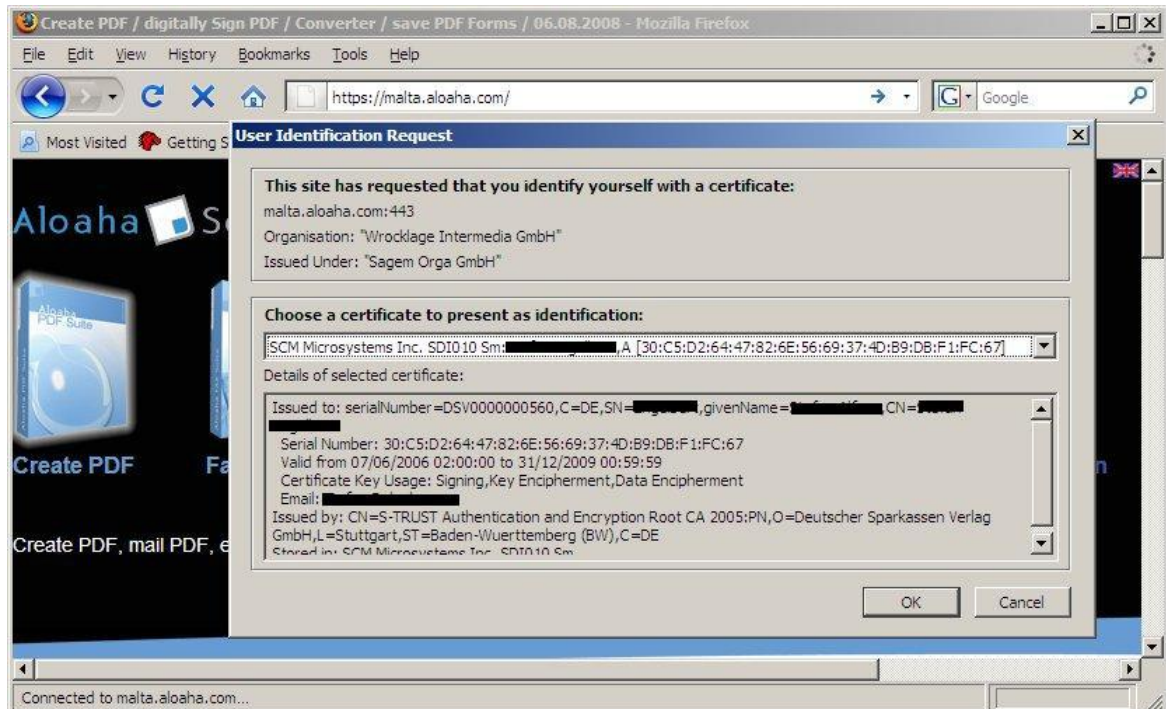
Im Feld **Modul Name** können Sie einen Namen ihrer Wahl eintragen, in **Modul-Dateinamen** ist das Verzeichnis zu wählen, in welchem die sich die Aloaha Dateien befinden. Die Der Dateiname lautet **aloaha\_pkcs11.dll** und ist im Verzeichnis **Windows\System32** abgelegt.



Nachdem Sie mit OK bestätigt haben, sehen Sie die registrierte Aloaha PKCS #11 im Kryptographie-Modul-Manager.



Wenn Sie nun eine Website besuchen, welche um ein Zertifikat verlangt, werden Sie den folgenden Dialog angezeigt bekommen. Sollten Sie diesen Dialog nicht angezeigt bekommen, vergewissern Sie sich, dass das Zertifikat gültig ist und die richtigen Attribute enthält.



## 4.3 Language.ini

### Aloaha Übersetzung / Software-Lokalisierung

Neuste Produkte von Aloaha lokalisieren und übersetzen verwendete Zeichenfolgen (strings) völlig automatisch. Die Zeichenfolgen werden als ini-Dateien gespeichert, um dem Anwender zu ermöglichen, sie zu ändern oder in eine andere Sprachen zu übersetzen, ohne den Aloaha-Code ändern zu müssen.

### Übersetzungs Mechanismus

- Beim Start des Aloaha-Smart Card SDK werden die Spracheinstellungen in der language.ini überprüft. Sollte diese Datei nicht existieren, fragt das Programm folgende Pfade ab:
  - o `HKCU\Software\Aloaha\language`
  - o `HKLM\Software\Aloaha\language`
  - o Die Betriebssystem Anwendersprache
- Basierend auf der "LanguageID" wird der Aloaha-Smart Card SDK die UserLanguage\_<ID>.ini für die Übersetzung der Zeichenfolge abfragen. Wenn diese Datei nicht die richtige Übersetzung nicht enthält, fragt der Aloaha-Smart Card SDK die Language\_<ID>.ini ab.
- Die Datei Language\_<ID>.ini wird durch jede(n) Neustart / Erweiterung überschrieben. Im Falle dass ein Benutzer Zeichenfolgen modifizieren möchte, wird darauf hingewiesen, die UserLanguage\_<ID>.ini zu verwenden.

### language.ini

Das Profil [Abbildung] weist eine Sprache an, sich in einer anderen abzubilden . Zum Beispiel 410=409 würde bedeuten, englische Sprache (409) auf italienischen (410) Systemen zu verwenden.

Das Profil [languageID] definiert welche ini Dateien gegenwärtig zu verwenden sind.

