Gesellschaft für Informatik (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into the fields of
• Seminars
• Proceedings
• Dissertations
• Thematics
current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi-ev.de/service/publikationen/lni/

ISSN 1617-5468
ISBN 978-3-88579-268-0

Upcoming solutions for cross-organizational telematic infrastructures are enabling intensive research in innovative security approaches for processing of person related medical and health data. Within the perspeGKtive 2010 workshop security and privacy aspects of both the used information technology of today and in future has been discussed by professionals and innovative solution approaches were presented.

A. Brömme, T. Eymann, D. Hühnlein, H. Roßnagel, P. Schmücker (Hrsg.): perspeGKtive 2010

174

GI-Edition

Lecture Notes in Informatics

Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)

perspeGKtive 2010

Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

8. September 2010, Mannheim

Proceedings

Arslan Brömme, Torsten Eymann, Detlef Hühnlein,
Heiko Roßnagel, Paul Schmücker (Hrsg.)

perspeGKtive 2010

Workshop
„Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

8. September 2010
Mannheim

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-174

ISBN 978-3-88579-268-0
ISSN 1617-5468

Volume Editors
Arslan Brömme
GI SIG BIOSIG, Gesellschaft für Informatik e.V.
E-Mail: arslan.broemme@aviomatik.de
Torsten Eymann
Universität Bayreuth
E-Mail: Torsten.Eymann@uni-bayreuth.de
Detlef Hühnlein
ecsec GmbH
E-Mail: detlef.huehnlein@ecsec.de
Heiko Roßnagel
Fraunhofer IAO
E-Mail: Heiko.Rossnagel@iao.fraunhofer.de
Paul Schmücker
Hochschule Mannheim
E-Mail: p.schmuecker@hs-mannheim.de

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

Die im Gesundheitswesen bereits eingesetzte Informationstechnologie und in naher Zukunft absehbar zum Einsatz kommenden Anwendungslösungen für organisationsübergreifende Telematikinfrastrukturen geben Anlaß, um einen vertiefenden Einblick in sicherheitsrelevante Neuerungen im Umgang mit sensiblen personenbezogenen und personenbeziehbaren medizinischen Daten und Gesundheitsdaten zu geben.

Der diesjährige perseGKtive Workshop widmet sich in diesem Zusammenhang mit seinen Beiträgen den Themengebieten:

- Datenschutz und Sicherheit bei der Dokumentation und Archivierung
- Sicherheitskonzepte und Authentisierung
- Sicherheitsanalysen im Bereich der Gesundheitstelematik
- PerspeGKtiven für die Sicherheit im Gesundheitswesen

perspeGKtive 2010 wird gemeinsam von der Gesellschaft für Informatik e.V. (GI) und der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e.V. (GMDS) unter Beteiligung verschiedener interner Fach- und Arbeitsgruppen veranstaltet:

Angewandte Kryptologie (KRYPTO, GI)
Archivierung von Krankenunterlagen (AKU, GI/GMDS)
Biometrik und elektronische Signaturen (BIOSIG, GI)
Datenschutz in Gesundheitsinformationssystemen (DGI, GI/GMDS)
Datenschutzfördernde Technik (PET, GI)
Informationssysteme im Gesundheitswesen (KIS, GI/GMDS)
Sicherheit in Netzen (NETSEC, GI)
Mobile Informationstechnologie in der Medizin (MOCOMED, GI/GMDS)
Mobilität und Mobile Informationssysteme (MMI, GI)
Sicherheit in Netzen (NETSEC, GI)

Mannheim, 8. September 2010

Arslan Brömme
GI SIG BIOSIG, GI e.V.

Torsten Eymann
Universität Bayreuth

Detlef Hühnlein
ecsec GmbH

Heiko Roßnagel
Fraunhofer IAO

Paul Schmücker
Hochschule Mannheim

Vorsitzende

Arslan Brömme
GI SIG BIOSIG, GI e.V.

Torsten Eymann
Universität Bayreuth

Detlef Hühnlein
ecsec GmbH

Heiko Roßnagel
Fraunhofer IAO

Paul Schmücker
Hochschule Mannheim

Programmkomitee

Oliver Berndt, Arslan Brömme, Herbert Bunz, Christoph Busch, Jörg Caumanns, Carl Dujat, Volkmar Eder, Torsten Eymann, Hannes Federrath, Ulrich Flegel, Lothar Fritsch, Christoph F.-J. Goetz, Thomas Grechenig, Peter Haas, Marit Hansen, Georg Heidenreich, Arno Herrmann, Alexander Hörbst, Detlef Hühnlein, Luigi Lo Iacono, Stefan Katzenbeisser, Werner Keil, Ulrike Korte, Jan Marco Leimeister, Wolfgang Loos, Sven Marx, Pablo Mentzinis, Gilbert Mohr, Jens Naumann, Alexander Nouak, Dittmar Padeken, Sachar Paulus, Hartmut Pohl, Klaus Pommerening, Kai Rannenberg, Georgios Raptis, Asarnusch Rashid, Heiko Roßnagel, Marcus Sasse, Paul Schmücker, Jörg Schwenk, Christoph Seidel, Sebastian C. Semler, Stefan Skonetzki-Cheng, Günter Steyer, Barbara Tappeiner, Ulrich Waldmann, Anette Weisbecker, Alex Wiesmaier, Jan Zibuschka

Unterstützende Partner

Berufsverband Medizinischer Informatiker e.V.
Bundesamt für Sicherheit in der Informationstechnik
Bundesministerium für Gesundheit
Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.
Competence Center für die Elektronische Signatur im Gesundheitswesen e.V.
Deutsche Gesellschaft für Gesundheitstelematik e.V.
Deutsche Gesellschaft für Telemedizin e.V.
Telematikplattform für Medizinische Forschungsnetze e.V.
TeleTrusT Deutschland e.V.
Verband der Hersteller von IT-Lösungen im Gesundheitswesen e.V.
Verband Organisations- und Informationssysteme e.V.

perspeGKtive 2010 – Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

“**perspeGKtive 2010 - Workshop**”
8. September 2010

Der vielfältige Einsatz von Informationstechnologien im Gesundheitswesen mit seinen Erweiterungen zu organisationsübergreifenden Telematikinfrastrukturen und den damit einhergehenden Neuerungen im Unfeld der Anwendungslösungen wird zahlreiche Möglichkeiten bieten, um personenbezogene Daten und personenbeziehbare Daten auszutauschen und zu verarbeiten.

Aufgrund des sehr hohen Schutzbedarfs der im Gesundheitswesen verarbeiteten medizinischen Daten und Gesundheitsdaten sowie dem Grundrecht der informationellen Selbstbestimmung der Patienten kommt den Aspekten des Datenschutzes, der Beweis- und IT-Sicherheit eine herausragende Bedeutung zu.

Der gemeinsam von GI e.V., GMDS e.V. und zahlreichen Partnern durchgeführte Workshop perspeGKtive 2010 ermöglicht Ihnen im Kreise von Experten zu aktuellen und zukünftigen Datenschutz-, Informationssicherheits- und IT-Sicherheitsaspekten Erfahrungen und Einschätzungen auszutauschen und vertiefend zu diskutieren.

Wir wünschen Ihnen einen erfolgreichen Workshop.

Inhaltsverzeichnis

perspeGKtive 2010 – Wissenschaftliche Beiträge 9

Ahmad-Reza Sadeghi, Thomas Schneider
Verschlüsselt Rechnen: Sichere Verarbeitung verschlüsselter medizinischer Daten am Beispiel der Klassifikation von EKG-Daten 11

Ali Sunyaev, Stefan Knipl, Sebastian Dünnebeil, Jan Marco Leimeister, Helmut Krcmar
RetrospeGKtive der bekannten sicherheitstechnischen Problematiken bei der Einführung der Elektronischen Gesundheitskarte und der Telematikinfrastruktur in Deutschland 27

Raik Kuhlisch, Jörg Caumanns
Deklarative Sicherheit zur Spezifikation und Implementierung der elektronischen Fallakte 39

Fabian Schwab, Jörg Lübbert, Peter Sakal, Hartmut Pohl
Rapid in-Depth Analysis für Telematikanwendungen im Gesundheitswesen – Identifizierung nicht erkannter Sicherheitslücken mit Threat Modeling und Fuzzing 53

Hagen Kosock, Judith Balfanz, Antje Brandner, Carl Dujat, Christopher Duwenkamp, Reinhold Haux, Nils Hellrung, Paul Schmücker, Christoph Seidel
Elektronische Signaturen in Versorgungseinrichtungen des Gesundheitswesens – Maßnahmen und Einführungsunterstützung 69

Daniel Eske, Detlef Hühnlein, Sachar Paulus, Johannes Schmölz, Tobias Wich, Thomas Wieland
OpeneGK - Benutzerfreundliche und sichere Authentisierung für Mehrwertdienste im Gesundheitswesen 83

Basel Katt, Thomas Trojer, Ruth Breu, Thomas Schabetsberger, Florian Wozak
Meeting EHR Security Requirements: Authentication as a Security Service 103

perspeGKtive 2010 – Kurzbeitrag zum Workshop 111

Raffael Rittmeier, Karsten Sohr
Grundzüge eines Sicherheitskonzeptes für Arztpraxen mit Hilfe von Attack Trees und unter Berücksichtigung der Gesundheitstelematik 113

perspeGKtive 2010

Wissenschaftliche Beiträge

Verschlüsselt Rechnen: Sichere Verarbeitung Verschlüsselter Medizinischer Daten am Beispiel der Klassifikation von EKG Daten

Ahmad-Reza Sadeghi, Thomas Schneider

{ahmad.sadeghi,thomas.schneider}@trust.rub.de

Abstract: Die rasant fortschreitende Modernisierung des Gesundheitswesens durch neuartige elektronische Dienste wie die elektronische Patientenakte und medizinische online Dienste erlaubt es, medizinische Prozesse effizienter zu gestalten. Andererseits birgt die digitale Verarbeitung sensibler Patientendaten Risiken und Gefahren hinsichtlich des Datenschutzes und des Missbrauchs.

Klassische kryptographische und IT sicherheitstechnische Methoden erlauben die sichere Verteilung und Speicherung medizinischer Daten. Allerdings erfordert die Verarbeitung der Daten deren Entschlüsselung. Zu diesem Zeitpunkt kann auf die Daten im Klartext (z.B. durch Systemadministratoren) zugegriffen und die Vertraulichkeit verletzt werden. Um dies zu verhindern, sollten Daten in verschlüsselter Form verarbeitet werden. Für ebendieses "Rechnen unter Verschlüsselung" wurden in den vergangenen 25 Jahren verschiedene kryptographische Verfahren vorgeschlagen.

Die Ziele dieser Arbeit sind folgende: 1) Der aktuelle Stand der Forschung im Bereich des effizienten Rechnens unter Verschlüsselung wird zusammengefasst. 2) Ein Werkzeug wird vorgestellt, das es erlaubt, effiziente Protokolle zum Rechnen unter Verschlüsselung abstrakt und ohne detaillierte kryptographische Kenntnisse zu beschreiben und automatisch zu generieren. 3) Es wird gezeigt, wie das vorgestellte Werkzeug exemplarisch zum Generieren eines medizinischen Web-Services verwendet werden kann, der EKG Daten in verschlüsselter Form klassifiziert.

1 Einleitung

Die Gesundheitsindustrie entwickelt in zunehmendem Maße medizinische Online-Dienste für Patienten, die ihnen schnell Auskunft über ihren Gesundheitszustand geben und gegebenenfalls weitere Maßnahmen wie das Aufsuchen eines Facharztes empfehlen können. Solche Technologien haben das Potential, medizinische Daten und Wissen für Millionen von Benutzern weltweit zu verwalten, verarbeiten, speichern, verteilen und allgegenwärtig zur Verfügung zu stellen. Prominente Beispiele für solche Online-Dienste sind Google Health [McB08] und Microsoft HealthVault [Bla08]. Da in diesen Diensten sensible Patientendaten verarbeitet werden, ist der Datenschutz von höchster Priorität. Falls es Zweifel seitens der potentiellen Nutzer an diesem Schutz gibt, wird die Verbreitung solcher neuer elektronischer Gesundheitsdienste verhindert oder zumindest verlangsamt. Auch großflächige Systeme zum Speichern von medizinischen Daten wie die elektronische Patientenakte könnten dahingehend erweitert werden, dass sie die vertrauliche Verarbeitung

medizinischer Daten ermöglichen.

Methoden der klassischen Kryptographie wie symmetrische und asymmetrische Verschlüsselung erlauben den sicheren Transport von Daten zwischen Rechnern und bieten Schutz gegen Angreifer von außen und unberechtigte Innentäter. Diese Methoden erlauben es, medizinische Daten sicher auf einen oder mehrere Server zu verteilen und dort verschlüsselt zu speichern. Zur Verarbeitung solcher verschlüsselter Daten müssten diese jedoch zunächst entschlüsselt und dann unverschlüsselt verarbeitet werden, bevor das Ergebnis wieder verschlüsselt wird. Zu dem Zeitpunkt, an dem die Daten unverschlüsselt vorliegen, könnten Innentäter wie zum Beispiel Systemadministratoren die Daten ausspionieren und somit die Vertraulichkeit des Systems verletzen. Um dies zu verhindern, sollten Daten in verschlüsselter Form verarbeitet werden, ohne sie zu entschlüsseln. Für ebendieses "Rechnen unter Verschlüsselung" wurden in den vergangenen 25 Jahren verschiedene kryptographische Verfahren vorgeschlagen.

1.1 Inhalt

In dieser Arbeit fassen wir zunächst in §2 den aktuellen Stand der Technik zum Rechnen unter Verschlüssung in, auch ohne kryptographischen Hintergrund, leicht verständlicher Form zusammen. Unsere Beschreibung basiert auf dem Modell von [KSS10], in dem Algorithmen zum Rechnen unter Verschlüsselung programmiert werden können, wobei von der zugrunde liegenden Kryptographie abstrahiert wird.

Anschließend stellen wir in §3 das zugehörige Werkzeug TASTY (Tool for Automating Secure Two-partY computations) [HKS+10] vor, das es Programmierern erlaubt, solche Algorithmen zum Rechnen unter Verschlüsselung in einer domänen-spezifischen Hochsprache zu beschreiben und mit einem Compiler automatisch in effiziente und sichere kryptographische Protokolle zu übersetzen.

Als Anwendung betrachten wir in §4 einen medizinischen Web-Service, der EKG Daten in verschlüsselter Form klassifiziert. Hierbei wird sowohl die Privatsphäre des Patienten als auch das geistige Eigentum des Dienstanbieters geschützt. Wir zeigen wie unter Verwendung von TASTY die diesem Web-Service zu Grunde liegenden kryptographischen Protokolle beschrieben und automatisch generiert werden können.

Abschließend diskutieren wir in §5 den beschriebenen Ansatz des Rechnens auf verschlüsselten Daten im Kontext medizinischer Anwendungen.

2 Verfahren zum effizienten Rechnen unter Verschlüsselung

Beim *Rechnen unter Verschlüsselung* wollen zwei Parteien, Client $\mathcal{C}$ und Server $\mathcal{S}$, eine beiden bekannte Funktion f auf ihren geheimen Eingaben x bzw. y so berechnen, dass lediglich das Ergebnis $f(x, y)$ bekannt wird, jedoch keine zusätzliche Information über die Eingabe der anderen Partei oder Zwischenergebnisse.

Sicherheitsmodell. Im Folgenden nehmen wir an, dass beide Parteien ehrlich aber neugierig sind (engl. honest-but-curious). Sie verhalten sich in dem Sinne *ehrlich*, daß sie die im Protokoll vorgegebenen Aktionen durchführen, sind aber *neugierig*, da sie versuchen, aus den empfangenen Nachrichten zusätzliche Informationen zu rekonstruieren. Die von uns vorgestellten Protokolle garantieren, dass falls sich beide Parteien ehrlich verhalten, sie keine zusätzlichen Informationen über die Eingabe der anderen Partei oder Zwischenergebnisse lernen. Dieses Modell ist nicht trivial und verhindert bereits viele realistische Angriffsszenarien wie "Insider Attacken", bei denen zum Beispiel Systemadministratoren Zugang zu geheimen Informationen erhalten. Für eine Übersicht zu erweiterten Protokollen, die sicher gegen stärkere Angreifer sind, verweisen wir auf [KSS10].

Ansätze. Es gibt zwei prinzipielle Ansätze zum Rechnen unter Verschlüsselung:

Beim *"Rechnen mit verschlüsselten Funktionen"* [Yao86] wird zusätzlich zu den verschlüsselten Daten, für jede der zu berechnenden Elementaroperationen eine verschlüsselte Übersetzungstabelle verwendet, die es erlaubt, eine Funktion *einmalig* auf den verschlüsselten Daten zu berechnen.

Beim *"Rechnen auf verschlüsselten Daten"* [Pai99] werden die Daten mit einem speziellen Verschlüsselungsverfahren, sogenannter "homomorpher Verschlüsselung" verschlüsselt, das es erlaubt, ohne Verwendung von Übersetzungstabellen, bestimmte Operationen auf den verschlüsselten Daten mehrmals zu berechnen.

Im Folgenden stellen wir zunächst effiziente Verfahren für diese beiden Ansätze vor und beschreiben anschließend ein Modell, das deren beliebige Kombination erlaubt.

2.1 Rechnen auf verschlüsselten Daten

Homomorphe Verschlüsselungsverfahren erlauben es, bestimmte Operationen auf verschlüsselten Daten durchzuführen.

Dies erlaubt das *Rechnen auf verschlüsselten Daten*: Client $\mathcal{C}$ generiert ein Schlüsselpaar für ein homomorphes Verschlüsselungsverfahren und schickt den öffentlichen Schlüssel gemeinsam mit der homomorphen Verschlüsselung seiner Inputs an $\mathcal{S}$. Dieser kann dank der homomorphen Eigenschaften auf den (homomorph) verschlüsselten Daten die gewünschte Funktion berechnen und schickt das verschlüsselte Ergebnis zurück an $\mathcal{C}$, der entschlüsselt.

Wir schreiben $[\![x]\!]$ für die homomorphe Verschlüsselung (Chiffretext) des Klartextes x unter $\mathcal{C}$'s öffentlichem Schlüssel.

Additiv homomorphe Verschlüsselungsverfahren erlauben die Addition unter Verschlüsselung, d.h. für alle möglichen Klartexte x, y aus dem Klartextraum P gilt:

$$\forall x, y \in P : [\![x]\!] \boxplus [\![y]\!] = [\![x + y]\!].$$

Wiederholtes Verdoppeln und Addieren erlaubt Multiplikation mit einer Konstanten a:

$$\forall a \in \mathbb{Z}, x \in P : a[\![x]\!] = [\![ax]\!].$$

Das am weitesten verbreitete additiv homomorphe Verschlüsselungsverfahren ist das von Paillier [Pai99] mit Optimierungen aus [DJ01]. Der öffentliche Schlüssel dieses Verfahrens ist ein RSA Modul n, d.h. das Produkt zweier sehr großer[1] Primzahlen p, q, und der geheime Schlüssel die Faktorisierung p, q von n; der Klartextraum P ist $\mathbb{Z}_n = [0, \ldots, n-1]$.

Multiplikation unter additiv homomorpher Verschlüsselung erfordert eine Runde Interaktion zwischen $\mathcal{S}$ und $\mathcal{C}$: Hierzu addiert $\mathcal{S}$ zu den zu multiplizierenden Chiffretexten $[\![x]\!]$ und $[\![y]\!]$ eine zufällig gewählte Maske r_x bzw. r_y und schickt $[\![\bar{x}]\!] = [\![x]\!] \boxplus [\![r_x]\!]$, $[\![\bar{y}]\!] = [\![y]\!] \boxplus [\![r_y]\!]$ an $\mathcal{C}$. $\mathcal{C}$ entschlüsselt, multipliziert und schickt $[\![\bar{x} \cdot \bar{y}]\!]$ zurück an $\mathcal{S}$. Schließlich berechnet $\mathcal{S}$ das gewünschte homomorph verschlüsselte Produkt als

$$[\![x \cdot y]\!] = [\![\bar{x} \cdot \bar{y}]\!] \boxplus (-r_x)[\![y]\!] \boxplus (-r_y)[\![x]\!] \boxplus [\![-r_x r_y]\!] = [\![(x+r_x)(y+r_y) - r_x y - r_y x - r_x r_y]\!].$$

Voll homomorphe Verschlüsselungsverfahren erlauben neben der Addition auch die (nicht-interaktive) Multiplikation unter Verschlüsselung. Da Addition und Multiplikation eine vollständige Boole'sche Basis bilden erlaubt ein solches Verschlüsselungsverfahren die Berechnung *beliebiger* Funktionen unter Verschlüsselung.

Lange Zeit waren lediglich voll homomorphe Verfahren mit begrenzter Anzahl an Multiplikationen bekannt, z.B. maximal eine Multiplikation [BGN05, GHV10] oder Chiffretexte, die exponentiell in der Anzahl der Multiplikationen wachsen [SYY99]. Die kürzlich vorgestellten voll homomorphen Verschlüsselungsverfahren [Gen09, SV10, DGHV10] unterliegen zwar theoretisch nicht mehr solchen Einschränkungen, sind jedoch für praktische Anwendungen noch viel zu ineffizient.

Zur Zeit sind die in der Praxis effizientesten Verfahren zum Rechnen auf verschlüsselten Daten somit die additiv homomorphen Verfahren mit interaktiver Multiplikation.

2.2 Rechnen mit verschlüsselten Funktionen

Beim Rechnen mit verschlüsselten Funktionen wird eine verschlüsselte Funktion (engl. "garbled functions") auf den verschlüsselten Daten berechnet. Hierbei wird die Funktion nicht auf Klartextwerten ('0' oder '1'), sondern auf zufällig gewählten symmetrischen Schlüsseln (engl. "garbled values") berechnet [Yao86]. Im Vergleich zum Rechnen auf verschlüsselten Daten mit homomorpher Verschlüsselung (vgl. §2.1) kann auf diesen symmetrischen Schlüsseln nicht direkt gerechnet werden, sondern nur unter Verwendung von verschlüsselten Übersetzungstabellen (engl. "garbled tables") für jede Elementaroperation, die in der Setup Phase übertragen werden. Die anschließende Online Phase

[1] Nach aktuellen Empfehlungen sollten die beiden Faktoren etwa tausend Bit lang sein [GQ09].

von Protokollen zum Rechnen mit verschlüsselten Funktionen ist sehr effizient, da lediglich kryptographische Hashfunktionen berechnet werden müssen und keine rechenaufwändigen public-key Operationen wie beim Rechen auf homomorph verschlüsselten Daten [KSS09, KSS10].

2.2.1 Yao's Protokoll

Zusammengefasst funktioniert Yao's Protokoll zum Rechnen mit verschlüsselten Schaltkreisen [Yao86] wie in Abbildung 1 gezeigt:

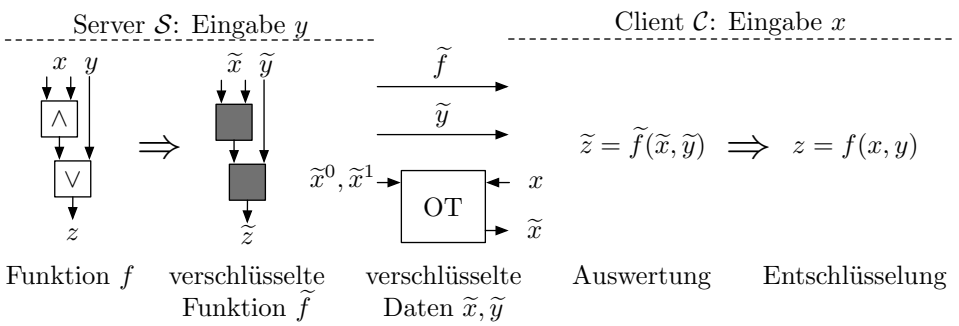

Abbildung 1: Yao's Protokoll zum Rechnen mit verschlüsselten Funktionen

In der Setup Phase generiert der Server $\mathcal{S}$ die verschlüsselte Funktion $\widetilde{f}$ aus der als Boole'scher Schaltkreis repräsentierten zu berechnenden Funktion f. Hierfür weist $\mathcal{S}$ jeder Kante von f zwei zufällige symmetrische Schlüssel zu $\widetilde{w}_i^0, \widetilde{w}_i^1$, die den entsprechenden Klartextwerten 0 bzw. 1 entsprechen. Da beide Schlüssel zufällig sind liefert $\widetilde{w}_i^j$ keine Information über den zugehörigen Klartextwert j. Nun berechnet $\mathcal{S}$ für jedes Gatter von f eine verschlüsselte Funktionstabelle, die es erlaubt, bei Kenntnis der Schlüssel für die Eingabekanten dieses Gatters, den entsprechenden Schlüssel für die Ausgabekante des Gatters zu berechnen. Die verschlüsselte Funktion $\widetilde{f}$ besteht aus den verschlüsselten Funktionstabellen für jedes Gatter und wird an $\mathcal{C}$ geschickt.

Später, in der Online Phase des Protokolls, erhält $\mathcal{C}$ die zu den Eingaben x und y gehörenden Schlüssel $\widetilde{x}$ und $\widetilde{y}$ und kann die verschlüsselte Funktion $\widetilde{f}$ Gatter für Gatter auswerten und erhält die verschlüsselte Ausgabe $\widetilde{z} = \widetilde{f}(\widetilde{x}, \widetilde{y})$, die mit Hilfe von $\mathcal{S}$ in die zugehörige Klartextausgabe $z = f(x, y)$ entschlüsselt werden kann.

Um ein Bit y_i der Eingabe von $\mathcal{S}$ zu verschlüsseln, sendet $\mathcal{S}$ einfach den zugehörigen Schlüssel $\widetilde{y}_i = \widetilde{y}_i^{y_i}$ an $\mathcal{C}$. Analog soll $\mathcal{C}$ den verschlüsselten Wert $\widetilde{x}_i = \widetilde{x}_i^{x_i}$ erhalten jedoch so, dass $\mathcal{S}$ nicht den Klartextwert x_i erfährt. Dies wird mit einem kryptographischen Protokoll genannt Oblivious Transfer (OT) erreicht (Details in §A).

Die verschlüsselte Funktion darf nur *genau einmal ausgewertet* werden, da ansonsten $\mathcal{C}$ aus zuvor gesehenen Schlüsseln Informationen über die Eingabewerte von $\mathcal{S}$ lernen kann. Für den Sicherheitsbeweis des skizzierten Protokolls von Yao verweisen wir auf [LP09], für effiziente Instanzierungen auf [PSSW09] und weitere Details auf [KSS10].

