Das eCard-API-Framework

Detlef Hühnlein[1] , Manuel Bach[2], Rainer Oberweis[2]

Zusammenfassung
Die eCard-Strategie der Bundesregierung zielt auf die breite Verwendbarkeit der im Rahmen der verschiedenen Kartenprojekte der Bundesverwaltung ausgegebenen und genutzten Chipkarten ab. Eine besondere Rolle bei der Realisierung dieses Zieles spielt das eCard-API-Framework [eCard-TR], durch das ein einfacher und einheitlicher Zugriff auf die Funktionen dieser unterschiedlichen Chipkarten ermöglicht wird. Dieser Beitrag liefert einen kompakten Überblick über die wichtigsten Aspekte des eCard-API-Frameworks.

1. Einleitung

Durch die eCard-Strategie der Bundesregierung [eCard-PM] werden die Kartenprojekte der Bundesverwaltung – die elektronische Gesundheitskarte (eGK), der elektronische Personalausweis (ePA), das JobCard-Verfahren (elektronischer Lohnnachweis (ELENA) und die elektronische Steuererklärung (ELSTER) – eng aufeinander abgestimmt. Gleiche Standards und die breite Verwendbarkeit der Chipkarten für den elektronischen Geschäftsverkehr sollen Effizienzgewinne und Kosteneinsparungen zum Nutzen von Bürgerinnen und Bürgern, Wirtschaft und Verwaltung gewährleisten. Eine zentrale Rolle bei der Umsetzung der eCard-Strategie spielt das eCard-API-Framework [eCard-TR], durch das ein einfacher und einheitlicher Zugriff auf die Funktionen der unterschiedlichen Chipkarten, die in den Kartenprojekten der Bundesverwaltung ausgegeben oder genutzt werden, ermöglicht wird. Dieser Beitrag liefert einen kompakten Überblick über die wichtigsten Aspekte des eCard-API-Frameworks und ist folgendermaßen gegliedert: In Abschnitt 2 werden die wichtigsten Anforderungen an das eCard-API-Framework zusammengetragen und die Erfüllbarkeit dieser Anforderungen durch existierende Application Programming Interfaces (APIs) untersucht. Abschnitt 3 liefert einen Überblick über die Architektur des eCard-API-Frameworks. Die Abschnitte 4 bis 6 erläutern die einzelnen Schichten des eCard-API-Frameworks (Application-Layer, eCard-Layer und Reader-Layer).

2. Anforderungen und existierende APIs

In diesem Abschnitt werden die Anforderungen an das eCard-API-Framework zusammengetragen und untersucht, ob diese Anforderungen durch existierende APIs erfüllt werden können.

[1] secunet Security Networks AG, Sudetenstraße 16, 96247 Michelau, detlef.huehnlein@secunet.com
[2] Bundesamt für Sicherheit in der Informationstechnik, Godesberger Allee 185-189, D-53175 Bonn, manuel.bach,rainer.oberweis@bsi.bund.de

2.1 Anforderungen an das eCard-API-Framework

Bei den Anforderungen an das eCard-API-Framework ist zwischen generellen Anforderungen und denen aus den verschiedenen eCard-Projekten zu unterscheiden.

2.1.1 Generelle Anforderungen

2.1.1.1 Zielsetzung

Durch das eCard-API-Framework soll der technische Rahmen für die Umsetzung der eCard-Strategie geschaffen werden. Insbesondere soll durch das eCard-API-Framework ein einheitlicher Zugriff auf die unterschiedlichen Chipkarten und weitere damit verbundene Funktionen ermöglicht werden.

2.1.1.2 Verfügbare Plattformen

Das eCard-API-Framework sollte idealer Weise für verschiedene gängige Entwicklungs- (C, C++, C#, Java, etc.) und Betriebssystemplattformen (MS Windows, Linux, Unix, etc.) zur Verfügung stehen. Darüber hinaus ist eine „Netzwerkfähigkeit“ des eCard-API-Frameworks wünschenswert, so dass unterschiedliche Komponenten des Frameworks auf verschiedenen Rechnersystemen verteilt ablaufen können.

2.1.1.3 Evaluierbarkeit

Bereits beim Design des eCard-API-Frameworks ist auf die Evaluierbarkeit und Zertifizierungsfähigkeit der zugehörigen Implementierungen gemäß Common Criteria zu achten. Hierbei sind insbesondere auch die Anforderungen an Produkte für qualifizierte elektronische Signaturen aus dem Signaturgesetz zu berücksichtigen, so dass neben der Zertifizierung auch eine Bestätigung durch eine Stelle gemäß § 18 SigG möglich ist.

