Douglas Cunningham, Petra Hofstedt, Klaus Meer, Ingo Schmitt (Hrsg.): INFORMATIK 2015
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2015

Haltung und Übertragung von Patientendaten im Cloud Computing: Anforderungserhebung und prototypische Implementierung eines Verschlüsselungsframeworks

Marc Walterbusch, Tim Hoffmann und Frank Teuteberg[1]

Abstract: Ziel des Beitrags ist es auf Basis eines systematischen Literaturreviews sowie eines Experteninterviews Anforderungen für die Übertragung und Haltung von Patientendaten zu identifizieren. Auf Basis dieser Anforderungen wird prototypisch ein Verschlüsselungsframework implementiert und anhand zweier Experteninterviews konzeptionell sowie bei einem Arzneimittellieferanten mittels zweier Anwendungsbeispiele (Verschlüsselte Backups sowie Arzneimittelbestellungen) technisch evaluiert. Auf Basis des systematischen Literaturreviews lässt sich statuieren, dass eine gründliche Betrachtung der deutschen Gesetzgebung hinsichtlich Sicherheit und Datenschutz in der Literatur bisher ausbleibt. Diese Lücke soll mit diesem Beitrag geschlossen werden.

Keywords: Patientendaten, Übertragung, Cloud Computing, Verschlüsselung, Framework, Literaturreview, Experteninterview, Prototypische Implementierung, Anwendungsfall.

1 Einleitung und Motivation

Cloud Computing Services nehmen einen wichtigen Platz im privaten sowie im Geschäftsleben ein. In diesem Zusammenhang bietet das Cloud Computing auch neue grundlegende Möglichkeiten im Gesundheitssektor [EFZ13]. Hierbei sind verschiedene Szenarien denkbar, wie der Austausch von Patientendaten bei der Überweisung von Patienten zwischen Arztpraxen, die Verwaltung von elektronischen Patientenakten oder das Einsehen aktueller medizinischer Daten auf mobilen Geräten [CH11, Ch12]. Zudem können die vom Cloud Computing bekannten Potentiale wie Kosteneinsparungen, Effizienzvorteile und die Entwicklung neuer Geschäftsmodelle auf den Gesundheitssektor transferiert werden [LSW10, ZL10]. Jedoch verhindern aktuell noch Bedenken bezüglich Sicherheit und Datenschutz die breite Einführung in diesem Bereich [EFZ14]. In diesem Kontext sei auf die regulatorischen Vorschriften des Gesetzgebers hingewiesen, der strikte Vorgaben für die Erhebung, Verarbeitung und Nutzung von Patientendaten macht. Aus diesen regulatorischen Vorschriften sind entsprechend Anforderungen an Cloud Computing Services im Gesundheitssektor abzuleiten. Eine essentielle Maßnahme zur Gewährleistung von Datensicherheit ist die Verschlüsselung von Daten. Hierfür gibt es Ansätze in der Literatur wie z. B. das Trusted Cloud Transfer Protocol (TCTP), welches entwickelt wurde um den Kommunikationskanal zwischen

[1] Universität Osnabrück, Unternehmensrechnung und Wirtschaftsinformatik, Katharinenstr. 1, 49096 Osnabrück, {marc.walterbusch;thoffmann;frank.teuteberg}@uni-osnabrueck.de

User Agent (bspw. Browser) und Cloud zu verschlüsseln [Sl13]. Weiterhin sind in der Literatur Ansätze zu Sicherheitsframeworks zu finden, die Anforderungen an Cloud Computing Systeme im Gesundheitssektor aus anderen Quellen zusammenfassen [EFZ13]. Jedoch gibt es keine Frameworks, die den gesamten Zyklus von Systemen bspw. zum Speichern von Patientendaten zu Backup-Zwecken in der Cloud betrachten. Außerdem bleibt eine gründliche Betrachtung der deutschen Gesetzgebung hinsichtlich Sicherheit und Datenschutz in der Literatur bisher aus. Diese Lücke soll mit diesem Beitrag geschlossen werden. Basierend auf einem Literaturreview, unter Berücksichtigung der deutschen Gesetzgebung sowie im Rahmen eines Experteninterviews mit einem Arzneimittellieferanten wird prototypisch ein Verschlüsselungsframework entwickelt und evaluiert.

2 Literaturreview

2.1 Verwandte Arbeiten

Es wurde ein systematischer Literaturreivew mit dem Suchterm *("Best Practice*" OR encryption OR cryptography OR Verschlüsselung) AND (health OR medical OR Gesundheit*) AND (patient OR "patient data" OR data) AND cloud** durchgeführt [Br09, WW02]. Durchsucht worden sind der Top10-Tier der AIS [As15] sowie die fünf internationalen Wirtschaftsinformatikkonferenzen ECIS, ICIS, AMCIS, HICSS, und PACIS mit dem Ergebnis von 71 Literaturquellen, von denen 23 als relevant eingestuft werden konnten. Aufbauend auf den initialen Ergebnissen wurde eine Rückwärtssuche mit 22 relevanten Ergebnissen sowie eine Vorwärtssuche mit 1 relevanten Artikel durchgeführt [WW02].

Verwandte Arbeiten sind das *Trusted Cloud Transfer Protocol* (TCTP) [Sl13] sowie die Abhandlung zum Thema *Security and Privacy System Requirements for Adopting Cloud Computing in Healthcare Data Sharing Scenarios* [EFZ13]. Das TCTP ist ein Datenübertragungsprotokoll, das hauptsächlich für die Verwendung von Browser- bzw. Hypertext Transfer Protocol (HTTP)-basierenden Software as a Service (SaaS) Cloud Computing Lösungen entwickelt wurde [Sl13]. Slawik et al. [Sl14] zeigen beispielhaft, wie das TCTP in einer medizinischen SaaS Lösung verwendet werden kann, sodass die sichere Übertragung sensibler Daten gewährleistet wird. Im Gegensatz zum TCTP wird in dieser Arbeit ein allein lauffähiges Verschlüsselungsframework entwickelt, das selbstverwaltend ein Verschlüsselungsverfahren anwendet. Weitergehend geht es bei den Anwendungsfällen des Verschlüsselungsframeworks nicht ausschließlich um die Verschlüsselung der Datenübertragung, sondern ebenfalls um die verschlüsselte Datenspeicherung in der Cloud, die das TCTP nicht unterstützt. Bei dem zweitgenanntem Artikel [EFZ13] handelt es sich um eine verwandte Forschungsarbeit, die Sicherheits- und Datenschutzanforderungen an das Cloud Computing im Gesundheitssektor behandelt. Dafür gehen Ermakova et al. [EFZ13] nach dem systematischen Design Science Ansatz vor, wonach zunächst mittels eines