2.3 Kombiniertes Rechnen unter Verschlüsselung

Das in [KSS10] vorgeschlagene Modell erlaubt es, Protokolle zum Rechnen unter Verschlüsselung als Sequenz von Operationen auf verschlüsselten Daten zu beschreiben wie in Abbildung 2 gezeigt: Beide Parteien haben zunächst Klartext Werte x als Eingaben. Diese werden zunächst in verschlüsselte Werte übersetzt, auf denen verschlüsselt gerechnet werden kann. Abschließend wird das verschlüsselte Endergebnis wieder zurück in einen Klartext Wert für Client $\mathcal{C}$ oder Server $\mathcal{S}$ umgewandelt.

Je nachdem, welche Operationen unter Verschlüsselung berechnet werden sollen stehen zwei verschiedene Arten der Verschlüsselung zur Verfügung: "Homomorphe Werte" $[\![x]\!]$ zum Rechnen unter homomorpher Verschlüsselung (vgl. §2.1) oder "Garbled Werte" $\widetilde{x}$ zum Rechnen mit verschlüsselten Funktionen (vgl. §2.2), die auch ineinander umgewandelt werden können ($\widetilde{x} \Leftrightarrow [\![x]\!]$).

Die Kombination dieser beiden Verfahren erlaubt es, das jeweils effizienteste Verfahren für eine bestimmte Teilfunktionalität zu verwenden. So erlaubt homomorphe Verschlüsselung beispielsweise effiziente (interaktive) Multiplikation für große Werte [HKS+10], während verschlüsselte Schaltkreise für Vergleichsoperationen besser geeignet sind [KSS09].

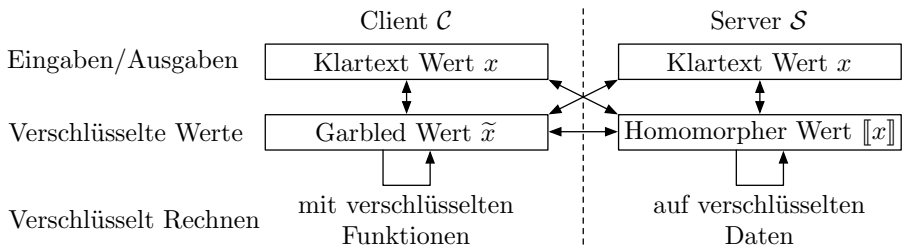

Abbildung 2: Modell zum kombinierten Rechnen unter Verschlüsselung

3 TASTY: Werkzeug zum Rechnen unter Verschlüsselung

Im Folgenden stellen wir TASTY (Tool for Automating Secure Two-partY computations) [HKS+10] vor, ein Werkzeug, das es erlaubt, Protokolle zum Rechnen unter Verschlüsselung zu beschreiben und automatisch zu generieren.

Existierende Ansätze. Im Gegensatz zu vorherigen Werkzeugen wie Fairplay [MNPS04, BDNP08] oder VIFF [DGKN09], die jeweils auf ein Paradigma zum Rechnen unter Verschlüsselung beschränkt waren (vgl. §2.1 bzw. §2.2), unterstützt TASTY die Kombination beider Verfahren, d.h. homomorphe Verschlüsselung und verschlüsselte Schaltkreise (vgl. §2.3), um hoch effiziente Protokolle zu erhalten.

TASTY verfolgt unter anderem die folgenden Design-Ziele:

1. Protokolle zum kombinierten Rechnen unter Verschlüsselung können in TASTYL programmiert werden, einer intuitiven Hochsprache, die solche Protokolle als Sequenz von Operationen auf verschlüsselten Daten beschreibt (vgl. §3.2).
2. Da besonders in Web-Applikationen die Antwortzeit kritisch ist optimiert TASTY die generierten Protokolle dahingehend, dass die Latenz der Online Phase, d.h. die Zeit von der Eingabe der Daten bis zur Ausgabe des Ergebnisses, minimiert wird. Dies wird dadurch erreicht, dass rechenaufwändige kryptographische Operationen und ein Großteil der Daten bereits in einer Setup Phase vorberechnet und ausgetauscht werden.

3.1 TASTY: Architektur und Workflow

Der Workflow zur Benutzung von TASTY ist wie in Abbildung 3 gezeigt:

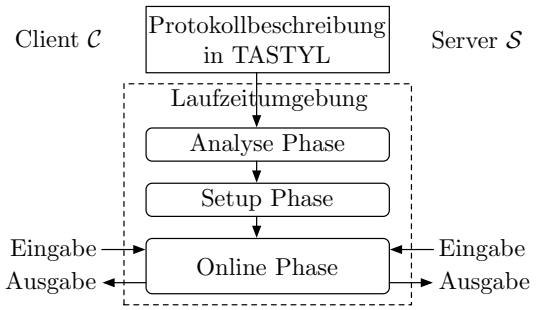

Abbildung 3: Architektur und Workflow von TASTY

1. Die beiden Benutzer, Client $\mathcal{C}$ und Server $\mathcal{S}$, einigen sich auf eine *Protokollbeschreibung* des zu berechnenden Protokolls in der TASTY Eingabesprache (TASTYL), die in §3.2 beschrieben wird.
2. Beide Benutzer rufen die *Laufzeitumgebung* von TASTY auf, ein Programm, welches das Protokoll analysiert und ausführt:
   (a) In der *Analyse Phase* überprüft die Laufzeitumgebung die syntaktische Korrektheit der Protokollbeschreibung, stellt sicher, dass beide Parteien das selbe Protokoll ausführen und ermittelt automatisch, welche Teile des Protokolls vorberechnet werden können.
   (b) In der *Setup Phase* führen beide Parteien die Vorberechnungen aus.
   (c) In der *Online Phase* geben beide Parteien ihre Eingaben ein und das Protokoll berechnet die jeweiligen Ausgaben unter Verschlüsselung.

3.2 TASTYL: Sprache zum kombinierten Rechnen unter Verschlüsselung

TASTYL, die "TASTY input Language", ist eine domänenspezifische Hochsprache zum Beschreiben von Protokollen zum kombinierten Rechnen unter Verschlüsselung als Sequenz von Operationen auf verschlüsselten Daten.

Im Folgenden fassen wir einige Grundelemente des Typsystems von TASTYL wie in Abbildung 4 zusammen. Ein Beispielprogramm in TASTYL ist in §B aufgelistet. Für Details zu TASTY und weitere Programmbeispiele verweisen wir auf [HKS+10].

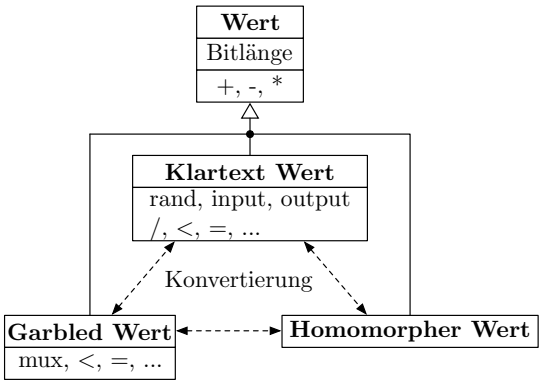

Abbildung 4: Typsystem von TASTYL (vereinfacht)

Jede Variable in TASTYL ist ein *Wert* (s. Abbildung 4) mit einer festen Bitlänge oder ein mehrdimensionaler Vektor bestehend aus mehreren gleichartigen Werten, auf die wir im Folgenden nicht näher eingehen werden. Jeder Wert kann entweder ein unverschlüsselter *Klartext Wert* oder ein verschlüsselter *Garbled Wert* bzw. *Homomorphic Wert* sein. Alle Werte unterstützen die Operationen Addition, Subtraktion und Multiplikation.

Zusätzlich zu diesen gemeinsam Operatoren werden weitere Operatoren und Konvertierungen angeboten:

Klartext Werte. Eingaben (`input`) und Ausgaben (`output`) der beiden Parteien sind Klartext Werte. Diese können zufällig gewählt werden (`rand`) und bieten zuätzliche Operationen wie Integer Division und Vergleiche.

Homomorphe Werte. Klartext Werte können in homomorph verschlüsselte *Homomorphe Werte* konvertiert werden und umgekehrt. Homomorphe Werte können nichtinteraktiv addiert, subtrahiert und mit einem Klartext Wert auf Seite von $\mathcal{S}$ multipliziert werden – die Multiplikation zweier Homomorpher Werte erfordert eine Runde Interaktion (vgl. §2.1).

Garbled Werte. Sowohl Klartext Werte als auch Homomorphe Werte können in verschlüsselte *Garbled Werte* konvertiert werden und umgekehrt. Der Vergleich zweier Garbled Werte liefert ein Garbled Bit (Garbled Wert mit Bitlänge 1), welches wiederum verwendet werden kann, um zwischen zwei Garbled Werten auszuwählen (Multiplexer `mux`). Für jede Operation auf Garbled Werten erzeugt TASTY dynamisch den entsprechenden verschlüsselten Schaltkreis.

4 Szenario: Klassifikation von EKG Daten unter Verschlüsselung

Als medizinisches Anwendungsszenario betrachten wir einen Web-Service, der biometrische Daten unter Verschlüsselung klassifizieren soll. Aus dem breiten Spektrum verschiedener biomedizinischer Daten [BCA+93, TWBT95, FLK08, KTB+03] wählen wir Elektrokardiogramm (EKG) Daten als einfaches aber repräsentatives Beispiel.

Ein Patient (Client $\mathcal{C}$) hat ein EKG erstellt und möchte dieses bei einem Service Anbieter (Server $\mathcal{S}$) klassifizieren lassen. Bezüglich der Privatsphäre ergeben sich folgende Anforderungen: $\mathcal{C}$ wünscht, dass $\mathcal{S}$ keinerlei Information über das EKG Signal erhält, da dies sensitive personenbezogene Daten sind). Andererseits möchte $\mathcal{S}$ Details über den verwendeten Klassifikationsalgorithmus vor $\mathcal{C}$ verbergen, da dies wertvolles geistiges Eigentum von $\mathcal{S}$ darstellt. Im Folgenden nehmen wir an, dass die Struktur des verwendeten Klassifikationsalgorithmus beiden Parteien bekannt ist und das schützenswerte geistige Eigentum von $\mathcal{S}$ die Parametrisierung des Algorithmus ist.

Eine solche sichere Klassifikation von EKG Daten kann dadurch erreicht werden, dass ein Algorithmus zur Klassifikation von EKG Daten unter Verschlüsselung berechnet wird: Hierfür wird zunächst ein bestehender Algorithmus zur EKG Klassifikation von Klartextdaten gewählt (in unserem Beispiel [ASSK07, GSK02]) und auf Integer-Arithmetik abgebildet (im Beispiel wie in [BFK+09a, BFK+09b] beschrieben). Anschließend kann dieser Algorithmus in TASTYL programmiert werden (s. §B). Schließlich wird dieses TASTYL Programm sicher von der TASTY Laufzeitumgebung berechnet.

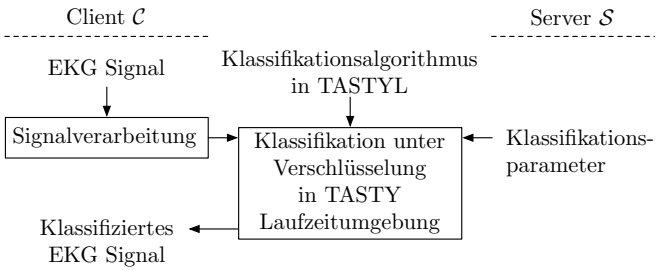

Abbildung 5: EKG Klassifikation unter Verschlüsselung mit TASTY

Der Gesamtablauf ist in Abbildung 5 gezeigt: Das EKG Signal von $\mathcal{C}$ wird zunächst vorverarbeitet (Rauschfilterung, Quantisierung, etc. – Details s. [BFK+09a, BFK+09b]). Der in TASTYL beschriebene Klassifikationsalgorithmus (s. §B) bekommt das verarbei-

tete EKG Signal als (geheime) Eingabe von $\mathcal{C}$ und die Parameter zur Konfiguration des Klassifikationsalgorithmus als (geheime) Eingabe von $\mathcal{S}$. Die TASTY Laufzeitumgebung berechnet die Klassifikation unter Verschlüsselung und gibt das klassifizierte EKG Signal an $\mathcal{C}$ aus.

5 Diskussion des Ansatzes

Abschließend möchten wir den Ansatz des Rechnens unter Verschüsselung und seine Verwendbarkeit insbesondere im Hinblick auf medizinische Anwendungen kurz diskutieren.

Effizienz. Protokolle zum Rechnen unter Verschlüsselung sind naturgemäß langsamer als wenn die Berechnungen unverschlüsselt durchgeführt werden. Als Faustregel für die Effizienz gilt, dass Protokolle zum Rechnen mit verschlüsselten Funktionen (s. §2.2) in der Online Phase in etwa eine symmetrische Entschlüsselung pro Bit-Operation kosten, während Berechnungen unter additiv homomorpher Verschlüsselung (s. §2.1) langsamer als bestehende public-key Verfahren wie RSA sind.

Dank der rasanten Zunahme verfügbarer Rechenleistung und rapide steigender Übertragungsraten von Datennetzen, in Kombination mit mehreren algorithmischen Verbesserungen der zugrunde liegenden Primitiven und Protokolle, ist Rechnen unter Verschlüsselung jedoch in den Bereich der Anwendbarkeit in der Praxis gerückt, wie beispielsweise ein System zur sicheren Gesichtserkennung zeigt [OPJM10].

Die sichere Klassifikation von EKG Daten auf aktueller PC Hardware benötigt ca. 19 s Rechenzeit und 64 kByte Datentransfer [BFK+09a].

Sicherheit. Der beim Rechnen unter Verschlüsselung verwendete Sicherheitsparameter bestimmt die Zeitspanne, bis zu der die Verschlüsselung als sicher angenommen werden kann. Die meisten existierenden prototypischen Implementierungen zum Rechnen unter Verschlüsselung basieren jedoch auf einem symmetrischen Sicherheitsparameter von 80 bit, dessen Verwendung nur noch bis Ende 2010 empfohlen wird [GQ09].

Insbesondere in medizinischen Anwendungen, in denen sensible Patientendaten unter Verschlüsselung verarbeitet werden, ist es notwendig, dass mitgeschnittene Protokollläufe auch in Zukunft nicht entschlüsselt werden können. Hierfür sollte ein ausreichend großer Sicherheitsparameter gewählt werden (z.B: 128 bit, dessen Sicherheit bis zum Jahr 2040 prognostiziert wird [GQ09]). Die Verwendung eines größeren Sicherheitsparameters hat jedoch negative Auswirkungen auf die Performanz der Protokolle. Beispielsweise dauert die Klassifikation von EKG Daten mit 112 bit Sicherheitsparameter (empfohlen bis zum Jahr 2030) ca. 41 s und 89 kByte.

Protokolle zum Rechnen unter Verschlüsselung mit *informationstheoretischer Sicherheit*, d.h. Protokolle, die mit unendlicher Rechenleistung nicht zu brechen sind, sind zwar bekannt [BOGW88, CCD88], jedoch zu komplex für den Einsatz in der Praxis.

In manchen Anwendungen ist die in vielen Papieren getroffene Annahme, dass die betei-

ligten Parteien ehrlich aber neugierig (engl. “semi-honest adversaries”) sind, also nicht in bösartiger Absicht vom Protokoll abweichen, ebenfalls nicht gerechtfertigt. Bestehende Protokolle zum Rechnen unter Verschlüsselung, die solche Betrugsversuche mit hoher (engl. “covert adversaries”) oder an Sicherheit grenzender (engl. “malicious adversaries”) Wahrscheinlichkeit erkennen sind jedoch deutlich aufwändiger. Beispielsweise benötigt die sichere Berechnung der aus 33.880 Bit-Operationen bestehenden AES Funktionalität mit Sicherheitsparameter 128 bit im semi-honest Fall 7 s ($0,5$ MByte), im covert Fall 60 s ($8,7$ MByte) und im malicious Fall 19 min (408 MByte) [PSSW09].

Benutzbarkeit. Neben der Performanz und Sicherheit der Protokolle zum Rechnen unter Verschlüsselung ist ihre Benutzbarkeit der wohl wichtigste Aspekt für den Einsatz in praktischen Anwendungen, um Entwicklungskosten zu minimieren.

Für den Entwurf effizienter und sicherer Protokolle war lange Zeit ein detailliertes Expertenwissen nötig. Auch bei der Implementierung der Protokolle konnten sich diverse Programmierfehler einschleichen, die die Sicherheit gefährdeten. Dies ist vergleichbar mit Zeiten, in denen ausschließlich in Maschinensprache (Assembler) programmiert wurde.

Compiler für kryptographische Protokolle (wie das in §3 beschriebene Werkzeug TASTY) abstrahieren von den kryptographischen Details und erlauben es so Anwendungsentwicklern auch ohne kryptographischem Hintergrundwissen die zu berechnende Funktionalität in einer Hochsprache zu beschreiben. Aus dieser Hochsprachenbeschreibung wird dann automatisch ein ausführbares, effizientes und sicheres Protokoll erzeugt. In Analogie abstrahieren Hochsprachen wie Java, C oder C++ von maschinenspezifischen Details und generieren aus Programmen effizienten und korrekten (im Sinne von äquivalent zur Hochsprachenbeschreibung) Maschinencode und erhöhen somit die Produktivität.

Im EU Projekt CACE (Computer Aided Cryptography Engineering)[2] werden neben TASTY eine Vielzahl von Werkzeugen zur automatischen Generierung von kryptographischen Protokollen und Primitiven entwickelt [BBB+10].

Nutzerfreundlichkeit. Zusammengefasst ist das Rechnen unter Verschlüsselung bereits heute technisch möglich und kann prinzipiell als Grundlage für vielerlei innovative und Datenschutz-konforme Anwendungen genutzt werden.

Um eine hohe Akzeptanz solcher Systeme zu erreichen ist neben dem Aspekt der Sicherheit jedoch vor allem die Nutzerfreundlichkeit von herausragender Bedeutung. Damit die Patienten solche neuartigen Dienste akzeptieren und als Ergänzung zum persönlichen Arztbesuch ansehen, ist vor allem eine einfach zu bedienende Benutzerschnittstelle wichtig, die sich nahtlos in den gewohnten Alltag der Patienten integriert.

[2] `http://www.cace-project.eu`

Danksagung

Diese Arbeit wurde gefördert durch das NRW Projekt MediTrust und das EU Projekt CACE (Computer Aided Cryptography Engineering).

Literatur

[ASSK07] U. R. Acharya, J. Suri, J. A. E. Spaan und S. M. Krishnan. *Advances in Cardiac Signal Processing*, Kapitel 8. Springer, 2007.

[BBB+10] E. Bangerter, M. Barbosa, D.J. Bernstein, I. Damgard, D. Page, J.I. Pagter, A.-R. Sadeghi und S. Sovio. Using Compilers to Enhance Cryptographic Product Development. In *Information Security Solutions Europe (ISSE'10)*, Seiten 291–301. Vieweg+Teubner, 2010.

[BCA+93] F. M. Bennett, D. J. Christini, H. Ahmed, K. Lutchen, J. M. Hausdorff und N. Oriol. Time series modeling of heart rate dynamics. In *Computers in Cardiology 1993. Proceedings.*, Seiten 273–276, 1993.

[BDNP08] A. Ben-David, N. Nisan und B. Pinkas. FairplayMP: a system for secure multi-party computation. In *ACM Conference on Computer and Communications Security (CCS'08)*, Seiten 257–266. ACM, 2008. `http://fairplayproject.net/fairplayMP.html`.

[Bea95] D. Beaver. Precomputing Oblivious Transfer. In *Advances in Cryptology – CRYPTO'95*, *LNCS*, Band 963, Seiten 97–109. Springer, 1995.

[BFK+09a] M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A. Paus, A.-R. Sadeghi und T. Schneider. Efficient Privacy-Preserving Classification of ECG Signals. In *IEEE International Workshop on Information Forensics and Security (IEEE WIFS'09)*, Seiten 91–95. IEEE, 2009.

[BFK+09b] M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A.-R. Sadeghi und T. Schneider. Secure Evaluation of Private Linear Branching Programs with Medical Applications. In *European Symposium on Research in Computer Security (ESORICS'09)*, *LNCS*, Band 5789, Seiten 424–439. Springer, 2009.

[BGN05] D. Boneh, E.-J. Goh und K. Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In *Theory of Cryptography Cronference (TCC'05)*, *LNCS*, Band 3378, Seiten 325–341. Springer, 2005.

[Bla08] D. Blankenhorn. Microsoft HealthVault is nothing like Google Health, 2008.

[BOGW88] M. Ben-Or, S. Goldwasser und A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *Symposium on Theory of Computing (STOC'88)*, Seiten 1–10. ACM, 1988.

[CCD88] D. Chaum, C. Crépeau und I. Damgård. Multiparty Unconditionally Secure Protocols (Extended Abstract). In *Symposium on Theory of Computing (STOC'88)*, Seiten 11–19. ACM, 1988.

[DGHV10] M. v. Dijk, C. Gentry, S. Halevi und V. Vaikuntanathan. Fully Homomorphic Encryption over the Integers. In *Advances in Cryptology – EUROCRYPT'10*, *LNCS*, Band 6110, Seiten 24–43. Springer, 2010.

[DGKN09] I. Damgård, M. Geisler, M. Krøigård und J. B. Nielsen. Asynchronous Multiparty Computation: Theory and Implementation. In *Public Key Cryptography (PKC'09)*, *LNCS*, Band 5443, Seiten 160–179. Springer, 2009. `http://viff.dk`.

[DJ01] I. Damgård und M. Jurik. A Generalisation, a Simplification and some Applications of Paillier's Probabilistic Public-Key System. In *Public-Key Cryptography (PKC'01)*, *LNCS*, Band 1992, Seiten 119–136. Springer, 2001.

[FLK08] P. Flor-Henry, J. L. Lind und Z. J. Koles. Quantitative EEG and source localization in fibromyalgia. *International Journal of Psychophysiology*, 69(3):142–142, 2008.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *ACM Symposium on Theory of Computing (STOC'09)*, Seiten 169–178. ACM, 2009.

[GHV10] C. Gentry, S. Halevi und V. Vaikuntanathan. A Simple BGN-type Cryptosystem from LWE. In *Advances in Cryptology – EUROCRYPT'10*, *LNCS*, Band 6110, Seiten 506–522. Springer, 2010.

[GQ09] D. Giry und J.-J. Quisquater. Cryptographic Key Length Recommendation, March 2009. `http://keylength.com`.

[GSK02] D. F. Ge, N. Srinivasan und S. M. Krishnan. Cardiac arrhythmia classification using autoregressive modeling. *BioMedical Engineering OnLine*, 1(1):5, 2002.

[HKS+10] W. Henecka, S. Kögl, A.-R. Sadeghi, T. Schneider und I. Wehrenberg. TASTY: Tool for Automating Secure Two-partY computations. In *ACM Conference on Computer and Communications Security (CCS'10)*. ACM, October 4-8, 2010. `http://tastyproject.net`.

[IKNP03] Y. Ishai, J. Kilian, K. Nissim und E. Petrank. Extending Oblivious Transfers Efficiently. In *Advances in Cryptology – CRYPTO'03*, *LNCS*, Band 2729, Seiten 145–161. Springer, 2003.

[KSS09] V. Kolesnikov, A.-R. Sadeghi und T. Schneider. Improved Garbled Circuit Building Blocks and Applications to Auctions and Computing Minima. In *Cryptology and Network Security (CANS'09)*, *LNCS*, Band 5888, Seiten 1–20. Springer, 2009.

[KSS10] V. Kolesnikov, A.-R. Sadeghi und T. Schneider. Modular Design of Efficient Secure Function Evaluation Protocols. Cryptology ePrint Archive, Report 2010/079, 2010. `http://eprint.iacr.org/2010/079/`.

[KTB+03] P. Kalra, J. Togami, G. Bansal, A. W. Partin, M. K. Brawer, R. J. Babaian, L. S. Ross und C. S. Niederberger. A neurocomputational model for prostate carcinoma detection. *Cancer*, 98(9):1849–1854, 2003.

[LP09] Y. Lindell und B. Pinkas. A Proof of Yao's Protocol for Secure Two-Party Computation. *Journal of Cryptology*, 22(2):161–188, 2009. Cryptology ePrint Archive: Report 2004/175.

[McB08] M. McBride. Google Health: Birth of a giant. *Health Management Technology*, 29:8–10, 2008.

[MNPS04] D. Malkhi, N. Nisan, B. Pinkas und Y. Sella. Fairplay — a secure two-party computation system. In *USENIX Security Symposium*, 2004. `http://fairplayproject.net/fairplay.html`.

[NP01] M. Naor und B. Pinkas. Efficient oblivious transfer protocols. In *ACM-SIAM Symposium On Discrete Algorithms (SODA'01)*, Seiten 448–457. Society for Industrial and Applied Mathematics, 2001.

[OPJM10] M. Osadchy, B. Pinkas, A. Jarrous und B. Moskovich. SCiFI - A System for Secure Face Identification. In *IEEE Symposium on Security & Privacy (S&P'10)*, Seiten 239–254. IEEE, 2010.

[Pai99] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology – EUROCRYPT'99*, *LNCS*, Band 1592, Seiten 223–238. Springer, 1999.

[PSSW09] B. Pinkas, T. Schneider, N. P. Smart und S. C. Williams. Secure Two-Party Computation is Practical. In *Advances in Cryptology – ASIACRYPT'09*, *LNCS*, Band 5912, Seiten 250–267. Springer, 2009.

[SV10] N. P. Smart und F. Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In *Public Key Cryptography (PKC'10)*, *LNCS*, Band 6056, Seiten 420–443. Springer, 2010.

[SYY99] T. Sander, A. Young und M. Yung. Non-Interactive CryptoComputing for NC^1. In *IEEE Symposium on Foundations of Computer Science (FOCS'99)*, Seiten 554–566. IEEE, 1999.

[TWBT95] S. Todd, M. Woodward, C. Bolton-Smith und H. Tunstall-Pedoe. An investigation of the relationship between antioxidant vitamin intake and coronary heart disease in men and women using discriminant analysis. *Journal of Clinical Epidemiology*, 48(2):297–305, 1995.