2.1.1.4 Skalierbarkeit

Das eCard-API-Framework soll möglichst skalierbar sein, so dass es auf einem Einzelplatz-System sowie in einer verteilten Serverumgebung eingesetzt werden kann.

2.1.2 Anforderungen aus den eCard-Projekten

2.1.2.1 Die elektronische Gesundheitskarte (eGK)

Die eGK soll zukünftig die bisherige Krankenversichertenkarte ersetzen. Zusätzlich werden schrittweise zunächst rund 300.000 elektronische Heilberufsausweise eingeführt. Ziel ist es, die Wirtschaftlichkeit und Leistungstransparenz im Gesundheitswesen zu steigern und die Arbeitsprozesse und die Bereitstellung von aktuellen gesundheitsstatistischen Informationen zu optimieren. Nähere Informationen zur elektronischen Gesundheitskarte finden sich unter [bIT4health].

Die wesentliche Anforderung an das eCard-API-Framework aus dem Blickwinkel dieses Projektes ist, dass die Funktionalität der eGK [eGK-Sp] und des HBA [HBA-Sp] vollumfänglich unterstützt wird. Beispielsweise muss das eCard-API-Framework in der Lage sein, mit verschiedenen Zertifikaten umgehen zu können, sowie die Card-to-Card-Authentisierung, das (De-)Aktivieren von Records, das Kartenapplikationsmanagement mittels Secure Messaging, logische Kanäle für die Secure Module Card (SMC), das Nachladen von qualifizierten Zertifikaten im Feld und SICCT-Kartenterminals zu unterstützen (vgl. [gemC-API]).

2.1.2.2 Der elektronische Personalausweis (ePA)

Neben der elektronischen Gesundheitskarte spielt der elektronische Personalausweis (ePA) [RRM05] für die eCard-Strategie eine wesentliche Rolle. Der ePA wird hierbei der Kartenspezifikation der European Citizen Card [CEN-ECC] genügen und somit neben der Funktionalität zur Speicherung biometrischer Merkmale wie beim ePass (vgl. Abschnitt 2.1.2.3) insbesondere auch generische Funktionen zur Authentisierung, Verschlüsselung und Signatur bereitstellen. Neben der Funktionalität der eGK sind beim ePA insbesondere die folgenden zusätzlichen Merkmale vorgesehen:

- Unterstützung kryptographischer Verfahren auf Basis elliptischer Kurven
- Funktionen zur Speicherung biometrischer Merkmale
- Spezifische Authentisierungsprotokolle für den Zugriff auf den ePA
- Kontaktloses Interface gemäß ISO/IEC 14443

2.1.2.3 Der elektronische Reisepass (ePass)

Seit dem 01. November 2005 werden in Deutschland Reisepässe ausgegeben, auf denen biometrische Merkmale gespeichert sind [ePass]. Für den standardisierten Zugriff auf den elektronischen Reisepass wurde die ePassport-API und das Golden Reader Tool entwickelt. Das eCard-API-Framework soll diese Entwicklungen berücksichtigen und muss zukünftig als Basis für Applikationen wie das Golden Reader Tool genutzt werden können.

2.1.2.4 Die elektronische Steuererklärung (ELSTER)

Seit Anfang 2005 müssen Umsatzsteuer-Voranmeldungen und Lohnsteuer-Anmeldungen
elektronisch erfolgen. Hierfür wurde im Auftrag der Finanzverwaltung eine neue Sicherheitsplattform für das ElsterOnline-Portal realisiert, die die Authentisierung, Verschlüsselung und elektronische Signatur für Web-Anwendungen über verschiedene zertifikatsbasierte Verfahren mit beliebigen Chipkarten über die [PKCS#11]-Schnittstelle unterstützt. Hierbei werden verschiedene Standardformate ([PKCS#7], [PKCS#10], [PKCS#12] und XML-DSig gemäß [RFC3275]) verwendet, die auch vom

eCard-API-Framework unterstützt werden sollten. Weitere Informationen zu diesem Projekt finden sich unter [ELSTER].