### Übersetzungs-Dateien

Zuerst wird die Aloaha UserLanguage\_<ID> für die Übersetzung abgefragt. Sollte keine Übersetzung gefunden werden, wird als nächster Schritt die Language\_<ID> für die Übersetzung abgefragt.

Wenn ein Benutzer Zeichenfolgen ändern möchte, wird empfohlen, die Änderungen in UserLanguage\_<ID>.ini durchzuführen, da die Language\_<ID>.ini mit jedem Neustart/Upgrade überschrieben wird.

Es ist auch möglich, Registrierungsschlüssel `HKLM\Software\Aloaha\pdf\WriteMissing` auf 1 zu setzen. In diesem Fall wird der Aloaha-Smart Card SDK alle Übersetzungsprobleme in der `MissedLanguage_<ID>.ini` protokollieren. Es kann sehr nützlich sein, dass Zeichenfolgen für andere Sprachen / Umgebungen übersetzt werden.



## 5. Aloaha CSP API

Der Aloaha Cryptographic Service Provider ist ein Multicard CSP um Standard-Windows-Anwendungen wie Outlook oder den Internet Explorer mit den kryptografischen Funktionen der unterstützten Smartkarten zu versorgen.

Man kann mittels Microsoft CAPICOM oder Crypto API auf den Aloaha CSP zugreifen.

Für Systemintegratoren könnte es manchmal nützlich sein, Aloahas native CSP APIs zu verwenden, um mehr Kontrolle über die kryptografischen Funktionen zu erlangen oder um eine Performanceverbesserung zu erzielen.

### Installation

Der Aloaha CSP und die CSP APIs sind in allen Signierfähigen Aloaha Tools enthalten:

#### **Aloaha PDF Suite:**

[http://www.aloaha.com/download/aloaha\\_pdf.zip](http://www.aloaha.com/download/aloaha_pdf.zip)

#### **Aloaha PDF Signator:**

[http://www.aloaha.com/download/aloaha\\_signator.zip](http://www.aloaha.com/download/aloaha_signator.zip)

#### **Aloaha PDF Saver:**

[http://www.aloaha.com/download/aloaha\\_saver.zip](http://www.aloaha.com/download/aloaha_saver.zip)

#### **Aloaha sign!:**

[http://www.aloaha.com/download/aloaha\\_sign.zip](http://www.aloaha.com/download/aloaha_sign.zip)

Um die Aloaha CSP API zu verwenden, ist ein gültiger Lizenzschlüssel erforderlich. Bitte kontaktieren Sie [aloaha@wrocklage.de](mailto:aloaha@wrocklage.de) um einen Test-Key zu erhalten!

### Programmier Modell

Alle Aloaha APIs sind als automatisierbare COM-Objekte implementiert, um sicher zu stellen, dass eine große Bandbreite an Programmiersprachen unterstützt wird.

Die Objekte können "in process" oder "out of process (OOP)" verwendet werden. Es wird empfohlen das "in process" Objekt zu verwenden. Falls Sie das OOP Objekt verwenden müssen, muss der Programmierer sicher stellen, dass Zwischenspeicher usw. korrekt gelöscht sind.

### In Process Objekt

Beide Objektmodelle benutzen die gleichen Schnittstellen, um sicher zu stellen, dass es für den Programmierer einfach ist zwischen den Objekten zu wechseln. Im Beispiel auf dieser Seite wird das "in process-Modell" verwendet.

Der "in process" Objektname lautet

AloahaCSPCore.provider. In VBS würden Sie das Objekt mit der folgenden Codezeile erzeugen:  
`set ACSP = CreateObject("AloahaCSPCore.provider")`

### Out of Process Objekt

Das "Out of Process" Modell wird empfohlen falls Sie PINs gecached bleiben müssen auch wenn das Objekt entladen wird. Dieses Modell ist von der Performance her ein wenig besser als damit auch der Caching-Mechanismus besser funktioniert.

Sobald das OOP Objekt erzeugt wird, wird ein entsprechendes System Tray Icon auf Ihrem Desktop sichtbar.

Der Objektname lautet: AloahaCertInstaller.provider

In VBS würden Sie das Objekt wie folgt erzeugen:  
`set ACSP = CreateObject("AloahaCertInstaller.provider")`

## 5.1 Laden der CSP API

Wie bereits erwähnt verwenden wir das "in process" Modell in unseren Beispielen. Das OOP Modell ist identisch mit Ausnahme des Objektnamens.

Nachdem ein Objekt erzeugt wurde, sollte die Funktion info beim Aufruf einen Wert > 0 zurückgeben.

### Zum Beispiel:

```
On Error Resume Next
```

```
Dim ACSP
```

```
Set ACSP = CreateObject("AloahaCSPCore.provider")
```

```
If ACSP.info>0 Then
```

```
    If err.number=0 Then
```

```
        MsgBox "Aloaha CSP API loaded"
```

```
    End If
```

```
End If
```

```
Set ACSP = nothing
```

### Trennen (Disconnect)

Um alle Kartenleser zu trennen und die internen Cache-Speicher zu leeren rufen Sie die Disconnect Funktion auf.

### Informationen sammeln

#### Funktion CertificateTypes Available

Zur Zeit unterstützt Aloaha 3 unterschiedliche Zertifikat-Typen.