[Yao86] A. C. Yao. How to Generate and Exchange Secrets. In *IEEE Symposium on Foundations of Computer Science (FOCS'86)*, Seiten 162–167. IEEE, 1986.

A Oblivious Transfer

Oblivious Transfer (OT) ist ein Protokoll, dessen Eingaben ein Bit b von $\mathcal{C}$ und zwei Nachrichten s^0 und s^1 von $\mathcal{S}$ sind. OT garantiert, dass $\mathcal{C}$ ausschließlich die gewählte Nachricht s^b erhält und keinerlei Information über s^{1-b}, während $\mathcal{S}$ die Wahl b nicht lernt.

Im Kontext des in §2.2.1 beschriebenen Protokolls von Yao wird für jedes Eingabebit x_i von $\mathcal{C}$ ein OT Protokoll ausgeführt, mit dem $\mathcal{C}$ den zugehörigen Schlüssel $\widetilde{x}_i$ erhält, ohne dass $\mathcal{S}$ den Wert x_i lernt: Die Eingabe von $\mathcal{C}$ ist hierbei das Auswahlbit $b = x_i$ und die Eingabe von $\mathcal{S}$ sind die beiden zugehörigen Schlüssel $s^0 = \widetilde{x}_i^0$ und $s^1 = \widetilde{x}_i^1$. Als Ausgabe erhält $\mathcal{C}$ den Wert $s^b = \widetilde{x}_i^{x_i} = \widetilde{x}_i$.

Durch Kombination verschiedener Optimierungen kann OT sehr günstig implementiert werden (für Details s. [KSS10]): Die Konstruktion von Beaver [Bea95] erlaubt es, alle OTs in der Setup Phase vor zu berechnen, so dass die Online Phase im Wesentlichen aus dem Versenden zweier Nachrichten besteht. Mit der Konstruktion von Ishai et al. [IKNP03] kann die Komplexität der Setup Phase weiter verringert werden, indem eine beliebige Anzahl paralleler OTs auf eine konstante Anzahl von OTs reduziert wird. Die verbleibende konstante Anzahl OTs in der Setup Phase kann mit effizienten OT Protokollen über elliptischen Kurven implementiert werden, z.B. [NP01].

B EKG Klassifikation in TASTYL

```
def protocol(c, s):
    L = 24
    N = 15
    D = 6

    # Eingaben
    c.x = SignedVec(bitlen=L, dim=N)
    s.A = SignedVec(bitlen=L, dim=(D, N))

    # Berechne y = Ax auf verschluesselten Daten
    s.hx <<= HomomorphicVec(val=c.x)
    s.hy = HomomorphicVec(bitlen=L, dim=D)
    for i in xrange(D):
        s.hy[i] = s.A[i].dot(s.hx)

    # Berechne y[i] > 0 mit verschluesselter Funktion
    c.gy <<= GarbledVec(val=s.hy)
    c.gs = GarbledVec(bitlen=1, dim=D)
    for i in xrange(D):
        c.gs[i] = c.gy[i] > 0

    # Berechne Entscheidungsdiagramm
    c.VF = Unsigned(val=(not c.gs[0]) and (not c.gs[2]))
    c.VT = Unsigned(val=(not c.gs[0]) and c.gs[2])
    c.SVT = Unsigned(val=c.gs[0] and (not c.gs[2]))
    c.gt = c.gs[0] and c.gs[1]
    c.PVC = Unsigned(val=c.gt and (not c.gs[3]) and (not c.gs[4]))
    c.APC = Unsigned(val=c.gt and c.gs[3] and (not c.gs[5]))

    # Ausgabe
    if c.VF:
        c.output("Ventricular Fibrillation (VF)")
    elif c.VT:
        c.output("Ventricular Tachycardia (VT)")
    elif c.SVT:
        c.output("SupraVentricular Tachycardia (SVT)")
    elif c.PVC:
        c.output("Premature Ventricular Contraction (PVC)")
    elif c.APC:
        c.output("Atrial Premature Contraction (APC)")
    else:
        c.output("Normal Sinus Rhythm (NSR)")
```

RetrospeGKtive der bekannten sicherheitstechnischen Problematiken bei der Einführung der elektronischen Gesundheitskarte und der Telematikinfrastruktur in Deutschland.

Ali Sunyaev[1], Stefan Knipl[1], Sebastian Dünnebeil[2],
Jan-Marco Leimeister[3], Helmut Krcmar[2]

[1] Wirtschafts- und Sozialwissenschaftliche Fakultät
Universität zu Köln
Deutschland
{sunyaev,knipl}@wiso.uni-koeln.de

[2] Fakultät für Informatik
Technische Universität München
Deutschland
{duennebe,krcmar}@in.tum.de

[3] Fakultät für Wirtschaftswissenschaften
Universität Kassel
Deutschland
leimeister@uni-kassel.de

Abstract: Die Sicherheitsmechanismen und das Sicherheitskonzept der Telematikinfrastruktur sind ein wesentlicher Bestandteil der veröffentlichten technischen Spezifikationen zur Einführung der elektronischen Gesundheitskarte in Deutschland. Für eine zuverlässige Handhabung der Gesundheitstelematik ist die Qualität der eingesetzten Sicherheitstechnik essentiell. Dieser Beitrag überprüft mögliche Sicherheitsproblematiken rund um die elektronische Gesundheitskarte, die in früheren Analysen der gematik-Spezifikationen festgestellt worden waren. Hierzu untersuchen wir ebenfalls, inwiefern und ob die Gesellschaft für Telematikanwendungen der Gesundheitskarte mbH (gematik) diese kommunizierten Fragestellungen in den aktuellen Spezifikationen der Telematikinfrastruktur berücksichtigt hat.

1 Einleitung

Die Einführung der Telematikinfrastruktur (TI) ist eines der weltweit größten Projekte im Informationstechnik-Bereich. Die Verwaltung von sensiblen medizinischen Daten aller Bürger der Bundesrepublik hat bereits zum jetzigen Zeitpunkt reges Interesse von Seiten vieler unterschiedlicher Interessensgruppen geweckt. Viele daraus hervorgegangene

Publikationen haben sich zumindest in Teilen mit der Sicherheit der medizinischen Daten beschäftigt. In diesem Beitrag werden exemplarisch einige der in [Su09b], [Su09c], [HSK08], [Kn09], [SLK10] und [Ma08] beschriebenen Bedenken betrachtet. Außerdem wird überprüft, ob diese Kritiken sinnvoll waren und wenn ja, ob sie in den aktuellsten Spezifikationen der gematik zu Änderungen geführt haben.

2 Ergebnisse

Trotz mancher genereller Bedenken gegenüber der elektronischen Gesundheitskarte und der Telematikinfrastruktur, welche durchaus diskutabel sein können, soll es in diesem Beitrag im Wesentlichen um konkrete Bedenken bezüglich der Sicherheit der TI gehen. So befürchtet beispielsweise [Ma08] Gesetzesänderungen nach der Einführung, die die Sicherheitsvorschriften zugunsten des Staates aufweichen könnten, sowie eine schlechte Kosten-Nutzen-Bilanz. Diese müssen natürlich auch berücksichtigt werden, sind aber aufgrund der anderen Thematik nicht Gegenstand dieses Beitrags.
Im Folgenden wird mehrere Male auf die von der Projektgruppe bit4Health erarbeiteten Sicherheitsanforderungen zur Einführung der Gesundheitskarte [NBB04] hingewiesen. Dieses Konsortium bestand aus Fachleuten von SAP, IBM Deutschland, dem Fraunhofer-Institut für Arbeitswirtschaft und Organisation, der InterComponentWare AG und der Sagem Orga (damals: ORGA Kartensysteme). Es wurde von der Bundesregierung beauftragt, eine herstellerneutrale Rahmenarchitektur und Sicherheitsinfrastruktur für ein vernetztes Gesundheitswesen zu entwerfen [In06]. Die daraus entstandenen Dokumente bilden die Basis, auf der die gematik eine konkrete und detaillierte Ausarbeitung aller Einzelaspekte erstellt hat.

2.1 Lichtbild

[Kn09] sieht ein Problem in der Handhabung der Lichtbilder, die auf eGK und HBA aufgebracht werden. Diese sollen zur visuellen Identifikation der Versicherten durch das medizinische Personal dienen und auf diese Weise Missbrauchsmöglichkeiten einschränken.
Den Autoren ist aber keine Rechtsgrundlage bekannt, welche die Versicherten zur Abgabe eines Lichtbilds verpflichten würde. Nach bisherigen Erfahrungen der Kartenherausgeber wird dies deshalb von einem hohen Prozentsatz der Versicherten auch nicht getan[1]. Außerdem werden die Bilder weder von den Kostenträgern selbst, bzw. durch von Ihnen beauftragte Fotografen, angefertigt. Auch wird auf keine andere Art und Weise verifiziert, dass ein Lichtbild die Versicherte oder den Versicherten selbst zeigt. Dadurch können Kostenträger nicht sicherstellen, dass die Lichtbilder den Anforderungen genügen, welche etwa an ein Lichtbild für einen deutschen Reisepass gestellt werden. Im Allgemeinen ist hier zwar kaum Mißbrauch zu erwarten. Dennoch sollte am besten durch den Gesetzgeber eine Regelung geschaffen werden, die für Klarheit sorgt.

[1] vgl. etwa [Sy07]

2.2 Verletzung der Informationshoheit der Versicherten

[Ma08, S.88f.] sieht in [NBB04] die Informationshoheit der Versicherten verletzt. Einerseits wird deutlich gemacht, dass Versicherte selbst entscheiden können müssen, wer Einblick in ihre Daten nehmen kann [NBB04, S.19], andererseits wird die Möglichkeit eingeräumt, Versichertendaten "in Ausnahmefällen" direkt den Gesetzlichen Krankenversicherungen zur Verfügung zu stellen [NBB04, S.19]. Hier wird weder gefordert, dass die Versicherten darüber informiert werden müssen, noch werden die Umstände, die dazu vorliegen müssen ("Schadensersatzansprüche gegenüber Dritten") genauer definiert. Ebenso wird die Anforderung aufgestellt, dass es eine Möglichkeit geben muss, pseudonymisierte Daten der Versicherten periodenübergreifend auszuwerten [NBB04, S.38]. Es steht zu befürchten, dass bei Vorliegen größerer Datenmengen über eine Versicherte oder einen Versicherten mit verhältnismäßig geringem Aufwand eine Depseudonymisierung durchgeführt werden kann.
In den Dokumenten der gematik wird die Datenhoheit der Versicherten betont. Es gibt keine Ausnahmefälle, in denen von diesem Prinzip abgewichen werden kann. Auch in [oV88] wird ausdrücklich betont, dass der Zugriff auf die Daten nur mit dem Einverständnis der Versicherten durchgeführt werden darf. Wird dieses nicht erteilt, so darf keine Benachteiligung erfolgen [oV88, 291a, Abs. 5 und 8]. Aufgrund der Pseudonymisierung gilt, dass für eine periodenübergreifende Auswertung die Mitwirkung der Kartenherausgeber nötig wäre, welche die Pseudonyme vergeben. Für eine über mehr als ein Pseudonym hinausgehende Begutachtung (also mehr als eine eGK, in der Regel dementsprechend 5 Jahre) müssten diese Daten von ihnen herausgegeben werden.
Da der Kartenherausgeber keinerlei Zugriff auf die in der TI gespeicherten Daten haben wird, kann er eine solche Auswertung keinesfalls selbst durchführen. Selbst dann könnten jedoch keine umfassenden Erkenntnisse gewonnen werden. Alle Datenobjekte sind mit dem Schlüssel der oder des Versicherten verschlüsselt sind und können nur von ihnen selbst entschlüsselt werden. Dies ändert sich nur dann, wenn Schlüsseltreuhänder eingeschaltet würden. Zum Einen erscheint der Aufwand dafür enorm hoch und zum Anderen erscheint es fraglich, ob dies den Vorgaben von [oV90] entspräche. Diese Kritik hat damit ihren Widerhall bei der Entstehung der durch die gematik aufgestellten Rahmenbedingungen gefunden.

2.3 Zusammenführung von administrativen und medizinischen Daten

[HSK08] fanden den Widerspruch, dass einerseits bei jedem Fachdienst zuständige Datenschutzbeauftragte einen Schlüssel zur Zusammenführung administrativer und medizinischer Daten besitzen sollen [NBB04, S.20]. Andererseits darf die Sicherheit nicht auf dem Vertrauen in einzelne Personen beruhen [NBB04, S.30]. Dies soll effektiv verhindern, dass etwa Erpressung oder Bestechung zu einer ernsthaften Gefährdung der medizinischen Daten führen könnten. Diese beiden Vorgaben widersprechen sich gegenseitig.
In den aktuellen Dokumenten der gematik wird diese Möglichkeit nicht mehr genannt. So wird definiert, dass "eine Zusammenführung der Metadaten von zwei Fachdiensten (...)

NICHT möglich sein [darf]" [ge08b, S.172]. Das schließt zwar die Zusammenführung bei nur einem einzelnen Fachdienst nicht aus, weist aber doch konzeptionell in die entgegengesetzte Richtung.
Das Treuhänderkonzept [ge09a, S.60] stellt sicher, dass Daten, die mit dem Schlüssel der oder des Versicherten chiffriert sind, auch nur von dieser oder diesem selbst oder unter Einbeziehung mehrerer Personen entschlüsselt werden können. Damit ist die Prämisse, dass Sicherheit nicht von einzelnen Personen abhängen darf, wieder weitgehend hergestellt. Könnte eine Zusammenführung von administrativen und medizinischen Daten stattfinden, so wäre es möglich, daraus Rückschlüsse zu ziehen. So könnten etwa Schätzungen über die Häufigkeit der Inanspruchnahme medizinischer Leistungen aufgestellt werden. Die eigentlichen medizinischen Daten könnten aber zunächst ohne Hinzuziehen der Treuhänder nicht entschlüsselt bzw. ausgewertet werden. Die ursprünglichen Bedenken können somit als geklärt bezeichnet werden.

2.4 Falsche Annahmen im Zonenkonzept

Durch [HSK08] wird eine zentrale Prämisse des von der gematik aufgestellten Zonenkonzepts in Frage gestellt, welche besagt, dass sich Bedrohungen nicht über die Zonengrenzen hinweg ausbreiten können, da hier die nötigen Vorkehrungen getroffen würden, um dies zu verhindern[2]. [HSK08] konstruieren einen Angriff, in dem eine Innentäterin oder ein Innentäter sich mittels eines HBAs von Zone 1 aus einen unbefugten Zugang zur TI verschafft. Dort wird der HBA erfolgreich authentifiziert. Auf dieser Basis kann nun ein Angriff auf die inneren Zonen durchgeführt werden. Damit wird dies als falsche Annahme dargestellt.
Tatsächlich wird die Bedrohungsquelle nicht wie proklamiert innerhalb der Grenzen von Zone 1 gehalten. Man sollte sich jedoch bewusst sein, dass in diesem Fall nicht von einer isolierten Bedrohung, sondern von zwei sich gegenseitig bedingenden gesprochen werden muss. Zum Einen muss eine Verletzung der Sicherheitsbedingungen in Zone 1 auftreten, also etwa das Entwenden eines HBAs und der zugehörigen PINs durch Innentäter[3]. Zum Anderen kann dann ausgehend auf dieser Basis in einem weiteren Schritt ein Angriff in den dadurch erreichten inneren Zonen durchgeführt werden.
Da unter bestimmten Umständen an den Zonengrenzen unbefugt eine erfolgreiche Authentifizierung erreicht werden kann, ist die durch die gematik angestellte Annahme tatsächlich etwas zu vollmundig. Im Kern kann sie dennoch als gültig angesehen werden. Eine Bedrohungsquelle, die ihren Ursprung in einer anderen Zone hat, hat weitaus eingeschränktere Möglichkeiten für darauf aufbauende Angriffe, als dies für Bedrohungsquellen in der gleichen Zone gilt.

[2] vgl. [ge08b, S.32]
[3] vgl. z.B. [Su08b] für verschiedene Angreiferprofile

2.5 Zeitabstände zwischen Anpassungen der Sicherheitsstandards

Die gematik erlegt sich selbst die Aufgabe auf, regelmäßig die Mindeststandards des Sicherheitskonzepts zu überprüfen, ob diese der jeweils gültigen Bedrohungslage weiterhin entsprechen. [SLK10] kritisieren hierbei lediglich, dass die Standards nur einmal jährlich überprüft und angepasst werden müssen[4]. Jährlich werden eine große Anzahl an Schwachstellen in Protokollen, Diensten und Chiffrierungsverfahren bekannt. Dies betrifft damit auch Komponenten und Bestandteile der TI. Unter diesem Gesichtspunkt und angesichts der zu schützenden Daten mutet die Mindesterfüllung der Empfehlungen der IT-Grundschutzkataloge des BSI von nur einer Überprüfung pro Jahr [Bs09, S.1] zumindest befremdlich an.
Auch in der aktuellsten Version des gematik-Sicherheitskonzeptes wurde dieser Zeitraum nicht verändert [ge08b, S.43ff.]. Eine Verringerung dieses Zeitraums müsste nicht zwingend zu einer Verbesserung der Sicherheit führen. Eine Aktualisierung des Sicherheitskonzepts darf auch außerhalb dieser regelmäßigen Überprüfung jederzeit stattfinden, wenn Probleme entdeckt werden. Eine häufigere Überprüfung dieses Konzepts würde jedoch zu einer regelmäßigen Auseinandersetzung mit neuen Bedrohungsquellen und Sicherheitsproblemen zwingen und erscheint damit dennoch empfehlenswert.

2.6 Fehlende Server-Authentifizierung

In ihrer Arbeit fanden [SLK10] eine fragwürdige Einschränkung der gegenseitigen Authentifizierung der Komponenten der TI. So wird prinzipiell durch den Konnektor keine Authentifizierung der Server des Zeitdiensts durchgeführt, “da die Systeme im VPN der Telematikinfrastruktur als sicher angesehen werden” [ge06, S.60]. Tatsächlich sind die inneren Zonen der TI aufgrund ihrer verhältnismäßig großen Zentralität deutlich einfacher zu kontrollieren, als dies für die dezentralen Systeme in Zone 1 gilt. Darüber hinaus gibt es speziell bei den Zeitservern eine große Redundanz, [ge08b, S.139] so dass für einen erfolgreichen, verwertbaren Angriff die Kompromittierung eines einzigen Servers nicht ausreicht [ge08b, S139f.]. Dies sollte allerdings nicht darüber hinwegtäuschen, dass die aufgestellte Prämisse deutlich weitgehender ist, als angemessen wäre. Des Weiteren wäre eine Authentifizierung an dieser Stelle mit keinerlei großen Nachteilen verbunden, wie erhöhter Prozesslaufzeit oder erhöhtem Datentransfervolumen, und mit nur geringem Aufwand realisierbar. Insofern ist es nicht nachzuvollziehen, warum diese These auch in den aktuellen Dokumenten der gematik aufrecht erhalten wird [ge09b, S.122]. Sinnvollerweise sollte auch hier zukünftig eine zumindest einseitige Authentifizierung des Servers durchgeführt werden.

[4] vgl. auch [Su09c]

2.7 Keine Verwendung quelloffener Software

Die gematik beteuert, dass "Security by Obscurity [...] nicht als zielführend [...] angesehen" wird [ge08d, S.80]. [HSK08] argumentieren, dass eben dieses Prinzip eingesetzt wird, indem erklärt wird, dass der Großteil der verwendeten Software einen hohen Schutzbedarf genießt, da es sich bei ihr um Firmengeheimnisse handeln würde. Etwas abgeschwächt wird dies dadurch, dass die Quellen den Mitarbeitern des BSI und der gematik zugänglich sein werden und darüber hinaus durch diese geprüft werden müssen.
Dies verhindert dennoch, dass freie Sicherheitsexperten den Quellcode auf potentielle Probleme überprüfen. Diese Regelung besteht nach wie vor in den aktuellen Dokumenten der gematik. Rein rechtlich sollte es durchaus möglich sein, die Zulassung einer neuen Komponente an die Offenlegung des Quellcodes der Software zu koppeln. Eine Offenlegung brächte zwar das Problem mit sich, dass Bedrohungsquellen ohne Aufwand an Material kommen. Sie könnten daher gezielt nach Schwachstellen suchen und diese anschließend möglicherweise ausnutzen. Basierend auf dem Prinzip von Kerckhoff [Ke83] muss aber immer davon ausgegangen werden, dass das System einem ernsthaften Angreifer bis ins Detail bekannt ist. Die Vorteile der Offenlegung übertreffen aus sicherheitstechnischer Sicht damit mit großer Wahrscheinlichkeit die möglichen Nachteile [MN03].

2.8 Unzureichende Definition der Sperrlistenverwaltung

[SLK10] kritisieren, dass zwar das notwendige Vorhandensein von Sperrinformationen durch die gematik vorgeschrieben wurde, nicht jedoch näher definiert worden ist, wie diese bereitzustellen sind. Auch ist nicht klar, ob Authentifizierung zum Zugriff erfolgen muss, wo die zuständigen Server angesiedelt werden und welche Funktionalitäten sie bieten.
Hier wurde seitens der gematik in einigen Punkten nachgebessert. So ist nun klar, dass die Integrität der Sperrlisten sicher gestellt werden muss und für den Anwender jederzeit nachprüfbar sein werden [ge08c, S.77]. Ist dies nicht gewährleistet, werden diese nicht genutzt, um die Gültigkeit eines Zertifikats zu überprüfen [ge08c, S.78]. Damit wird verhindert, dass ein unbefugter Schreibzugriff auf die Sperrliste die beliebige Sperrung und Entsperrung von Zertifikaten ermöglicht.
Darüber hinaus wird nur noch in geringen Umfang mit Sperrlisten gearbeitet. Diese können zwar ("in Ausnahmefällen" [ge09a, S.145]) eingesetzt werden, üblicherweise wird aber OCSP (Onlince Certificate Status Protocol) benutzt, um die Gültigkeit von Zertifikaten zu überprüfen; diese Möglichkeit muss immer gegeben sein [ge09a, S.45]. Alles Weitere zur Funktionsweise der Gültigkeitsprüfung wird durch die Server-Betreiber (Trusted Service Provider, TSP) festgeschrieben.[5] Damit können die ursprünglich aufgestellten Kritikpunkte als im Wesentlichen geklärt gelten.

[5] vgl. [ge08a, S.39f.]

2.9 Unzureichende Spezifikation des Trusted Viewers

[HSK08] sind der Meinung in der Spezifikation der vertrauenswürdigen Anzeigekomponente, dem Trusted Viewer (mittlerweile extended Trusted Viewer, xTV), eine Inkonsistenz und eine Sicherheitslücke entdeckt zu haben. Der Trusted Viewer stellt sicher, dass tatsächlich nur Inhalte signiert werden, von denen die Versicherten und Leistungserbringer glauben, sie zu signieren. In den Konnektorspezifikationen wurde nur ungenau darauf eingegangen, wie und in welcher Form der Trusted Viewer realisiert werden sollte. So konnte dieser Service als direkter Bestandteil des Konnektors oder als eigene Hardwarekomponente oder gar als Funktionalität der angeschlossenen Clients realisiert werden. Darüber hinaus musste der Konnektor eine Schnittstelle zum Trusted Viewer enthalten, auch wenn der Trusted Viewer Teil des Konnektor selbst sein sollte. Außerdem war es aus Sicherheitssicht als kritisch einzustufen, dass der Trusted Viewer keine eigene Identität erhalten sollte, also nicht durch den Konnektor oder die Primärsysteme authentifiziert werden konnte.
In diesen Punkten wurden mittlerweile durch die gematik mehrere Änderungen durchgeführt. Die Wahlfreiheit, wie der Trusted Viewer realisiert werden kann, wurde eingeschränkt[6]. Damit wird der xTV in zwei Bestandteile zerfallen: einen, welcher im Konnektor beheimatet sein wird und eine Komponente für das Primärsystem [ge09b, S.93]. Bei letzterer ist wie bisher nicht vorgeschrieben, ob dies in Form einer Softwarelösung auf dem Primärsystem durchgeführt werden soll oder in Form einer angeschlossenen Hardwarekomponente geschieht, jedoch ist immerhin die konkrete Ausgestaltung der Schnittstelle zwischen diesen beiden Komponenten definiert [ge09b, S.307].
Dies ist eine Verbesserung gegenüber dem Zustand, den [HSK08] kritisiert haben, da es somit weitestgehend möglich ist, den xTV-Dienst aus dem Praxisverwaltungssystem (PVS) heraus über eine fest definierte Schnittstelle zu nutzen. Eine Festlegung darauf, die Komponente des Primärsystems entweder als Softwarelösung oder als Hardwarelösung zu fordern, hätte jedoch verhindert, dass nun jeder Anbieter eines PVS für jeden zertifizierten Konnektor beide Möglichkeiten unterstützen muss. Dies erhöht die Komplexität und damit auch die Fehleranfälligkeit deutlich. Diese hätte damit ohne jeglichen Sicherheitsverlust des Systems reduziert werden können. Letztlich hätte dies zu weniger Problemen bei Inbetriebnahme und laufendem Betrieb geführt und wäre damit der generellen Akzeptanz der eGK zuträglich gewesen. Dennoch ist diese Präzisierung gegenüber der von [HSK08] betrachteten Version zu begrüßen. Positiv muss insbesondere erwähnt werden, dass mittlerweile auch der xTV-Komponente im Konnektor eine eigene Identität zugedacht wird [ge09a, S.173]. Damit können Man-in-the-Middle-Angriffe und Spoofing-Attacken weitgehend ausgeschlossen werden, da dadurch alle Gegenstellen die Integrität des xTVs überprüfen können.