2.1.2.5 Der elektronische Einkommensnachweis (ELENA)

Der elektronische Einkommensnachweis (ELENA), früher Jobcard-Verfahren [Jobcard], ist ein weiteres wichtiges Anwendungsgebiet von Signaturkarten. Ziel des Projektes ist die Entlastung der Arbeitgeber von der Ausstellung papierbezogener Bescheinigungen (zum Beispiel Verdienstbescheinigungen) und die Modernisierung von Verwaltungsabläufen. Hierbei handelt es sich nicht um ein Projekt, in dem eine spezielle Karte (wie die elektronische Gesundheitskarte oder der elektronische Personalausweis) ausgegeben wird, sondern das Verfahren soll mit beliebigen Signaturkarten mit qualifiziertem Zertifikat genutzt werden können. Deshalb muss das eCard-API-Framework beliebige, unterschiedlich personalisierte Chipkarten unterstützen können.

2.2 Existierende APIs

Im Rahmen einer Vorstudie [eCard-VS] wurde untersucht, ob die in Abschnitt 2.1 zusammengetragenen Anforderungen mit existierenden APIs[3] - insbesondere von der im Signaturbündnis entwickelten [SASCIA] - erfüllt werden können und welche etwaigen Anpassungen notwendig sind. Auf Basis dieser Analyse wurde die Entscheidung getroffen, die Kartendienste des eCard-API-Frameworks auf Basis des ISO/IEC 24727-Standards zu realisieren und neben dem [ISO24727-3]-Interface weitere Schnittstellen zu den Kartenterminals (unten) und zur Anwendungsebene (oben) anzubieten. Im Vergleich zu [SASCIA] haben sich beim eCard-API-Framework insbesondere folgende Änderungen ergeben:

1. Zusätzliche Schnittstellen für PKI-Funktionen und (kryptographische) Support-Funktionalität außerhalb der Karte

   Für eine leichte Integration kryptographischer Funktionen in Anwendungssysteme ist es hilfreich, neben den Funktionen für den direkten Chipkartenzugriff, wie sie auch von [SASCIA] angeboten werden, zusätzliche Schnittstellen auf höherer Ebene (vgl. Abschnitt 5.1) und für weitere (kryptographische) Support-Funktionen (symmetrische Verschlüsselung, Berechnung von Hashwerten, XML-Schema-Validierung, Datenkompression etc.) (vgl. Abschnitt 5.4) anzubieten, die nicht auf einer Chipkarte realisiert werden.

2. Unterstützung der vollen Funktionalität von eGK / HBA und ePA über ISO/IEC 24727-3-Schnittstelle

   Im Vergleich zu einer gewöhnlichen Signaturkarte sind bei der eGK, dem HBA und dem ePA eine Reihe von zusätzlichen Funktionen (Card-to-Card-Authentisierung, ePA-spezifische Authentisierungsprotokolle, Speicherfunktionen,

[3] Ein Überblick über die bis zum Jahr 2005 standardisierten kryptographischen Programmierschnittstellen findet sich beispielsweise in Kapitel 2.3 von [Hueh05].

Kartenapplikationsmanagement etc.) vorhanden, die bei [SASCIA] nicht bzw. nur auf sehr umständliche Art und Weise mit dem sigall_forward_command_apdu-Kommando genutzt werden könnten. Nicht zuletzt deshalb wird beim eCard-API-Framework der Zugriff auf die Kartenfunktionen nicht über die [SASCIA]-Kartenschnittstelle, sondern die in [ISO24727-3] definierte Schnittstelle realisiert.

3. XML-basierte CardInfo-Datei statt ausführbarer CardProvider

   Die Unterstützung der unterschiedlichen Chipkarten wird bei [SASCIA] durch die so genannte SASCIA-Factory und entsprechende CardProvider realisiert. Für die Unterstützung einer neuen Chipkarte muss mit dem CardProvider, der vom Kartenherausgeber bereitgestellt werden soll, fremder ausführbarer Code auf das System geladen werden. Da in [SASCIA] keine Mechanismen zur Authentifizierung der verschiedenen Komponenten (Factory, CardProvider, ReaderProvider, Karte) vorgesehen sind, ist Viren und Trojanern bei [SASCIA] Tür und Tor geöffnet. Auch nach entsprechenden Anpassungen wäre ein [SASCIA]-basiertes System, bei dem verschiedene Parteien ausführbaren Code bereitstellen, nur mit sehr großem Aufwand oder gar nicht evaluier-, zertifizier- und gemäß SigG bestätigbar. Deshalb wird beim Card-API-Framework ein anderer Ansatz verfolgt und für eine neue Chipkarte kein ausführbarer Code nachgeladen, sondern lediglich eine XML-basierte CardInfo-Datei bereitgestellt, in der die detaillierten Eigenschaften der Chipkarte dokumentiert sind (vgl. Abschnitt 5.3).

