Langfristig beweiskräftige Signaturen mit dem eCard-API-Framework

Detlef Hühnlein[1] · Stefanie Fischer-Dieskau[2] · Utz Gnaida[2]
Ulrike Korte[2] · Peter Rehäußer[3] · Wolf Zimmer[3]

[1]secunet Security Networks AG, detlef.huehnlein@secunet.com
[2]Bundesamt für Sicherheit in der Informationstechnik,
{stefanie.fischer-dieskau,utz.gnaida,ulrike.korte}@bsi.bund.de
[3]CSC Deutschland Solutions GmbH {prehaeus,wzimmer2}@csc.com

Zusammenfassung

Durch das eCard-API-Framework [BSI-TR03112] wird ein einheitlicher und einfacher Zugriff auf die unterschiedlichen am Markt angebotenen Signaturkarten ermöglicht und somit die Integration der elektronischen Signatur in unterschiedliche Anwendungen signifikant erleichtert. Darüber hinaus kann dieses auf internationalen Standards (vgl. [HuBa08]) basierende Rahmenwerk bei der vertrauenswürdigen Langzeitarchivierung gemäß [BSI-TR03125] genutzt werden, damit die Beweiskraft von qualifizierten elektronischen Signaturen langfristig bewahrt werden kann. Dieser Beitrag liefert einen Überblick über die wichtigsten, für die elektronische Signatur relevanten Funktionen des eCard-API-Frameworks und geht näher auf die Nutzung des Frameworks zur vertrauenswürdigen Langzeitarchivierung ein.

1 Einleitung

Neben der Authentisierung zählt die elektronische Signatur zu den wesentlichen Stützpfeilern der eCard-Strategie der Bundesregierung [Kowa07]. Eine wichtige Rolle bei der Umsetzung dieses Vorhabens wiederum spielt das eCard-API-Framework [BSI-TR03112], durch das beliebige Applikationen in einheitlicher Weise auf die unterschiedlichen, in den Projekten der Bundesregierung ausgegebenen oder genutzten Chipkarten zugreifen können.

Durch die unterschiedlichen Schichten des Frameworks (vgl. [BSI-TR03112], Teil 1-8, jeweils Abbildung 1) wird eine grundsätzliche Entkopplung der Anwendung von den eingesetzten Signaturerstellungseinheiten ermöglicht, wobei die nötigen Detailinformationen für die Abbildung der generischen Funktionen auf chipkartenspezifische APDUs dem Framework in Form von XML-basierten CardInfo-Files (vgl. [HuBa07] und [CEN15480-3]) bereit gestellt werden. Oberhalb der in [BSI-TR03112] spezifizierten Schnittstellen können so genannte „Convenience-Interfaces“ existieren, in denen die generischen Sicherheitsfunktionen des eCard-API-Frameworks in einer anwendungsspezifischen Weise genutzt und gekapselt werden.

Neben einem kompakten Überblick über die wichtigsten für die elektronische Signatur relevanten Funktionen des eCard-API-Frameworks liefert der vorliegende Beitrag eine solche

anwendungsspezifische Ergänzung des eCard-API-Frameworks für Zwecke der vertrauenswürdigen Langzeitarchivierung gemäß [BSI-TR03125].

Der Rest des Beitrags ist folgendermaßen gegliedert: Abschnitt 2 beschreibt die wesentlichen Funktionen des eCard-API-Frameworks im Umfeld der elektronischen Signatur. Abschnitt 3.2 liefert einen Überblick über die in [BSI-TR03125] definierte Referenzarchitektur zur vertrauenswürdigen Langzeitarchivierung und skizziert deren Umsetzung auf Basis des eCard-API-Frameworks. Abschnitt 4 enthält schließlich einen kurzen Ausblick auf zukünftige Entwicklungen.

2 Signaturen mit dem eCard-API-Framework

Bei der Betrachtung der wesentlichen Signaturfunktionen kann zunächst zwischen dem Erzeugen und dem Prüfen von Signaturen unterschieden werden.

2.1 Erzeugen von elektronischen Signaturen

Beim Erzeugen von elektronischen Signaturen wirken typischer Weise verschiedene Funktionen in den unterschiedlichen Schichten des eCard-API-Frameworks zusammen.