[6] vgl. [ge09b, S.38]

2.10 Keine kryptographische Identität der Primärsysteme

[SLK10] kritisieren, dass kein eigenes Identitätszertifikat für die mit dem Konnektor verbundenen Primärsysteme vorgesehen ist. Dies führt dazu, dass zwar die Primärsysteme den Konnektor einseitig authentifizieren können. Sie können aber nicht sicherstellen, dass kein Man-in-the-Middle-Angriff stattfinden kann. Der Konnektor kann zur Zeit nicht überprüfen, ob eine Verbindung mit einem Primärsystem direkt aufgebaut wurde oder unter Einbeziehung eines Dritten. Dieser könnte sich dabei dem Konnektor gegenüber als Primärsystem und dem Primärsystem gegenüber als Konnektor ausgeben. Daher könnten Anfragen des Primärsystems an den Konnektor verändert oder abgehört werden[7].
Hieran hat sich auch in den aktuellen Spezifikationen nichts geändert. Dieses Problem ließe sich auch nur durch das Hinzufügen kryptographischer Identitäten für die Primärsysteme lösen. Hier wäre zu überlegen, ob dies hardwaretechnisch gelöst werden könnte. Denkbar wäre der Einsatz eines Dongles, eines Kopierschutzsteckers, der die Identität enthält und meist an den USB-Port eines Clients angesteckt wird. Da ansonsten die kryptographischen Identitäten manuell auf jedem Client eingerichtet werden müssten, wäre dies eine einfach umzusetzende Lösung, die auch ohne größeres technisches Verständnis durchgeführt werden kann. Da jedoch der Schutz vor Diebstahl eines Dongles in einer Arzt-Praxis vermutlich nicht ausreichend gewährleistet werden kann, ist dies keine wirklich praktikable Lösung.
Der größte Teil der ablaufenden Kommunikation zwischen Primärsystemen und Konnektor wird bereits mit den geheimen Schlüsseln von HBA und eGKs verschlüsselt. Daher können auch ohne kryptographische Identität der Primärsysteme zunächst nur wenige konkrete Daten abgehört werden. Gefälschte Anfragen an den Konnektor sind zwar in der Lage, das Ziel der Verfügbarkeit zu gefährden, können aber nicht die Integrität oder Authentizität des Datenaustauschs mit der TI verletzen. Eine eigene Identität der Clients ist damit wünschenswert, jedoch nur mit hohem Aufwand realisierbar.

2.11 Ungenügende Vorgaben und Richtlinien für Primärsysteme

[HSK08] beklagen, dass gerade im Hinblick auf die Primärsystemen keine Vorgaben oder zumindest Richtlinien seitens der gematik gibt. Speziell, da dort alle medizinischen Daten erstellt und zum großen Teil gespeichert werden, geht damit eigentlich ein sehr hoher Schutzbedarf einher. Derartige Vorschriften oder Richtlinien findet man auch in den aktuellen Versionen der Spezifikationen nicht. Gerade die Primärsysteme in der Zone 1 stellen die am unmittelbarsten angreifbaren Systeme dar. Zugleich besitzen die hier vertretenen Akteure die geringste Fachkompetenz im Bereich der IT-Sicherheit. Dennoch wurden bisher lediglich von Dritten Vorschläge zur Verbesserung der IT-Sicherheit erarbeitet[8]. Da jedoch Richtlinien und besonders Vorschriften von offizieller Seite eine deutlich höhere Beachtung und Umsetzung finden, sollte in diesem Bereich auch noch eine offizielle Veröffentlichung durch die gematik erfolgen.

[7] vgl. auch [Su09b] und [Su09a]
[8] vgl. z.B. [Su08a], [Ka10]

3 Ausblick

Die vorgestellten Bedenken können rechtzeitig vor der flächendeckenden Einführung der eGK in Deutschland behoben werden, bzw. wurden bereits größtenteils behoben. Bei weiteren, hier nicht näher aufgeführten, Bedenken[9] sehen die Autoren von einer expliziten Behauptung ab.
Die gematik hat vorwiegend widerspruchsarme Konzepte zum Schutz der Sicherheitsziele entworfen. Diese Dokumente sind öffentlich einsehbar und stellen damit eine Einladung an alle interessierten Parteien dar, an der Weiterntwicklung der Konzepte teilzunehmen. Nur so kann kann ein bestmöglichen Schutz gegen Bedrohungsquellen angestrebt werden. Dass sich dies auch in einem gewissen Rahmen auszahlt, lässt sich am Beispiel dieser Arbeit beobachten.

Literatur

[Bs09] BSI (2009): IT-Grundschutzkataloge - 11. Ergänzungslieferung. M 2.199 Aufrechterhaltung der Informationssicherheit. In: https://www.bsi.bund.de/cae/servlet/contentblob/478422/publicationFile/55552/massnahmen.zip, zugegriffen am 10. März 2010.

[ge06] gematik (2006): Konnektorspezifikation. Teil 1 - Allgemeine Funktionen und Schnittstellen des Konnektors. Version 0.6.0. o.O., 2006.

[ge08a] gematik (2008a): Certificate Policy. Gemeinsame Zertifizierungs-Richtlinie für Teilnehmer der gematik-TSL zur Herausgabe von X.509-ENC/AUTH/OSIG-Zertifikaten. Version 1.3.0, o.O, 2008.

[ge08b] gematik (2008b): Übergreifendes Sicherheitskonzept der Telematikinfrastruktur. Version 2.4.0, o.O., 2008.

[ge08c] gematik (2008c): Übergreifendes Sicherheitskonzept der Telematikinfrastruktur - Anhang B - Sicherheitsanforderungen. Version 2.4.0, o.O., 2008.

[ge08d] gematik (2008d): Übergreifendes Sicherheitskonzept der Telematikinfrastruktur - Anhang C - Schutzbedarfsanalyse. Version 2.4.0, o.O, 2008.

[ge09a] gematik (2009a): Gesamtarchitektur. Version 1.7.0, o.O., 2009.

[ge09b] gematik (2009b): Konnektorspezifikation. Version 3.0.0, o.O, 2009.

[HSK08] Huber, M.; Sunyaev, A.; Krcmar, H. (2008): Security Analysis of the Health Care Telematics Infrastructure in Germany. In: ICEIS 2008 - Proceedings of the Tenth International Conference on Enterprise Information Systems, Vol. ISAS-2, pp. 144-153. Barcelona, Spain.

[In06] Initiative D21 (2006): Die neue Gesundheitskarte - ein Themenservice. 03/06. In: http://www.old.initiative21.de/fileadmin/files/themenservice/themenservice_3/NEU_Themenservice_03_06.pdf, zugegriffen am 22. März 2010.

[9] z.B. bei der an sich nicht schädlichen Möglichkeit, den Konnektor zum Signieren und Verschlüsseln zu mißbrauchen, vgl. [Su10] oder der Frage der langfristigen Vertraulichkeit verschlüsselter medizinischer Daten, vgl. [SLK09]

[Ka10] Kassenärztliche Bundesvereinigung (2010): Anforderungen an Hard- und Software in der Praxis. Hinweise zum Datenschutz. Ein Leitfaden für Ärzte und Psychotherapeuten. In: http://daris.kbv.de/daris/link.asp?ID=1003760425, Berlin 2010, zugegriffen am 02. März 2010.

[Ke83] Kerckhoffs A. (1883): La cryptographie militaire. In: Journal des sciences militaires, vol. IX, S. 5-38, Januar 1883, S. 161-191, Februar 1883.

[Kn09] Knüttel, A. (2009): Probleme und Lösungsansätze zur eGK/HBA aus der Sicht eines Testkrankenhauses und der NKG. In: conhIT 2009 - Satellitenveranstaltung GMDS/BVMI, Workshop 2: GMDS-Projektgruppe "Einführung von eGK und HBA in Krankenhäusern".

[Ma08] Maus, T. (2008): Risiken + Nebenwirkungen. Der Beipackzettel zur Gesundheitskarte. In: http://www.medi-deutschland.de/datei.php?id=1080, zugegriffen am 03. Februar 2010.

[MN03] Mercuri, R.T.; Neumann, P.G.(2003): Security by Obscurity. In: Communications of the ACM, Vol.46, No. 11, S.160.

[NBB04] Neeb, J.; Bunz, H.; Biltzinger, P. (2004): Erarbeitung einer Strategie zur Einführung der Gesundheitskarte. Sicherheitsanforderungen. In: Projektgruppe bit4Health, http://www.inso.tuwien.ac.at/uploads/media/b4h_sicherheitsanforderungen_v1-1.pdf, zugegriffen am 01. Februar 2010.

[oV88] o.V. (1988): Sozialgesetzbuch (SGB) Fünftes Buch (V) - Gesetzliche Krankenversicherung -(Artikel 1 des Gesetzes v. 20. Dezember 1988, BGBl. I S. 2477). In: http://bundesrecht.juris.de/sgb_5/BJNR024820988.html, zugegriffen am 08.Februar 2010.

[oV90] o.V. (1990): Bundesdatenschutzgesetz in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), das zuletzt durch Artikel 1 des Gesetzes vom 14. August 2009 (BGBl. I S. 2814) geändert worden ist. In: http://www.gesetze-im-internet.de/bdsg_1990/BJNR029550990.html, zugegriffen am 18. Februar 2010.

[Su08a] Sunyaev, A; von Beck, J.; Jedamzik, S.; Krcmar, H. (2008): IT-Sicherheitsrichtlinien für eine sichere Arztpraxis. 1. Auflage, Shaker Verlag, Aachen 2008.

[Su08b] Sunyaev, A.; Huber M.J.; Mauro, C.; Leimeister J.M.; Krcmar, H. (2008): Bewertung und Klassifikation von Bedrohungen im Umfeld der elektronischen Gesundheitskarte. In: Proceedings of Informatik 2008 - Beherrschbare Systeme - dank Informatik, Band 1, Hrsg: GI - Gesellschaft für Informatik, GI Lecture Notes in Informatics, München, S.65-70.

[Su09a] Sunyaev, A.; Dünnebeil, S.; Mauro, C.; Leimeister, J.M.; Krcmar, H. (2009): Sicherheitsbetrachtung der Primärsysteme in der Deutschen Gesundheitstelematik. In: Proceedings of 54. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie (GMDS). Essen, 07.-10.09.2009.

[Su09b] Sunyaev, A.; Kaletsch, A.; Mauro, C.; Krcmar, H. (2009): Security Analysis of the German electronic Health Card's Peripheral Parts. In: ICEIS 2009 - Proceedings of the 11th International Conference on Enterprise Information Systems. Milan, Italy, 6-10 May 2009. Volume ISAS, pp. 19-26.

[Su09c] Sunyaev, A.; Leimeister, J.M.; Krcmar, H. (2009): Security Analysis of the German Healthcare Telematics. In: AMCIS 2009 - Proceedings of the 15th Americas Conference on Information Systems. San Francisco, California, 6-9 August 2009. Paper 232.

[Su10] Sunyaev, A.; Kaletsch, A.; Duennebeil, S.; Krcmar, H. (2010): Attack Scenarios for possible Misuse of Peripheral Parts in the German Health Information Infrastructure. In: Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS 2010). Funchal, Madeira - Portugal, 8 - 12 June, 2010. Volume DISI, pp. 229-235.

[SLK09] Sunyaev, A.; Leimeister, J.M.; Krcmar, H. (2009): Telematik im Gesundheitswesen: Sichere Autobahn mit unsicheren Auffahrten? In: Krankenhaus-IT Journal, Nr. 2/2009, S. 46-47, 2009.

[SLK10] Sunyaev, A.; Leimeister, J.M.; Krcmar, H. (2010): Open Security Issues in German Healthcare Telematics. In: Proceedings of the Third International Conference on Health Informatics (HealthInf 2010), January 20-23, 2010, Valencia, Spain, pp. 187-194.

[Sy07] systemform Media Card GmbH (2007): Einfürung der elektronischen Gesundheitskarte (eGK) - Erste Erfahrungen aus der Praxis. In: ECM Tag Fokus Gesundheit 2007, http://www.pentadoc.de/fileadmin/_temp_/documents/ecmtag/20070612_rheingau/Einfuehrung_der_elektronischen_Gesundheitskarte_-_Erste_Erfahrungen_aus_der_Praxis.pdf, zugegriffen am 16. Juli 2010.

Deklarative Sicherheit zur Spezifikation und Implementierung der elektronischen Fallakte

Raik Kuhlisch, Jörg Caumanns

Fraunhofer-Institut für Software- und Systemtechnik ISST
Steinplatz 2
10623 Berlin
raik.kuhlisch@isst.fraunhofer.de
joerg.caumanns@isst.fraunhofer.de

Abstract: Anwendungen zur sektorübergreifenden Kommunikation im Gesundheitswesen stellen für deren Betreiber (z. B. Kliniken) große Investitionen mit oftmals unsicherem Marktpotenzial dar. Es ist daher wesentlich, dass die einmal aufgebauten Dienste für eine Vielzahl von Anwendungen mit- und nachnutzbar sind. In diesem Papier wird am Beispiel der elektronischen Fallakte (eFA) dargestellt, wie Konzepte einer deklarativen Sicherheit dazu beitragen, bestehende Dienste flexibel an sich verändernde Sicherheitsanforderungen anzupassen bzw. parallel in unterschiedlichen Sicherheitskontexten zu nutzen.

1 Motivation

Die elektronische Fallakte (eFA) erlaubt einen fallbezogenen Austausch von medizinischen Daten zwischen Leistungserbringerorganisationen. Hierzu werden bestehende Systeme in Kliniken und Praxen über ein föderiertes Peer-to-Peer Netzwerk von Registern und Speichern virtuell integriert [CBN07]. Der Patient willigt explizit in das Anlegen und Nutzen einer Fallakte ein und berechtigt Ärzte oder Einrichtungen für den Zugriff auf seine Daten. Die durch den besonders schützenswerten Status der ausgetauschten Daten bedingten Anforderungen werden durch ein Datenschutzkonzept und ein übergreifendes Sicherheitskonzept adressiert[1].

Die elektronische Fallakte basiert auf einer serviceorientierten Architektur, die strikt in Sicherheits- und Anwendungsdienste getrennt ist. Verschiedene Schutzmechanismen in der eFA adressieren die hohen Datenschutzanforderungen an die Verarbeitung medizinischer Daten mit Patientenbezug. Adäquate Schutzmechanismen beantworten Fragen wie etwa: Wer darf welchen Dienst nutzen bzw. wer darf welche Daten zuordnen, zusammenführen, lesen oder ändern? Dedizierte Sicherheitsdienste übernehmen die Aufgaben der Identifizierung/Authentifizierung, Pseudonymisierung, sowie Autorisierung von Leistungserbringern [BSI07].

[1] Siehe http://www.fallakte.de für die vollständigen Spezifikationen zur elektronischen Fallakte.

1.1 Deklarative Sicherheit

Sicherheitsfunktionen zur Erfüllung von Anforderungen an die Vertraulichkeit und Authentizität von geschützten Daten folgen oftmals dem gleichen Muster: Eine die geschützte Ressource kapselnde Anwendung erhält einen Dienstaufruf. Der Aufrufende muss authentisiert und anschließend autorisiert werden. Dies führt zum Zugriff auf entsprechende Ressourcen oder eben nicht. Hier liegt es nahe, solche wiederkehrende Aufgaben in *Security Frameworks* auszulagern und zu beschreiben (deklarieren), welche Sicherheitsfunktionen zu aktivieren sind, um einer übergeordneten Sicherheitsstrategie gerecht zu werden. Die Ausübung dieser Sicherheitsfunktionen ist dann Sache des Frameworks. Dieser Schutzziele und Sicherheitsdienste betonende deklarative Sicherheitsansatz steht im Gegensatz zur programmierten Sicherheit, bei der konkrete Sicherheitsmechanismen und –objekte fest an die Implementierung der Anwendungslogik gebunden werden.

Deklarativen Elemente zur Vereinfachung und Flexibilisierung der Umsetzung von Sicherheitsvorgaben im Bereich der Authentisierung und Autorisierung sind mittlerweile Bestandteil verschiedener *Business-Level Frameworks* (z. B. OASIS ebXML Business Processes [ebXMLBP]) und werden auch von J2EE unterstützt [SUN08]. Ziel ist es hierbei immer, die für eine Komponente oder Kommunikationsbeziehung geltenden Sicherheitsziele zu beschreiben und die Auswahl des geeigneten Mechanismus dem Framework zu überlassen.

Im Umfeld von Web Services stellen *WS-Policy* [WSPOL1.5] und der darauf aufbauende *WS-SecurityPolicy* Standard [WSSPOL1.2] Sprachen bereit, um Anforderungen an einen Dienstaufruf zu beschreiben. In sogenannten *Policy Assertions* wird so das Verhalten des Web Service deklarativ gesteuert. Die Implementierung des Web Service kann sich somit auf die Fachlogik konzentrieren und die Durchsetzung der Sicherheitsanforderungen als vorausgesetzt ansehen.

Diese Nutzung deklarativer Sicherheit bietet eine Reihe von Vorteilen:

- Sicherheitsdienste und –mechanismen können ohne Änderungen am Code der Anwendungsdienste weiterentwickelt und an neue Anforderungen angepasst werden

- In einem Framework enthaltene Stubs der Sicherheitsdienste können sehr einfach an bestehenden Anwendungen angebunden werden. Hierdurch wird eine Entkopplung von Sicherheitsdiensten - z. B. für Authentifizierung und Autorisierung – unterstützt [EDOC07].

- Die Kongruenz der Umsetzung von Sicherheit zu ihrer Spezifikation und einer übergeordneten Sicherheitsrichtlinie ist explizit gegeben, d. h. deklarative Sicherheit kann unmittelbar aus dem Sicherheitskonzept abgeleitet werden

- Die umgesetzten Sicherheitsmechanismen und die genutzten Objekte werden an der Schnittselle der Dienste sichtbar und damit überprüfbar.

Ein Nachteil der deklarativen Sicherheit sind die durch eine Interpretation der Deklarationen bedingten Performanzeinbußen. Performanzanalysen der elektronischen Fallakte am Fraunhofer ISST und bei Siemens haben jedoch gezeigt, dass schon durch einfache Mechanismen wie z. B. das Caching von Deklarationen auch bei einer intensiven Nutzung deklarativer Sicherheit eine sehr gute Performanz erzielt werden kann.

1.2 Deklarative Sicherheit am Beispiel der elektronischen Fallakte

Die Sicherheitsdienste der elektronischen Fallakte sind so ausgelegt, dass sie auch für eine Vielzahl weiterer Anwendungen zur Kooperation von Leistungserbringern genutzt werden können. Um dieses Ziel zu erreichen, erfolgt die Kopplung zwischen Sicherheits- und Anwendungsdiensten soweit als möglich deklarativ. D. h. für jede Schnittstelle wird beschrieben, welche Sicherheitsobjekte beim Aufruf bereitzustellen sind und welche der eFA-Sicherheitsdienste diese Objekte bereitstellen. Das daraus abgeleitete Konzept des *Assertion-Chaining* ist in Kapitel 2 dieses Papiers beschrieben.

Die Hauptaufgabe der eFA-Sicherheitsarchitektur ist es, einen Satz authentischer Berechtigungsregeln (*Access Policy)* eines authentisierten Nutzers als Teil des *SOAP Security Headers* bei jedem Aufruf eines Anwendungsdienstes mitzuliefern. Vor jeder geschützten Ressource ist ein *Policy Enforcement Point* (PEP) positioniert, der sicherstellt, dass nur die *Access Policy* erfüllende Zugriffe zugelassen werden. Da *Policy* und Nutzerattribute an der Schnittstelle des Anwendungsdienstes bereitgestellt werden, muss dieser für eine *Policy*-Entscheidung lediglich noch Attribute der angeforderten Ressource beifügen. In Kapitel 3 dieses Papiers wird beschrieben, wie die hierzu genutzten XACML-Policies aufgebaut sind, um ein rollenbasiertes Berechtigungsmanagement durchzusetzen.

Ein wesentliches Element von service-orientierten Sicherheitsarchitekturen sind sog. *Security Token*, die eine Entkopplung von Sicherheitsdiensten erlauben [EDOC07]. Standards wie *WS SecurityPolicy* erlauben eine deklarative Steuerung des Austauschs von *Security Token* (siehe Kapitel 2). Leider gibt es momentan jedoch keinen dedizierten Standard, um den einer Sicherheitsrichtlinie entsprechenden Aufbau eines *Security Token* zu beschreiben, so dass die Ausstellung und Verifizierung der *Token* zumindest in Teilen deklarativ erfolgen könnte. In der Referenzimplementierung der elektronischen Fallakte wird der XACML-Standard genutzt, um die Verifizierung eines *Security Token* am konsumierenden Dienst zu flexibilisieren. Wie hiermit eine deklarative Verifizierung von Elementen einer Sicherheitsrichtlinie umsetzbar ist, wird in Kapitel 4 dieses Papiers beschrieben.

2 Deklarative Sicherheit bei der Schnittstellenspezifikation

Die Sicherheitsarchitektur der elektronischen Fallakte definiert für jeden Anwendungsdienst mittels *WS-Policy* und *WS-SecurityPolicy* welche Sicherheitsnachweise (*Security Assertions,* z. B. kodiert als *Security Token)* notwendig sind, um die Operationen aufrufen zu können. SAML *Assertions* [SAML2.0] kodieren die notwendigen

Authentisierungs- und Autorisierungsinformationen, welche von speziellen *Security Token Services* ausgestellt werden. Hierbei kann ein *Security Token Service* zur Ausstellung eines geforderten Sicherheitsnachweises (*Security Assertion*) die Vorlage eines *Security Token* verlangen, dass in die Zuständigkeit eines anderen *Security Token Service* fällt. Auf diese Weise können Abhängigkeiten in Sicherheitsdiensten (z. B. Autorisierung erfordert Authentifizierung) auf sequentielle Ketten von Sicherheitsnachweisen abgebildet werden (siehe Abbildung 1).

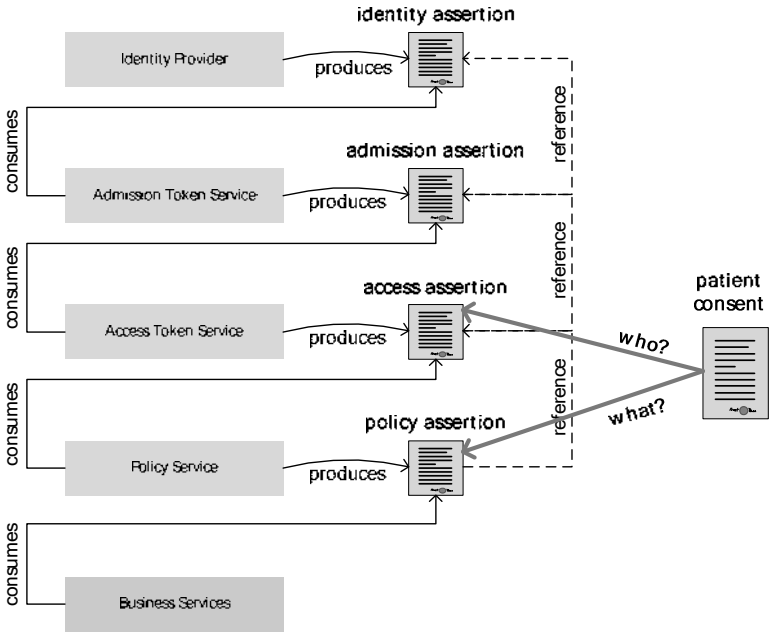

Abbildung 1: Authentisierungs- und Autorisierungsinformationen als Assertion-Kette

In der Deklaration der zur Umsetzung der Schutzziele beizubringenden Sicherheitsobjekte können „klassische" X.509 Token und *Security Context Token* mit SAML Token kombiniert werden. Die eFA-Sicherheitsarchitektur erlaubt es auch, mehrere SAML Token an einen Web Service zu versenden, d. h. iterativ ganze Ketten von aufeinander verweisenden Sicherheitsnachweisen aufzubauen. Hiermit können die Nachweise der Nutzung einzelner Sicherheitsdienste (Authentisierung, Pseudonymisierung, Autorisierung, etc.) als vom unterliegenden Security Framework zu bearbeitende Bestandteile des Aufrufs eines Anwendungsdienstes kodiert werden[2].

Die Einbettung mehrerer SAML *Assertions* in das SOAP *Security Header* stellt auf der Implementierungsseite vielerlei Ansprüche: Zum einen muss die Semantik der

[2] Dieses Muster wird u. a. auch im europäischen epSOS-Projekt (www.epsos.eu) genutzt, um einem Anwendungsdienst Nachweise zu der in einem anderen Land erfolgten Nutzerauthentisierung und –autorisierung zu übermitteln.

Assertions selbst (strukturierte Elemente in Attributen) und zum anderen die korrekte Kombination einer *Assertion*-Kette überprüft werden können. Einem Aufrufenden muss zudem ein Besitznachweis für diese SAML *Assertions* abverlangt werden. Um unterschiedliche Sicherheitsanforderungen für die verschiedenen Vertrauensbeziehungen zwischen Dienstnutzern und -anbietern deklarieren zu können, wird für jede Vertrauensbeziehung in der Schnittstellenspezifikation ein eigener Port definiert. So können z. B. sehr einfach unterschiedliche *Policies* für Zugriffe aus verschiedenen Sicherheitszonen heraus definiert werden (z. B. Zugriffe innerhalb eines *Circle-of-Trust* und in einen *Circle-of-Trust* hinein).

Die folgende Abbildung verdeutlicht die Komplexität verschiedener Schnittstellen mit den Policies, in denen deklariert ist, welche Sicherheitsnachweise bei Aufruf eines Dienstes aus verschiedenen Sicherheitskontexten heraus jeweils beizubringen sind (z. B. wird bei einem Aufruf von „außen“ ein expliziter Authentisierungsnachweis verlangt, während beim Aufruf durch Dienste innerhalb eines definierten *Circle-of-Trust* aufgrund von vorab aufgebauten *WS-SecureConversation* Kanälen solche Nachweise nicht erforderlich sind).

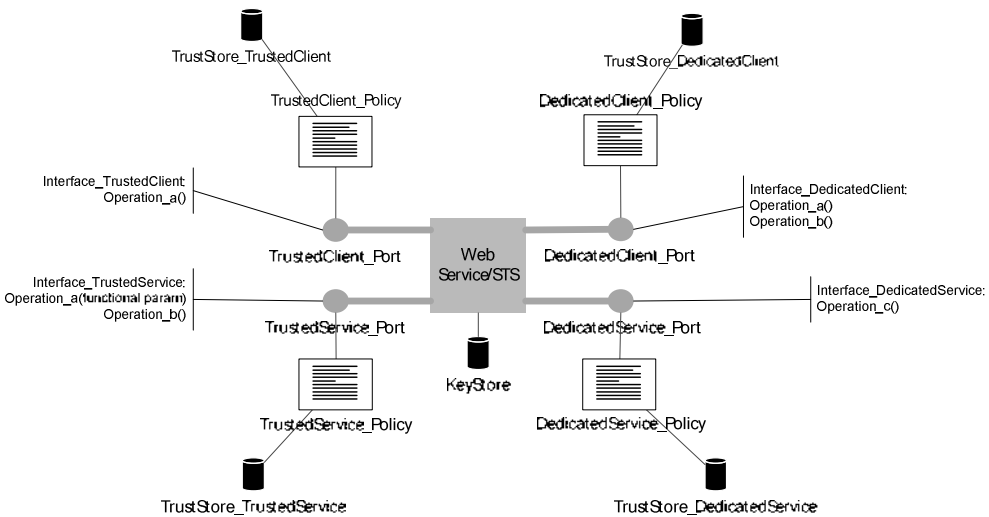

Abbildung 2: Web Service Schnittstellen mit differenzierten Security Policies

3 Deklarative Autorisierung

Das eFA Berechtigungsmanagement setzt Autorisierungen über den Standard XACML v2.0 [XACML2.0] um. Die Zugriffsrechte einzelner Rollen auf Fallakten werden in Zugriffsregeln (Access *Policies)* festgelegt. Durch Bindung von Rollen an Personen kann ein dedizierter Dienst (*eFA Access Token Service*) die für den jeweiligen Nutzer

geltenden Zugriffsregeln ermitteln und einen entsprechenden Autorisierungsnachweis ausstellen, der an die *Assertion*-Kette des authentisierten Nutzers angefügt wird.