0 = (Non Repudiation) Nachweisbarkeitszertifikat

1 = Signatur / Authentifizierungszertifikat

2 = Verschlüsselungszertifikat.

Zur Zeit gibt die Funktion CertificateTypesAvailable immer den Wert 3 zurück. Zukünftig sollen jedoch mehrere Typen vorgestellt werden, die dann auch mit dieser Funktion überprüft werden können.

#### Funktion CSP\_License

Diese Funktion signalisiert ob der CSP und/oder die CSP API in lizenzierte oder NICHT-lizenzierte Betriebsart laufen.

#### Funktion Readers

Diese Funktion zeigt Ihnen an, wieviele Kartenleser an Ihr System angeschlossen sind.

#### Funktion Readername

Diese Funktion weist einer Kartenleser-Nummer einen richtigen Textnamen zu. Zum Beispiel: ReaderName = ACSP.ReaderName(1) gibt den Namen des zuerst angeschlossenen Kartenlesers zurück.

## 5.2 Nützliche Hilfsfunktionen für Skriptsprachen

Aloaha wünscht natürlich, dass so viele Programmiersprachen wie möglich von den Aloaha APIs unterstützt werden.

Da nicht alle Programmiersprachen in der Lage sind mit Speicher Pointern oder byte Arrays zu arbeiten, bietet der Aloaha CSP einige Hilfsfunktionen zur Typanpassung.

**Alle Funktionen die auf dieser Seite erklärt werden, benötigen KEINE Lizenzschlüssel und sind Freeware!**

### BA2STR

Diese Funktion konvertiert ein ByteArray in einen one byte string.

Z.B.:

```
Dim OutputString as string
Dim InputArray(0 to 1) as byte
```

```
InputArray(0)=asc("A")
InputArray(1)=asc("B")
```

```
OutputString = ACSP.BA2STR(InputArray)
```

OutputString wird "AB"

### STR2BA

Diese Funktion konvertiert einen String in ein ByteArray.

Z.B.:

```
Dim OutputArray() as Byte
Dim InputString as String
```

```
InputString = "AB"
```

```
OutputArray = ACSP.STR2BA(InputString)
```

Das OutputArray wird 2 bytes mit dem Ascii Wert "A" und "B" enthalten.

### HEX2STR

Diese Funktion konvertiert einen HEX String in einen "normalen" String. z.B. "4142" in "AB".

### STR2HEX

Diese Funktion konvertiert einen "normalen" String in einen HEX String. z.B. "AB" in "4142".

## 5.3 Digitale Signatur Funktionen

Funktionen und Eigenschaften auf dieser Seite beschreiben wie man eine Smartkarte benutzt um digitale Signaturen zu erzeugen. Digitale Signatur Funktionen sind KEINE FREEWARE. Bitte kontaktieren Sie [aloaha@wrocklage.de](mailto:aloaha@wrocklage.de) für einen Test-Lizenzschlüssel.

### Funktionen

Die Funktion hash verarbeitet die übergebene Information in einen HASH-Wert.

```
hash(ByVal StrInputBA As Variant, ByRef StrOutputBA As Variant, ByVal Algo As AvailableHashAlgos) As Boolean
```

### Eigenschaften

signHASH\_BA gibt die digitale Signatur eines übergebenen Hash-Wertes und einem Zertifikat-Fingerprints zurück. Schauen Sie sich die Rubrik "Zertifikat Management" an, um heraus zu finden, wie man den Fingerprint einer gegebenen Karte oder eines Kartenlesers heraus bekommt.

```
signHASH_BA(ByVal CertificateThumbPrint As String, ByVal ctype As CertificateType, ByVal inputHASH As Variant, ByVal SignatureStandard As SignatureType, ByVal SignatureHashType As AvailableHashAlgos, ByVal SignatureEncType As AvailableEncryptionAlgos) As Variant
```

## 6. APIs und Beispiele

Die Aloaha Smartkarten API kann ohne einen gültigen Lizenzschlüssel ausprobiert werden. Aber beachten Sie, dass jede X Signatur fehlerhaft sein wird, wenn kein Lizenzschlüssel vorhanden ist.

Kontaktieren Sie unseren Support, um einen gültigen Test-Key zu erhalten!

Um die Smartkarten APIs benutzen / testen zu können, müssen Sie die Aloaha PDF Suite, Aloaha PDF Saver oder Aloaha PDF Signator installieren.

**VBS Beispiel Digitale Signatur**  
**Weitere APIs erhalten Sie auf Anfrage!**

## 7. Zertifikat Parser

Der Zertifikat Parser ist als Klasse in den Objekten AloahaCSP und AloahaCertInstaller implementiert. Um darauf zugreifen zu können, müssen Sie das Objekt referenzieren oder es beispielsweise wie folgt benutzen:  
`createobject("AloahaCertInstaller.certparser")`

Eine detailliertere Liste aller Funktionen in Pseudo Code können Sie auf Anfrage erhalten.

## 8. PKCS#7 / S/Mime

Der Unterschied zwischen PKCS #1 und PKCS #7 ist, dass PKCS #7 zusätzlich das Signaturzertifikat und optional die signierten Daten enthält. PKCS #7 ist auch bekannt unter S/Mime und p7m.

Ein Codebeispiel darüber wie man eine PKCS #7 oder PKCS #1 Signatur erzeugt, finden Sie in unserer Sektion **Aloaha CSP API**

Ein Beispiel eines Standalone S/Mime Mailers ist im Pfad samplesSMime von der **Aloaha PDF Suite, Aloaha PDF Signator** und **Aloaha PDF Saver** enthalten.