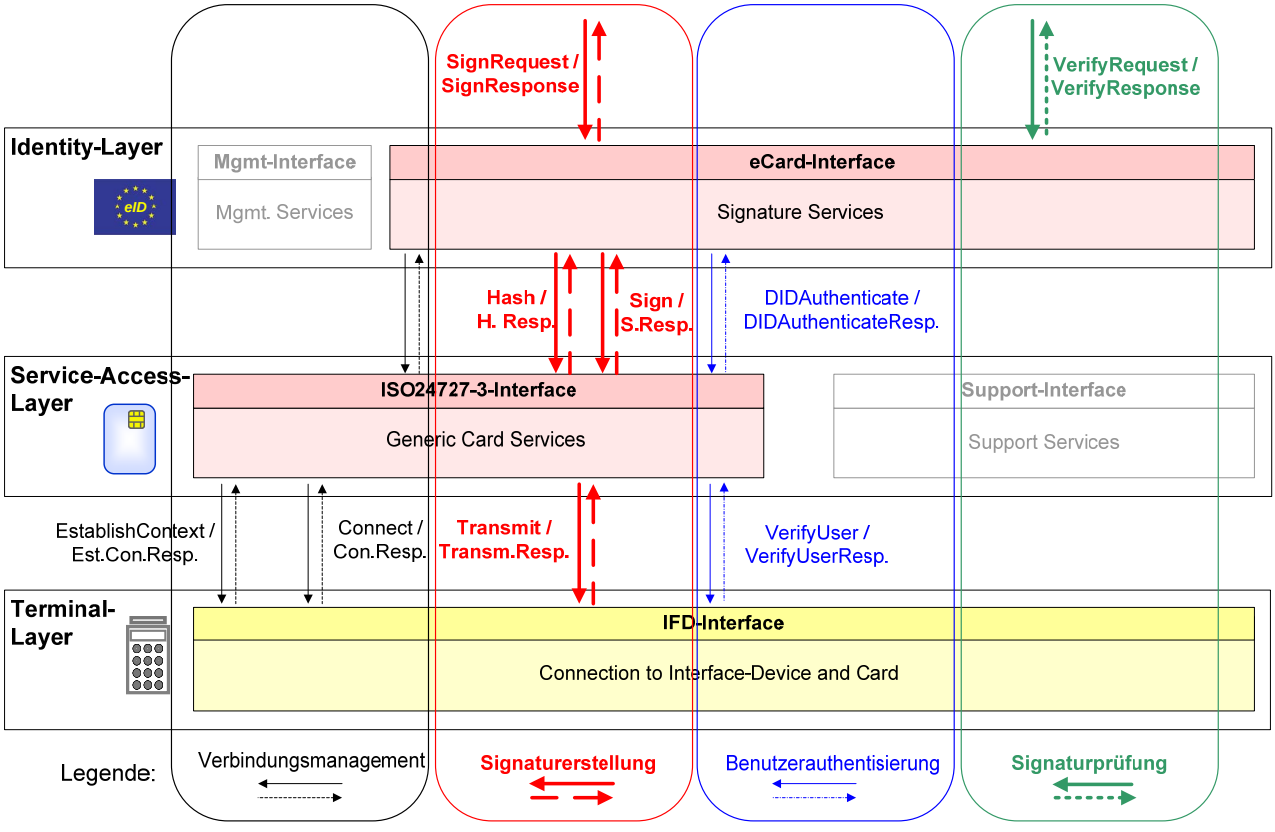

Abbildung 1: Signatur-relevante Funktionen des eCard-API-Frameworks

2.1.1 Schichtenübergreifende Abläufe bei der Signaturerstellung

Wie in Abbildung 1 angedeutet, löst hierbei ein `SignRequest` am eCard-Interface des ***Identity Layers*** (siehe [BSI-TR03112] (Teil 2) bzw. [OASIS-DSS]) den Aufruf der Funktionen `Hash` und `Sign` aus.

Für diese Aufrufe am ***Service Access Layer*** (SAL) (siehe [BSI-TR03112] (Teil 4) bzw. [ISO24727-3]) wird jedoch zumindest eine logische Verbindung zur Chipkartenapplikation benötigt, die mittels `CardApplicationConnect` aufgebaut wird. Sofern für die Kommunikation mit der Chipkartenapplikation ein kryptographisch geschützter Kanal benötigt wird, kann dieser mit `CardApplicationStartSession` aufgebaut werden, so dass fortan die für die Chipkarte vorgesehenen Application Protocol Data Units (APDUs) mittels „Secure Messaging“ gemäß [ISO7816-8] (Abschnitt 8) geschützt werden können.

Im Rahmen der Initialisierung des Frameworks wird für die Kommunikation zwischen dem SAL und dem ***Interface Device (IFD) – Layer*** (siehe [BSI-TR03112] (Teil 6) bzw. [ISO24727-4]) mittels `EstablishContext` ein Kommunikationskontext aufgebaut, in dem schließlich mit `Connect` technische Verbindungen zu den erfassten Chipkarten aufgebaut werden. Über eine solche technische Verbindung werden die im SAL gebildeten APDUs mit der `Transmit`-Funktion an den IFD–Layer gereicht und dort schließlich über das Chipkartenterminal und ein geeignetes Transportprotokoll zu einer kontaktbehafteten[1] oder kontaktlosen[2] Chipkarte übertragen. Anders als bei [PC/SC-5] wird bei der IFD-API gemäß [ISO24727-4] bewusst von den verwendeten Transportprotokollen abstrahiert, so dass der SAL von der Einführung neuartiger Übertragungsprotokolle gänzlich unberührt bleiben würde.

In der Regel ist vor dem Zugriff auf den privaten Signaturschlüssel eine ***Benutzerauthentisierung*** erforderlich. Deshalb wird vor dem Aufruf der SAL-Funktion `Sign` die Funktion `DIDAuthenticate` aufgerufen, die beliebige Authentisierungsprotokolle (vgl. [BSI-TR03112] (Teil 7) und [ISO24727-3] (Annex A)) nutzen kann. Somit könnte die Benutzerauthentisierung nicht nur mittels der klassischen PIN oder einem PACE-Passwort (vgl. [BSI-TR03110], Abschnitt 4.2) sondern auch mittels biometrischer Verfahren oder durch den Nachweis des Besitzes eines RFID-Tokens (vgl. [BSI-TR03115]) erfolgen. Sofern die Eingabe der PIN am Kartenterminal erfolgen soll oder ein biometrisches Merkmal über einen entsprechenden Sensor erfasst werden muss, wird die Funktion `VerifyUser` im IFD-Layer genutzt.