Die XACML-Spezifikation definiert sowohl die *Policy*-Sprache mit entsprechendem Datenmodell als auch ein *Request/Response*-Protokoll für die Anfrage von Autorisierungsentscheidungen. Eine *Access Policy* besteht aus einer oder mehreren Regeln. Um konkurrierende Regeln zu synchronisieren, werden Kombinationsalgorithmen definiert (z. B. deny-overrides). Jede Regel gewährt oder verweigert den Zugriff auf eine Kombination aus Subjekt (Aufrufender Nutzer), Ressource, Benutzeraktion und Ausführungsumgebung. An eine Regel können noch weitere optionale Bedingungen gestellt werden. Die Zugriffskontrolle gestaltet sich vereinfacht wie folgt: Ein *Policy Enforcement Point* (PEP) unterbricht den Kontrollfluss vor der geschützten Ressource und stellt eine Autorisierungsanfrage (XACML *Request*) an einen *Policy Decision Point*[3] (PDP). Der PDP lädt im System vorhandene Zugriffsregeln über einen *Policy Administration Point*, welcher für die Erstellung und Verwaltung der *Policies* zuständig ist. Die Auswertung der Zugriffsregeln gegen einen konkreten Zugriffsversuch durch den PDP resultiert in einer Autorisierungsentscheidung, welche dem PEP in einem XACML *Response* zurückgesandt wird. Nur bei einer positiven Autorisierungsentscheidung gibt der PEP die weitere Ausführung des Kontrollflusses – und damit den Zugriff auf die geschützte Ressource - frei.

Um ein flexibles *Policy*-Design zu ermöglichen, sieht die Sicherheitsarchitektur der elektronischen Fallakte einen Indirektionsschritt bei der Ausstellung einer Zugriffsberechtigung vor. Der Autorisierungsnachweis bzw. die *Access Assertion* enthält ein XACML *PolicySet* mit Subjekt und Fallakten-ID, welcher ein oder mehrere Zugriffs-Policies zugewiesen werden. Die *Policy* selber ist in einer sog. *Policy Assertion* enthalten, die anhand einer *Access Assertion* von einem dedizierten Dienst (*Policy Token Service*) abgerufen werden kann. Diese Trennung von Policy-ID und eigentlicher Policy erlaubt es, mit den gleichen Sicherheitsdiensten sowohl Sicherheitsarchitekturen aufzubauen, die (wie die eFA) deklarativ kodierte *Policies* nutzen als auch Anwendungsdiensten die Umsetzung impliziter Regelwerke zu überlassen (z. B. IHE BPPC, wo Consent-Modelle über IDs ausgerückt werden und die Abbildung dieser Modelle auf den Daten zugeordnete *HL7 Confidentiality Codes* durch das Dokumenten-Register erfolgt).

```
<xacml:PolicySet
  PolicyCombiningAlgId="policy-combining-algorithm:permit-overrides">
  <xacml:Target>
    <xacml:Subjects>
      <xacml:Subject>
        <xacml:SubjectMatch MatchId="xacml:1.0:function:base64Binary-
         equal">
          <xacml:AttributeValue>
            VYDKTmgZ6JarY2xJ8RFUO5e35bYgewAIhLVFtfEN0Cg=
          </xacml:AttributeValue>
```

[3] Fehlen Informationen für die Evaluierung einer Policy oder ist ein PEP nicht in der Lage, einen XACML Request zu erzeugen, kann ein Context Handler genutzt werden, welcher die Daten z. B. durch Nutzung eines Policy Information Points aufbereitet und dann an den PDP sendet.

```
          <xacml:SubjectAttributeDesignator
            AttributeId="urn:ecr:attributes:user-information-hash"/>
        </xacml:SubjectMatch>
      </xacml:Subject>
    </xacml:Subjects>
    <xacml:Resources>
      <xacml:Resource>
        <xacml:ResourceMatch MatchId="xacml:1.0:function:string-equal">
          <xacml:AttributeValue>
            1.3.6.1.4.1.778.51.623.3.1.20.163f20c8
          </xacml:AttributeValue>
          <xacml:ResourceAttributeDesignator
            AttributeId="urn:ecr:attributes:record-id"/>
        </xacml:ResourceMatch>
      </xacml:Resource>
    </xacml:Resources>
  </xacml:Target>
  <xacml:PolicyIdReference>
    urn:ecr:names:xacml:2.0:default:policyid:read-only
  </xacml:PolicyIdReference>
  <xacml:PolicyIdReference>
    urn:ecr:names:xacml:2.0:default:policyid:deny-all
  </xacml:PolicyIdReference>
</xacml:PolicySet>
```

Abbildung 3: Deklarative Rechtezuweisung mit Referenzen

In diesem Beispiel wird einem Subjekt (in diesem Falle ein Hashcode über einen Benutzer mit seinen Rollen)[4] für eine Fallakte das Leserecht gewährt. Die Referenzen (read-only, deny-all) können genauso gut implizite Zugriffs-Policies und keine weiteren XACML Policies bedeuten, welche dann jedoch bei der Zugriffsprüfung im Programmcode ausgewertet werden können. Dies ist umsetzungsspezifisch. Die eFA-Sicherheitsarchitektur lässt jedoch noch komplexere und damit feingranularere Zugriffsbeschreibungen mit diesem Mechanismus zu.

Dieser Aspekt der deklarativen Sicherheit geschieht in der Datenbasis des *Access Token Service*, bei der einer Kombination aus Subjekt und Fallakte semantisch die Zugriffsrechte zugewiesen werden. Wie die referenzierten Zugriffs-Policies tatsächlich ausgestaltet sind, ist bei der Vergabe der Berechtigungen an dieser Stelle nicht von Interesse.

Das *Policy Management* definiert, was sich hinter diesen sogenannten Referenzen verbirgt. Dazu werden sogenannte Building Blocks, also wiederverwendbare Policy-Bausteine, definiert. Die Policy-IDs aus der *Access Assertion* zeigen jeweils genau auf solch einen Block. Das folgende Beispiel zeigt einen Ausschnitt einer Zugriffs-Policy für Leseoperationen – in diesem Fall die Berechtigung, eine Ordnerliste einer Fallakte abzurufen.

```
<xacml:Policy
  PolicyId="urn:ecr:names:xacml:2.0:default:policyid:read-only">
```

[4] Mehr Flexibilität kann dadurch erreicht werden, wenn den zugewiesenen funktionalen Rollen Zugriffsrechte zugewiesen werden.

```
  <xacml:Target />
  <xacml:Rule Effect="Permit"
    RuleId="urn:ecr:names:xacml:2.0:default:rules:
    ecrfolderregistry_ecrfolderregistrydedicatedclient_getfolderlist">
    <xacml:Target>
      <xacml:Resources>
        <xacml:Resource>
          <xacml:ResourceMatch MatchId=" xacml:1.0:function:string-equal">
          <xacml:AttributeValue>
            {http://isst.fhg.de/ecr/application/ecrfolderregistry}
            ECRFolderRegistry
          </xacml:AttributeValue>
          <xacml:ResourceAttributeDesignator
            AttributeId="xacml:2.0:profile:webservices:v1.1:service"/>
        </xacml:ResourceMatch>
        <xacml:ResourceMatch MatchId="xacml:1.0:function:string-equal">
          <xacml:AttributeValue>
            {http://isst.fhg.de/ecr/application/ecrfolderregistry}
            ECRFolderRegistryDedicatedClientSoap12HttpPort
          </xacml:AttributeValue>
          <xacml:ResourceAttributeDesignator
            AttributeId="xacml:2.0:profile:webservices:v1.1:port"/>
        </xacml:ResourceMatch>
        <xacml:ResourceMatch MatchId="xacml:1.0:function:string-equal">
          <xacml:AttributeValue>
            {http://isst.fhg.de/ecr/application/ecrfolderregistry}
            getFolderList
          </xacml:AttributeValue>
          <xacml:ResourceAttributeDesignator
            AttributeId="xacml:2.0:profile:webservices:v1.1:operation"/>
          </xacml:ResourceMatch>
        </xacml:Resource>
      </xacml:Resources>
    </xacml:Target>
  </xacml:Rule>
  <!—
    Add more permit rules with read operations.
  -->
</xacml:Policy>
```

Abbildung 4: XACML Policy als Building Block

Um die Policy-Evaluierung in einem *Security Provider* eines Anwendungsdienstes zu vereinfachen, brauchen die Zugriffs-Policy-Referenzen aus dem XACML *PolicySet* der *Access Assertion* lediglich mit den Building Blocks ersetzt zu werden. Das Ergebnis ist eine von einen XACML Framework direkt verarbeitbare Zugriffs-Policy. Eine entsprechende Autorisierungsanfrage (XACML *Request*) an einen *Policy Decision Point* übergibt die Fallakten-ID (Ressource), das aufrufende Subjekt und die aufzurufende Aktion (Web Service Operation) auf der Ressource.

4 Deklarative Sicherheit bei der Prüfung von Security Token

Wie oben geschildert, bieten *WS-Policy* und *WS-SecurityPolicy* die Beschreibungsmöglichkeit, um nichtfunktionale Anforderungen (z. B. Sicherheit, Datenschutz oder

garantierte Zustellung) in XML für eine Kommunikation zu formulieren.[5] Ein Operationsaufruf eines Web Service wird entsprechend einer Anforderungsbeschreibung geprüft und ggf. zugelassen oder verweigert (*WS-SecurityPolicy Enforcement* und *Decision*). Der Einsatz von Web Service Frameworks bietet hier schon eine umfangreiche Unterstützung. Dazu zählen u. a. die Prüfung von Zeitstempeln und Signaturen oder Möglichkeit der Anforderung von *Security Token*.

Spezielle *Security Token*[6] wie das SAML Token können von einem Web Service Framework lediglich auf Wohlgeformtheit, allgemeine Attribute (Gültigkeit, etc.) oder Signatur geprüft werden. Hinzu kommt noch die Möglichkeit, die Authentizität des Einreichers eines Token über *Holder-of-Key SAML Assertions* nachprüfen zu können. Alles was darüber hinaus geht, wird von Frameworks nicht mehr adäquat adressiert. Die Semantik der SAML Token kann mit bestehenden WS-* Spezifikationen derzeit beispielsweise nicht beschrieben werden. Auch deren Überprüfung muss folglich manuell d. h. im Programmcode erfolgen.

XACML Policies für die Anforderungsbeschreibung von Security Token

Das Autorisierungsframework XACML lässt sich nicht nur dazu einsetzen, Web Service Operationen zu speziellen Ports zu schützen (*Web Services Profile of XACML* [WS-XACML]), sondern auch um die Korrektheit von SAML Token festzustellen. Existierende Open-Source XACML Frameworks wie SunXACML oder Enterprise Java XACML erlauben die schnelle Evaluierung von Autorisierungsentscheidungen bei einem gegebenen Pool von Zugriffs-Policies. Komfortable Funktionen wie das *Decision* und *Policy Caching* ermöglichen gute Antwortzeiten auch bei umfangreichen Sicherheitsprüfungen.

SAML Token bzw. SAML *Assertions* entsprechen immer einem definierten Profil, so dass die verarbeitenden Systeme auf die Existenz bestimmter Attribute in einer *Assertion* vertrauen. Beispielsweise akzeptiert ein Service Provider eine SAML *Assertion* als Authentisierungsnachweis nur, wenn diese von einem vertrauten *Identity Provider* ausgestellt und der Nutzer sich mit einem X.509 Zertifikat authentisiert hat (In diesem Fall existiert dann ein Attribut *AuthenticationMethod* in einem *AuthenticationStatement* mit dem Wert „urn:oasis:names:tc:SAML:1.0:am:X509-PKI“).

Entsprechend dem Profil der SAML *Assertion* kann eine XACML *Policy* erstellt werden, wonach die einem Dienst vorgelegte SAML *Assertion* mit den existierenden Mechanismen des Sicherheitsframeworks validiert wird. Als Beispiel dient im Folgenden die *Identity Assertion* aus der elektronischen Fallakte, welche eine Ausprägung eines SAML Token ist und einen Authentisierungsnachweis mit Attributen für einen Nutzer (Leistungserbringer) von Fallakten darstellt [EFASEC1.2]. Eine wichtige Eigenschaft von XACML ist die Möglichkeit, Werte aus einer

[5] Dies kann mit entsprechender Tool-Unterstützung (z. B. NetBeans) Assistenten-basiert oder manuell in der Policy (XML-Datei) erfolgen.
[6] Dazu zählen z. B. Username Token, Binary Security Token (X509Token), XML Security Token (SAML Token).

Autorisierungsanfrage mittels XPath-Ausdrücken zu ermitteln. Dies macht es möglich, ein Anforderungsprofil eines beliebigen XML-Dokuments (in unserem Fall die *Identity Assertion*) maschinenlesbar zu definieren und zu überprüfen.

Das folgende Beispiel zeigt eine XACML *Policy* für eine *Identity Assertion*. Mittels XPath-Ausdrücken wird z. B. geprüft, ob dem Subjekt eine entsprechende eMail-Adresse zugeordnet wurde. Das in der eFA definierte Profil für die *Identity Assertion* ist wesentlich komplexer, so dass die korrespondierende XACML *Policy* die Attributwerte der SAML *Attribute Statements* ebenfalls beschreiben muss. Aus Platzgründen wurde hier auf die vollständige Darstellung verzichtet.

```
<Policy PolicyId="identity-assertion-policy"
        RuleCombiningAlgId="deny-overrides">

  <PolicyDefaults>
    <!-- Define the URI for the XPath 1.0 specification. -->
    <XPathVersion>
       http://www.w3.org/TR/1999/Rec-xpath-19991116
    </XPathVersion>
  </PolicyDefaults>

  <Target>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="xacml:1.0:function:string-equal">
          <AttributeValue>
            urn:oasis:names:tc:SAML:1.0:assertion
          </AttributeValue>
          <ResourceAttributeDesignator
            AttributeId="..target-namespace" />
        </ResourceMatch>
      </Resource>
    </Resources>
  </Target>

  <Rule RuleId="identity-assertion-rule" Effect="Permit">
    <Description>The assertion is valid, if ...</Description>
    <Target/>
    <Condition>
      <Apply FunctionId="xacml:1.0:function:and">
        <Apply FunctionId="xacml:1.0:function:integer-equal">
          <Apply FunctionId="xacml:1.0:function:xpath-node-count">
            <AttributeSelector
                  RequestContextPath="//xacml-context:Request/node()"/>
            </Apply>
              <AttributeValue>1</AttributeValue>
            </Apply>

            <!-- Node Values -->
            <Apply FunctionId="xacml:1.0:function:string-equal">
              <Apply
                FunctionId="xacml:1.0:function:string-one-and-only">
                <AttributeSelector RequestContextPath=
                      "//*[local-name()='ConfirmationMethod']/text()"/>
              </Apply>
              <AttributeValue>
                urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
```

```
              </AttributeValue>
            </Apply>

            <!-
            Define value for attribute 'format' in
            subject name identifier
            -->
            <Apply FunctionId="xacml:1.0:function:string-equal">
              <Apply
                FunctionId="xacml:1.0:function:string-one-and-only">
                <AttributeSelector
                  RequestContextPath="/Request/Resource/ResourceContent/
                  Assertion/AttributeStatement/Subject/NameIdentifier/
                  Format" />
              </Apply>
              <AttributeValue>
                urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
              </AttributeValue>
            </Apply>

            <!-
            Check other values according to the assertion's profile
            -->
          </Apply>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

Abbildung 5: XACML Policy für eine SAML Assertion

Token-Prüfung mit XACML

Einem XACML Framework muss nun das *Security Token* bzw. die SAML *Assertion* übergeben werden. Hierzu wird von einem XACML *Policy Enforcement Point* (PEP) bzw. einem *Context Handler* des genutzten Frameworks ein XACML *Request* erstellt und dann einem XACML *Policy Decision Point* (PDP) für eine Autorisierungsprüfung (in unserem Fall eine Token-Prüfung) zugesandt. Dieser XACML *Request* enthält als *Resource Content* die SAML *Assertion* selbst. Ein weiteres Attribut definiert den Namensraum der SAML *Assertion*, um das entsprechende *ResourceMatch* der XACML *Policy* zu erwirken.

```
<xacml:Request>
  <xacml:Subject/>
  <xacml:Resource>
    <xacml:ResourceContent>
      <saml:Assertion> ... </saml:Assertion>
    </xacml:ResourceContent>
    <xacml:Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:target-namespace"
      DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <xacml:AttributeValue>
        urn:oasis:names:tc:SAML:1.0:assertion
      </xacml:AttributeValue>
```