Literaturreviews eine Anforderungserhebung durchgeführt wird. Danach erfolgt die Anwendung der festgestellten Sicherheitsanforderungen mittels Szenarioanalyse. Die von Ermakova et al. [EFZ13] identifizierten Sicherheits- und Datenschutzanforderungen betreffen unmittelbar auch diese Arbeit, in der ein Verschlüsselungsframework für das Cloud Computing im Gesundheitssektor entwickelt werden soll, welches die genannten Anforderungen zum Teil abdecken muss. Hierbei gilt es jedoch zu differenzieren, da nicht alle Sicherheitsproblematiken durch eine Verschlüsslung gelöst werden können. Schließlich kann bspw. das Fehlen eines Notfallzugriffs auf das System oder das Logging des Benutzerverhaltens nicht durch ein Verschlüsselungsframework übernommen werden. Jedoch muss eine Verschlüsselung die Verhinderung eines unautorisierten Zugriffs unterstützen und bei Angriffen oder Datendiebstahl ausreichend stark sein, damit Dritte keinen Zugriff auf den Klartext erlangen. Hierbei grenzt sich die vorliegende Arbeit durch die zusätzliche Identifikation von Regularien und Best Practices ab. Außerdem erfolgt durch die Konsolidierung der Regularien und Best Practices die Ableitung konkreter Anforderungen an ein Verschlüsselungsframework für das Cloud Computing im Gesundheitssektor.

2.2 Best Practices

Auf Basis der Ergebnisse des systematischen Literaturreviews ist eine Konzeptmatrix der Best Practices erstellt worden.[2] Die am häufigsten genannte Best Practice ist die Unterstützung eines adäquaten (i) *Schlüsselmanagements* seitens des Verschlüsselungsframeworks innerhalb der Kommunikation zwischen Sender und Empfänger [AB09, BK14, OS12, Sl14, ST13, ZL10, Ca11, CH11, Ch12, DNP12, DP11, Ja11, Li10, LSW10]. Hierbei wird eine sichere und sinnvolle Methode zum Austausch der Schlüssel zwischen Sender und Empfänger empfohlen. Weiterhin wird eine sichere Verwaltung und Speicherung der Schlüssel als sinnvoll erachtet, die bspw. durch eine Verschlüsselung der Schlüssel erfüllt werden kann [AB09, KL10, Li10]. Das (ii) *hybride Verschlüsselungsverfahren* wurde am zweithäufigsten genannt [AB09, CH11, DNP12, Ja11, Li10, LSW10, LYZ13, Mi13, ST13, We14], wobei diese Best Practice mit dem *Schlüsselmanagement* korrespondiert, da hierbei das asymmetrische Verschlüsselungsverfahren genutzt werden soll, um einen symmetrischen Sitzungsschlüssel zwischen Sender und Empfänger auszutauschen. Des Weiteren empfehlen sieben der analysierten Artikel die Verwendung von verbreiteten und gut getesteten (iii) *Verschlüsselungsstandards* für Verschlüsselungsoperationen im Bereich Cloud Computing im Gesundheitssektor [AB09, BK14, Ca11, DNP12, KL10, Sl14, ST13]. Explizit werden der *Advanced Encryption Standard* (AES) mit einem 256 Bit Schlüssel als symmetrischer Verschlüsselungsalgorithmus [AB09, DNP12, KL10, ST13] sowie der nach den Entwicklern Rivest, Shamir und Adleman benannte asymmetrische Verschlüsselungsalgorithmus *RSA* [AB09, Ch12, DP11, ZL10] vorgeschlagen. Die weiteren identifizierten Best Practices umfassen die (iv) *separate Verschlüsselung* (Einträgen, Dateien und Datensätze nicht in Containern zusammengefasst verschlüsseln,

[2] Dem interessierten Leser wird die entsprechende Konzeptmatrix auf Anfrage gerne zur Verfügung gestellt.

sondern lediglich logisch zusammengehörige Daten – bspw. eine Bestellung bestehend aus Artikel, Preis usw. – zusammen zu verschlüsseln) [AB09, BK14, Ch12, LSW10, LYZ13, Pr06], die (v) *Schlüsselerzeugung und Speicherung beim Client* durchzuführen (Geheimhaltung der Schlüssel nach dem *Need-to-know-Principle*[3] [JP00]; es ist nicht nötig, dass geheime Schlüssel in Kontakt mit der Cloud Computing Umgebung kommen) [AB09, BK14, DP11, Ja11, KL10, KS11], das (vi) *homomorphe Verschlüsselungsverfahren* (ein Verfahren, das speziell für das Cloud Computing entwickelt wurde; dieses macht es möglich, verschlüsselte Daten innerhalb einer Cloud Computing Umgebung zu verarbeiten ohne dabei Zugriff auf den Klartext preiszugeben) [Bl11, Me13, We14], sowie die Verwendung eines (vii) *Tagging Systems* (die verschlüsselten Daten mit Bezeichnern versehen, wodurch diese mit geringem Aufwand identifiziert und verarbeitet werden können) zum leichten Auffinden von verschlüsselten Einträgen in der Cloud [AB09, KL10].

2.3 Regularien

Die Identifikation der Regularien erfolgte durch die im systematischen Literaturreview ermittelte Literatur sowie durch weitere Literatur, die mittels einer offenen Recherche nach Regularien im Cloud Computing via Google Scholar identifiziert worden ist. Da der Gesetzgeber eine Differenzierung vorsieht (vgl. §3 Abs. 9 BDSG) erfolgt die Unterteilung in *patientenbezogene* und *personenbezogene Daten*. Die einzelnen Regularien stammen aus der (Muster-)Berufsordnung für die in Deutschland tätigen Ärztinnen und Ärzte (MBO-Ä), dem Bundesdatenschutzgesetz (BDSG), der Strafprozessordnung (StPO), den Technischen Richtlinien des Bundesamts für Sicherheit in der Informationstechnik (BSI-TR), dem Sozialgesetzbuch (SGB) sowie dem Bürgerlichen Gesetzbuch (BGB).[4]

Hinsichtlich *patientenbezogener Daten* besagt § 9 Abs. 1 MBO-Ä [BK14, DEG11, Du11], dass Ärzte einer Schweigepflicht über sämtliche Informationen ihre Patienten betreffend unterliegen. Dieser Paragraph korrespondiert mit § 203 StGB, der das Strafmaß für die Verletzung von Privatgeheimnissen festlegt, und §§ 630a ff. BGB, wonach der Behandlungsvertrag das Patientengeheimnis schützt [BK14]. Weiterhin verpflichtet § 10 Abs. 2 MBO-Ä den Behandelnden auf Verlangen des Patienten, diesem Einsicht in seine Patientendaten zu gewähren [BK14, Du11, Me13, We10]. Zusätzlich korrespondiert dieses Gesetz mit § 630g BGB [BK14], der die Einsichtnahme in die Patientenakte beschreibt und § 34 BDSG, der dem Betroffenen das Recht auf Auskunftsanspruch über gespeicherte personenbezogene Daten zuspricht. Eine grundsätzliche Aufbewahrungsfrist von patientenbezogenen Daten von zehn Jahren ergibt sich aus § 10 Abs. 3 MBO-Ä [BK14] und § 630f Abs. 3 BGB, sofern keine verlängerten Aufbewahrungsfristen durch andere gesetzliche Vorschriften bestehen [BK14]. Außerdem ist laut § 10 Abs. 5 MBO-Ä eine elektronische externe Speicherung