4. Webservice- statt C-Schnittstelle

   Während bei [SASCIA] eine C-Schnittstelle vorgesehen ist, werden beim eCard-API-Framework Webservice-Schnittstellen verwendet werden, so dass bestimmte Sicherheitsprobleme (z.B. mögliche Buffer-Overflow-Angriffe) vermieden werden und eine größtmögliche Plattformunabhängigkeit und Netzwerkfähigkeit erreicht werden kann. Hierdurch kann das eCard-API-Framework auf gleichermaßen einfache Weise in Java- oder .NET-basierte Applikationen integriert werden.

3. Architektur des eCard-API-Frameworks im Überblick

Hauptziel des eCard-API-Frameworks ist es, Anwendungsentwicklern eine möglichst einfache und homogene Schnittstelle für aktuelle und zukünftige kartenbasierte Anwendungen im Bereich ePassport, eHealth, ELSTER, ELENA etc. und die damit verbundenen Chipkarten zur Verfügung zu stellen. Gleichzeitig sollen aber möglichst geringe Einschränkungen für bereits bestehende Middleware-Infrastrukturen und kartenbasierte Applikationen entstehen.

Die Architektur des eCard-API-Frameworks unterteilt sich in drei wesentliche Schichten (Application-Layer, eCard-Layer und Reader-Layer), die nachfolgend beschrieben werden.

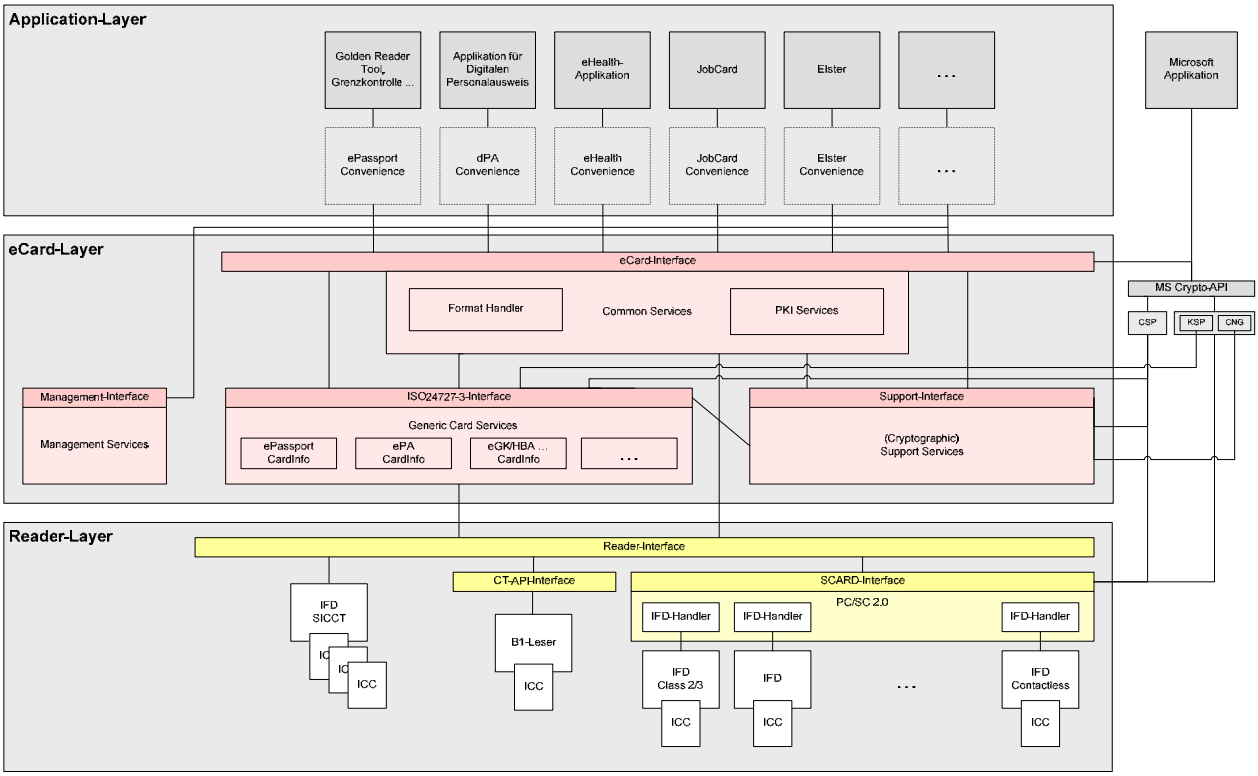

Abbildung 1: Architektur des eCard-API-Frameworks

4. Application-Layer

Im Application-Layer finden sich unterschiedlichste Anwendungen. Ziel des eCard-API-Frameworks ist es, innerhalb dieser Anwendungen dennoch eine möglichst homogene Schnittstelle zu den kartenbasierten Diensten anzubieten, so dass eine Anwendung möglichst wenig oder kein Spezialwissen über die anzusprechende Chipkarte besitzen muss.