2.1.2 Erstellen von Signaturen und Zeitstempeln

Wie in [BSI-TR03112] (Teil 2) erläutert, kann beim Aufruf von `SignRequest` angegeben werden, welcher Signatur- oder Zeitstempel-Typ erzeugt und wie die Signatur im Detail gestaltet werden soll.

Deshalb kann mit dem `SignatureType`-Parameter grundsätzlich angegeben werden, welche der folgenden Signatur- und Zeitstempelformate (vgl. [HuKo06]) erstellt werden soll:

- urn:ietf:rfc:3275 für XML-Signaturen gemäß [RFC3275]

[1] Z.B. mittels dem T=0 oder T=1 Protokoll gemäß [ISO7816-3].

[2] Z.B. mit dem in [ISO14443-4] definierten Transportprotokoll.

- urn:ietf:rfc:3369 für CMS-Signaturen gemäß [RFC3369]
- http://ns.adobe.com/pdf für integrierte PDF-Signaturen gemäß [AI-PDF]
- urn:ietf:rfc:3161 für Zeitstempel gemäß [RFC3161]
- urn:oasis:names:tc:dss:1.0:core:schema:XMLTimeStampToken für XML-basierte Zeitstempels gemäß [OASIS-DSS] Abschnitt 5.1.1
- urn:ietf:rfc:4998 für Archivzeitstempel gemäß [RFC4998]

Außerdem kann für die Erzeugung von fortgeschrittenen XML- oder CMS-basierten Signaturen gemäß [XAdES] oder [CAdES] über eine zusätzliche URI im `SignatureForm`-Element angegeben werden, welche der standardisierten Signaturtypen erzeugt werden soll. Beispielsweise würde man die komplexen, zur langfristigen Archivierung geeigneten Signaturformate XAdES-A gemäß [XAdES] Anhang B.3 bzw. CAdES-A gemäß Abschnitt 4.4.4 von [CAdES] einfach über die in Abschnitt 7.1 von [OASIS-AdES] spezifizierte URI urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:ES-A anfordern können.

2.2 Prüfen von elektronischen Signaturen

Wie in Abbildung 1 ersichtlich ist beim Prüfen von elektronischen Signaturen typischer Weise nur der Identity Layer involviert, da bei der Prüfung von elektronischen Signaturen in der Regel[3] nicht auf eine Hardwaretoken zugegriffen werden muss. In der Funktion `VerifyRequest` wird dem eCard-API-Framework eine Folge von Dokumenten und/oder Signaturen übergeben und man erhält einen mehr oder weniger ausführlichen `VerificationReport` zurück.

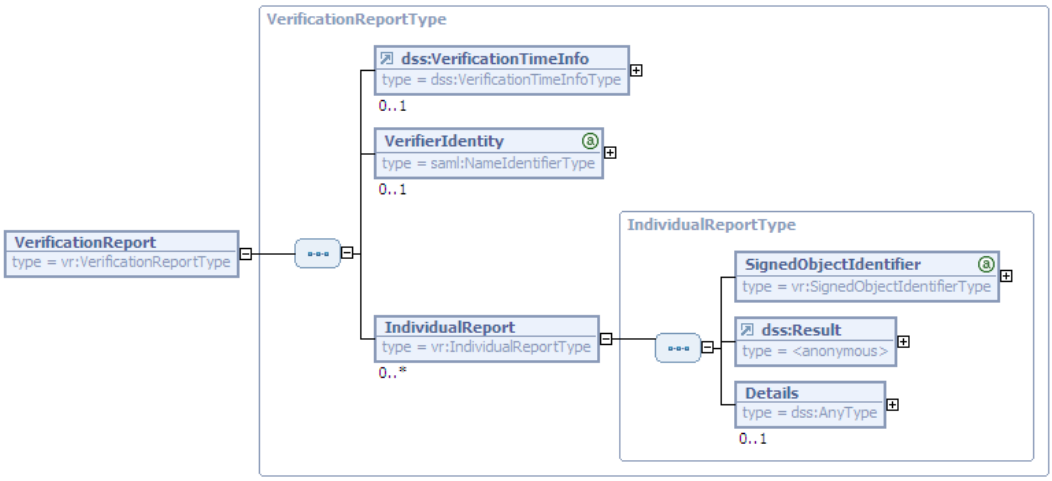

Abbildung 2: Struktur eines `VerificationReport`

Wie in Abbildung 2 ersichtlich, kann ein solcher Prüfbericht neben Angaben zum Verifikationszeitpunkt (`dss:VerificationTimeInfo`) und zur Identität des Prüfers (`VerifierI-`

[3] Eine Ausnahme würde hier beispielsweise ein Szenario bilden, in dem ein auf einer sicheren Signaturerstellungseinheit abgelegtes Wurzelzertifikat der Bundesnetzagentur als Vertrauensanker für die Signaturprüfung genutzt wird, da in diesem Fall auch der Service-Access-Layer und der IFD-Layer benötigt werden würde.

dentity) insbesondere für jedes übergebene individuelle[4] Signaturobjekt ein `IndividualReport`-Element enthalten, in dem ein eindeutiger Bezeichner des signierten Objektes (`SignedObjectIdentifier`), das Prüfergebnis (`dss:Result`) und ggf. weitere Detailinformationen (`Details`) enthalten sind. Die detaillierten Strukturen sind abhängig vom Typ des Signaturobjektes.