[3] Das *Need-to-know-Principle* besagt, dass ein Subjekt nur Zugriff auf Informationen haben sollte, wenn diese zur Wahrnehmung der Aufgaben des Subjekts notwendig sind [SS94].
[4] Dem interessierten Leser wird die entsprechende Konzeptmatrix auf Anfrage gerne zur Verfügung gestellt.

von Patientendaten grundsätzlich möglich, jedoch sind hierbei durch besondere Sicherungs- und Schutzmaßnahmen die Veränderung, Vernichtung und unrechtmäßige Verwendung zu verhindern [BK14]. Weiterhin kann hier der § 203 StGB angeführt werden, wodurch der Arzt verpflichtet ist den Zugriff des Dienstleisters auf patientenbezogene Daten zu verhindern. In § 3 Abs. 9 BDSG werden Gesundheitsdaten als eine besondere Art personenbezogener Daten klassifiziert, weshalb es hierfür spezifische Gesetzgebungen gibt [BK14, Du11, We10]. § 97 Abs. 2 StPO schreibt indirekt die getrennte Speicherung von medizinischen Daten und anderen Datenformen vor, damit der Beschlagnahmeschutz gewährleistet ist [BK14]. Als Kryptografische Vorgaben für Projekte der Bundesregierung hat das Bundesamt für Sicherheit in der Informationstechnik (BSI) die Richtlinie BSI-TR-03116 [Bs14] herausgegeben. Der erste Teil BSI-TR-03116-1 dieser Richtlinie legt die im Gesundheitswesen verbindlichen Sicherheitsanforderungen und -vorgaben für den Einsatz kryptographischer Verfahren für die elektronische Gesundheitskarte, den Heilberufsausweis und die technischen Komponenten der Telematikinfrastruktur fest [BK14]. Dabei macht das BSI konkrete Empfehlungen zur Verwendung von Verschlüsselungsalgorithmen, zum Schlüsselvereinbarungsverfahren, zur Datenauthentisierung und Schlüsselerzeugung. Weiterhin werden Empfehlungen bezüglich Instanzauthentisierung, elektronischer Signatur, Verschlüsselung von Dokumenten, Kommunikation und Schlüsselmanagement ausgesprochen. Zudem verlangen das Sozialgesetzbuch (§ 67b SGB X [BK14, DEG11, Du11]) und das Bundesdatenschutzgesetz (§ 4a BDSG) in bestimmten Fällen für die Übermittlung von patientenbezogenen Daten eine Einwilligung des Betroffenen in schriftlicher Form. Durch § 42a BDSG wird eine nicht-öffentliche oder öffentliche Stelle, die besondere Arten personenbezogener Daten (z. B. Patientendaten) speichert, bei unrechtmäßiger Übermittlung an Dritte oder wenn Dritte unrechtmäßig Zugriff auf diese Daten erlangen, verpflichtet, diesen Vorfall unverzüglich den Betroffenen und der Aufsichtsbehörde zu melden [Du11, We10]. Der § 630f BGB macht das Führen einer elektronischen Patientenakte möglich und legt Anforderungen (bspw. Aufzeichnung von Änderungen) und die spezifischen Daten, die gespeichert werden sollen, fest.

Zu den *personenbezogenen Daten* betreffenden Regularien zählt § 11 BDSG, welcher die vertraglichen Rahmenbedingungen für die Erhebung, Verarbeitung und Nutzung personenbezogener Daten einer externen Stelle festlegt [BK14, BS12, DEG11, DRS10, Du11, Me13, MS11, Se13, We10]. Hierbei gilt vor allem zu beachten, dass der Auftraggeber für die Einhaltung der Vorschriften des BDSG verantwortlich ist. Weiterhin legen §§ 4b, c BDSG Vorschriften für die Übermittlung personenbezogener Daten in das Ausland fest [BK14, BS12, DRS10, MS11, We10]. Diese ermöglichen die Übertragung personenbezogener Daten in Mitgliedsstaaten der europäischen Union, Vertragsstaaten des Abkommens über den Europäischen Wirtschaftsraum sowie Organe und Einrichtungen der Europäischen Gemeinschaften. Bei nicht Gewährleistung eines angemessenen Schutzniveaus ist die Übermittlung untersagt, es sei denn eine der in § 4c Abs. 1 BDSG genannten Ausnahmen trifft zu. Außerdem wird durch § 4f BDSG die Bestellung eines Beauftragten für Datenschutz für öffentliche und nicht-öffentliche Stellen, die personenbezogene Daten automatisiert verarbeiten oder bei dessen Erhebung, Verarbeitung und Nutzung mindestens 20 Personen beschäftigt sind,

festgelegt [BK14]. Der § 28 BDSG reguliert die Datenerhebung und -speicherung für eigene Geschäftszwecke [BS12, DEG11, Du11, We10]. Das Erheben, Speichern, Verändern oder Übermitteln personenbezogener Daten zur Erfüllung der Geschäftszwecke ist zulässig. Die Übermittlung oder Nutzung für einen anderen Zweck ist unter den Voraussetzungen aus § 28 Abs. 2 BDSG zulässig. Schließlich macht § 35 BDSG Vorschriften für das Berichtigen, Löschen und Sperren von Daten [BK14]. Personenbezogene Daten sind zu berichtigen, wenn sie unrichtig sind und können u. U. jederzeit gelöscht werden, außer in den Fällen des § 35 Abs. 3.

3 Anforderungen

Aus den Best Practices und Regularien lassen sich argumentativ-deduktiv explizit und implizit Anforderungen ableiten [WH07]. Es werden Anforderungen, die im Allgemeinen die Datenhaltung, -verarbeitung und -übertragung betreffen, sowie Anforderungen, die unmittelbar für ein Verschlüsselungsframework für das Cloud Computing im Gesundheitssektor anwendbar sind, unterschieden.