Ein weiteres Ziel des eCard-API-Frameworks ist es, keine Interferenzen zu bereits bestehenden etablierten Standards, wie z.B. dem Microsoft Crypto-API-Konzept [MS-CAPI] oder zukünftigen Entwicklungen in diesem Bereich [MS-CNG], zu generieren. Das konkrete Konzept sieht sogar eine Verbindung dieser unterschiedlichen Welten vor, so dass es z. B. problemlos möglich ist, einen Crypto Service Provider auf die herkömmliche Art oder unter Benutzung des eCard-API-Framework zu realisieren. Selbst ein Parallelbetrieb beider Konzepte wäre durch die gewählte Kartenkommunikation möglich.

5. eCard-Layer

Etabliert man ein Framework mit einer möglichst generischen Schnittstelle zum Application-Layer (oben) und einer möglichst generischen Schnittstelle zum Reader-Layer (unten), so muss dennoch an einer Stelle das Spezialwissen über den verwendeten Chipkarten-Typ, die physikalischen und kryptographischen Eigenschaften einer Karte und die Personalisierung der Karte mit Kartenapplikationen vorhanden sein.

Diese Karten- und applikationsspezifischen Besonderheiten werden im eCard-Layer berücksichtigt.

Der eCard-Layer untergliedert sich in die folgenden verschiedenen Interfaces, welche von der Anwendung bei entsprechender Berechtigung als öffentliche Schnittstelle genutzt werden können:

- eCard-Interface
- Management-Interface
- ISO24727-3-Interface
- Support-Interface

5.1 eCard-Interface

Das eCard-Interface stellt die wesentlichen Funktionen des eCard-API-Frameworks in einer anwendungsnahen Art und Weise bereit. Da diese Schnittstelle in der Regel zur Integration der Chipkartenfunktionalität in Anwendungen genutzt wird, wird die Funktionalität in diesem Interface näher betrachtet. Die Funktionen des eCard-Interfaces lassen sich in folgende Gruppen einteilen:

- Verbindungsfunktionen
- Kartenfunktionen
- Lese- und Schreibfunktionen
- Signaturfunktionen
- Verschlüsselungsfunktionen

5.1.1 Verbindungsfunktionen

Mit der Funktion OpenServerSession kann eine gesicherte Verbindung (z.B. über TLS [RFC4346]) von einer als Client agierenden eCard-API-Framework-Instanz zu einer anderen als Server agierenden Instanz aufgebaut und mit CloseServerSession wieder abgebaut werden. Hierdurch können die anderen Funktionen des eCard-API-Frameworks nicht nur lokal, sondern auch über ein Netzwerk genutzt werden. Beispielsweise kann dadurch ein elektronisches Rezept in einer Internet-Apotheke eingelöst werden.

5.1.2 Kartenfunktionen

In dieser Gruppe existieren Funktionen, mit denen die verfügbaren Kartenterminals (GetCardTerminals) und Karten (GetCards) aufgelistet werden oder weitere Informationen über eine spezifische Karte (GetCardInfo) in Form einer CardInfo-Struktur erhalten werden können. Außerdem existieren hier einfache Funktionen zum Überprüfen (VerifyPin), Ändern (ChangePin) und Reaktivieren (UnblockPin) einer PIN. Schließlich existieren hier auch sehr einfache Funktionen, mit denen die unter Umständen

sehr komplexen Vorgänge zur Authentisierung zwischen zwei Karten (Card2CardAuthenticate) oder der Aufbau eines vertrauenswürdigen Kanals zwischen zwei Karten (OpenCardChannel) angestoßen werden kann.

5.1.3 Lese- und Schreibfunktionen

Durch die Lese- (ReadData) und Schreibfunktionen (WriteData) können einzelne oder mehrere Datenobjekte auf eine Karte geschrieben bzw. von ihr gelesen werden.