```
    </xacml:Attribute>
  </xacml:Resource>
  <xacml:Action/>
  <xacml:Environment/>
</xacml:Request>
```

Abbildung 6: XACML Request für eine Token-Prüfung

Die Vorteile der Nutzung deklarativer Sicherheit zur Prüfung von SAML *Assertions* liegen insbesondere in der dadurch erreichbaren Flexibilität in Bezug auf die definierten SAML Profile. So können z. B. bestehende Profile weiterentwickelt oder sogar verschiedenen Profile parallel genutzt werden, ohne dass eine Anpassung des Programmcodes der verarbeitenden Dienste erforderlich wäre. Soll beispielsweise in der Zukunft der Zugriff auf eine Fallakte nur nach Authentifizierung mit einem Heilberufsausweis oder aus einem über eine SMC geschützten Kontext erfolgen können, muss lediglich die XACML *Policy* zur Verifizierung des entsprechenden Attributs der *Identity Assertion* geändert werden. Hiermit ist im Endeffekt die Anbindung bestehender eFA-Implementierungen an die Sicherheitsobjekte der Telematikinfrastruktur möglich, ohne auch nur eine Zeile Programmcode ändern zu müssen. Insbesondere können Anpassungen dieser Art auch vom Sicherheitsadministrator des eFA-Providers ohne Erfordernis der Einbeziehung des KIS-Herstellers vorgenommen werden, wodurch zusätzliche Flexibilität gewonnen wird.

Die folgende Abbildung fasst den Einsatz von XACML noch einmal zusammen:

(I) Ein SOAP *Request* wird einem Service Provider zugesandt und das Web Service Framework prüft diesen entsprechend der *WS-Policy/WS-SecurityPolicy* des Web Service;

(II) Ein Security Provider oder *Interceptor* triggern einen PEP, welcher die im SOAP-Aufruf vorhandene SAML *Assertion* in einen XACML *Request* überführt und an den PDP sendet.

(III) Ist die Autorisierungs- bzw. Token-Prüfung positiv, wird die eigentliche Autorisierung der Web Service Operation überprüft.

(IV) Die Web Service Operation wird ausgeführt und der Zugriff auf die Ressource gewährt.

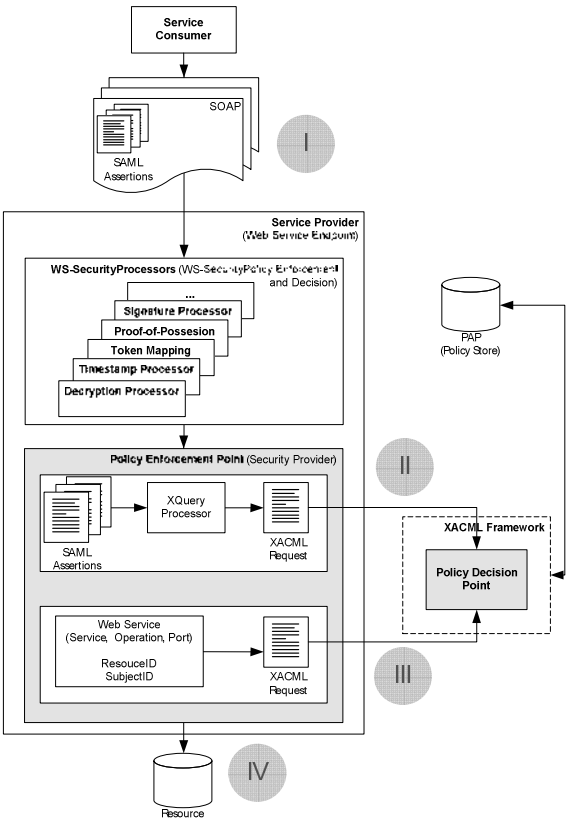

Abbildung 7: Token-Prüfung und Zugriffskontrolle mit XACML

4. Zusammenfassung

In diesem Papier wurde am Beispiel der Sicherheitsarchitektur der elektronischen Fallakte beschrieben, wie durch Nutzung deklarativer Sicherheit eine hoher Grad an Flexibilität und Evolutionsfähigkeit erreicht werden kann. Hierbei wird deklarative Sicherheit nicht nur zur Externalisierung von Berechtigungsregeln genutzt, sondern auch um die Umsetzung von Sicherheitsanforderungen an *Security Token* zu verifizieren. Dadurch dass Zugriffe auf einen Web Service, die aus verschiedenen Sicherheitszonen kommen, auf verschiedene Ports abgebildet werden, ist dieser Ansatz insbesondere auch geeignet, um Anwendungen einer sektorübergreifenen Kommunikation zu vereinfachen. Gerade hier erfolgen Zugriffe auf geschützte Ressourcen sowohl von internen als auch von externen Systemen, die jeweils unterschiedlichen Sicherheitsannahmen unterliegen und potenziell auch unterschiedliche Sicherheitsobjekte nutzen.

Literaturverzeichnis

[BC07] Boehm, O.; Caumanns, J.: Föderatives Identitätsmanagement am Beispiel der elektronischen Fallakte. Informatik-Spektrum, Nr.4/07, 2007; S. 240-250.

[BSI07] Caumanns, J.; Boehm, O.; Neuhaus, J.: Elektronische Fallakten zur sicheren einrichtungsübergreifenden Kommunikation. In (Bundesamt für Sicherheit in der Informationstechnik): Innovationsmotor IT-Sicherheit. SecuMedia Verlag, 2007.

[CBN07] Caumanns, J.; Boehm, O.; Neuhaus, J.: Elektronische Fallakten zur einrichtungsübergreifenden Kooperation. E-HEALTH-COM, Nr.1/07, 2007; S. 68-70.

[ebXMLBP] Dubray, J.; Amand, S.; Martin, M. (Eds.): ebXML Business Process Specification Schema, Technical Specification v2.0.4. OASIS Standard, Dezember 2006. http://docs.oasis-open.org/ebxml-bp/2.0.4/OS/spec/ebxmlbp-v2.0.4-Spec-os-en.pdf

[Edoc08] Boehm, O.; Caumanns, J.; Franke, M.; Pfaff, O.: Federated Authentication and Authorization: A Case Study. In: Proc. 12th IEEE International EDOC Conference. 16.8.08. München, 2008;.S. 356-362.

[EFASEC1.2] Boehm, O.; Kuhlisch, R.: Sicherheitsarchitektur der elektronischen Fallakte. Version 1.2. Februar 2008.

[SAML2.0] Cantor, S.; Kemp, J.; Philpott, R.; Maler, E. (Eds.): Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, März 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SUN08] Singh, I.; Stearns, B.; Johnson, M.: Designing Enterprise Applications with the J2EE Platform, Second Edition. http://java.sun.com/blueprints/guidelines/ designing_enterprise_applications_2e/ security/security4.html

[WS-XACML] Anderson, A. (Ed.): Web Services Profile of XACML Version 1.0. Working Draft 8, Dezember 2006. http://www.oasis-open.org/committees/download.php/21490/ xacml-3.0-profile-webservices-spec-v1.0-wd-8-en.pdf

[WSPOL1.5] Vedamuthu, A. et al (Eds.): Web Services Policy 1.5 – Framework. W3C Recommendation. September 2007. http://www.w3.org/TR/ws-policy/

[WSSPOL1.2] Nadalin A. et al (Eds.): WS SecurityPolicy 1.2. OASIS Standard, Juli 2007. http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.pdf

[XACML2.0] Moses, T. (Ed.): eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard, Februar 2005. http://docs.oasis-open.org/xacml/ 2.0/ access_control-xacml-2.0-core-spec-os.pdf

Rapid in-Depth Analysis für Telematikanwendungen im Gesundheitswesen

Identifizierung nicht erkannter Sicherheitslücken mit Threat Modeling und Fuzzing

Fabian Schwab, B.Sc.; Jörg Lübbert, B.Sc.;
Peter Sakal, B.Sc.; Prof. Dr. Hartmut Pohl

Informationssicherheit, Fachbereich Informatik
Hochschule Bonn-Rhein-Sieg
Grantham-Allee 20
53757 St. Augustin
Fabian.Schwab@h-brs.de
Joerg.Luebbert@h-brs.de
Peter.Sakal@h-brs.de
Hartmut.Pohl@h-brs.de

Abstract: Die Normen DIN EN 61508 und DIN EN 62304 beschreiben Sicherheitsanforderungen für die Entwicklung von Software im medizinischen Umfeld. Diese beinhalten u.a. Vorschriften zur Verifikation und Diagnose (Kapitel C5, DIN EN 61508-7 [Di09]), zur Beurteilung der funktionalen Sicherheit (Kapitel C6, DIN EN 61508-7 [Di09]), zur Implementierung und Verifikation von Software-Einheiten (Kapitel 5.5, DIN EN 62304 [Di07]) und zur Prüfung des Softwaresystems (Kapitel 5.7, DIN EN 62304 [Di07]). Durch die kosteneffektiven Verfahren Threat Modeling und Fuzzing wird diesen Forderungen entsprochen und insbesondere die Identifizierung unveröffentlichter Sicherheitslücken ermöglicht. In einem Forschungsprojekt[1] werden Tools für beide Verfahren analysiert und bewertet. Im Projekt werden mit beiden Verfahren sehr erfolgreich bislang nicht identifizierte (unveröffentlichte) Sicherheitslücken in Individual- und Standardsoftware identifiziert und auch behoben. Im Rahmen der Gesundheitstelematik können durch beide Verfahren die Anforderungen zur Softwareentwicklung und -verifizierung erfüllt und darüber hinaus kann ein weit höheres Sicherheitsniveau erreicht werden.

[1] Förderung des Projektes SoftSCheck durch das Bundesministerium für Bildung und Forschung (Förderkennzeichen 01 IS 09030)

1 Motivation

Software kann nicht fehlerfrei erstellt werden [SP10]. Dies macht das Testen von Software erforderlich, wobei eine enge Bindung zur Qualitätssicherung besteht. Manuelles Testen von Software mit einer großen Menge an Programmcode ist nicht praktikabel [SGA07]. Threat Modeling ermöglicht die Identifizierung von Sicherheitslücken bereits in der Design-Phase. Fuzzing kann sowohl als Black-Box Test (ohne vorliegenden Quellcode) als auch als White-Box Test (mit vorliegendem Quellcode) in der Verification-Phase durchgeführt werden.

Das heuristische Tool-gestützte Verfahren Threat Modeling unterstützt die Identifizierung von Sicherheitslücken. Durch Bewertung und Kategorisierung von Sicherheitslücken können Schutzmechanismen und Gegenmaßnahmen bestimmt werden, um Sicherheitsrisiken zu mindern, zu minimieren, zu beheben oder auch zu akzeptieren [HL06]. Die Erstellung eines Threat Models für einen Systementwurf erfolgt idealerweise bereits in der Design Phase; dadurch ist eine vertrauenswürdige Implementierung von Software bereits lange vor der Veröffentlichung (Release) möglich. Threat Modeling ist ein kosteneffizientes Verfahren zur Identifizierung, Vermeidung und Minderung von Sicherheitslücken und Bedrohungen [My05, MLY05].

Mit Fuzzing steht ein Tool-gestütztes Verfahren zur Identifizierung von Softwarefehlern in der Verification-Phase zur Verfügung. Fuzzing kann dazu beitragen, unveröffentlichte sicherheitsrelevante Fehler zu identifizieren. Dazu werden die Eingabeschnittstellen der zu testenden Software identifiziert, um automatisiert und zielgerichtet Daten an diese zu senden, während die Software durch einen Monitor auf auftretende Fehler überwacht wird. So kann der Identifizierung von Sicherheitslücken durch Dritte und somit der Entwicklung von (Less-Than-)Zero-Day-Exploits (Angriffsprogramme auf unveröffentlichte Sicherheitslücken) entgegengewirkt werden [Po07]; (Less-Than-)Zero-Day-Exploits sind eine der zwanzig häufigsten Angriffsformen [Sa10].

Die Kosten zur Behebung von Sicherheitslücken nehmen im Verlauf des Softwareentwicklungsprozesses exponentiell zu [Ni02]. Wenn Fehler in der Testing- bzw. Verifikationsphase identifiziert werden, steigen die Kosten im Vergleich zur Identifizierung in der Design-Phase um den Faktor 15. Werden Fehler erst in der Release-Phase (oder später) entdeckt, steigen die Kosten um den Faktor 100 (Siehe Abbildung 1: Kosten zur Behebung von Sicherheitslücken im Verlauf der Softwareentwicklung [nach: Jo96]).

Microsoft nutzt die Vorteile von Threat Modeling [HL06] und Fuzzing [GKL08] seit 2003 als festen Bestandteil des eigenen `sicheren` Softwareentwicklungsprozess, dem Security Development Lifecycle (SDL) [HL06].

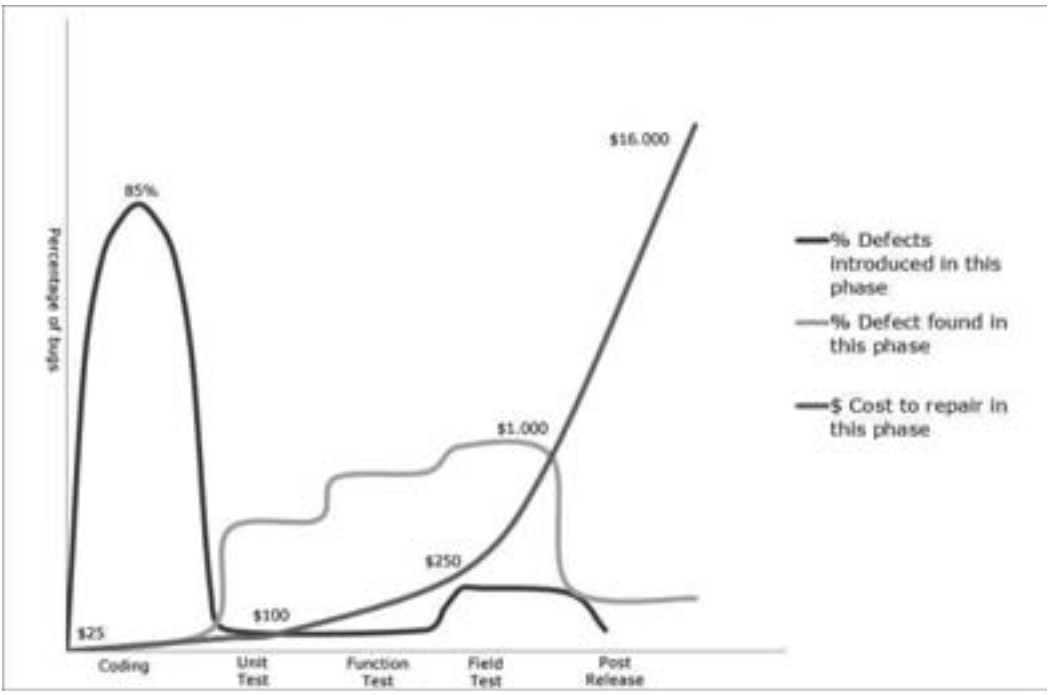

Abbildung 1: Kosten zur Behebung von Sicherheitslücken im Verlauf der Softwareentwicklung [nach: Jo96]

2 Threat Modeling

2.1 Einführung Threat Modeling

Nach vollständiger Identifizierung schützenswerter Komponenten (Assets) sowie zugehöriger Bedrohungen und Sicherheitslücken ist ihre Bewertung erforderlich. Identifizierung und Bewertung der Bedrohungen und Sicherheitslücken kann z.B. durch Attack Trees erfolgen [Sc99]. Auf Grundlage dieser Bewertung kann eine Minderung (Mitigation) der Bedrohungen und eine Behebung der Sicherheitslücken erfolgen. Neben den unterschiedlichen in den Threat Modeling Tools implementierten Maßnahmen (z.B. Redesign, Standard Mitigation, Custom Mitigation oder Accept Risk) ist eine individuelle Behandlung einzelner Bedrohungen und Sicherheitslücken sowie die Kontrolle der implementierten Verfahren erforderlich. Zur Durchführung von Threat Modeling können mehrere Ansätze gewählt werden. Hierbei wird jeweils ein unterschiedlicher Ausgangspunkt gewählt. Neben dem Software-Centric und dem Asset-Centric Ansatz existiert der Attacker-Centric Ansatz. Der systematische Ablauf des Attacker-Centric Ansatzes mit seinen drei Stufen – Sicht eines Angreifers verstehen, Sicherheit charakterisieren und Threats bestimmen [SS04] – ist in Abbildung 2 grafisch dargestellt. In jeder Stufe werden zugehörige Aktionen, mit dem Ziel, das Threat Model genauer zu spezifizieren und weiter auszubauen, durchgeführt. Die Entstehung neuer Bedrohungen für ein bereits durch Threat Modeling spezifiziertes System ist nach [Ow09] zu vernachlässigen.

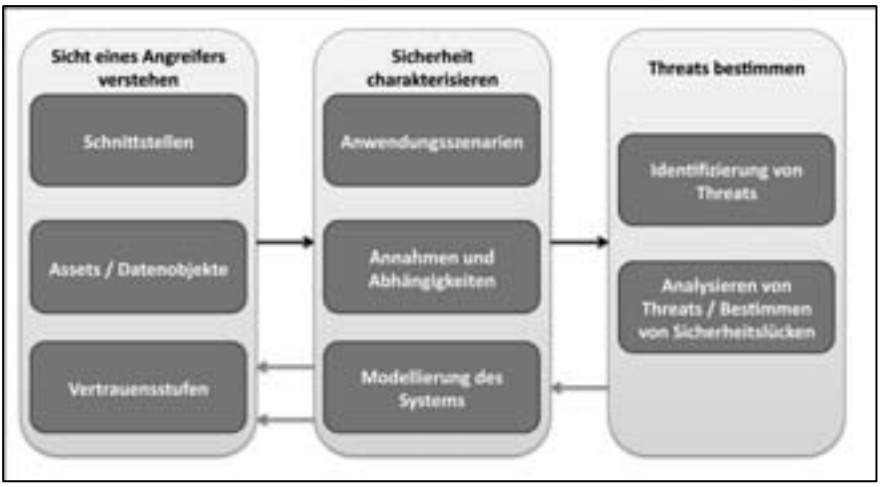

Abbildung 2: Threat Modeling Ablauf des Attacker-Centric Ansatzes [nach: SS04]

Durch Threat Modeling gewonnene Erkenntnisse können ebenso zur Entwicklung strategischer Penetration Tests genutzt werden. Hierbei wird auf Erkenntnisse über Angriffspunkte und Eingabeschnittstellen zurückgegriffen, welche aus der modellierten Systemarchitektur gewonnen werden können.

Neben der Anwendung in der Design Phase der Softwareentwicklung ist es auch möglich ein Threat Model für eine bereits implementierte Systemarchitektur – und auch für bereits ausgelieferte Software – zu erstellen.

2.2 Bewertungsverfahren für Threat Modeling Tools

Um eine einheitliche Klassifizierung der zu untersuchenden Threat Modeling Tools zu gewährleisten, ist die Entwicklung und Begründung differenzierender Bewertungsparameter unabdingbar. Diese müssen unter Berücksichtigung der im Projekt festgelegten Ziele neben einer eindeutigen Produktzuordnung die anfallenden Kosten, den Funktionsumfang, die Entwicklungsmöglichkeiten, den Installationsaufwand, die Benutzerfreundlichkeit und die Qualität der Dokumentation berücksichtigen. Die hier vorgelegten Bewertungsparameter sind in unterschiedliche Kategorien unterteilt. Kategorien beinhalten jeweils Parameter einer Funktionsgruppe. Parameter sind entsprechend ihrer Relevanz gewichtet und fließen in die Berechnung der jeweiligen Kategorie ein. Einzelne Kategorien werden ebenso auf Grundlage Ihrer Relevanz unterschiedlich gewichtet und fließen dementsprechend in das Gesamtergebnis ein.

Die Produkte werden anhand der folgenden Kategorien bewertet:

- Erfassung der Modellierungsmöglichkeiten hinsichtlich Assets, Threats, Threat Mitigation, Vulnerabilities und dem eingesetzten System bzw. der Systemumgebung.
- Folgende Visualisierungsmöglichkeiten werden erfasst:
  - Datenflussdiagramm
  - Bedrohungsbaum

  - Anwendungsdiagramm
  - Sonstige Möglichkeiten

- Das Berichtswesen (Reporting) erfasst Möglichkeiten des grafischen und textuellen Exports sowie den Reportumfang.
- Folgende Entwicklungsmöglichkeiten werden berücksichtigt:
  - Vorhandener Quellcode
  - Entwicklungsschnittstellen: Möglichkeit zur Anbindung eigener Erweiterungen und/oder zusätzlicher Module)
  - Community-Projekt: Entwicklung und/oder Support durch eine Community
  - Dokumentation der Entwicklungstools
- Beim Installations- und Nutzungsaufwand wird unterschieden nach der Grundinstallation, speziellen Voraussetzungen (z.B. erforderlicher Software), Einarbeitungszeit, Usability und Anwendung (vollständige Modellierung eines Testszenarios).

Die Lizenzkosten der Tools wurden nicht berücksichtigt, da mit Ausnahme eines Tools alle entgeltfrei verfügbar sind. Detailliertere Informationen zum entwickelten Bewertungsverfahren sowie dessen Anwendung wurden bereits gesondert veröffentlicht [Sc10].

2.3 Marktanalyse und Bewertung

Zum Zeitpunkt der Untersuchung waren sieben Threat Modeling Tools (sowohl entgeltfreie, als auch entgeltpflichtige) verfügbar. Um die Interessen von Herstellern zu wahren werden die Ergebnisse im weiteren Verlauf anonymisiert dargestellt. Die Tools können durch folgende Kategorien klassifiziert werden:

- Universal: Das Produkt eignet sich zur Erstellung eines Threat Models ohne spezifische Ausrichtung.
- Netzwerk: Das Produkt eignet sich zur Erstellung eines Threat Models für ein Netzwerkszenario. Es werden insbesondere Elemente eines Netzwerks betrachtet.
- Software Design: Das Produkt eignet sich zur Erstellung eines Threat Models im Rahmen des Software Designs und berücksichtigt spezifische Elemente der Software Entwicklung. Die Integration in den Software-Entwicklungsprozess von Microsoft - dem Security Development Lifecycle (SDL) - wird unterstützt.

Zu den untersuchten Produkten sowie deren Einsatzkategorie siehe Abbildung 3: Verfügbare Threat Modeling Tools.

Im Rahmen der durchgeführten Untersuchungen wurden große Diskrepanzen beim Funktionsumfang - explizit bei Modellierungs- und Visualisierungsmöglichkeiten - festgestellt. Alle untersuchten Tools bieten Funktionen zur Modellierung elementarer Bestandteile [Sc06] eines Threat Models an - Assets, Threats und Vulnerabilities - in unterschiedlichem Umfang. Eines der Tools (Tool C) lässt ausschließlich die Modellierung von Assets zu. Bei dem Tool handelt es sich um eine Sammlung von Symbolen zur Darstellung eines Threat Models. Lediglich drei Tools (Tool

A, Tool E und Tool G) bieten umfangreiche Visualisierungsmöglichkeiten in Form von Datenflussdiagrammen, Bedrohungsbäumen und Anwendungsdiagrammen. Die anderen Tools bieten ausschließlich eigene Visualisierungsformen in unterschiedlichem Umfang.

Name	Hersteller	Kategorie	Version
The CORAS Method	CORAS	Universal	20060714
Microsoft SDL Threat Modeling Tool	Microsoft	Software Design	3.1.3.1
Microsoft Threat Analysis & Modeling	Microsoft	Software Design	3.0
Practical Threat Analysis	PTA Technologies	Universal	1.6 Build 1212
SeaMonster	SeaMonster	Software Design	3.1.3.1
Skybox Secure	Skybox Security	Netzwerk	4.5
Trike	Dymaxion	Universal	1.12a

Abbildung 3: Verfügbare Threat Modeling Tools

Die Bewertung erfolgte auf Grundlage des eigens entwickelten Bewertungsverfahrens, welches die im Projekt festgelegten Ziele und die Anwendung in einem Testszenario abbildet. Eine Bewertung kann ausschließlich subjektiv für einen bedarfsspezifischen Fall erfolgen – es kann daher keine objektive Aussage über eine allgemeingültige Platzierung der einzelnen Tools erfolgen.Die Ergebnisse der einzelnen Parameter, Kategorien und der Gesamtbewertung werden in Prozent der möglichen Punkte dargestellt. Lediglich drei Tools erzielten in der Gesamtbewertung mehr als 70% der möglichen Punkte (Tool A, Tool B und Tool E). Die weiteren Tools weisen große Defizite (weniger als 10% der möglichen Punkte) in mindestens einer Kategorie auf und erzielen in der Gesamtbewertung weniger als 70% der möglichen Punkte. Die Erstellung komplexer und vollständiger Threat Models ist mit drei Produkten möglich. Die von den Tools erzielten Punkte in den einzelnen Kategorien sind in Abbildung 4 grafisch dargestellt.

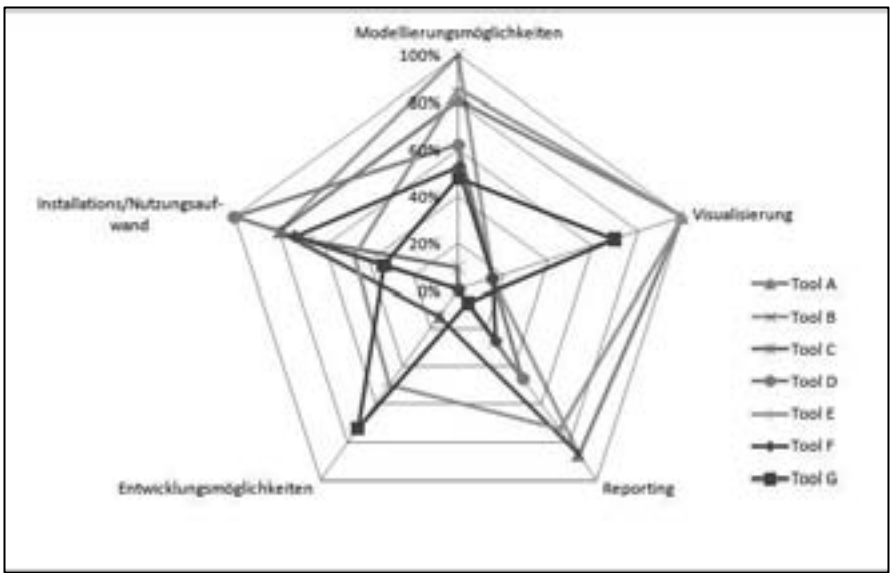

Abbildung 4: Bewertung verfügbarer Threat Modeling Tools

Die Stärken und Schwächen sowie die Gesamtbewertung und der Funktionsumfang der einzelnen Tools werden verdeutlicht:

- Tool A zeichnet sich durch gute Modellierungsmöglichkeiten sowie durch eine sehr gute Visualisierung aus; das Reporting ist gut ausgebildet; der Installations- und Nutzungsaufwand ist gut und Entwicklungsmöglichkeiten dieses Community-Projekts sind befriedigend.
- Tool B zeichnet sich durch gute Modellierungsmöglichkeiten und ein gutes Reporting aus; die Entwicklungsmöglichkeiten sind in geringen Umfang vorhanden und gut dokumentiert; die Visualisierung ist ausreichend (nur eigene Visualisierungsformen) und der Installations- und Nutzungsaufwand ist gut.
- Tool C bietet unzureichende Modellierungsmöglichkeiten und Berichtsfunktionen; Funktionen zur Visualisierung und Entwicklungsmöglichkeiten sind nicht vorhanden; der Installations- und Nutzungsaufwand ist befriedigend; es handelt sich um eine Sammlung von Symbolen zur Darstellung eines Threat Models.
- Tool D bietet befriedigende Modellierungsmöglichkeiten und ein ausreichendes Reporting; die Visualisierung ist mangelhaft und Entwicklungsmöglichkeiten sind nicht vorhanden; der Installations- und Nutzungsaufwand ist sehr gering und daher als sehr gut zu bewerten.
- Tool E zeichnet sich durch gute Modellierungsmöglichkeiten und eine sehr gute Visualisierung aus; das Reporting ist gut ausgebildet; die Entwicklungsmöglichkeiten dieses Community-Projekts sind gut und es sind Entwicklungsschnittstellen vorhanden; der Installations- und Nutzungsaufwand ist hoch - jedoch noch mit ausreichend zu bewerten.

- Tool F bietet befriedigende Modellierungsmöglichkeiten und einen guten Installations- und Nutzungsaufwand; die Visualisierung und das Reporting sind mangelhaft; Entwicklungsschnittstellen sind nicht vorhanden.
- Tool G bietet gute Entwicklungsschnittstellen und eine gute Visualisierung; die Modellierungsmöglichkeiten sind befriedigend; der Installations- und Nutzungsaufwand ist mit ausreichend zu bewerten und das Reporting ist ungenügend.

Zusammenfassend ergibt sich eine Spitzengruppe gebildet aus den Tools A, B und E. Die Tools C, D, F und G zeigen - bei Anwendung der hier zugrunde gelegten Bewertungsparameter - nur mittlere oder sogar unzureichende Leistungen. Abbildung 5 stellt die bewerteten Tools hinsichtlich Ihrer Effektivität bei der Erstellung eines Threat Models sowie hinsichtlich Ihres Funktionsumfangs dar.

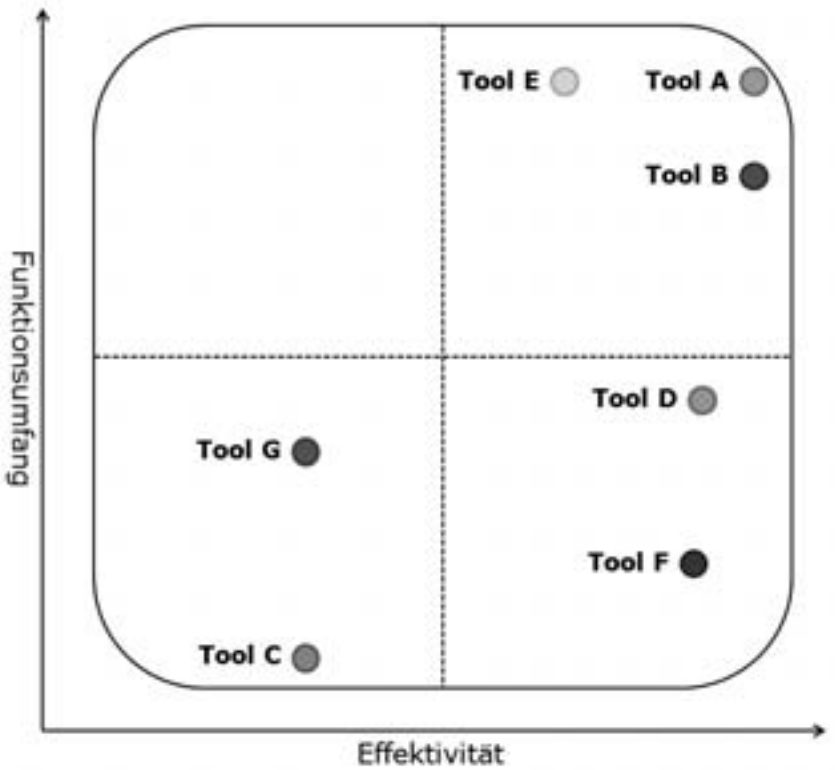

Abbildung 5: Vergleich der untersuchten Threat Modeling Tools

2.4 Fallstudie: Erfolgreiches Threat Modeling