- Allgemeine Anforderungen an Datenhaltung, -verarbeitung und -übertragung
  - *Verschlüsselte Datenübertragung*: Verschlüsselung der Daten beim Versand; Authentifizierung von Sender und Empfänger; Sicherstellung der Integrität der Daten durch Signaturverfahren; geeignetes Verfahren zum Schlüsselaustausch
  - *Verfügbarkeit*: stetiger Zugriff; zu jeder Zeit bereitstehende Möglichkeit der Datenoperationen Berichtigen, Sperren und Löschen
  - *Aufbewahrungsfrist*: Einhaltung gesetzlicher Aufbewahrungsfristen von Gesundheitsdaten (grundsätzlich zehn Jahre; je nach Art der Daten durch andere Gesetze auch länger)
  - *Verschlüsselte Datenspeicherung*: Wahrung des Patientengeheimnisses und Wahrung der ärztlichen Schweigepflicht durch Verhinderung des Zugriffs Dritter durch eine ausreichend starte Verschlüsselung; Verschlüsselungsoperationen nur lokal durchführen
  - *Datentrennung*: Trennung von medizinischen Daten und anderen Datenformen
- Anforderungen an ein Verschlüsselungsframework für das Cloud Computing im Gesundheitssektor
  - *Schlüsselmanagement*: zur Verschlüsselung verwendete Schlüssel müssen so gespeichert werden, dass ein Zugriff Dritter verhindert wird und diese den verschlüsselten Daten zuzuordnen sind
  - *Verschlüsselungsstandards*: Verwendung gängiger Verschlüsselungs-algorithmen (vgl. BSI TR-03116-1 [Bs14])

- *Separate Verschlüsselung*: keine Containerdateien sondern lediglich logisch zusammengehörende Daten zusammengefasst verschlüsseln
- *Instanzauthentisierung*: Authentisierung des Empfängers gegenüber dem Sender vor der Übermittlung von Daten
- *Datenauthentisierung*: Sicherstellung der Integrität der übermittelten Daten
- *Schlüsselerzeugung*: Sicherstellung, dass Schlüssel ausschließlich clientseitig erzeugt werden; mind. 100 Bit Entropie[5] für die Generierung von Schlüsseln

Damit einerseits weitere Anforderungen für ein Verschlüsselungsframework für das Cloud Computing im Gesundheitssektor identifiziert und in den Abschnitten zuvor eruierte Erkenntnisse verifiziert werden können, wurde ein leitfadengestütztes Experteninterview durchgeführt. Gesprächspartner war eine Expertin, welche in der Stabsstelle Abrechnung eines Dienstleisters im Gesundheitswesen tätig ist, wo sie für die Abrechnung und Abrechnungsschwierigkeiten (bspw. Retaxierung und Regressvermeidung) sowie für datenschutzrechtliche Angelegenheiten (bspw. Weitergabe von Patientendaten an Krankenkassen) zuständig ist. Vorher arbeitete die Expertin als Sachbearbeiterin in der Bundesopiumstelle, wo Sie u. A. die Integration des Betäubungsmittelrezepts in die Gesundheitskarte auf datenschutzrechtliche Problematiken untersuchte. Zusammenfassend ist zu sagen, dass durch das Experteninterview keine neuen Anforderungen für ein Verschlüsselungsframework für das Cloud Computing im Gesundheitssektor identifiziert werden konnten. Da sich jedoch die Aussagen der Expertin mit denen durch die Literaturrecherche identifizierten Best Practices und Regularien decken, konnten diese verifiziert werden.

4 Implementierung

Wesentliches Einsatzgebiet des Frameworks soll die Verschlüsselung jeglicher in Arztpraxen anfallender Daten sein. Hierbei geht es u. a. um Daten der Personalverwaltung, Warenwirtschaft sowie insbesondere um patientenbezogene Bestellungen von Medikamenten. Ziel des Verschlüsselungsframeworks ist es dabei eine hohe Sicherheit zu bieten, generisch einsetzbar zu sein sowie eine gewisse Plattformunabhängigkeit bei gleicher Codebasis zu bieten. Letzteres ergibt sich aus der Tatsache, dass Arztpraxen eine heterogene Betriebssystem- als auch Gerätelandschaft darstellen. Damit eine möglichst große Spanne von Betriebssystemen und Geräten abgedeckt werden kann, wird das Verschlüsselungsframework in der Programmiersprache C# als Portable Class Library (PCL) in der Entwicklungsumgebung Visual Studio 2013 entwickelt. Bei der Entwicklung des Verschlüsselungsframeworks sind die konsolidierten Anforderungen entsprechend berücksichtigt worden. Weiterhin sind diverse Verschlüsselungsalgorithmen hinsichtlich der Verwendung im

[5] *Entropie* beschreibt in der Kryptologie den Informationsgehalt von Daten bei der Erzeugung von Zufallszahlen.

Gesundheitssektor anhand eines Kriterienkatalogs (*Empfehlung des BSI Verschlüsselungsstandard*, *256 Bit Schlüssellänge*, *keine schwachen Schlüssel*, *keine bekannten Angriffe*, *Sicherheits-Performance-Verhältnis*, *freie Verfügbarkeit*), welcher sich z. T. wiederum aus den Anforderungen ergibt, bewertet worden.[6] Dem Ergebnis entsprechend wurde zur symmetrischen Verschlüsselung AES und zur asymmetrischen Verschlüsselung RSA verwendet. Dies ist auch der Grund, warum das Framework dem Anwender keine Auswahl zwischen verschiedenen Algorithmen bietet. Hiermit folgen wir u. a. der Empfehlung der Gesellschaft für Telematikanwendungen der Gesundheitskarte mbH [Ge08]. Im Folgenden soll verkürzt auf die Implementierung von Schlüsselgenerierung, -bereitstellung, -austausch und -speicherung eingegangen werden.

Die Klasse `KeyFactory`, die der Schlüsselgenerierung dient, wurde als statische Klasse implementiert. Die Methode `DeriveAESKeyFromString()` leitet einen für den AES kompatiblen Schlüssel von der übergebenen Zeichenkette mit Hilfe eines übergebenen Salts ab. Weiterhin verfügt diese Methode über zwei weitere optionale Parameter, die die Länge des abgeleiteten Schlüssels in Bytes sowie die Anzahl der Iterationen beim Ableiten des Schlüssels angeben. Dabei repräsentieren die Iterationen die Anzahl der Durchläufe, die beim sogenannten Key Stretching [YY05] durchgeführt werden; eine hohe Anzahl an Iterationen erschwert dabei Brute Force Angriffe [Ap12]. Des Weiteren erfolgt durch die Methode `GenerateRSAKeyPair()` die Generierung eines asymmetrischen Schlüsselpaares, das mit dem RSA Algorithmus kompatibel ist. Hierbei muss als Parameter die Länge des Schlüsselpaares übergeben werden. Als Rückgabewert ergibt sich eine Instanz der Schlüsselklasse `CryptographicKey`, die den generierten privaten und öffentlichen Schlüssel vorhält.