5.1.4 Signaturfunktionen

Ein wesentliches Element der eCard-Strategie ist die (bedarfsweise) Nutzung der qualifizierten elektronischen Signatur. Zur Nutzung der elektronischen Signatur sieht das eCard-API-Framework drei Funktionen vor:

- SignDocument – ermöglicht die Erzeugung von komplexen Signaturen (z.B. gemäß [CAdES] und [XAdES]) oder Zeitstempeln (z.B. gemäß [RFC3161]). Hierbei kann die Gestalt der Signatur beim Aufruf der Funktion oder über konfigurierte Standard-Parameter gesteuert werden.
- VerifySignedObject – dient der Prüfung von Signaturen, Zeitstempeln oder anderen signierten Objekten (z.B. Zertifikaten, CRLs und OCSP-Responses). Hierbei wird ein Prüfbericht (VerificationReport) zurückgeliefert, dessen Detaillierungsgrad beim Aufruf der Funktion oder über Standard-Parameter gesteuert werden kann.

  ShowViewer – dient der vertrauenswürdigen Anzeige von Dokumenten und kann in Verbindung mit den beiden oben genannten Funktionen – oder bei entsprechender Berechtigung - eigenständig genutzt werden.

5.1.5 Verschlüsselungsfunktionen

Mit den Funktionen EncryptDocument und DecryptDocument können schließlich Daten gemäß [RFC3369] und [XML-Enc] ver- und entschlüsselt werden.

5.2 Management-Interface

Im Management-Interface sind die Funktionen zur Verwaltung und Konfiguration des eCard-API-Frameworks enthalten. Dies umfasst beispielsweise die Verwaltung von (vertrauenswürdigen) Zertifikaten, Identitäten, Chipkarten, Kartenterminals und Diensten, die vom eCard-API-Framework genutzt werden.

5.3 ISO24727-3-Interface

Das ISO24727-3-Interface ist eine Webservice-basierte Umsetzung des gleichnamigen Standards [ISO24727-3] und bietet eine generische Schnittstelle für alle kartenbasierten Funktionen der verschiedenen eCards. Entgegen existierenden Schnittstellen für kryptographische Token, wie z.B. [PKCS#11], unterstützt die [ISO24727-3]-

Schnittstelle insbesondere auch spezifische Authentisierungsfunktionen für Chipkarten und Funktionen für das Kartenapplikationsmanagement. Um die generischen Anfragen der Applikationsebene auf die spezielle Personalisierung einer spezifischen Karte zu wandeln, werden parametrisierbare Services implementiert, die mittels XML-basierter CardInfo-Dateien konfiguriert werden können. Diese CardInfo-Dateien sind Karten-Typ-spezifisch und enthalten Informationen gemäß [ISO24727-2] und [ISO7816-15], sowie eine Folge charakteristischer Merkmale anhand derer der Typ einer Karte bestimmt werden kann. Die Nutzung der CardInfo-Dateien ist insbesondere deshalb notwendig, da bei den für die eCard-Strategie wichtigen Chipkarten (eGK, HBA, ePass, Signaturkarten etc.) diese Information derzeit nicht auf der Karte aufgebracht sind. Weitere Informationen zu den Inhalten einer CardInfo-Dateien und zu den Abläufen beim Erkennen eines Kartentyps und der Abbildung eines generischen Aufrufs auf kartenspezifische APDUs finden sich in [HuBa07].

5.4 Support-Interface

Das Support-Interface enthält eine Reihe von unterstützenden (kryptographischen) Funktionen, die nicht auf einer eCard ausgeführt werden. Beispielsweise finden sich in dieser Schnittstelle Funktionen zur symmetrischen Verschlüsselung von Daten oder zur Berechnung von Hashwerten. Darüber hinaus existieren hier auch Funktionen für das Komprimieren von Daten zur speicherplatzeffizienten Ablage von Daten auf Chipkarten und zur XML-Schema-Validierung.