### Mime 2 S/Mime Konverter

Die Funktion SignMessage\_native konvertiert mime Objekte in S/Mime Objekte.

Die Variable reader legt fest welcher Kartenleser dazu verwendet wird. Mögliche Werte sind 0 für den zuerst angeschlossenen Kartenleser bis 9 für den zehnten Kartenleser.

```
dim mime
set mime = createobject("aloaha_smime.mailer")
    if mime.SignMessage_native(ByRef oMsg As CDO.Message,
    true, reader As Long) = true then
        msgbox "sucess"
    else
        msgbox "problem"
    end if
set mime = nothing
```

Anstatt das CDO Objekt zu übergeben, ist es auch möglich die mime Email als String zu übergeben!

```
dim mime
set mime = createobject("aloaha_smime.mailer")
call msgbox(ctr(mime.mime2smime(ByVal mime As String, reader
As Long)))
set mime = nothing
```

Aber auch das einfache Übergeben der Dateinamen von den .eml Dateien ist möglich!

```
dim mime
set mime = createobject("aloaha_smime.mailer")
if mime.mime_eml2smime(EML_Path As String, SMime_path As String,
reader As Long) = true then
    msgbox "sucess"
else
    msgbox "problem"
end if
set mime = nothing
```

## 9. PKCS#7 erzeugen / überprüfen

Manchmal ist es notwendig alle möglichen Dateitypen signieren zu können. Beispielsweise um Veränderungen an einer Datei feststellen zu können oder um rechtsverbindliche NICHT PDF Dokumente zu signieren.

PKCS7 Signaturen können sehr einfach mithilfe der Shell Erweiterung erzeugt werden.

Aloaha stellt eine leicht bedienbare API zur Verfügung um mit unterstützten Smartkarten PKCS7 Signaturen zu erzeugen.

**Nachfolgend finden Sie ein Codebeispiel dazu:**

```
Dim csp
Dim FileToBeSigned
Dim CardReader          'verwendeter Kartenleser. Kann den Wert 0 bis 9 haben
Dim CardPIN             'kann die PIN der Karte enthalten. Falls leer, öffnet sich ein
Dim Eingabedialog
```

```
FileToBeSigned = "d:\mymailer.exe"
CardReader = 2      'für den dritten angeschlossenen Kartenleser
CardPIN = "123456"
```

```
Set csp = CreateObject("aloahacsp.aloaha_csp")
```

```
If csp.sign_file(CStr(FileToBeSigned), CLng(CardReader), CStr(CardPIN)) = true Then
    MsgBox "Datei signiert"
Else
    MsgBox "Problem"
End If
```

```
Set csp = nothing
```

### PKCS7 Signaturdatei überprüfen

Schauen Sie das folgende Beispiel an, um zu lernen wie Sie mit Aloaha PKCS7 Signaturdateien überprüfen können.

```
Dim csp
Dim SignedFile
Dim SigFilePATH
Dim Signers
```

```
SignedFile = "d:mymailer.exe"
SigFilePATH = "d:mymailer.exe.pkcs7"
```

```
Set csp = CreateObject("aloahacsp.aloaha_csp")
```

```
If csp.VerifyPKCS7File(CStr(SignedFile), CStr(SigFilePATH), true, signers) = true Then
    MsgBox "File Signed"
Else
    MsgBox "Problem"
End if
```

```
Set csp = nothing
```

## 10. Smartkarten Zertifikate anzeigen

Mit dem folgenden Beispiel ist es möglich die Signaturzertifikate einer Smartkarte aufzulisten:

```
Dim csp
Dim reader

'Kartenlesernummer kann von 0 bis 9 sein
'0 ist der erste angeschlossene Kartenleser
reader = 0

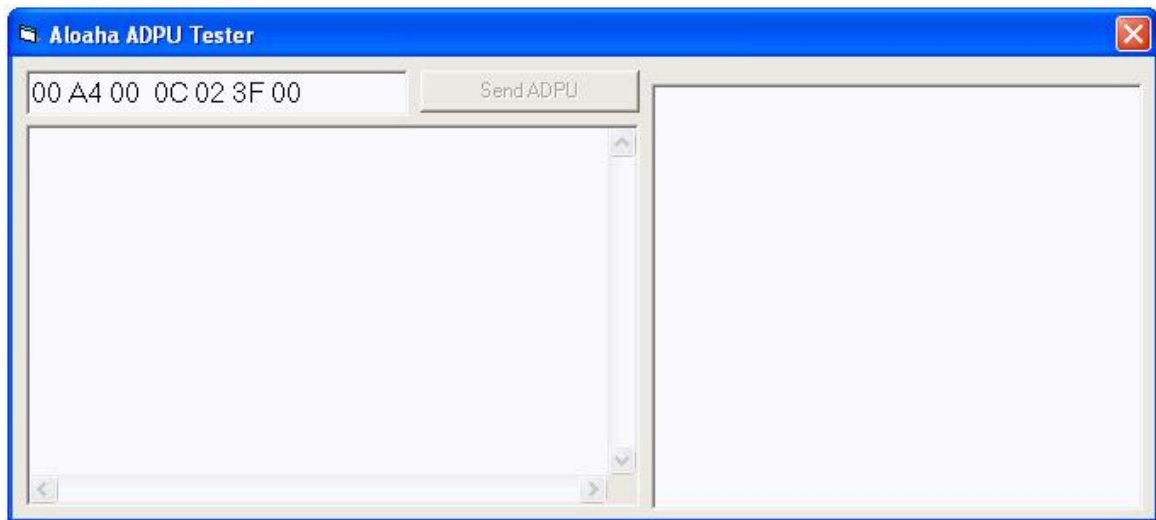
Set csp = CreateObject("aloahacsp.aloaha_csp")

If csp.set_reader(CLng(reader)) = True Then
    Call csp.show_ROOTCertificate
    Call csp.show_Certificate
End If

Set csp = nothing
```