Für das Vorhalten und die Verwaltung einzelner Schlüssel sind die Klassen `AESKeyManager` und `RSAKeyManager` zuständig. Die Klasse `AESKeyManager` dient dazu der Klasse `AESCryption` zur symmetrischen Verschlüsselung auf Verlangen den Schlüssel zum Ver- bzw. Entschlüsseln zu liefern. Für diesen Zweck wurde ein öffentliches Property `Key` implementiert, das den symmetrischen Schlüssel zurückgibt. In der Methode zum Import des Schlüssels `ImportKey()` wird als Parameter der symmetrische Schlüssel als Byte Array übergeben, woraufhin eine Instanz der Schlüsselklasse `CryptographicKey` initialisiert wird. Als weitere Klasse zur Schlüsselbereitstellung wurde die Klasse `RSAKeyManager` implementiert. Diese stellt der Klasse `RSACryption` das Schlüsselpaar zur asymmetrischen Verschlüsselung zur Verfügung; hierfür wird der öffentliche separat vom privaten Schlüssel vorgehalten.

Der Kern der Implementierung ist der Austausch eines Schlüssels zwischen Client und Server, der zum Verschlüsseln sämtlicher Nachrichten einer Sitzung angewendet wird. Um den Ablauf des Handshakeprozesses (siehe *Abb. 1*) in einzelne Schritte aufzuteilen, wurden bei der Implementierung Nachrichtencodes eingeführt, die innerhalb der Klasse `SessionKeyExchange` genutzt werden, um entsprechende Operationen auszuführen.

[6] Die Bewertung ist nicht Teil dieser Publikation. Dem interessierten Leser werden die entsprechenden Informationen auf Anfrage gerne zur Verfügung gestellt.

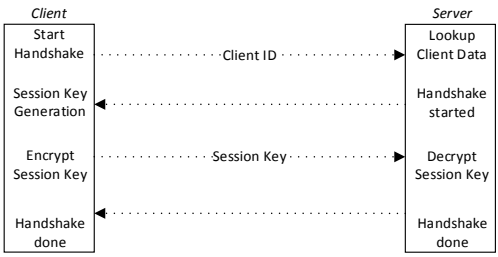

Abb. 1: Prozess des Schlüsselaustauschs

Die Klasse `EncryptedKeyStore` fungiert als Speicher für symmetrische und asymmetrische Schlüssel. Zudem hält die Klasse den Pfad zu einer XML Datei vor, in der sämtliche Schlüssel chiffriert gespeichert werden. Die Speicherung der Schlüssel zur Laufzeit erfolgt mittels zweier Attribute vom Typ `Dictionary<String, AESKeyManager>` und `Dictionary<String, RSAKeyManager>`. Dies ermöglicht das komfortable Hinzufügen, Lesen und Löschen von Schlüsseln. Der Konstruktor erwartet die Parameter `AESCryption`, mit der die XML Datei, in der die Schlüssel gespeichert werden, verschlüsselt wird, sowie eine Zeichenkette mit dem Pfad zur XML Datei. Der Schlüssel für die Instanz der `AESCryption` sollte von dem Benutzerpasswort abgeleitet werden. Nach der Initialisierung der Klasse `EncryptedKeyStore` muss zunächst die Methode `Load()` aufgerufen oder neue Schlüssel hinzugefügt werden. Weiterhin ist es möglich durch den Aufruf der Methode `Safe()` sämtliche Schlüssel persistent und verschlüsselt in die zuvor übergebene XML Datei zu schreiben. Der Aufbau der XML Datei kann in *Abb. 2* eingesehen werden, wobei übersichtshalber die Schlüssel gekürzt wurden.

```
<Keys>
   <AES>
     <key id="backupkey">/VILoJt034ypkzhQEbsKJ5kikey>
   </AES>
   <RSA>
     <keypair id="rsa_client_1">
       <public>Z02bQP7k+uLv4bFHWBtCJDmb+</public>
       <private>C47p3Maltbcl1uK8SoBseI4UY</private>
     </keypair>
   </RSA>
</Keys>
```

Abb. 2: Struktur der XML Datei zum Speichern von Schlüsseln (gekürzt)

Um der Klasse Schlüssel hinzuzufügen kann die Methode `Add`, von der es zwei Überladungen gibt, die einerseits einen Schlüssel in der Form eines `AESKeyManager` und andererseits in der Form eines `RSAKeyManager` annimmt, verwendet werden. Weiterhin ist das Lesen von Schlüsseln durch die Methoden `GetRSAKeyManager()` und `GetAESKeyManager()` möglich, die jeweils den Identifikator eines Schlüssels als Zeichenkette übergeben bekommen müssen. Schließlich kann durch den Aufruf der Methode `Remove()` mit einer entsprechenden Zeichenkette, die einen Schlüssel identifiziert, ein Schlüssel gelöscht werden, sofern dieser in der Datenstruktur existiert.

5 Konzeptuelle und technische Evaluation

Die konzeptuelle Evaluation des Verschlüsselungsframeworks ist mit Hilfe von zwei leitfadengestützten Experteninterviews durchgeführt worden [Wa06]. Interviewpartner A arbeitet seit drei Jahren als Softwareentwickler mit dem Fokus der Entwicklung von Webseiten mit HTML, CSS und JavaScript sowie der Entwicklung von Backends mit C#, womit auch das Verschlüsselungsframework programmiert worden ist. Interviewpartner B arbeitet seit vier Jahren als Softwareentwickler mit dem Fokus zum einen auf Web Anwendungen, Webseiten und Windows Anwendungen, zum anderen auf mobilen Anwendungen für Windows Phone, iOS und Android. Letztere werden in der Programmiersprache C# entwickelt. Da in diesem Kontext vor allem auch Portable Class Libraries eine Rolle spielen, eignet sich der Experte zur Evaluation des Frameworks durch seine Erfahrung im Umgang mit diesen.[7]

Das Verschlüsselungsframework ist mit den beiden Interviewpartnern A und B hinsichtlich der *Plattformabhängigkeit*, dem zugrundeliegenden *Klassendiagramm* sowie hinsichtlich der *Umsetzung der Implementierung* (u. a. verwendete Bibliotheken, Schlüsselgenerierung, Ver- und Entschlüsselung, Schlüsselaustausch) evaluiert worden. Aufgrund der Implementierung des Frameworks als PLC ist eine universelle Einsetzbarkeit in verschiedenen Softwareprojekten möglich [A, B]. Weiterhin erlaubt dies den Einsatz des Frameworks auf verschiedenen Betriebssystemen und Endgeräten wie Smartphones mit Android, Windows Phone oder iOS sowie auf Desktop-PCs mit MacOS oder Windows. Außerdem ermöglicht die Ver- und Entschlüsselung verschiedener Datentypen und Dateien einen vielseitigen Einsatz für verschiedene kryptografische Aufgaben. Auch ist der Anwender hier nicht eingeschränkt, da ebenfalls sehr generelle Datentypen wie Byte Arrays ver- und entschlüsselt werden können, in die praktisch jeder Datentyp konvertiert werden kann [A]. Des Weiteren wird der Einsatz des Verschlüsselungsframeworks als Middleware innerhalb des Anwendungsfalls der Bestellung von Arzneimitteln als sinnvoll erachtet. Dies wird damit begründet, dass der Anwender des Frameworks nicht selbst direkt die Funktionaltäten zur Ver- und Entschlüsselung aufrufen muss, sondern dieser wie gewohnt mit den Operationen der Web API arbeiten kann [A; B]. Einige Verbesserungspotentiale sind bereits im Prototypen umgesetzt worden. Weitere Anpassungen werden in der nächsten Iteration der Entwicklung des finalen Verschlüsselungsframeworks umgesetzt.