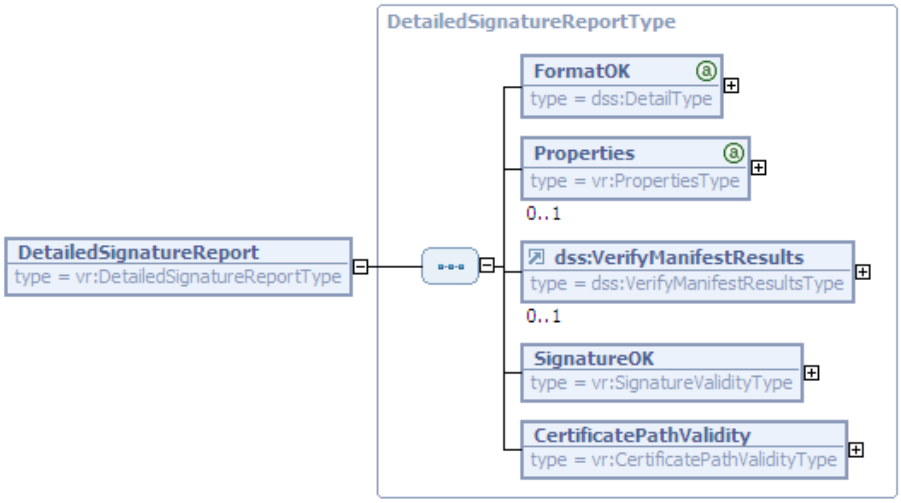

Abbildung 3: Struktur eines `DetailedSignatureReport`

Wie in Abbildung 3 ersichtlich, kann ein `DetailedSignatureReport` Informationen über verschiedene Aspekte der Gültigkeit enthalten:

- `FormatOK` – gibt an, ob die Signatur syntaktisch korrekt ist.
- `Properties` – enthält Informationen über die Gültigkeit der signierten und unsignierten Attribute / Eigenschaften einer Signatur und kann bei einer fortgeschrittenen elektronischen Signatur gemäß [CAdES] oder [XAdES] äußerst vielfältige Informationen über die Gültigkeit von weiteren signierten Objekten (z.B. Signaturen, Zeitstempeln, Public-Key-Zertifikaten, Attribut-Zertifikaten, CRLs, OCSP-Responses und Trust-Service Status-Lists) enthalten.
- `dss:VerifyManifestResults` – kann bei einer XML-Signatur das Ergebnis der Prüfung eines Manifests enthalten.
- `SignatureOK` – gibt an, ob die digitale Signatur mathematisch korrekt ist.
- `CertificatePathValidity` – enthält Informationen über die Gültigkeit des Zertifikatspfades der Signatur bis hin zu einer vertrauenswürdigen Wurzel.

[4] Ein „individuelles Signaturobjekt“ ist ein Signaturobjekt, das nicht bereits in einem anderen Signaturobjekt enthalten ist oder auf das in einem anderen übergebenen Signaturobjekt verwiesen wird. Beispielsweise wäre ein Zertifikat, auf dem eine fortgeschrittene elektronische Signatur basiert und auf das mit einem `SigningCertificate`-Attribut (vgl. [CAdES], Abschnitt 5.7.3) bzw. einer gleichnamigen Eigenschaft (vgl. [XAdES], Abschnitt 7.2.2) verwiesen wird, *kein individuelles* Signaturobjekt.

3 Langzeitarchivierung mit eCard-API-Framework

In diesem Abschnitt wird erläutert, wie das eCard-API-Framework zur vertrauenswürdigen Langzeitarchivierung genutzt werden kann. Hierfür wird in Abschnitt 3.1 zunächst die abstrakte Referenzarchitektur aus [BSI-TR03125] vorgestellt, bevor in Abschnitt 3.2 näher auf die konkrete Umsetzung auf Basis des eCard-API-Frameworks eingegangen wird.

3.1 Referenzarchitektur zur Langzeitarchivierung

Die in [BSI-TR03125] entwickelte Referenzarchitektur (siehe auch [RZK+09]) basiert auf den Vorarbeiten des ArchiSig[5]- (siehe [RoSc05]) und ArchiSafe[6]-Projektes (siehe [HaRo08, ZiLH08]) und stützt sich insbesondere auf die inzwischen in [RFC4998] standardisierte Evidence Record Syntax.