## 11. ADPU Tester

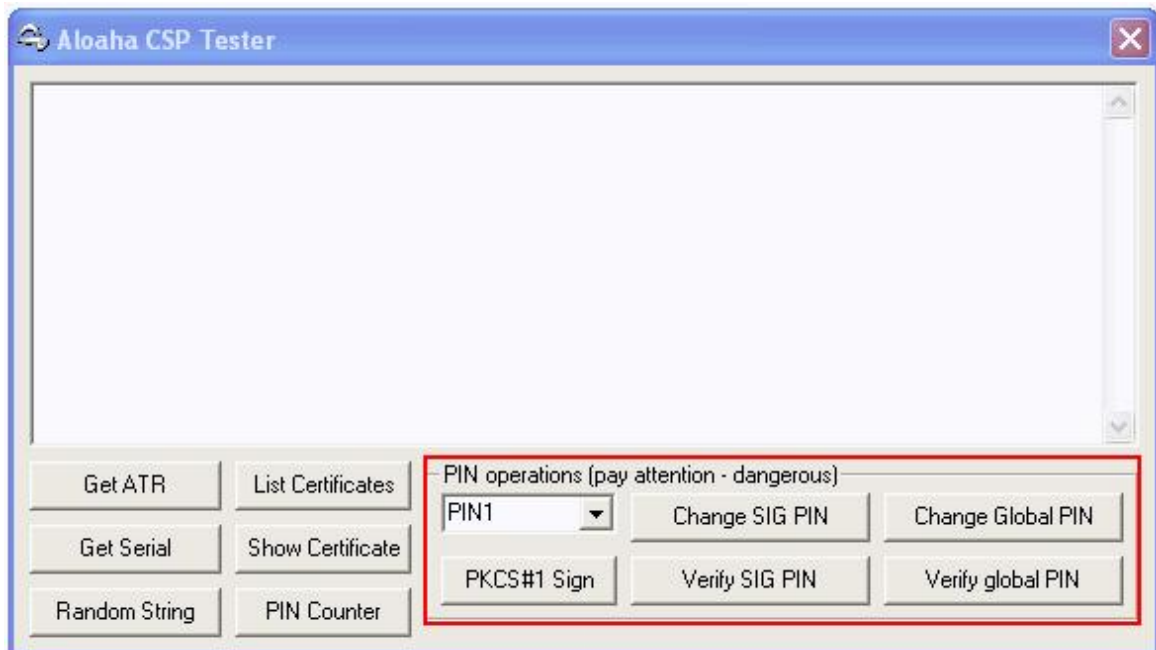


Der Aloaha ADPU Tester ist das ideale Tool um ADPUs direkt an eine ausgewählte Karte / Kartenleser zu senden.

Der Aloaha ADPU Tester ist als VB6 Quellcode Projekt in allen Aloaha PDF Tools enthalten. Sie finden ihn unter

<Installationsverzeichnis>\samples\ADPU Tester.

## 12. Smartkarten Tester



Der Aloaha Card Tester ist das Ideale Tool, um festzustellen ob Ihre Karte vom Aloaha SmartCard Connector unterstützt wird. Falls nicht anders definiert in HKLMSoftwareAloaha eader verbindet der Card Tester mit dem ersten Kartenleser (0) und sendet die ADPU Kommandos die auf den Schaltflächen stehen.

### **BITTE SEIEN SIE VORSICHTIG MIT DEN ROT MARKIERTEN SCHALTFLÄCHEN!**

Es ist auch möglich eine Textdatei mit der Endung .csp zu erzeugen welche in jeder Zeile einen ADPU Befehl enthält. Diese Datei können Sie via Drag&Drop auf die Oberfläche des Card Testers ziehen, um die darin enthaltenen ADPU Befehle auszuführen.

### **Was muss ich tun damit meine Karte unterstützt wird?**

Der Aloaha Smart Card Connector enthält schon ein großes Spektrum an unterstützten Kartentypen. Beispielsweise die SparkassenCard (SECCOS), T-Systems Telesec (TCOS), German Health Professional Card (HBA), Belgium digital ID Card (Belpic), D-Trust (Sagem-Orga Micardo), Sicrypt, CardOS und viele mehr.

Dennoch ist es möglich, dass Ihre Karte noch nicht dabei ist. Um heraus zu finden ob Ihre Karte unterstützt wird, betätigen Sie ALLE Buttons die NICHT ROT markiert sind und senden die Ausgabe an unser Support-Team. Sie können dazu auch das folgende Formular verwenden.