Die technische Evaluation erfolgte mittels der Nutzung des Verschlüsselungsframeworks bei einem Arzneimittellieferanten in zwei exemplarischen aber realen Anwendungsfällen. Der Arzneimittellieferant gibt seinen Kunden (in diesem Fall Arztpraxen) ein speziell entwickeltes Praxissystem an die Hand. Dieses Praxissystem übernimmt u. a. die Warenwirtschaft und Bedarfsplanung, sodass hiermit elektronische Bestellungen möglich sind. Später soll diese Software noch um weitere Module erweitert werden, sodass sämtliche Aufgabenbereiche einer Arztpraxis abgedeckt werden. Weiterhin soll die Software später serverseitig an eine Cloud angebunden werden,

[7] Im Folgenden wird *Interviewpartner A* bzw. *Experte A* mit *A* abgekürzt; *Interviewpartner B* analog.

weshalb diese zukünftigen Pläne bei den folgenden Ausführungen bereits berücksichtigt werden. Arzneimittelbestellungen sind oftmals patientenbezogen und enthalten damit personenbezogene Daten. Deshalb müssen diese speziell gesichert bzw. Ende-zu-Ende verschlüsselt werden.

Szenario 1: Verschlüsselte Backups

Im Folgenden soll der Einsatz des entwickelten Verschlüsselungsframeworks beim Anlegen und Wiederherstellen von verschlüsselten Backups beschrieben werden.[8] Zu Beginn des Backups muss der entsprechende Schlüssel geladen werden, welcher bei der Installation der Software angelegt wurde. Zum Verschlüsseln der Backups wird der symmetrische Verschlüsselungsalgorithmus AES verwendet, da dieser bei einer Schlüssellänge von 256 Bit als sehr sicher gilt und für die Ver- bzw. Entschlüsselung großer Datenmengen geeignet ist. Für die Verschlüsselung von Dateien, die auf dem System gespeichert werden, nehmen die entsprechenden Methoden jeweils als Parameter den Pfad zur Datei, die verschlüsselt werden soll, und den Zielpfad der verschlüsselten Datei. Hierbei gilt zu beachten, dass die vorhandene Ordnerstruktur beibehalten wird und zu entscheiden, ob Dateinamen verschlüsselt werden sollen oder im Klartext gesichert werden. Das Wiederherstellen eines Backups erfolgt entsprechend auf dem entgegengesetzten Weg mittels der entsprechenden Methode. Sollen Softwarekonfigurationen und Benutzereinstellungen gesichert werden, handelt es sich oftmals um Datenbankdatensätze die per Datenbankabfrage gelesen werden. Sofern ein Backup dieser Konfigurationsdaten nicht mittels Datenbankauszug in eine Datei erfolgt, gestaltet sich die Verwendung der vorherigen Methoden als eher unhandlich, da hierfür die zusätzliche Zwischenspeicherung der aus der Datenbank extrahierten Daten in einer Datei nötig ist. Daher empfiehlt sich für diesen Zweck die Verwendung von Methoden, die einen Datenstrom, der die entsprechenden Konfigurationsdaten aus der Datenbank enthält, ver- bzw. entschlüsseln. Die Speicherung der verschlüsselten Konfigurationsdaten kann schließlich in einer Textdatei oder ähnlichem auf dem externen Medium erfolgen.

Szenario 2: Arzneimittelbestellungen

In diesem Abschnitt wird die Anwendung des Verschlüsselungsframeworks bei der Bestellung von Arzneimitteln erläutert. Hierfür soll ein über HTTP erreichbarer Webserver verwendet werden, der Bestellungen entgegennimmt. Das Aufgeben von Bestellungen soll mit einem einfachen HTTP-Client möglich sein. Der detaillierte technische Aufbau ist in *Abb. 3* ersichtlich.

[8] Da zunächst die Daten des Praxissystems für Arztpraxen auf externen Festplatten gesichert werden und erst zu einem späteren Zeitpunkt in der Software das Speichern von Backups in der Cloud ermöglicht werden soll, wird im Folgenden die Sicherung auf externen Medien betrachtet. Jedoch sind die Unterschiede zwischen dem Speichern in der Cloud oder auf externen Medien vom Standpunkt der Verschlüsselung betrachtet marginal, weshalb die folgenden Beschreibungen ebenfalls auf das Sichern von Backups in der Cloud anwendbar sind.