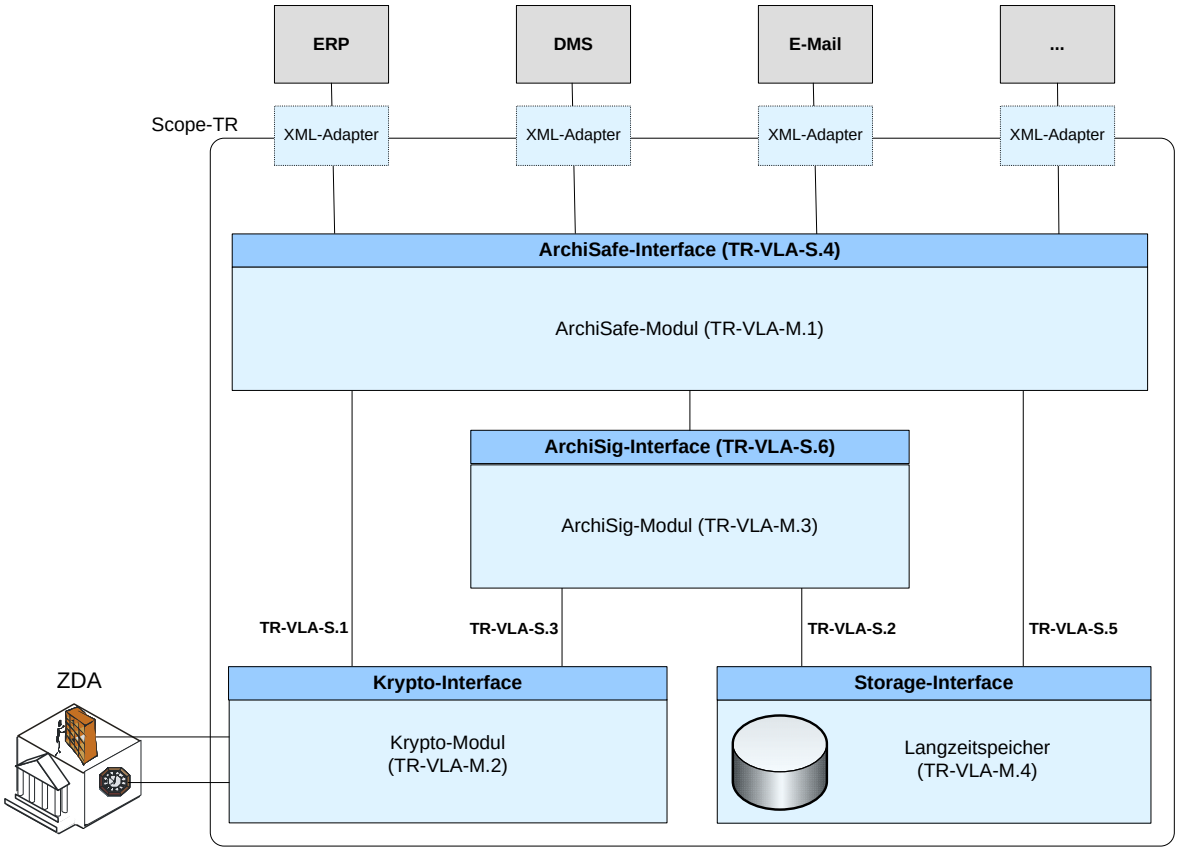

Abbildung 4: Referenzarchitektur zur vertrauenswürdigen Langzeitarchivierung

3.1.1 Module

Wie in Abbildung 4 dargestellt, besteht die Referenzarchitektur aus den folgenden Modulen:

[5] Siehe http://www.archisig.de/ .

[6] Siehe http://www.archisafe.de/ .

- Das ***ArchiSafe-Modul*** nimmt die Archivanfragen der Geschäftsanwendungen[7] entgegen und steuert die wesentlichen Abläufe im Archiv, indem das ArchiSig-Modul oder der Langzeitspeicher angesprochen werden. Sofern die zu archivierenden Daten signiert sind, wird i.d.R. mittels des Krypto-Moduls die Prüfung der im Archivdatenobjekt enthaltenen Signaturen angestoßen.
- Das ***ArchiSig-Modul*** verwaltet die Merkle-Hashbäume [Merk80] zu den registrierten Archivdatenobjekten, fordert bei Bedarf über das Krypto-Modul (qualifizierte) Zeitstempel gemäß [RFC3161] an und erzeugt auf Anforderung technische Beweisdaten gemäß [RFC4998].
- Das ***Krypto-Modul*** ist zumindest in der Lage, Hashwerte zu berechnen sowie Signaturen und Zeitstempel samt der zugehörigen Zertifikatsketten zu prüfen. Außerdem können (qualifizierte) Zeitstempel bei einem Zertifizierungsdiensteanbieter (ZDA) angefordert werden.
- Im ***Langzeitspeicher*** werden schließlich die Archivdatenobjekte abgelegt und können bei Bedarf dort wieder ausgelesen oder – z.B. nach Ablauf der Aufbewahrungsfrist – gelöscht werden.