## 13. PKCS7 Signatur mit Zeitstempel

Mit dem / der Aloaha Smart Card Connector / API ist es möglich, jede PKCS #7 Signatur mit Zeitstempel zu versehen. Die Zeitstempel sind RFC 3161 kompatibel.

Beispiel:

### TSA Server konfigurieren

```
dim tsa
set tsa = createobject("aloahacsp.aloaha_csp")

        call tsa.show_TSAConfig(cbool(false))

set tsa = nothing
```

### Eine PKCS#7 Struktur zeitstempeln. Ein- und Ausgabe müssen in HEX-Format sein

```
dim tsa
dim input          'input contains PKCS7 signature in HEX
dim output         'output will hold PKCS7 stamped in HEX

set tsa = createobject("aloahacsp.aloaha_csp")

        output = tsa.TimeStamp_PKCS7(cstr(input))

set tsa = nothing
```

## 14. Allgemeine CSP Informationen

### Objektnamen

#### In Process:

AloahaCSPCore.provider

Zum Beispiel: set ACSP = CreateObject("AloahaCSPCore.provider")

#### Out of Process (OOP):

AloahaCertInstaller.provider

Zum Beispiel: set ACSP = CreateObject("AloahaCertInstaller.provider")

### Unterstützte Hashing Algorithmen

#### Folgende Hashing Algorithmen werden unterstützt:

SHA1 = 1

SHAMD5 = 3

SHA256 = 4

### Verfügbare Verschlüsselungs Algorithmen

Zur Zeit wird nur RSA = 1 von Aloaha unterstützt.

### Zertifikat Typen

Üblicherweise enthalten Signaturkarten unterschiedliche Zertifikate. Der Typ eines qualifizierten Zertifikats ist 0. "Normalerweise" sind erweiterte Zertifikate vom Typ 1 und Verschlüsselungszertifikate vom Typ 2.

## 15. Zertifikat Management

Der Vorteil der Aloaha CSP API ist der, dass der Benutzer nicht wissen muss, in welchem angeschlossenen Kartenleser die gewünschte Karte steckt. Aloaha benutzt verschiedene Algorithmen, um den richtigen Kartenleser zu finden.

### Funktionen

**Die Funktion `get_certificate_by_reader` gibt das Zertifikat einer Karte zurück die in einen bekannten Kartenleser eingesteckt ist.**

```
function get_certificate_by_reader(ByVal lngReader As Long, ByRef CertificateBA As Variant, ByRef CA_BA As Variant, ByVal ctype As CertificateType) as boolean
```

z.B.

**Die Funktion `FindCertificate` durchsucht alle angeschlossenen Kartenleser nach einem bestimmten Zertifikat.**

```
Function FindCertificate(ByVal SearchString As String, ByVal SearchFilter As SearchFilter, ByVal ctype As CertificateType, ByRef fingerprint As String, ByRef CertificateBA As Variant) As Long
```

Der Suchfilter kann so aussehen:

- `CertificateSubject = 0`  
Sucht nach einem Zertifikat mit dem gegebenen Subject.
- `CertificateSimpleSubject = 1`  
Sucht nach einem Zertifikat mit dem gegebenen SimpleSubject.
- `CertificateFingerprint = 2`  
Sucht nach einem Zertifikat mit dem gegebenen fingerprint.
- `CertificateIssuer = 3`  
Gibt das erste Zertifikat mit dem gegebenen Issuer zurück.
- `CertificateEmail = 4`  
Gibt das erste Zertifikat mit der gegebenen Emailadresse zurück.
- `CertificateBSTR = 5`  
Findet das übergebene Zertifikat.
- `CertificateSerialNumber = 6`  
Gibt das Zertifikat mit der übergebenen Seriennummer zurück.
- `FreeText = 7`  
Macht eine Volltextsuche nach Zertifikaten.
- `Dialog = 8`  
Zeigt einen Dialog mit einer Liste der physikalisch verfügbaren Zertifikate an.

Das folgende Beispiel findet das erste Nachweisbarkeitszertifikat (non-repudiation), welches den String "Stefan Engelbert" enthält. Es gibt als Ergebnis den Kartenleser, die Karte und den Fingerprint des Zertifikats zurück.

```
Const FreeText = 7
Const SignatureCertificate = 0

Dim ACSP As Object
Dim SearchString As String
Dim SearchFilter As Long
Dim FingerPrint As String
Dim CardReader As Long

SearchString = "Stefan Engelbert"
SearchFilter = FreeText
ctype = SignatureCertificate

Set ACSP = CreateObject("AloahaCSPCore.provider")

CardReader = ACSP.FindCertificate(SearchString, SearchFilter, ctype, FingerPrint, vbNull)

If CardReader > -1 Then

    MsgBox "Found Certificate with Fingerprint: " + FingerPrint + " in Reader " + ACSP.
    ReaderName(CardReader)

End If

Set ASP = Nothing
```

### Eigenschaften

**FingerPrint\_by\_Reader** gibt den Fingerprint des Zertifikats der Karte zurück die in dem übergebenen Kartenleser steckt.

Z.B. fingerprint = ACSP.FingerPrint\_by\_Reader(0,1)  
gibt den Fingerprint des Signatur-/ Authentifizierungszertifikats aus den zuerst angeschlossenen Kartenleser zurück.