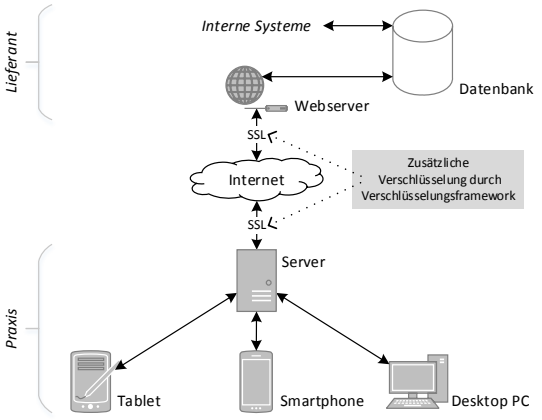

Abb. 3: Technischer Aufbau der Kommunikation

Zur Illustration eines solchen Systems wurde ein ASP .NET MVC Web API [Mi14] Projekt erstellt, das die serverseitige Annahme von Bestellungen übernimmt. Dies ermöglicht das Versenden von Objekten als JavaScript Object Notation (JSON), welche vor dem Versenden mittels HTTP serialisiert werden, sodass diese nach dem Empfangen von der ASP .NET Web API deserialisiert werden und dem Entwickler ohne Zutun als Instanzen zur Verfügung stehen. Sendet also ein Client bspw. eine Instanz der Klasse Bestellung an den Server, so muss der Entwickler keine Zeichenkette parsen sondern erhält eine Instanz der Klasse mit denselben Attributen, wie sie vom Client versendet wurde. Um das Verschlüsselungsframework möglichst anwenderfreundlich einzubetten, wird die Verschlüsselung in der Web API als Middleware implementiert. Hierfür bieten sich sogenannte Delegating Handler an, die in den Kommunikationskanal eingefügt werden und das Lesen und Manipulieren von HTTP-Anfragen sowie Antworten ermöglichen. Innerhalb der Methode findet zunächst die Identifikation des Clients anhand der Session ID statt, woraufhin geprüft wird, ob mit diesem Client bereits der Handshake abgeschlossen ist. Sofern dieses der Fall ist, erfolgt die Entschlüsselung der Anfrage mittels des Sitzungsschlüssels und die Weitergabe zur Verarbeitung. Nachdem die Anfrage verarbeitet und eine Antwort zurückgegeben wurde, wird diese mit dem Sitzungsschlüssel verschlüsselt sowie zum Versand an den Client zurückgegeben. Ist der Handshake bzw. Austausch des Sitzungsschlüssels mit diesem Client noch nicht abgeschlossen, wird, sofern es sich bei der HTTP-Anfrage um eine Handshake Nachricht handelt, der nächste Schritt im Handshakeprozess ausgeführt und die entsprechende HTTP-Antwort zurückgegeben. Dies geschieht solange bis der Handshake erfolgreich war oder fehlgeschlagen ist. Auf Seiten des Clients wird eine Verbindung zu der ASP .NET Web API hergestellt. Weiterhin erfolgt die Verschlüsselung der HTTP-Anfrage, die gesendet werden soll, und die Entschlüsselung der HTTP-Antwort mittels des Sitzungsschlüssels. Da der ASP .NET Web API Server eine Initiierung des Handshakes seitens des Clients erwartet, wurde eine entsprechende Klasse implementiert, die den Handshake startet und den gesamten Handshake durchführt, bis dieser erfolgreich war oder fehlschlägt.

6 Zusammenfassung und Ausblick

Auf Basis eines Literaturreviews und den daraus resultierenden Best Practices und Regularien sind Anforderungen an ein Verschlüsselungsframework im Gesundheitssektor abgeleitet worden. Unter Anwendung dieser Anforderungen ist ein solches Verschlüsselungsframework prototypisch in C# als PLC implementiert worden, welches auf den meisten gängigen Computer- und Handheldplattformen verwendbar ist. Mit Hilfe von Experteninterviews ist das Verschlüsselungsframework konzeptionell sowie in zwei Anwendungsfällen technisch evaluiert worden.

Mit Blick auf Limitationen sei angemerkt, dass der systematische Literaturreview primär auf Beiträge aus der Wirtschaftsinformatik abgezielt hat, korrespondierende Beiträge aus dem Bereich der Medizininformatik/eHealth sind nicht aufgegriffen worden. Weitergehen sei erwähnt, dass jede Verschlüsselung mit entsprechender Rechenleistung zu entziffern ist. Allerdings bietet die Wahl von hohen Schlüssellängen eine hohe Sicherheit. Hinsichtlich der Implementierung ist das Fehlen des Ablaufs eines Sitzungsschlüssels nach einer bestimmten Zeit als kritisch zu betrachten. Hier sollte der Schlüssel seitens des Webservers mit einem Zeitstempel versehen werden, sodass nach einer bestimmten Zeit ein neuer Sitzungsschlüssel ausgetauscht wird. Zudem wurde u. a. die Anforderung der *Datenauthentisierung* an ein Verschlüsselungsframework gestellt, diese allerdings nicht explizit umgesetzt. Hierbei handelt es sich jedoch um kein schwerwiegendes Problem, da bei einer Manipulation der verschlüsselten Daten die Entschlüsselung schlichtweg fehlschlägt und die Datenübertragung abgebrochen wird. Des Weiteren empfiehlt es sich das gesamte Verschlüsselungsframework sowie den vollständig implementierten Anwendungsfall gängiger kryptografischer Angriffe zu unterziehen und zum anderen zu versuchen bspw. als Angreifer den Handshakeprozess zu manipulieren, damit ggf. Schwächen oder Fehler in der Implementierung gefunden und daraufhin behoben werden können.

Bezüglich zukünftigen Forschungsbereichen sollten Vorschläge für Standards bei der Übertragung und Speicherung von Patientendaten (in der Cloud) entwickelt werden, um so ein einheitliches Sicherheitsniveaus zu erreichen. Diese Arbeit hat dafür erste Schritte aufgezeigt, jedoch sind weitere Arbeiten hinsichtlich Übertragungsprotokolle sowie der Integration der Verschlüsselung in die Datenübertragung nötig. Ein weiterer Bereich ist das Teilen von Patientendaten mit Dritten (bspw. bei der Überweisung von Patienten).

Danksagung

Die Autoren danken sowohl der LM IT Services AG, den Gutachtern sowie den Experten, die für die Interviews zur Verfügung standen. Darüber hinaus gilt unser Dank explizit Frau Linster-Hoffmann für ihre Unterstützung. Diese Arbeit ist Teil des Projekts „Nachhaltiger Konsum von Informations- und Kommunikationstechnologie in der digitalen Gesellschaft - Dialog und Transformation durch offene Innovation“. Das

Projekt wird vom Ministerium für Wissenschaft und Kultur des Landes Niedersachsen und der VolkswagenStiftung aus Landesmitteln des Niedersächsischen Vorab gefördert (Projektnummer VWZN3037).

Literaturverzeichnis

[AB09] Archer, J.; Boehm, A.: Security guidance for critical areas of focus in cloud computing. Cloud Security Alliance S. 1–176, 2009.

[Ap12] Apostol, K.: Brute-force Attack. SaluPress, 2012.

[As15] Association for Information Systems, MIS Journal Rankings, http://aisnet.org/general/custom.asp?page=JournalRankings, Stand: 18.01.2015.

[BK14] Bundesärztekammer; Kassenärztliche Bundesvereinigung: Empfehlungen zur ärztlichen Schweigepflicht , Datenschutz und Datenverarbeitung in der Arztpraxis. Deutsches Ärzteblatt 21, S. 963–972, 2014.

[Bl11] Blumenthal, M.: Is Security Lost in the Clouds? Communications and Strategies 2011.

[Br09] vom Brocke, J.; Simons, A.; Niehaves, B.; Riemer, K.; Plattfaut, R.; Cleven, A.: Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. 17th European Conference on Information Systems. Verona, Italy, 2009.

[BS12] Borges, G.; Schwenk, J.: Daten- und Identitätsschutz in Cloud Computing, E-Government und E-Commerce. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Bs14] BSI, BSI-TR-03116, https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03116/index_htm.html, Stand: 21.10.2014.

[Ca11] Carroll, M.: Secure cloud computing: Benefits, risks and controls. Information Security South Africa 2011.

[CH11] Chen, L.; Hoang, D.B.: Novel Data Protection Model in Healthcare Cloud. 2011 IEEE International Conference on High Performance Computing and Communications. Ieee, S. 550–555, 2011.

[Ch12] Chen, T.-S.; Liu, C.-H.; Chen, T.-L.; Chen, C.-S.; Bau, J.-G.; Lin, T.-C.: Secure Dynamic access control scheme of PHR in cloud computing. Journal of medical systems 36/6, S. 4005–20, Dec. 2012.