3.1.2 Archivierungsprozesse

Ein Archivsystem im Sinne der [BSI-TR03125] unterstützt insbesondere die folgenden von der Geschäftsanwendung initiierten Archivierungsprozesse:

- Archivierung elektronischer Daten (vgl. Abschnitt 3.2.1)
- Abfrage archivierter Daten (vgl. Abschnitt 3.2.2)
- Löschen von Archivdaten (vgl. Abschnitt 3.2.3)
- Rückgabe technischer Beweisdaten (vgl. Abschnitt 3.2.4)

Außerdem muss das ArchiSig-Modul oder der Betreiber des vertrauenswürdigen Archivs die eingesetzten kryptographischen Signatur- und Hashalgorithmen überwachen und bei Bedarf neue Zeitstempel anfordern oder den Hashbaum neu aufbauen (vgl. [RFC4998]).

3.2 Realisierung auf Basis des eCard-API-Frameworks

Während für die Umsetzung der in [BSI-TR03125] entwickelten abstrakten Referenzarchitektur verschiedene Möglichkeiten existieren, ist es besonders naheliegend und empfehlenswert, dies auf Basis des eCard-API-Frameworks [BSI-TR03112] durchzuführen.

Hierfür sind zwei einfache Schritte notwendig:

1. Wie in Abbildung 5 ersichtlich, wird das in der Referenzarchitektur vorgesehene Krypto-Modul (siehe Abbildung 4 und Abschnitt 3.1.1) mittels einer Implementierung des eCard-API-Frameworks realisiert, wodurch auf die eigenständige und aufwändige Implementierung, Evaluierung und Zertifizierung dieser Komponente verzichtet werden kann.

[7] Sofern die Geschäftsanwendung (Enterprise Ressource Planning (ERP), Document Management System (DMS), E-Mail etc.) nicht bereits selbst die zu archivierenden Daten in Form von XML-basierten Archivdatenobjekten (XML formatted Archival Information Package (`XAIP`), vgl. [ISO14721]) liefern kann, sorgt ein optionaler *XML-Adapter* für die Konvertierung.

2. Außerdem werden die weiteren, aus den notwendigen Archivierungsprozessen (vgl. Abschnitt 2.2) abgeleiteten Schnittstellen (siehe Abschnitte 3.2.1-3.2.4) auf Basis der auch dem eCard-API-Framework [BSI-TR03112] zu Grunde liegenden Basistypen (`RequestBaseType` und `ResponseBaseType`) aus [OASIS-DSS] umgesetzt. Hierdurch können bewährte Werkzeuge und Bibliotheken nahtlos weiterverwendet werden und es entsteht eine harmonisch aufeinander abgestimmte Gesamtlösung, die zudem das Potenzial besitzt, in die internationale Standardisierung einzufließen.