**Mit publickeyBA wird der Public Key eines Zertifikats abgefragt. Das kann bei Public Key Verschlüsselung oder bei manueller Signaturüberprüfung notwendig sein.**

```
Dim PublicKey() as byte
Dim PublicKeyString as string
Dim FingerPrint as string
Dim CType
```

```
Fingerprint = <Fingerprint of Certificate>
CType = Type of Certificate (0, 1 or 2)
```

```
PublicKey = ACSP.publickeyBA(Fingerprint, CType)
PublicKeyString=ACSP.BA2STR(PublicKey)
```

## 16. Verwendung von Zertifikaten

In manchen Fällen kann es wichtig sein zu wissen, für welche Verwendung ein Zertifikat erstellt wurde. Aloaha bietet zwei Möglichkeiten um diese Eigenschaften auszugeben. Der Inhalt dieser Eigenschaften sind bitmaskierte Werte.

### CertCapabilities

CertCapabilities(fingerprint, ctype) gibt die Zertifikat Ressourcen in "CSP Notation" zurück. Fingerprint muss der Fingerprint des Zertifikats sein und ctype kann vom Typ 0-2 sein.

Die Ergebnis Bitmaske verschlüsselt die folgenden Parameter:

```
IsCritical = 1
CRLSignEnabled = 2
DataEnciphermentEnabled = 4
DecipherOnlyEnabled = 8
DigitalSignatureEnabled = 16
EncipherOnlyEnabled = 32
KeyAgreementEnabled = 64
KeyCertSignEnabled = 128
KeyEnciphermentEnabled = 256
NonRepudiationEnabled = 512
IsPresent = 1024
```

### CertUsage

CertUsage(fingerprint, ctype) gibt die Zertifikat Ressourcen in "CSP Notation" zurück. Fingerprint muss der Fingerprint des Zertifikats sein und ctype kann vom Typ 0-2 sein.

Die Ergebnis Bitmaske verschlüsselt die folgenden Parameter:

```
decrypt = 1
sign = 2
SignRecover = 4
Derive = 8
Unwrap = 16
NonRepudiation = 32
```

### Codebeispiel

## 17. RSA Datenverschlüsselung

Die meisten Smartkarten (allerdings nicht alle) enthalten auch ein Zertifikat, um mit Public Key verschlüsselte Daten zu entschlüsseln. Diese Funktionen sind direkt mit den Aloaha EncryptedStringBA und DecryptedString Eigenschaften verbunden.

**Beachten Sie, dass RSA Verschlüsselung nur für relativ kleine Daten einsetzbar ist. Die Länge des RSA verschlüsselbaren Strings hängt von der Länge des Schlüssels ab. Um auf der sicheren Seite zu sein, sollte er nicht länger als annähernd 160 bytes sein!**

### EncryptedStringBA

EncryptedStringBA verschlüsselt jeden String mit dem übergebenen Public Key. Für diese Funktion wird keine Smartkarte benötigt!

EncryptedStringBA(ByVal publicKeyBA As Variant, ByVal inputstring As String, ByVal RSA\_Padding As RSAPadding) As Variant.

RSA\_Padding definiert den verwendeten Padding Typ. Unterstützte Typen sind:

NONE = 0  
PKCS = 1  
OAEP = 2  
SSL = 3

Verschlüsselung ist eine FREEWARE Funktion von Aloaha!

### DecryptedString

DecryptedString gibt den entschlüsselten Daten von InputBA zurück. Dieser Vorgang wird auf der Smartkarte ausgeführt!

DecryptedString(ByVal CertificateThumbPrint As String, ByVal InputBA As Variant, ByVal ctype As CertificateType, ByVal RSA\_Padding As RSAPadding) As String

### Beispielcode

## 18. FAQ

(siehe auch <http://www.aloaha.de/support>)

### **Was ist Aloaha Smart Card SDK?**

Aloaha Smart Card SDK ist für die Verwaltung von Smartkarten zuständig. Weiterhin erlaubt er die Kommunikation zwischen den verschiedenen Aloaha Prozessen und Produkten. Er ist in jedem Aloaha Produkt enthalten. Für die enthaltenen PKCS11 und CSP Module wird eine gültige Lizenz benötigt.

### **Welche Signaturkarten werden unterstützt?**

Aloaha unterstützt gängige Karten wie die neuen Krankenversicherungskarten, Heilberufsausweise, D-Trust Signaturkarten, belgische e-ID, Infocamere, Actalis und viele mehr. Mehr Informationen finden Sie auf der Aloaha-Webseite unter Support -> Nativ unterstützte SmartCards.

### **Gibt es eine HBA/eGK spezifische FAQ?**

Ja, die gibt es. Gehen Sie bitte zu:

<http://www.aloaha.de/wi-software/hba---egk-faq.php>

### **Meine Karte wird von Aloaha nicht unterstützt. Was kann ich tun?**

Bitte kontaktieren Sie unseren Support. Wir freuen uns immer neue Karten zu unterstützen!

### **Wozu ist die Option "Automatisch De-registrieren" gut?**

Wenn diese Option aktiviert ist werden alle Referenzen von Zertifikaten die von Aloaha registriert wurden gelöscht sobald keine Karte mehr in keinem angeschlossenen Leser steckt.

### **Wozu hat Aloaha die Option "Konfiguriere Unterschrift"?**

Damit kann das Standard Zertifikate oder der Standard Leser konfiguriert werden. Diese Einstellung wird zum Beispiel genutzt wenn man mit der rechten Maustaste auf eine Datei klickt und diese PKCS #7 signiert. Auch die Aloaha PDF Tools greifen auf diese Einstellung zu.