[DEG11] Duisberg, A.; Eckhardt, J.; Grudzien, W.: Rechtliche Anforderungen an Cloud Computing – Sichere Cloud Dienste. IT-Gipfel 2011 - Rechtliche Anforderungen an Cloud Computing S. 1–67, 2011.

[DNP12] Deng, M.; Nalin, M.; Petković, M.: Towards trustworthy health platform cloud. Secure Data Management S. 162–175, 2012.

[DP11] Deng, M.; Petkovic, M.: A Home Healthcare System in the Cloud--Addressing Security and Privacy Challenges. 2011 IEEE International Conference on Information Systems, Cloud Computing (CLOUD). S. 1–8, 2011.

[DRS10] Doelitzscher, F.; Reich, C.; Sulistio, A.: Designing Cloud Services Adhering to Government Privacy Laws. International Conference on Computer and Information Technology Cit, S. 930–935, Jun. 2010.

[Du11] Duisberg, A.: Gelöste und ungelöste Rechtsfragen im IT-Outsourcing und Cloud Computing. In (Picot, A., Götz, T., and Hertz, U., Hrsg.): Trust in IT. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 49–70, 2011.

[EFZ13] Ermakova, T.; Fabian, B.; Zarnekow, R.: Security and Privacy System Requirements for Adopting Cloud Computing in Healthcare Data Sharing Scenarios. Americas Conference on Information Systems. S. 1–9, 2013.

[EFZ14] Ermakova, T.; Fabian, B.; Zarnekow, R.: ACCEPTANCE OF HEALTH CLOUDS-A PRIVACY CALCULUS PERSPECTIVE. European Conference on Information Systems. S. 1–13, 2014.

[Ge08] gematik, Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur, https://www.gematik.de/cms/media/dokumente/release_0_5_3/release_0_5_3_datensc hutz/gematik_GA_Spezifikation_Kryptographischer_Algorithmen_V1_3_0.pdf, Stand: 03.03.2015.

[Ja11] Jansen, W. a: Cloud Hooks: Security and Privacy Issues in Cloud Computing. 2011 44th Hawaii International Conference on System Sciences. Ieee, S. 1–10, 2011.

[JP00] Janczewski, L.J.; Portougal, V.: "Need-to-know" principle and fuzzy security clearances modelling. Information Management & Computer Security 8/5, S. 210–217, 2000.

[KL10] Kamara, S.; Lauter, K.: Cryptographic Cloud Storage. Proceedings of the 1st Workshop on RealLife Cryptographic Protocols and Standardization S. 1–14, 2010.

[KS11] Kaletsch, A.; Sunyaev, A.: Privacy engineering: personal health records in cloud computing environments. International Conference on Information Systems. S. 1–11, 2011.

[Li10] Li, M.; Yu, S.; Ren, K.; Lou, W.: Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. Security and Privacy in Communication Networks S. 89–106, 2010.

[LSW10] Löhr, H.; Sadeghi, A.-R.; Winandy, M.: Securing the e-health cloud. Proceedings of the ACM International Conference on Health informatics - IHI '10. ACM Press, New York, New York, USA, 2010.

[LYZ13] Li, M.; Yu, S.; Zheng, Y.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Transactions on Parallel and Distributed Systems 24/1, S. 131 – 143, 2013.

[Me13] de Meer, H.; Diener, M.; Herkenhöner, R.; Kucera, M.; Niedermeier, M.; Reisser, A.; Schryen, G.; Vetter, M.; Waas, T.; Yasasin, E.: Sicherheitsherausforderungen in hochverteilten Systemen. PIK - Praxis der Informationsverarbeitung und Kommunikation 36/3, Jan. 2013.

[Mi13] Mitchell, S.; Ridley, S.; Tharenos, C.; Varshney, U.: Investigating Privacy and Security Challenges of mHealth Applications. Americas Conference on Information Systems. S. 1–9, 2013.

[Mi14] Microsoft Corporation, ASP.NET Web API, http://www.asp.net/web-api, Stand: 03.12.2014.

[MS11] Marnau, N.; Schlehahn, E.: Cloud computing und safe harbor. Datenschutz und Datensicherheit - DuD S. 311–316, 2011.

[OS12] Oetzel, M.; Spiekermann, S.: Privacy-by-design through systematic privacy impact assessment-a design science approach. European Conference on Information Systems. 2012.

[Pr06] Pratt, W.; Unruh, K.; Civan, A.; Skeels, M.: Personal health information management. Communications of the ACM 49/1, S. 51, Jan. 2006.

[Se13] Selzer, A.: Die Kontrollpflicht nach § 11 Abs. 2 Satz 4 BDSG im Zeitalter des Cloud Computing. Datenschutz und Datensicherheit - DuD 37/4, S. 215–219, 2013.

[Sl13] Slawik, M.: The Trusted Cloud Transfer Protocol. 2013 IEEE International Conference on Cloud Computing Technology and Science. Ieee, S. 203–208, 2013.

[Sl14] Slawik, M.; Ermakova, T.; Repschläger, J.; Küpper, A.: Securing Medical SaaS Solutions Using a Novel End-To-End Encryption Protocol. European Conference on Information Systems. S. 1–9, 2014.

[SS94] Sandhu, R.S.; Samarati, P.: Access control: principle and practice. IEEE Communications Magazine 32/9, S. 40–48, 1994.

[ST13] Stark, L.; Tierney, M.: Lockbox: mobility, privacy and values in cloud storage. Ethics and Information Technology 16/1, S. 1–13, Oct. 2013.

[Wa06] Walsham, G.: Doing Interpretive Research. European Journal of Information Systems 15/3, S. 320–330, 2006.

[We10] Weichert, T.: Cloud Computing und Datenschutz. Datenschutz und Datensicherheit - DuD 34/10, S. 679–687, Oct. 2010.

[We14] Wenge, O.; Lampe, U.; Müller, A.; Schaarschmidt, R.: Data Privacy in Cloud Computing–An Empirical Study in the Financial Industry. Americas Conference on Information Systems. S. 1–10, 2014.

[WH07] Wilde, T.; Hess, T.: Forschungsmethoden der Wirtschaftsinformatik: Eine empirische Untersuchung. Wirtschaftsinformatik 49/4, S. 280–287, 2007.

[WW02] Webster, J.; Watson, R.T.: Analyzing the past to prepare for the future: Writing a Literature Review. MIS Quarterly 26/2, S. 13–23, 2002.

[YY05] Yao, F.F.; Yin, Y.L.: Design and analysis of password-based key derivation functions. IEEE Transactions on Information Theory 51/9, S. 3292–3297, Sep. 2005.

[ZL10] Zhang, R.; Liu, L.: Security Models and Requirements for Healthcare Application Clouds. International Conference on Cloud Computing S. 268–275, Jul. 2010.