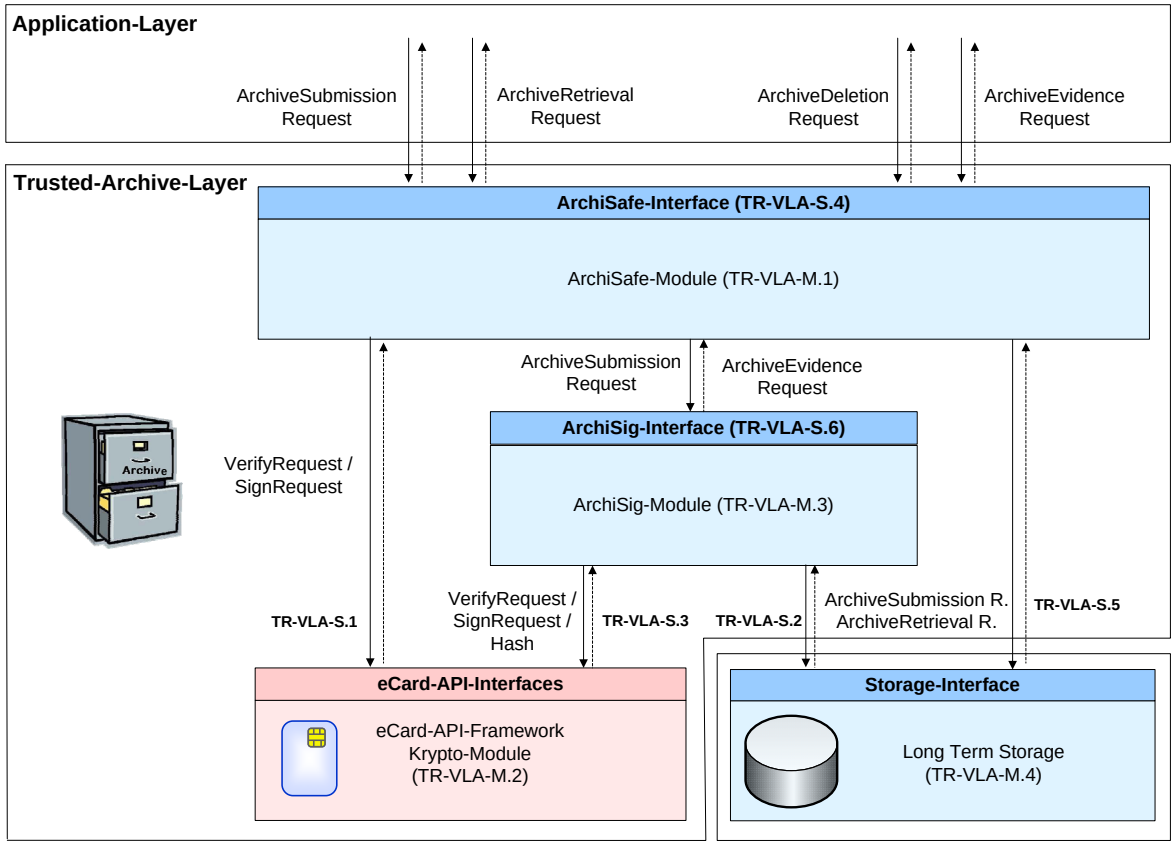

Abbildung 5: Realisierung auf Basis des eCard-API-Frameworks

Aus den durch die Geschäftsanwendung initiierten Archivierungsprozessen (vgl. Abschnitt 2.2) leiten sich die im Folgenden näher betrachteten Schnittstellen ab:

- ArchiveSubmissionRequest / -Response
- ArchiveRetrievalRequest / -Response
- ArchiveDeletionRequest / -Response
- ArchiveEvidenceRequest / -Response

3.2.1 ArchiveSubmissionRequest / -Response

Hierbei wird im Wesentlichen ein Archivdatenobjekt (XML-formatted Archive Information Package, `XAIP`) übergeben und man erhält ein so genanntes `ArchiveToken` zurück, mit dem man – analog einer „Garderobenmarke“ – später das dadurch eindeutig bezeichnete Ar-

chivdatenobjekt wieder auslesen (vgl. Abschnitt 3.2.2) und löschen (vgl. Abschnitt 3.2.3) oder zugehörige Beweisdaten gemäß [RFC4998] (vgl. Abschnitt 3.2.4) anfordern kann. Dieser Funktionstyp wird vom ArchiSafe- und ArchiSig-Modul sowie in ähnlicher Form vom Langzeitspeicher angeboten. Sofern das übergebene Archivdatenobjekt elektronische Signaturen enthält, werden diese unter Verwendung der `VerifyRequest`-Funktion gemäß [BSI-TR03112] (Teil 2) vor der Archivierung geprüft (vgl. Abschnitt 2.2).

3.2.2 ArchiveRetrievalRequest / -Response

Nach Übergabe eines `ArchiveToken` wird das entsprechende `XAIP` zurückgeliefert. Diese Funktion wird vom ArchiSafe-Modul und vom Langzeitspeicher angeboten.

3.2.3 ArchiveDeletionRequest / -Response

Nach Übergabe des `ArchiveToken` und ggf. einer entsprechenden Begründung für das Löschen wird das entsprechende `XAIP` im Langzeitspeicher gelöscht. Diese Funktion wird vom ArchiSafe-Modul und vom Langzeitspeicher angeboten.

3.2.4 ArchiveEvidenceRequest / -Response

Für ein mittels `ArchiveSubmissionRequest` (vgl. Abschnitt 3.2.1) archiviertes `XAIP` kann bei Bedarf mittels `ArchiveEvidenceRequest` ein entsprechender `EvidenceRecord` gemäß [RFC4998] angefordert werden, durch den die Integrität und Authentizität des Archivdatenobjekts langfristig gewahrt und somit die Beweiskraft der archivierten Daten erhalten werden kann.

4 Zusammenfassung

In diesem Beitrag wurden die wesentlichen für die elektronische Signatur relevanten Funktionen des eCard-API-Frameworks vorgestellt und erläutert, wie auf Basis dieser Funktionen eine Umsetzung der derzeit entstehenden Technischen Richtlinie [BSI-TR03125] erfolgen kann. Dadurch kann ein in sich geschlossenes Signatursystem geschaffen werden, bei dem der ***komplette Lebenszyklus der (qualifizierten) elektronischen Signatur*** – von der Erstellung über die Prüfung bis hin zur dauerhaften Aufbewahrung – durch harmonisch aufeinander abgestimmte Module unterstützt werden kann.

Literatur

[AI-PDF] Adobe Inc.: *Portable Document Format Reference Manual*, Version 1.3-1.6, via http://partners.adobe.com/asn/tech/pdf/specifications.jsp

[BSI-TR03110] Bundesamt für Sicherheit in der Informationstechnik (BSI): *Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI)*, Technical Guideline TR-03110, http://www.bsi.bund.de/english/publications/techguidelines/tr03110/index.htm, 2008

[BSI-TR03112] Bundesamt für Sicherheit in der Informationstechnik (BSI): *eCard-API-Framework*, Technische Richtlinie (TR) des BSI Nr. 03112, Teil 1-8, http://www.bsi.de/literat/tr/tr03112/index.htm, 2008

[BSI-TR03115] Bundesamt für Sicherheit in der Informationstechnik (BSI): *Komfortsignatur mit dem Heilberufsausweis*, Technische Richtlinie (TR) des BSI Nr. 03115, Version 2.0, http://www.bsi.bund.de/literat/tr/tr03115/BSI-TR-03115.pdf , 2007

[BSI-TR03125] Bundesamt für Sicherheit in der Informationstechnik (BSI): *VLA – Vertrauenswürdige elektronische Langzeitarchivierung*, Technische Richtlinie (TR) des BSI Nr. 03125, in Vorbereitung, 2008