### **Kann ich ein Zertifikat anzeigen bevor ich es registriere?**

Ja, dazu benutzen Sie die rechte Maustaste und klicken auf ein Zertifikat in der Liste

### **Ich möchte gerne sicher PDF Massensignaturen erstellen. Was brauche ich dazu?**

Sie benötigen:

1. Einen Kartenleser mit Pin Pad
2. Eine Multisignaturkarte
3. Den Aloaha Multisignator

Nun stellen Sie bitte in "Konfiguriere Unterschrift" den zu benutzenden Leser ein.

Wenn Sie nun X PDF Dokument mit der rechten Maustaste auswählen und "PDF Signatur Erstellen" auswählen werden Sie nur einmal via PIN Pad zur PIN Eingabe aufgefordert.

### **Ist es möglich mit Aloaha einen bestimmten Kartenleser zu ignorieren?**

Ja, Sie müssen dafür die Datei ReaderIni.ini editieren.

Der Eintrag IgnoreReader weist Aloaha dazu an einen bestimmten Kartenleser zu ignorieren!

Beispiel:

```
[SCM Microsystems Inc. SPRx32 USB Smart Card Reader 0]  
IgnoreReader=1
```

### **Ist es möglich auf die lästige PIN Eingabe zu verzichten?**

Die PIN einer Smartcard kann nicht deaktiviert werden. Aloaha unterstützt jedoch kontaktlose (RFID/Mifare) PIN Token.

### **Gibt es ein Diskussions Forum?**

Ja, auf <http://portal.aloaha.com>

Sollten Sie die Antwort Ihrer Frage nicht gefunden haben, kontaktieren Sie [aloaha@wrocklage.de](mailto:aloaha@wrocklage.de).



## 19. Hilfe

So gelangen Sie zur Online Hilfe und dem Online Forum im Internet: <http://www.aloaha.de/support/aloaha-smartcard-connector.php>

Die Online Hilfe bzw. das Online Forum kann über das Windows Startmenü oder das Schnellstartmenü gestartet werden.

Wählen Sie anschließend den entsprechend benötigten Menüpunkt.





# Index

## - A -

ADPU Tester 65  
Alle entfernen 10  
Allgemeine CSP Informationen 67  
Aloaha PDF Signator 23  
Aloaha PDF Suite 23  
Anwender Support 44  
Anwendung 10  
APIs und Beispiele 61  
Art des Zertifikats 17  
Authentifizierungszertifikat 39  
Autoremove 10

## - B -

BA2STR 59  
Bild Unterschrift 17, 23

## - C -

CAP 45  
CertCapabilities 70  
CertUsage 70  
Cryptographic Application Programming Interface 45  
CSP / Kartenleser 39

## - D -

DecryptedString 71  
Digital Signieren 15  
Digitale Signatur Funktionen 60

## - E -

Einleitung 5  
Einstellungen für den Zeitstempel 17  
EncryptedStringBA 71

## - F -

FAQ 72  
Firefox Einstellungen 52  
Funktion CSP\_License 58  
Funktion Readername 58  
Funktion Readers 58

## - G -

Gesetzliche Regelungen 15

## - H -

HEX2STR 59  
Hilfe 73

## - I -

In Process Objekt 57  
Inkrementelle Signatur 23  
Installation 7

## - K -

Karten-Assistent 13  
Kartenleser 39  
Konfiguration digitale Unterschrift 17

## - L -

Laden der CSP API 58

## - M -

Manuelle Registrierung 10  
Microsoft Cryptography API 45  
Microsoft Outlook 46  
Mime 2 S/Mime Konverter 62  
Mozilla Firefox 52  
MS Crypto API 45

## - N -

Nützliche Hilfsfunktionen für Skriptsprachen 59

## - O -

Objektnamen 67  
Out of Process Objekt 57  
Outlook Einstellungen 46

## - P -

PIN Verwaltung 26  
PKCS #7 Bestätigung 25  
PKCS #7 Signatur erstellen 25  
PKCS#11 51  
PKCS#7 / S/Mime 62  
PKCS#7 erzeugen / überprüfen 63  
PKCS7 Signatur mit Zeitstempel 67  
Position der Signatur 17  
Public Key Verfahren 15

## - R -

Registrierung 10  
RSA Datenverschlüsselung 71

## - S -

Signatureinstellungen 23  
Smartkarten Tester 66  
Smartkarten Zertifikate anzeigen 64  
Sprachauswahl 12  
STR2BA 59  
STR2HEX 59

**- T -**

Text Unterschrift 17, 23  
TSA Server konfigurieren 67

**- U -**

Übersetzungs Mechanismus 56  
Übersetzungs-Dateien 56  
Umschlag erstellen 25  
Unterschriftszertifikat 39  
Unterstützte Hashing Algorithmen 67

**- V -**

Verfügbare Verschlüsselungs Algorithmen 67  
Verschlüsselungszertifikat 39  
Verwendung von Zertifikaten 70

**- Z -**

Zeitstempel 42  
Zertifikat auswählen 17  
Zertifikat Management 68  
Zertifikat Parser 61  
Zertifikate 28  
Zertifikatquelle 17  
Zweck der Signatur 17