6. Reader-Layer

Die generelle Karten- und Kartenleseransteuerung wird über eine einheitliche Schnittstelle - das Reader-Interface - realisiert, durch die von der konkreten Ausprägung des Lesegerätes abstrahiert wird. Hierbei sollen unterschiedlichste Lesegeräte (konktaktbehaftete Leser (Klasse 1, Klasse 2/3), SICCT-Leser, kontaktlose Leser etc.) und existierende Schnittstellen ([CT-API], [PC/SC] und [SICCT]) unterstützt werden, so dass im eCard-API-Framework beliebige Chipkarten gemäß ISO7816 ohne Spezialwissen über Übertragungsprotokolle oder Chipkartenleser genutzt werden können. Auch die kontaktlosen ISO14443 Protokolle sowie reine Speicherkarten werden komplett durch die API gekapselt. Eine Applikation kann nach wenigen Initialisierungsbefehlen beispielsweise sofort APDUs über beliebige Kartenleser an ebenfalls beliebige ISO7816-kompatible Chipkarten senden und die spezifischen Eigenschaften der Terminals (Netzwerkfähigkeit, Multislot-Eigenschaften und die Ansteuerung von Displays, Eingabefeldern und biometrischen Sensoren) nutzen.

7. Zusammenfassung und Ausblick

Dieser Beitrag lieferte einen Überblick über die Architektur des eCard-API-Frameworks, durch das beliebige Applikationen in einfacher und einheitlicher Art und Weise auf die Funktionen beliebiger eCards zugreifen können. Um die Interoperabili-

tät in den kommenden Pilotvorhaben sowie im Produktivbetrieb zu gewährleisten wird eine entsprechende Testinfrastruktur geschaffen, durch die die Konformität der unterschiedlichen Realisierungen mit der Spezifikation [eCard-TR] geprüft werden kann. Um die einfache und einheitliche Nutzung der eCards auch über die Grenzen der Bundesrepublik Deutschland hinaus zu ermöglichen, werden Teile des eCard-API-Frameworks derzeit der internationalen Normung zugeführt.

Literatur

[bIT4health] Bundesministerium für Gesundheit: *Die elektronische Gesundheitskarte, via* http://www.bit4health.de , 2005

[CAdES] ETSI: *Electronic Signature Formats*, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V1.5.1, 2003-12, via http://portal.etsi.org/docbox/EC_Files/EC_Files/ts_101733v010501p.pdf

[CEN-ECC] Comité Européen de Normalisation (CEN): Identification card systems — European Citizen Card — Part 2: Logical data structures and card services, TC 224 WI 188 Draft für prCEN 15480, November 2005

[CT-API] J. Attrott, L. Eckstein, B. Kowalski, R. Moos, H. Reimer, B. Struif: *CT-API - Anwendungsunabhängiges CardTerminal-Application Programming Interface für Chipkartenanwendungen*, Version 1.1.1, via https://www.secure.trusted-site.de/Download/CTAPI/CTAPI111.pdf

[eCard-PM] Bundesregierung: *eCard-Strategie der Bundesregierung*, Pressemitteilung vom 09.03.2005, via http://www.bmwi.de/Navigation/Presse/pressemitteilungen ,did=60006.html , 2005

[eCard-TR] Bundesamt für Sicherheit in der Informationstechnik: *eCard-API-Framework – Technische Richtlinie des BSI 03112 (Teil 1 – 6)*, 2007

[eCard-VS] Bundesamt für Sicherheit in der Informationstechnik: *eCard-API-Framework – Vorstudie*, 2006

[eGK-Sp] *Die Spezifikation der elektronischen Gesundheitskarte – Teil 1-3*, Version 1.1.0, vom 07.02.2006, via http://www.bmg.bund.de/cln_040/nn_893332/DE/Gesetze-und-Verordnungen/zur-Gesundheit/zur-gesundheitskarte/spezifikationen/spez-egk.html

[ELSTER] Bayrisches Landesamt für Steuern: *ELSTER – die elektronische Steuererklärung*, https://www.elster.de, 2005.

[ePass] Bundesamt für Sicherheit in der Informationstechnik: *ePass - Der Reisepass mit biometrischen Merkmalen*, via http://www.bsi.bund.de/fachthem/epass/index.htm

[gemC-API] gematik: Interaktion zwischen eGK und HBA - Karten-Service-Schnittstelle für die Einbindung der eSign-Anwendung in eine Client-Anwendung, Version 0.4, 05.10.2005

[HBA-Sp] *Heilberufs-Ausweis und Security Module Card – Teil 1-3*, Version 2.1.0, 28.05.2006, via http://www.bmg.bund.de/cln_040/nn_893332/DE/Gesetze-

und-Verordnungen/zur-Gesundheit/zur-gesundheitskarte/spezifikationen/spez-Heilberufeausweis.html