Bei der exemplarischen Anwendung von Threat Modeling auf eine bereits implementierte und an Kunden ausgelieferte Individualsoftware konnten zahlreiche, mit klassischen Verfahren nicht erkannte, Sicherheitslücken identifiziert und behoben werden. Bei der untersuchten Software handelt es sich um eine komplexe E-Commerce-Plattform. Vor Release der Software wurde diese u.a. durch einen Web-Application-Scanner überprüft. Mit diesem wurden 30 schwerwiegende Sicherheitslücken gefunden, welche im Anschluss behoben wurden. Durch eine erneute Untersuchung wurde dies verifiziert. Durch die Anwendung von Threat Modeling auf die bereits implementierte, veröffentlichte und mit etablierten Verfahren untersuchte Software konnten acht weitere Sicherheitslücken, davon 5 schwerwiegende, identifiziert werden. Die Klassifizierung erfolgte anhand der SDL Security Bug Bar [HL06]. Bei den Sicherheitslücken handelt es sich sowohl um Implementierungsfehler als auch um Designfehler (z.B. der unberechtigte Zugriff auf eine Datenbank), welche durch die frühzeitige Anwendung von Threat Modeling hätten vermieden werden können. Für eine detaillierte Aufstellung der identifizierten Sicherheitslücken siehe Abbildung 6: Durch Threat Modeling identifizierte Sicherheitslücken in veröffentlichter Individualsoftware [nach: Ju10].

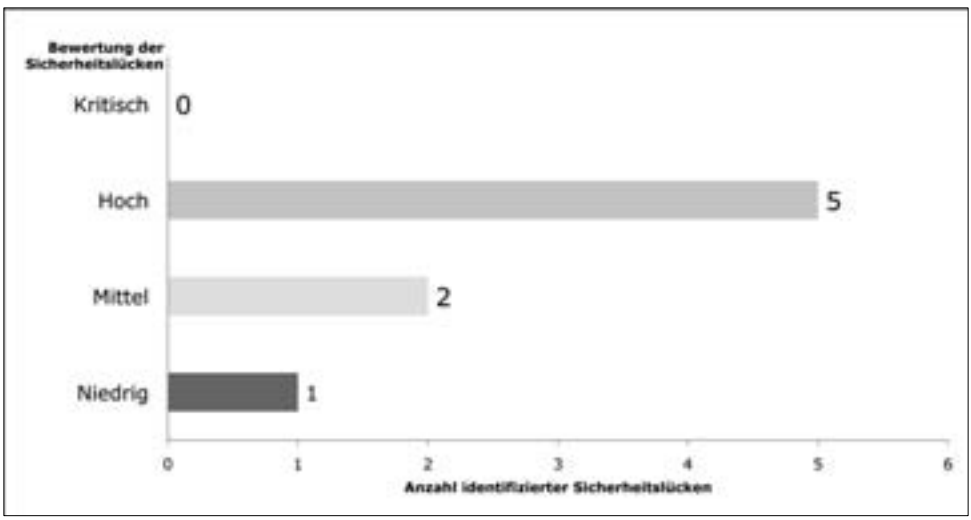

Abbildung 6: Durch Threat Modeling identifizierte Sicherheitslücken in veröffentlichter Individualsoftware [nach: Ju10]

Im Rahmen eines weiteren Forschungsprojektes[2] wurde ein zu entwickelnder Prototyp in der Design-Phase durch Threat Modeling untersucht. Auf Grundlage der Ergebnisse konnten im Entwicklungszyklus zahlreiche potentielle Bedrohungen identifiziert und vermieden werden (z.B. Denial of Service oder Information Disclosure)

[2] KMU-innovativ - Verbundprojekt EOMS: RFID in der Logistik - Offene Systeme auf der Basis von EPC und ONS (Förderkennzeichen 01IS08043B)

3 Fuzzing

3.1 Einführung Fuzzing

Fuzz-Testing (Fuzzing) ist eine Software-Überprüfungsmethode, die im Rahmen des Security Development Lifecycle (SDL) idealerweise in der Verifikationsphase eingesetzt wird [LH05], aber auch später nach Auslieferung der Software erfolgreich ist. Die Verifikationsphase befindet sich zwischen der Implementierungs- und der Release-Phase des SDL (Abbildung 1).

Der Fuzzing-Lifecycle beschreibt die Durchführung von Fuzzing [TDM08]:

1. Identifikation von Eingabeschnittstellen,
2. Generierung von Eingaben,
3. Versand von Eingaben,
4. Überwachung der Zielsoftware (Monitoring),
5. Durchführung einer Exception-Analyse und
6. Erstellung von Berichten (Reporting).

Nach erfolgreicher Identifizierung von Schnittstellen können Eingabedaten durch einen Fuzzer generiert und an die zu testende Software gesendet werden. Durch einen Monitor wird die Software während des Fuzzing auf auftretende Fehler überwacht, die durch möglichst gezielt gewählte Eingabedaten provoziert werden.

3.2 Bewertungsverfahren

Für die Bewertung von Fuzzern wird ein mehrstufiges Schema aus Bewertungsparametern, Kategorien und Gewichtungen herangezogen. Folgende Kategorien von Bewertungsparametern werden in diesem Schema verwendet und sind unterschiedlich gewichtet:

- Fuzzing-Methoden: Bewertet wird die Möglichkeit, einen Fuzzer für unterschiedliche Aufgaben einzusetzen. Darunter fallen z.B. die Unabhängigkeit des Produkts, Schnittstellen zu interpretieren, das Erlernen von Protokollspezifikationen oder die Möglichkeiten, die Zielsoftware zu interpretieren.
- Sonstiger Umfang: Bewertung sonstiger von Fuzzern bereitgestellter Methoden und Funktionen, um die Qualität des Fuzzing zu erhöhen. Hier werden insbesondere Möglichkeiten bemessen, aufgetretene Fehler zu identifizieren, einzugrenzen, auszuwerten und zu präsentieren.
- Software-Ergonomie: Bewertung insbesondere der Effektivität, Effizienz und Zufriedenstellung bei der Durchführung von Fuzzing im Umgang mit dem Produkt. Weiter werden Funktionale-, Dialog- sowie Ein- und Ausgabekriterien bewertet.

- Dokumentation: Bewertung des Umfangs und der Qualität der bereitgestellten Dokumentationsressourcen, wie Benutzerhandbuch, technische Dokumentation, integrierte Hilfesysteme etc.
- Entwicklungsmöglichkeiten: Bewertung der durch das Produkt bereitgestellten Funktionen zur Ergänzung eigener Funktionen oder zur Erweiterung bestehender.
- Kosten: Bewertung der anfallenden Kosten, insbesondere Anschaffungskosten, Wartungskosten und Personalkosten werden exemplarisch anhand von Fallstudien quantifiziert.

3.3 Marktanalyse

Es existieren über 100 Fuzzer. 25% aller Fuzzer können Web-Applikationen und 34% sonstige Netzwerkprotokolle testen. Das Testen von Dateiformaten wird von 15% aller Fuzzer unterstützt. Web-Browser können durch 10% und APIs durch 7% aller Fuzzer untersucht werden. Vgl. Abbildung 7: Marktübersicht von Fuzzern.

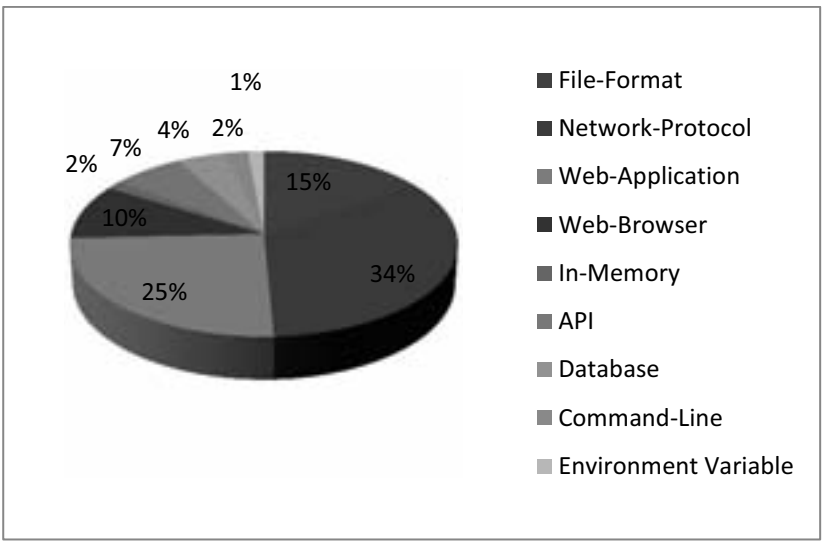

Abbildung 7: Marktübersicht von Fuzzern

Im Folgenden werden zwei Fuzzer von je zwei Fuzzer-Arten aufgrund ihrer Verbreitung exemplarisch untersucht, die das Testen mehrerer Schnittstellen unterstützten. Die untersuchten und bewerteten Fuzzer können der Abbildung 8: Einige untersuchte Fuzzer – entnommen werden.[3].

Fuzzer der Kategorie „Multi-Protocol Fuzzer“ unterstützen im Lieferumfang bereits mehrere Protokolle und können somit mehrere Schnittstellen untersuchen. Fuzzer der Kategorie „Fuzzing Framework“ müssen vor Verwendung auf die jeweiligen Protokolle angepasst werden.

[3] Die Ergebnisse der Untersuchung aller verfügbaren Fuzzer ist Teil des Forschungsprojekts und wird über diese Veröffentlichung hinaus nochmals veröffentlicht

Name	Kategorie	Version
beSTORM	Multi-Protocol Fuzzer	3.7.5 (4480)
Peach	Fuzzing Framework	2.3.4
Defensics	Multi-Protocol Fuzzer	3.8.3
Sulley	Fuzzing Framework	r156

Abbildung 8: Einige untersuchte Fuzzer

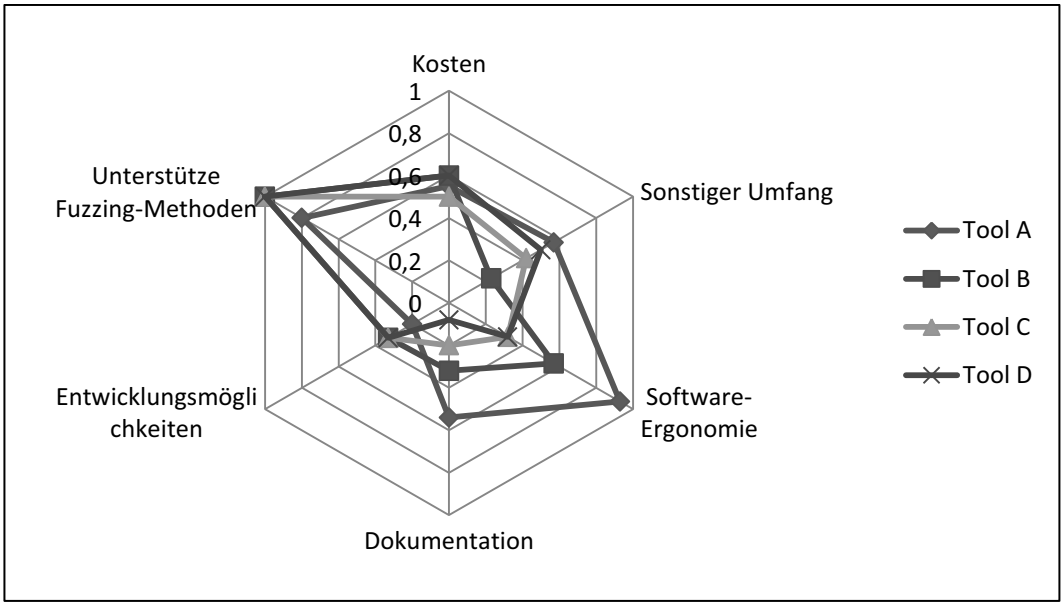

Abbildung 9: Bewertung der Fuzzer

Tool A zeichnet sich insbesondere durch sehr hohe Software-Ergonomie aus. Die Dialoge sind sehr benutzerfreundlich und selbsterklärend gestaltet; die Dokumentation ist vollständig und ausführlich, der Einsatz weiterer Medien ist jedoch wünschenswert. Die Weiterentwicklungsmöglichkeiten sind hier allerdings gering. Im Vergleich zu den anderen Tools ist der Umfang unterstützer Fuzzing-Methoden geringer.

Tool B bietet befriedigende Software-Ergonomie und Dokumentation. Wenig ausgeprägt ist der sonstige Umfang. Dafür werden sehr viele Fuzzing-Methoden unterstützt, die eine große Bandbreite von Anwendungsgebieten ermöglicht.

Tool C und D erlangen sehr geringe Wertungen in der Benutzerfreundlichkeit. Dafür sind die Einsatzmöglichkeiten sowie der Umfang bei beiden Produkten hoch. Sie unterscheiden sich nur geringfügig voneinander, wobei sich Tool C leicht von Tool D abhebt.

Bei den Kosten der Produkte gibt es geringe Unterschiede. Tool A und B weisen höhere Anschaffungskosten, Tool C und D höhere Personalkosten auf.

Insgesamt hebt sich Tool A am meisten ab, unterstützt jedoch nicht alle Fuzzing-Methoden. Keines der Produkte erhält mehr als 60% der möglichen Punkte in der Kategorie Umfang. Daher empfiehlt sich grundsätzlich die Nutzung von mindestens zwei Fuzzern.

Abbildung 9 – Bewertung der Fuzzer – stellt die Ergebnisse grafisch dar.

Sollen Fuzzer für mehrere Schnittstellen eingesetzt werden, sollte der Grad unterstützter Schnittstellen berücksichtigt werden. Die benötigte Expertise zum Einsatz von Fuzzern ist ein weiteres Maß, anhand dessen Fuzzer betrachtet werden können. Die Produkte D und C unterstützen mehr Schnittstellen als Produkte A und B, setzen jedoch eine höhere Expertise voraus. Vgl. Abbildung 10: Vergleich von Fuzzern anhand von Schnittstellen und benötigter Expertise.

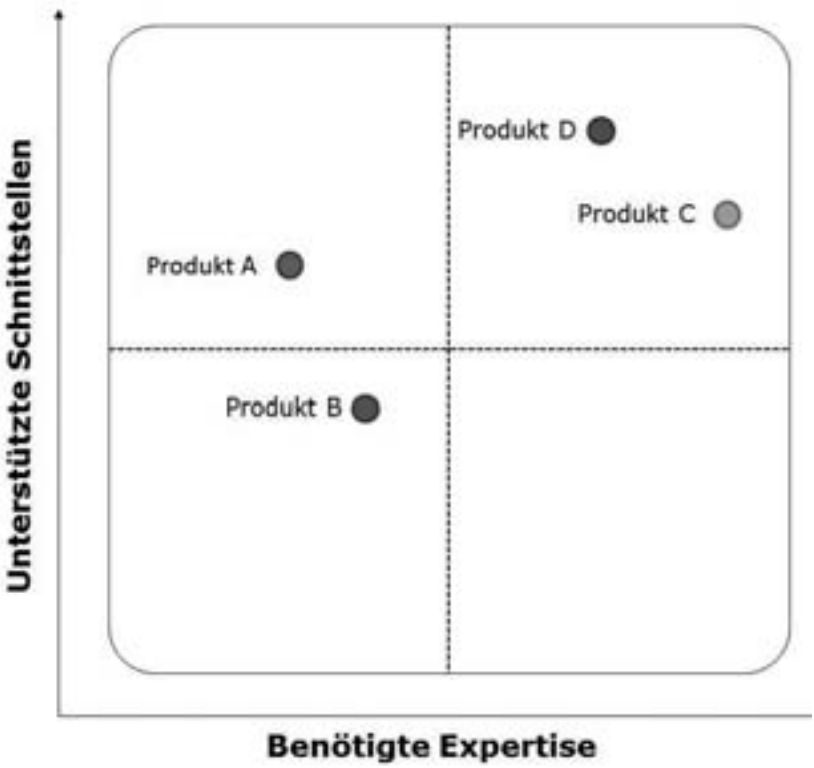

Abbildung 10: Vergleich von Fuzzern anhand von Schnittstellen und benötigter Expertise

3.4 Fallstudie: Erfolgreiches Fuzzing

In einer Fallstudie des Projekts SoftSCheck der Hochschulle Bonn-Rhein-Sieg wurde unter Einsatz von Fuzzing-Tools in einer kommerziellen (daher namentlich nicht genannten) Web-Applikation eine Vielzahl von Sicherheitslücken identifiziert. Darunter fallen u.A. Cross Site Scripting-, Cross Site Request Forgery-, Session Fixation- und Authentifizierungssicherheitslücken [Ow10]. Die Schweregrade der Sicherheitslücken wurde nach dem Common Vulnerability Scoring System [MSR07] errechnet und auf die Kritikalitäten „Kritisch“, „Hoch“, „Mittel“, „Niedrig“ abgebildet (Abbildung 11: Kritikalität identifizierter Sicherheitslücken eines Fallbeispiels). Je kritischer eine Sicherheitslücke ist, desto größer sind der potenzielle Schaden und desto geringer der Aufwand zur Ausnutzung der Sicherheitslücke.

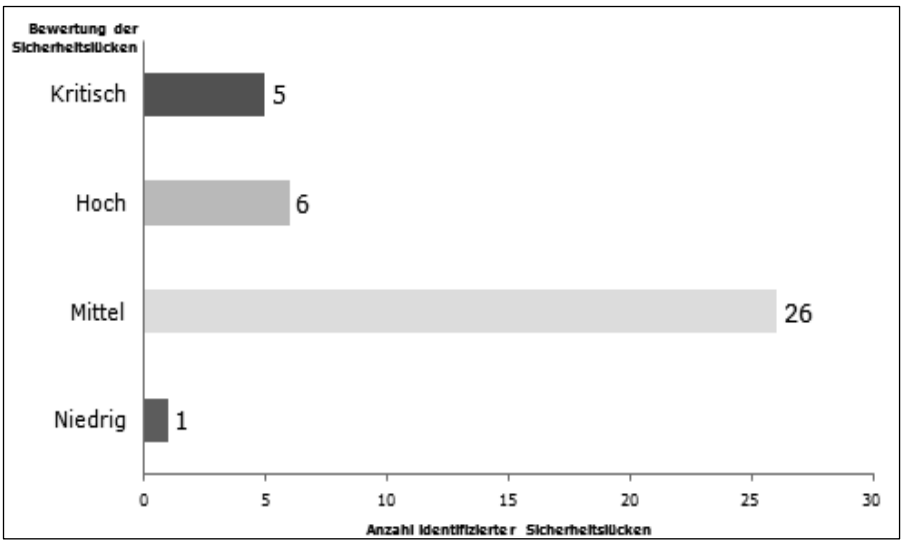

Abbildung 11: Kritikalität mit Fuzzing identifizierter Sicherheitslücken eines Fallbeispiels

4 Anwendung auf Telematikanwendungen im Gesundheitswesen

Die Normen DIN EN 61508 und DIN EN 62304 beschreiben Sicherheitsanforderungen für die Entwicklung von Software im medizinischen Umfeld. Diese beinhalten u.a. Vorschriften zur Verifikation und Diagnose (Kapitel C5, DIN EN 61508-7 [Di09]), Beurteilung der funktionalen Sicherheit (Kapitel C6, DIN EN 61508-7 [Di09]), Implementierung und Verifikation von Software-Einheiten (Kapitel 5.5, DIN EN 62304 [Di07]) und Prüfung des Softwaresystems (Kapitel 5.7, DIN EN 62304 [Di07]). Diese Forderungen und auch höherwertige Anforderungen an das Sicherheitsniveau können durch die Anwendung von Threat Modeling und Fuzzing in vollem Umfang erfüllt werden. Durch Anwendung von Threat Modeling auf die Gesamtstruktur der Gesundheitskarte, sowie auf einzelne Bestandteile dieser, können Bedrohungen und Sicherheitslücken frühzeitig erkannt und mit minimalen Kosten behoben werden. Durch Anwendung von Fuzzing können - durch klassische Verfahren nicht erkannte - Sicherheitslücken identifiziert werden. Die Anwendung der Verfahren Threat Modeling und Fuzzing ist kosteneffizient und

ergänzend zu der Anwendung formalisierter Sicherheitsüberprüfungen (z.B. Common Criteria) durchzuführen. Derzeit wird eine Integration von Threat Modeling in die Common Criteria geprüft mit dem Ziel den Zertifizierungsprozess sicherer Software zu verkürzen. Durch den Einsatz von Threat Modeling und Fuzzing kann ein höheres Sicherheitsniveau der Telematikanwendungen im Gesundheitswesen erreicht werden.

5 Fazit / Ausblick

Die bisherigen Erfolge haben gezeigt, dass mit den Verfahren Threat Modeling und Fuzzing selbst bei Standardsoftware sehr viele aus dem Internet ausnutzbare kritische Sicherheitslücken gefunden werden konnten - trotz eines hohen Sicherheitsstandards in den Programmierrichtlinien [SP10]. Es ist zu erwarten, dass auch bei Software für die Gesundheitstelematik ähnlich hervorragende Ergebnisse erreicht werden.

Literaturverzeichnis

[Di07] DIN Deutsches Institut für Normung e.V. (Hrsg.): Medizingeräte-Software - Software-Lebenszyklus-Prozesse (IEC 62304:2006). Berlin 2007.

[Di09] DIN Deutsches Institut für Normung e.V. (Hrsg.): Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme - Teil 7: Anwendungshinweise über Verfahren und Maßnahmen (IEC 65A/528/CDV:2008). Berlin 2009.

[GKL08] Godefroid, P.; Kiezun, A.; Levin, M.Y.: Grammar-based Whitebox Fuzzing. o.O. 2008. http://research.microsoft.com/en-us/projects/atg/pldi2800.pdf

[HL06] Howard, M.; Lipner, Steve: The Security Development Lifecycle. SDL: A Process for Developing Demonstrably More Secure Software. Redmond 2006.

[Jo96] Jones, C.: Applied Software Measeurement. New York 1996.

[Ju10] Juhasz, S.: Dokumentation und Überprüfung des Sicherheitsniveaus von Individualsoftware: Einsatz der Bedrohungsmodellierung (Threat Modeling). Sankt Augustin 2010.

[LH05] Lipner, S.; Howard, M.: The Trustworthy Computing Security Development Lifecycle. o.O. 2005. http://msdn.microsoft.com/en-us/library/ms995349.

[Lü10] Lübbert, J.: Bewertung von Netzwerkprotokoll-Fuzzern. Entwicklung einer Taxonomie zur Klassifizierung und Aufbau von Parametern zur Evaluation von Fuzzern. Sankt Augustin 2010.

[MLY05] Myagmar, S.; Lee, Adam J.; Yurcik, W.: Threat Modeling as a Basis for Security Requirements. Symposium on Requirements Engineering for Information Security. Paris 2005. http://www.suvda.com/papers/threat_sreis05.pdf.

[MSR07] Mell, P; Scarfone, K; Romanosky, S: A Complete Guide to the Common Vulnerability Scoring System Version 2.0. o.O. 2007. http://www.first.org/cvss/cvss-guide.pdf

[My05] Myagmar S.: Threat Modeling networked and data-centric systems, Urbana 2005, http://www.projects.ncassr.org/threatmodeling/myagmar-msthesis.pdf

[Ni02] National Institute of Standards and Technology (NIST) (Ed.): The Economic Impacts of Inadequate Infrastructure for Software Testing. Gaithersburg 2002. http://www.nist.gov/director/prog-ofc/report02-3.pdf

[Ow09] OWASP Foundation (Ed.): Threat Risk Modeling. Columbia 2009. http://www.owasp.org/index.php/Threat_Risk_Modeling

[Ow10] OWASP Foundation (Ed.): Category:Vulnerability – OWASP. Columbia 2010. http://www.owasp.org/index.php/Category:Vulnerability

[Po07] Pohl, H.: Zur Technik der heimlichen Online Durchsuchung. DuD, Ausg. 31. 2007, 684 - 688.

[Sa09] Sakal, P.: Identifikation und Evaluation von Fuzzing Tools, zur Nutzenmaximierung bei der Identifikation und Lokalisierung sicherheitsrelevanter Softwarefehler. Sankt Augustin 2009.

[Sa10] SANS (Ed.): The Top Cyber Security Risks. o.O. 2010 http://www.sans.org/top-cyber-security-risks/?ref=top20

[Sc99] Schneier, B.: Attack Trees: Modeling Security Threats. In: Dr. Dobb's Journal, v. 24, n.12. San Francisco 1999. http://www.schneier.com/paper-attacktrees-ddj-ft.html

[Sc06] Schumacher, M.; Fernandez-Buglioni, E.; Hybertson, D.; Buschmann, F.; Sommerlad, P.: Security Patterns. Integrating Security and Systems Engineering. Hoboken 2006.

[Sc10] Schwab, F.; Findeisen, A.; Sakal, P.; Pohl, H.: Bedrohungsmodellierung (Threat Modeling) in der Softwareentwicklung.GI-Edition: Lecture Notes in Informatics. Berlin 2010.

[SGA07] Sutton, M.; Greene, A.; Amini, P.: Fuzzing – Brute Force Vulnerability New York 2007.

[SP10] Sakal, P.; Pohl, H.: Entwicklungshelfer und Stresstester - Tool-gestützte Identifizierung von Sicherheitslücken in verschiedenen Stadien des Softwarelebenszyklus. In: <kes> - Die Fachzeitschrift für Informations-Sicherheit, 2, 2010

[SS04] Swiderski, F.; Snyder, W.: Threat Modeling. Redmond 2004.

[TDM08] Takanen, A.; Demott, J.D.; Miller, C.: Fuzzing for Software Security Testing and Quality Assurance. Norwood 2008.

Elektronische Signaturen in Versorgungseinrichtungen des Gesundheitswesens - Maßnahmen und Einführungsunterstützung

Hagen Kosock, Judith Balfanz, Antje Brandner, Carl Dujat, Christopher Duwenkamp, Reinhold Haux, Nils Hellrung, Paul Schmücker, Christoph Seidel

Peter L. Reichertz Institut für Medizinische Informatik der Technischen Universität Braunschweig und der Medizinischen Hochschule Hannover
Mühlenpfordtstraße 23
D-38106 Braunschweig
Hagen.Kosock@plri.de
Judith.Balfanz@authentidate.de
Antje.Brandner@med.uni-heidelberg.de
dujat@promedtheus.de
christopher.duwenkamp@plri.de
reinhold.haux@plri.de
nils.hellrung@plri.de
p.schmuecker@hs-mannheim.de
c.seidel@klinikum-braunschweig.de

Abstract: Die Relevanz sicherer elektronischer Dokumentation nimmt im Gesundheitswesen immer weiter zu, dennoch entwickelt sich eine passende IT-Infrastruktur nur langsam. Um die Einführung sicherer elektronischer Dokumentation zu unterstützen wurde das Competence Center für die Elektronische Signatur e.V. (CCESigG) als neutrale Plattform des Gesundheitswesens gegründet. Dieser Artikel beschreibt die Zielsetzung des CCESigG, die zugrundeliegende Problematik bei der Einführung sicherer elektronischer Dokumentation und die geplanten Maßnahmen, um diese zu unterstützen. Darüber hinaus werden die Ergebnisse der „AG Dokumente“ des CCESigG vorgestellt und diskutiert. Ziel dieser Arbeitsgruppe ist eine praxistaugliche Systematik zusammenzustellen, die aussagt, welche patientenbezogenen Dokumente mit welchen Ausprägungen der elektronischen Signatur oder des Zeitstempels, basierend auf rechtlichen Anforderungen sowie praxisbasierten Empfehlungen, zu versehen sind. Hierbei wird einzelnen Dokumentengruppen eine Minimalanforderung an ein zu verwendendes Sicherheitsniveau zugeordnet. Das CCESigG erhofft sich, Entscheider im Gesundheitswesen bei der Einführung sicherer elektronischer Dokumentation und Archivierung zu unterstützen und Unsicherheiten bezüglich der rechtlichen Anforderungen zu beseitigen.

1 Einleitung

Integrierte rechnerunterstützte Dokumentenmanagement- und Archivierungssysteme können viele Probleme lösen, die aus den Anforderungen an die Dokumentation und Kommunikation in Einrichtungen des Gesundheitswesens resultieren [SDH08]. Um Sicherheit und Datenschutz bei der Umsetzung von elektronischen Patientenakten (EPA) zu gewährleisten, ist die Sicherstellung der grundlegenden Anforderungen der IT-Sicherheit, Vertraulichkeit, Verfügbarkeit, Authentizität und Integrität [Bs10] zwingend erforderlich. Integrität beinhaltet die Hauptaspekte Vollständigkeit und Unveränderheit. Zur Sicherstellung der Integrität müssen Attribute wie der Autor und der Zeitpunkt, zu dem ein elektronisches Dokument erzeugt beziehungsweise signiert wurde, eindeutig mit den digitalen Daten verknüpft werden [Bs10]. Gesetzliche Regelungen und technische Möglichkeiten ermöglichen durch den Einsatz qualifizierter elektronischer Signaturen beweis- und rechtssicheres elektronisches Dokumentenmanagement und rechtssichere elektronische Langzeitaufbewahrung [RS06, HR08]. Ein erheblicher Teil der Dokumentation in deutschen Krankenhäusern wird bereits originär elektronisch erzeugt und mehr als 70 Prozent der deutschen Krankenhäuser implementieren oder planen aktuell die Implementierung einer EPA [Hü10]. Dennoch entwickeln sich im Gesundheitswesen nur langsam IT-Strukturen, die den Einsatz von elektronischen Signaturen unterstützen, obwohl die Vorteile von elektronischer Datenverarbeitung und Aufbewahrung von Patientenakten allgemein bekannt sind. Neben anderen Vorteilen sind elektronische Patientenakten zu jedem Zeitpunkt, unabhängig vom Standort, schnell und für verschiedene autorisierte Personen gleichzeitig verfügbar [Le06]. Darüber hinaus besteht ein wirtschaftliches Potential. Elektronische Signaturen sind bei der Kommunikation und Aufbewahrung von elektronischen Patientenakten notwendig, um rechtliche Risiken zu vermeiden und die Verkehrsfähigkeit über lange Zeiträume zu gewährleisten. Durch unklare technische, prozedurale und organisatorische Anforderungen, wenige praktische Umsetzungen und komplexe gesetzlichen Grundlagen im Gesundheitswesen resultieren Unsicherheiten, die viele Krankenhäuser veranlassen, nur zögerlich ausschließlich auf elektronische Dokumentenmanagement- und Archivierungssysteme zu setzen. Papier- oder mikrofilmbasierte Archivsysteme werden weiterhin eingesetzt, selbst wenn integrierte elektronische Systeme vorhanden sind. Diese doppelte Archivierung von Patientenakten führt zu erheblichen Kosten.

Basierend auf dem Signaturgesetz (SigG) und weiteren gesetzlichen Regelungen, wie z.B. § 126 Abs. 3 Bürgerliches Gesetzbuch (BGB) und § 371a der Zivilprozessordnung (ZPO), sind elektronische Signaturen ein zentrales Element, um sichere und gesetzeskonforme elektronische Dokumentenmanagement- und Archivierungssysteme zu implementieren. Verschiedene Institutionen und Projektgruppen haben den Umgang mit elektronischen Signaturen im Gesundheitswesen untersucht. Dennoch gibt es aktuell nur wenige praktische Umsetzungen im Gesundheitswesen. Problemlösungen und Anforderungsanalysen müssen von jeder Einrichtung erneut eigenständig erstellt werden, auch wenn diese bereits von anderen gelöst wurden. Demzufolge ist ein möglichst weitgreifender Austausch von Erfahrungen erforderlich, um den Lernprozess zu beschleunigen [Le08].