[CAdES] ETSI: *Electronic Signature Formats*, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V1.7.4, http://www.etsi.org

[CEN15480-3] Comité Européen de normalisation (CEN): *Identification card systems - European Citizen Card – Part 3: European Citizen Card Interoperability using an application interface*, CEN/TS 15480-3 (Working Draft), 2008

[HaRo08] Siegfried Hackel und Alexander Roßnagel: *Langfristige Aufbewahrung elektronischer Dokumente*, in *Informationelles Vertrauen für die Informationsgesellschaft*, (Hrsg.) D. Klumpp, H. Kubicek, A. Roßnagel und W. Schulz, Springer-Verlag, 2008, SS. 199-207

[HuBa07] Detlef Hühnlein, Manuel Bach: *How to Use ISO/IEC 24727-3 with Arbitrary Smart Cards*, TrustBus 2007, LNCS 4657, Springer, Seiten 280–289, 2007, http://www.ecsec.de/pub/2007_TrustBus.pdf

[HuBa08] Detlef Hühnlein, Manuel Bach: *Die Standards des eCard-API-Frameworks – Eine deutsche Richtlinie im Konzert internationaler Normen*, Datenschutz und Datensicherheit (DuD), 06/2008, SS. 379-382, http://www.ecsec.de/pub/2008_DuD_eCard.pdf

[HuKo06] Detlef Hühnlein, Ulrike Korte: *Signaturformate für elektronische Rechnungen*, in Horster P. (Hrsg.): Tagungsband „D•A•CH Security“, 2006, Seiten 1-14, http://www.ecsec.de/pub/2006_DACH_Signaturformate.pdf

[ISO7816-3] ISO/IEC 7816: *Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols*, International Standard, 2006

[ISO7816-8] ISO/IEC 7816: *Identification cards — Integrated circuit cards — Part 8: Security related interindustry commands*, International Standard, 2008

[ISO14443-4] ISO/IEC 14443: *Identification cards — Contactless integrated circuit cards — Proximity cards — Part 4: Transmission protocol*, International Standard, 2008

[ISO14721] ISO 14721: *Space data and information transfer systems – Open archival information system – Reference model*, International Standard, 2003 (vgl. http://public.ccsds.org/publications/archive/650x0b1.pdf)

[ISO24727-3] ISO/IEC 24727-3: *Identification cards – Integrated circuit cards programming interfaces – Part 3: Application programming interface*, International Standard, 2008

[ISO24727-4] ISO/IEC 24727-4: *Identification cards – Integrated circuit cards programming interfaces – Part 4: Application programming interface (API) administration*, International Standard, 2008

[Kowa07] Bernd Kowalski: *Die eCard-Strategie der Bundesregierung im Überblick*, in BIOSIG 2007: Biometrics and Electronic Signatures, Proceedings of the Special Interest Group on Biometrics and Electronic Signatures, LNI 108, Seiten 87–96, 2007

[Merk80] Ralph Merkle: *Protocols for Public Key Cryptosystems*, Proceedings of the 1980 IEEE Symposium on Security and Privacy (Oakland, CA, USA), SS. 122-134, 1980

[OASIS-AdES] OASIS: *Advanced Electronic Signature Profiles of the OASIS Digital Signature Service Version 1.0*, OASIS Standard, http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-spec-v1.0-os.pdf, 2007

[OASIS-DSS] OASIS: *Digital Signature Service Core Protocols, Elements, and Bindings, Version 1.0*, OASIS Standard, http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf, 2007

[PC/SC-5] PC/SC Workgroup: *Interoperability Specification for ICCs and Personal Computer Systems - Part 5. ICC Resource Manager Definition*, Revision 2.01.01, September 2005, http://pcscworkgroup.com

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

[RFC3275] D. Eastlage, J. Reagle, D. Solo: *(Extensible Markup Language) XMLSignature Syntax and Processing*, IETF RFC 3275, http://www.ietf.org/rfc/rfc3275.txt

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

[RFC4998] T. Gondrom, R. Brandner, U. Pordesch: *Evidence Record Syntax (ERS)*, IETF RFC 4998, http://www.ietf.org/rfc/rfc4998.txt

[RoSc05] Alexander Roßnagel und Paul Schmücker: *Beweiskräftige elektronische Archivierung - Bieten elektronische Signaturen Rechtssicherheit? Ergebnisse des Forschungsprojekts „ArchiSig“*, Economica Verlag, 2005

[RZK+09] P. Rehäußer, W. Zimmer, U. Korte, D. Hühnlein, S. Fischer-Dieskau, U. Gnaida: *Technische Richtlinie zur vertrauenswürdigen Langzeitarchivierung*, im vorliegenden Tagungsband

[XAdES] ETSI: *Technical Specification XML Advanced Electronic Signatures (XAdES)*, ETSI Technical Specification TS 101 903, Version 1.3.2. http://webapp.etsi.org/action/PU/20060307/ts_101903v010302p.pdf

[ZiLH08] Wolf Zimmer, Thomas Langkabel und Carsten Hentrich: *ArchiSafe: Legally Compliant Electronic Storage*, in IT Professional, vol. 10, no. 4, SS. 26-33, Jul/Aug, 2008