[Hueh05] D. Hühnlein: *Die CCES Signature API – eine offene Programmierschnittstelle für langfristig beweiskräftige elektronische Signaturen*, in Federrath H. (Hrsg.): Sicherheit 2005: Sicherheit - Schutz und Zuverlässigkeit, Beiträge der 2. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 5.-8. April 2005 in Regensburg. LNI 62 GI, 2005, Seiten 361-374, via http://www.secunet.com/download/fachartikel/cces-api.pdf

[HuBa07] D. Hühnlein, M. Bach: *A generic alternative to ISO/IEC 24727-2*, in Vorbereitung[ISO7816-15] ISO/IEC: Information technology — Identification cards - Integrated circuit(s) cards with contacts — Part 15: Cryptographic information application, ISO/IEC 7816-15, 2003

[ISO24727-2] ISO/IEC: Identification cards — Integrated circuit cards programming interfaces — Part 2: Generic Card Interface, ISO/IEC 24727-2, Final Committee Draft (2006-07-30), 2006

[ISO24727-3] ISO/IEC: Identification Cards — Integrated Circuit Cards Programming Interfaces — Part 3: Application Interface, ISO/IEC 24727-3, Committee Draft (2006-09-07), 2006

[Jobcard] ITSG Informationstechnische Servicestelle der gesetzlichen Krankenversicherung GMBH: *Das JobCard-Verfahren*, via http://www.itsg.de/download/BroschuereJobcard.pdf , 2004.

[MS-CAPI] Microsoft Inc.: *Cryptography Reference (Microsoft CryptoAPI), Platform SDK: Security*, via http://msdn.microsoft.com/library/en-us/security/security/cryptography_reference.asp

[MS-CNG] Microsoft Inc.: *Cryptography API: Next Generation* http://msdn.microsoft.com/library/en-us/seccrypto/security/cryptography_api__next_generation.asp

[PC/SC] PC/SC Workgroup, *PC/SC Workgroup Specifications 1.0/2.0*, via http://pcscworkgroup.com

[PKCS#7] RSA Labs: *PKCS #7: Cryptographic Message Syntax Standard*, via http://www.rsalabs.com/pkcs/pkcs-ia7/index.html

[PKCS#10] RSA Labs: *PKCS#10: Certification Request Syntax Standard*, Version 1.7, viahttp://www.rsasecurity.com/rsalabs/pkcs/pkcs-10/index.html

[PKCS#11] RSA Labs: *PKCS#11: Cryptographic Token Interface Standard*, Version 2.11, viahttp://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/index.html

[PKCS#12] RSA Labs: *PKCS#12: Personal Information Exchange Syntax Standard*, Version 1.0, via http://www.rsasecurity.com/rsalabs/pkcs/pkcs-12/index.html

[RFC3161] C. Adams, P. Cain, D. Pinkas, R. Zuccherato: *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*. IETF RFC 3161, August 2001, via http://www.ietf.org/rfc/rfc3161.txt

[RFC3275] D. Eastlake, D.; Reagle, J.; Solo, D.: *(Extensible Markup Language) XML-Signature Syntax and Processing*, RFC 3275, via http://www.ietf.org

[RFC3369] R. Housley: *Cryptographic Message Syntax (CMS)*, IETF RFC 3369., via http://www.ietf.org/rfc/rfc3369.txt

[RFC4346] T. Dierks, E. Rescorla: *The Transport Layer Security (TLS) Protocol*, Version 1.1, IETF RFC 4346., via http://www.ietf.org/rfc/rfc4346.txt

[RRM05] H. Reichl, A. Roßnagel, G. Müller (Hrsg.): *Digitaler Personalausweis – Eine Machbarkeitsstudie*, Deutscher Universitätsverlag, 2005, ISBN: 3835000543.

[SICCT] TeleTrusT e.V.: *Secure Interoperable ChipCard Terminal (SICCT)*, Version 1.0.0 vom 28.02.2006, via http://www.bmg.bund.de/cln_040/nn_893332/SharedDocs/Gesetzestexte/Gesundheitskarte/SICCT-Spezifikaton,templateId=raw,property=publicationFile.pdf/SICCT-Spezifikaton.pdf

[SASCIA] Signaturbündnis / SRC GmbH: Specification of the Application Programming Interface to the Signature Card, Version 1.0 vom 14.01.2005

[XAdES] W3C Note: *XML Advanced Electronic Signatures (XAdES)*, 20. Februar 2003, via http://www.w3.org/TR/XAdES/

[XML-Enc] W3C Recommendation: *XML Encryption Syntax and Processing*, 10. Dezember 2002, via http://www.w3.org/TR/xmlenc-core/