Im März 2009 wurde das Competence Center für die Elektronische Signatur im Gesundheitswesen (CCESigG) e.V. gegründet [Cc10]. Der Verein unterstützt Entscheider, Projektleiter, Projektdurchführende und Berater bei der Einführung von elektronischen Signaturen und Zeitstempeln. Die Akkumulation von bundesweiten Informationen kann verhindern, dass einzelne Institutionen des Gesundheitswesens Problemstellungen im Zusammenhang mit elektronischen Signaturen von Grund auf selbständig lösen müssen. Als neutrale Plattform für Kliniken, Institutionen und Arbeitsgruppen, Hersteller, Trustcenter und Dienstleister vermittelt das CCESigG plausible Methoden und erprobte Lösungen, um einen Markt zu schaffen, damit die Effizienzvorteile gesetzeskonformer und sicherer digitaler Kommunikation sich zügig im Sektor durchsetzen und allen Beteiligten zugute kommen. Das CCESigG trägt dazu bei, Unsicherheiten bezüglich der elektronischen Signatur im Gesundheitswesen abzubauen. Der Verein kooperiert mit anderen Institutionen, Organisationen und Arbeitsgruppen, welche sich mit elektronischen Signaturen befassen. Er erarbeitet "Best Practice"-Lösungen und unterstützt Bemühungen zur Entwicklung von allgemeinen, wenn möglich internationalen Standards für gegebenenfalls gesundheitswesenspezifische Schnittstellen im Zusammenhang mit elektronischen Signaturen, deren Validierung und Archivierung. Das CCESigG bietet Versorgungseinrichtungen des Gesundheitswesens Unterstützung an um diese zu motivieren, elektronische Signaturen einzusetzen sowie an Pilotprojekten und am Informationsaustausch mit Hilfe der neutralen Plattform teilzunehmen.

Aktuell setzt sich der Verein aus über 20 Mitgliedern verschiedener Interessengruppen zusammen. Über diese Mitglieder hinaus bestehen Kooperationen und Kontakte mit Mitgliedern und Arbeitsgruppen der GMDS, des BSI und des VHitG. Notwendige Grundlagen werden in Arbeitsgruppen erarbeiten. Es sind existierende Kommunikationsstandards und Schnittstellen auf ihre Fähigkeit zur Unterstützung der elektronischen Signatur zu prüfen und Richtlinien und Empfehlungen für zukünftige Entwicklungen zu erarbeiten. Ferner ist eine systematische Aufarbeitung von Dokumenten einer elektronischen Patientenakte bezüglich der Unterschrifts- bzw. Signaturnotwendigkeit zwingend erforderlich.

2 Zielsetzung

Laut Seidel [Se10] ist der entscheidende Vorteil elektronischer Dokumenten-management- und Archivierungssysteme im Gesundheitswesen die bessere Unterstützung der Behandlungsprozesse durch:

- die Erhöhung der Verfügbarkeit, durch Einsehbarkeit elektronischer Dokumente ohne Warte- und Lieferzeit nahezu an jedem Ort eines Krankenhauses,

- die nahezu vollständige Reduzierung des Verlustes von Dokumenten einschließlich der Vermeidung von z.B. Doppeluntersuchungen und

- die Qualitätssteigerung des Behandlungsprozesses durch die Erhöhung der Verfügbarkeit der Daten sowie die Einbettung der Daten in entscheidungsunterstützende Systeme.

Ferner wird herausgestellt, dass diese Vorteile jedoch nur dann zum Tragen kommen, „wenn nicht gleichzeitig ein Papierarchiv zur Gewährleistung der Beweissicherheit geführt wird. Ein Verzicht auf Papierarchivierung wird nur durch den Einsatz elektronischer Signaturen in elektronischen Dokumentenmanagement- und Archivierungssystemen ermöglicht“ [Se10].

Die Umsetzung elektronischer Signaturen und Zeitstempel in Prozessen der Patientenversorgung stellt für Kliniken eine Herausforderung dar. Dabei ist die Verwendung von elektronischen Signaturen mit finanziellem und organisatorischem Aufwand verbunden. Verschiedene Verfahren bergen unterschiedliche Aufwände und erzeugen Dokumentation mit unterschiedlich hohem Beweiswert. Somit ist es sinnvoll als Basis zu prüfen, welche Dokumente, z.B. einer elektronischen Patientenakte, mit welcher Qualität einer elektronischen Signatur bzw. eines Zeitstempels zu versehen sind. Die geltenden gesetzlichen Regelungen sind im Hinblick auf die Signatur- bzw. Unterschriftsnotwendigkeit im Gesundheitswesen unübersichtlich und uneinheitlich. Eine systematische Betrachtung der notwendigen Anforderungen an die Sicherungsmaßnahmen für klinische Dokumente existierte bisher nicht.

Infolgedessen ist es Ziel eine praxistaugliche Systematik für Dokumente einer Patientenakte hinsichtlich ihrer Unterschrifts- bzw. Signaturnotwendigkeit zu erstellen und dabei die entsprechenden gesetzlichen Vorschriften, Regelungen, Ausnahmen und Besonderheiten zu beachten.

3 Methoden

Um die Zielsetzung zu erreichen wurde eine Ist-Analyse der Sicherungsmaßnahmen hinsichtlich elektronischer Signaturen und Zeitstempel durchgeführt. Zur Identifikation möglicher Sicherungsmaßnahmen und deren Anforderungen sind die dazu bestehenden Gesetze, Normen, Leitfäden, Standards und Konzepte im Detail betrachtet worden. Mit Hilfe der Ergebnisse der Analyse wurde eine gesetzeskonforme Einteilung verschiedener Sicherungsmaßnahmen hinsichtlich elektronischer Signaturen und Zeitstempel definiert.

Darüber hinaus wurden die Dokumente papierbasierter und gescannter Patientenakten in mehreren Kliniken systematisch untersucht. Hierzu sind über 1000 verschiedene Papierdokumente einer Patientenakte[1] hinsichtlich der verwendeten Unterschriftenart und Rolle analysiert worden. Jedes papierbasierte Dokument wurde dahingegen überprüft welche Unterschriften oder Handzeichen, mit oder ohne Datumsangabe von welcher Gruppe des medizinischen Personals erbracht wird. Die auf den Papierdokumenten aufgebrachten Signaturen wurden der entsprechenden Inhalts- bzw. Ereignisbekundung zugeordnet.

Um die rechtlich festgelegten Mindestanforderungen hinsichtlich der notwendigen elektronischen Signaturen und Zeitstempel zu bestimmen wurden mehr als 50 in Deutschland relevante Werke systematisch analysiert. Diese umfassten Gesetze (z.B. Betäubungsmittelgesetz, Bundesdatenschutzgesetz, SGB V, Landeskrankenhausgesetze), Verordnungen (z.B. Arzneimittelverschreibungsverordnung), Verträge (z.B. Bundesmantelvertrag für Ärzte), Regelungen und Richtlinien (z.B. Richtlinie nach der Verordnung über den Schutz vor Schäden durch ionisierende Strahlen). Die Werke wurden mit Hilfe von Rechtsexperten bezüglich enthaltener Signatur- und Aufbewahrungspflichten geprüft.

Auf Grundlage der zuvor ermittelten Ergebnisse wurden in mehreren Expertenworkshops Dokumentengruppen festgelegt und diesen die mindestens notwendigen Sicherungsmaßnahmen zugeordnet. Dazu wurden die rechtlichen Grundlagen von den teilnehmenden Vertreter aus Krankenhäusern, Ärztekammern, Krankenhausgesellschaften, Industrie/Herstellern, wissenschaftlichen Institutionen, Fachgesellschaften und Rechtsexperten herangezogen und durch praktische Einflussfaktoren bei der Definition der Systematik ergänzt.

4 Ergebnisse

Grundlegend können sieben verschiedene Sicherungsmaßnahmen im Zuge der Einführung von elektronischen Signaturen im Gesundheitswesen unterschieden werden. Diese erstrecken sich von einem geeigneten Authentifizierungsverfahren nach dem aktuellen Stand der Technik bis hin zu qualifizierten elektronischen Signaturen mit Anbieterakkreditierung nach Signaturgesetzt. Diese qualifizierten elektronischen Signaturen können die Authentizität, Integrität, Vollständigkeit und Verkehrsfähigkeit der signierten Dokumente über einen Aufbewahrungszeitraum von mindestens 30 Jahren sicherstellen. Jedoch gehört auch die weiterhin zusätzliche Aufbewahrung in Papierform zu den möglichen Sicherungsmaßnahmen. Die ausführliche Auflistung der definierten Maßnahmen kann der Tabelle 1 entnommen werden.

[1] Diese Analyse umfasste den Großteil in einem Klinikum vorhandenen bzw. auftretenden papierbasierten Dokumententypen, welche in einer kategorisierten Musterpatientenakte zusammengefasst vorlagen.

Grafik	Bezeichnung	Kurzbeschreibung
	Papierform	Das Dokument sollte weiterhin in Papierform aufbewahrt werden. Es kann parallel elektronisch zur Einsicht vorgehalten werden.
	Geeignetes Authentifizierungsverfahren	Eine gesicherte Identifikation einer Person gegenüber einem DV-System und eine daraus resultierende eindeutige Zuordnung der durch diese Person erzeugten Informationen und Dokumente zu dieser.
elektronische Signaturen:		
	Keine elektronische Signatur	Es werden keine Anforderungen bezüglich einer elektronischen Signatur an das Dokument gestellt.
	Einfacher Zeitstempel	Eine Verknüpfung der Systemzeit oder Serverzeit des Institutionsrechenzentrums mit dem Dokument bzw. die Angabe des Zeitpunkts im Dokument wird empfohlen. Dies unterliegt der freien Beweiswürdigung des Richters. Für die Erhöhung des Beweiswerts werden zusätzliche Sicherheitsmaßnahmen empfohlen, z.B. Protokollierung und Handlungsanweisungen.
	fortgeschrittene elektronische Signatur	Eine fortgeschrittene elektronische Signatur gemäß § 2 Nr. 2 SigG wird empfohlen.
	Qualifizierter Zeitstempel mit Anbieterakkreditierung	Ein qualifizierter elektronischer Zeitstempel mit Anbieterakkreditierung (ausgestellt durch einen Zertifizierungsdiensteanbieter gemäß § 2 Nr. 14 SigG) wird empfohlen.
	Qualifizierte elektronische Signatur mit Anbieterakkreditierung	Eine qualifizierte elektronische Signatur mit Anbieterakkreditierung gemäß § 2 Nr. 3 SigG wird empfohlen (d.h. Smart Card, Kartenleser, gesetzeskonforme Signatursoftware und Akkreditierung des Zertifizierungsdiensteanbieters erforderlich)

Tabelle 1: Übersicht verschiedener Sicherungsmaßnahmen [Se10]

Herauszustellen ist, dass Sicherungsmaßnahmen wie die fortgeschrittene bzw. qualifizierte elektronische Signatur durch das Signaturgesetz und die Signaturverordnung definiert und z.B. durch die Bundesnetzagentur (BNetzA) und das Bundesamt für Sicherheit in der Informationstechnik (BSI) konkretisiert sind. Hingegen bieten sich bei der Ausgestaltung des geeigneten Authentifizierungsverfahrens unterschiedliche Möglichkeiten, die jedoch ein definiertes Maß an Sicherheit auch über längere Zeiträume hinweg gewährleisten sollten.

Die aufgelisteten Verfahren bergen unterschiedliche Aufwände und erzeugen Dokumentationen mit unterschiedlich hohem Beweiswert. Der Beweiswert beeinflusst die Wahrscheinlichkeit, eine Tatsache in einem Prozess vor Gericht beweisen zu können. So hat z.B. der Beweis durch Sachverständige einen höheren Beweiswert als die Aussagen von Zeugen [Ro07]. Eine Übersicht bezüglich des Beweiswertes der verschieden Sicherungsmaßnahmen bietet Abbildung 1. Der Großteil der möglichen elektronischen Verfahren unterliegt vor Gericht der freien Beweiswürdigung durch den Richter. Je nach Ausgestaltung bei der Umsetzung kann der Beweiswert dieser Verfahren verschieden ausfallen. Darüber hinaus sind durch das Gesetz Beweisregeln definiert. Beweismittel zu denen Beweisregeln bestehen, wie z.B. Urkunden (Erklärungen die z.B. durch eine qualifizierte elektronische Signatur bestätigt wurden), besitzen einen höheren Beweiswert als Beweismittel die der freien Beweiswürdigung unterliegen. Herauszuheben ist, dass die freie Beweiswürdigung vor Gericht kein minderwertiges Mittel der Beweisführung darstellt.

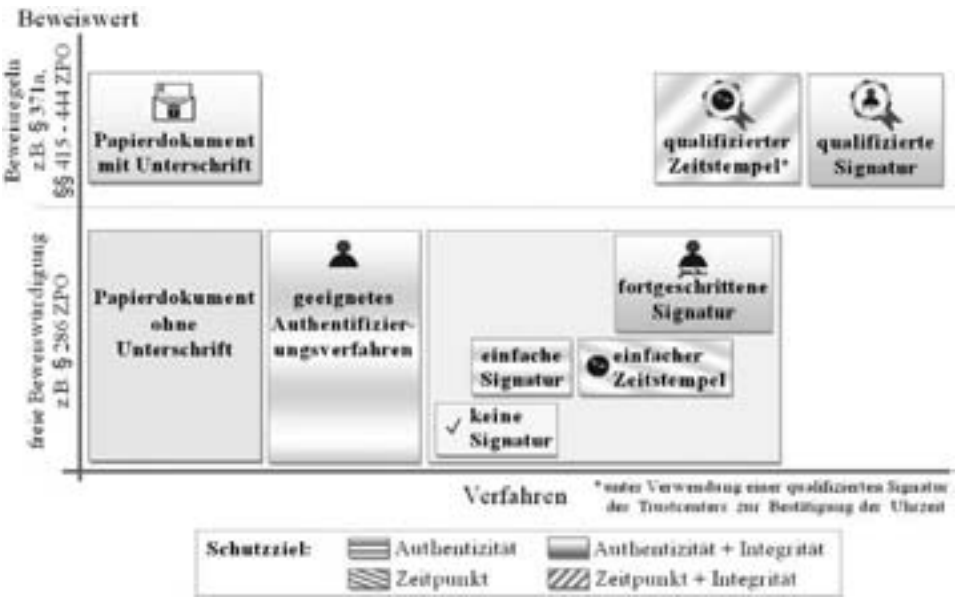

Abbildung 1: Einstufung der Signaturverfahren bezüglich des Beweiswerts [Se10]

Die Dokumentenanalyse in den Kliniken hat ergeben, dass die meisten Dokumente einer Patientenakte eine oder mehrere Unterschriften bzw. Handzeichen tragen. So werden z.B. Arztbriefe und andere nach extern übermittelte Dokumente in Papierform in der Regel von mehreren Ärzten unterschrieben. Ebenso werden Papierdokumente der Funktionsdiagnostik, Laborbefunde (Pathologie, Histologie, Zytologie) und Befunde im Allgemeinen, Briefe, Bereichte, Konsildokumentation, teils separate Diagnosedokumentation (nicht in Arztbrief eingebettet), Leistungsanforderungen (z.B. Blutbild), Anordnungen (z.B. Verordnung von Arzneimitteln) und Therapieplanung jeweils mit einer Unterschrift bestätigt. Patienteneinwilligungen, Patientenverträge und Anamnesebögen, die der Patient ausfüllt werden, müssen generell vom Patienten unterschrieben werden. Hingegen werden ca. 90% der papierbasierten Anamneseaufzeichnungen vom medizinischen Personal nicht unterzeichnet. Die papierbasierte Maßnahmendokumentation (z.B. Pflegekurven und Verlaufdokumentation) wird hauptsächlich durch Handzeichen des medizinischen Personals bestätigt. Die papierbasierte Entlassungs- und Verlegungsdokumentation innerhalb einer Klinik wird ebenfalls vorwiegend per Handzeichen oder Unterschrift abgezeichnet. Administrative Informationen werden wenn nur vom Patienten mit einer Unterschrift bestätigt. Automatisch erzeugte Bilddaten, Signale und andere technische Aufzeichnungen werden meist nicht unterschrieben, es sei denn diese sind auffällig bzw. in einen Befund/Bereicht eingebettet. Herauszuheben ist, dass die Dokumentation von Anforderungen, Planung und Befundung der Radiologie, Nuklearmedizin und des klinischen Labors den prozentual größten Anteil der unterschriebenen papierbasierten Dokumentation einer Patientenakte ausmachen.

„Wird in Gesetzestexten oder Verordnungen die Schriftform gefordert, bedeutet dies, dass das jeweilige Papierdokument mit einer Unterschrift versehen werden muss. Die Schriftform kann grundsätzlich gemäß § 126 Abs. 3 BGB durch die elektronische Form unter Verwendung einer qualifizierten elektronischen Signatur gemäß Signaturgesetz ersetzt werden. Generell ist es somit möglich, im elektronischen Rechtsverkehr den Papierurkunden gleichwertige elektronische Dokumente zu erzeugen, wenn diese mit einer qualifizierten elektronischen Signatur versehen sind.“ [Se10]. Die Gegenüberstellung in Tabelle 2 macht deutlich, wie gesetzlich geregelte Signaturvorschriften für Papierdokumente äquivalent rechtssicher für originär elektronische Dokumente umgesetzt werden können.

Papierdokument	Elektronisches Dokument
Unterschrift gefordert, z.B. durch die Schriftform in einer gesetzlichen Regelung	Gesetzlich festgeschrieben: Qualifizierte elektronische Signatur gemäß Signaturgesetz notwendig
Papierdokument explizit gefordert, z.B. durch Musterpapierformulare in einer gesetzlichen Verordnung	Gesetzlich festgeschrieben: Verbot der elektronischen Form
Handzeichen / Namenskürzel	Nicht gesetzlich festgeschrieben: Umsetzung durch fortgeschrittene

	elektronische Signatur oder geeignetes Authentifizierungsverfahren möglich

Tabelle 2: Äquivalente der Signaturen für Papier- und elektronische Dokumente [Se10]

Die weitere Analyse der rechtlichen Anforderungen hat ergeben, dass zwar Aufbewahrungsfristen für die medizinische Dokumentation umfangreich geregelt sind, jedoch nur wenige gesetzliche Regelungen explizit die Schriftform und somit eine Unterschrift fordern. Am häufigsten wird die Unterschrift und somit die qualifizierte elektronische Signatur des Patienten und teils des Arztes bei Patienteneinwilligungen, -aufklärungen und -verträgen gefordert.

Expertenworkshops ergaben, dass sich bei der Umsetzung von elektronischen Signaturen im Gesundheitswesen „zahlreiche Gestaltungsmöglichkeiten und Freiheiten innerhalb der Einrichtungen unter Berücksichtigung der Vorgaben des Gesetzgebers und der Fachgesellschaften ergeben“ [Se10]. Die über die gesetzlichen Regelungen hinaus in der Praxis etablierte umfangreiche Verwendung von Unterschriften auf Papierdokumenten sollte nicht eins zu eins bei der Umstellung auf ein elektronisches Dokumentenmanagement- und Archivierungssystem durch qualifizierte elektronische Signaturen ersetzt werden. Vielmehr sind häufig keine Signaturen erforderlich oder ein geeignetes Authentifizierungsverfahren ausreichend. Einrichtungen des Gesundheitswesens sollten ihr individuelles Beweisinteresse für die jeweilige Dokumentation und somit die zu verwendende Sicherungsmaßnahme festlegen.

Als Unterstützung bei der Umsetzung von elektronischen Signaturen kann das Resultat der Expertenworkshops, die „Empfehlungen für den Einsatz elektronischer Signaturen und Zeitstempel in Versorgungseinrichtungen des Gesundheitswesens“ [Se10], herangezogen werden. Diese pragmatische und individuell verwendbare Empfehlung nimmt eine Klassifizierung von nahezu allen patientenbezogenen Dokumenten vor, die im Klinikalltag anfallen, spricht anhand der Systematik aus Tabelle 3 Signaturempfehlungen aus und gibt Hinweise zum ersetzenden Scannen von papierbasierten Patientenakten. Die in Tabelle 3 aufgeführten Signaturempfehlungen beziehen sich auf die in Tabelle 1 definierten Sicherungsmaßnahmen. Existieren innerhalb einer Dokumentengruppe Abweichungen von der empfohlenen Sicherungsmaßnahme, z.B. durch spezielle Gesetze, ist dies in der Spalte „Spezielle Anforderungen“ vermerkt. In die Systematik sind neben den gesetzlichen Grundlagen besonders praktische Erfahrungen, technische Voraussetzungen und Überlegungen bezüglich des jeweiligen Beweisinteresses aus Sicht der Einrichtungen, des medizinischen Personals, aber auch des Patienten eingeflossen. So wird z.B. für Arztbriefe die Verwendung einer qualifizierten elektronischen Signatur empfohlen, obwohl keine gesetzliche Notwendigkeit einer Unterschrift bzw. elektronischen Signatur besteht. Dies begründet sich vor allem darin, dass beim (externen) Empfänger vom Arztbrief ärztliche Entscheidungen für die Patientenbehandlung abhängig gemacht werden. Hingegen ist bei der elektronischen Pflegedokumentation meist ein geeignetes Authentifizierungsverfahren ausreichend, um die entsprechende Dokumentation, z.B. in einer digitalen Pflegekurve, einer Person zuzuordnen. Ausführliche Erläuterungen der allgemeinen Signaturmaßnahmenempfehlungen aus Tabelle 3 und die Auflistung vorhandener Sonderfälle kann den „Empfehlungen für den Einsatz elektronischer Signaturen und Zeitstempel in Versorgungseinrichtungen des Gesundheitswesens“ [Se10] entnommen werden.

Herkunft	Kategorie	Dokumenten-gruppe	Signatur-empfehlung	Spezielle Anforderungen
Extern erhaltene elektronische Dokumente		Beliebige	 Eingangs-zeitstempel	--
Intern erstellte elektronische Dokumente (vom Leistungserbringer selbst erstellte Dokumente)	Aufnahme	Anamnese		--
	Diagnostik / Therapie	Anforderung		vorhanden
		Bilder, Signale, technische Aufzeichnungen etc.		vorhanden

		Diagnosen		--
		Anordnungen und Therapieplanung		vorhanden
		Befunde, Berichte und Konsile	Interne:  Externe: 	--
	Pflege	Pflegedokumentation		vorhanden
	Entlassung	Arztbriefe		--
	Adminis-tration	Bescheinigungen / Atteste		--
		Einwilligungen		--
		Verträge		--
	Sonstige		Einzeln zu betrachten	möglich

Tabelle 3: Übersicht der Hauptdokumentengruppen mit allgemeinen Signaturempfehlungen [Se10]

Die Quintessenz der in den Expertenworkshops erarbeiteten Ergebnisse kann in den folgenden zehn „Braunschweiger Regeln zur Archivierung mit elektronischen Signaturen im Gesundheitswesen“ [Se10] zusammengefasst werden:

1. Generelle Verwendung archivgeeigneter Dateiformate (z.B. PDF/A) sowie qualifizierter elektronischer Signaturen und Zeitstempel mit Anbieterakkreditierung durch die Bundesnetzagentur (nachfolgend als akkreditierte Signatur bzw. akkreditierter Zeitstempel bezeichnet).

2. Akkreditierte Signatur originär elektronischer Dokumente, für die gesetzliche Regelungen, die Schriftform fordern (grundsätzlich kann die Schriftform – unterschriebenes Papierdokument – gemäß § 126a Abs. 1 BGB durch die elektronische Form ersetzt werden).

3. Akkreditierte Signatur für Dokumente zur externen Verwendung und für interne Dokumente, die einen besonders hohen Stellenwert (z.B. Beweisinteresse) haben.

4. Akkreditierter („Eingangs-“) Zeitstempel für Dokumente externer Einsender. (Kann auch durch Regel Nr. 6 umgesetzt werden).

5. Geeignetes Authentifizierungsverfahren für alle sonstigen Dokumente.

6. Zeitnahe Archivierung der Dokumente, Protokoll- und Verifikationsdaten in einem revisionssicheren Archiv mit akkreditiertem („Archiv-“) Zeitstempel, in jedem Fall innerhalb von maximal 24 Stunden nach Erstellung oder Erhalt.

7. Absicherung des Betriebes des elektronischen Archivs nach dem Stand der Technik durch Umsetzung allgemein anerkannter Regelungen und Normen (z.B. ISO 27001, BSI) - im Idealfall Nachweis durch ein Zertifikat.

8. Hash- und Signaturerneuerungen gemäß den Vorgaben der Bundesnetzagentur; Datei- und Medienkonvertierungen gemäß den Empfehlungen der BMWi-Studie TransiDoc[Ro09].

9. Generelle Vermeidung von Medienbrüchen. Falls dennoch ersetzendes Scannen erforderlich ist:

    a. Aufbewahrung der Originaldokumente, für die gesetzliche Regelungen die Schriftform fordern.

    b. Verwendung eines abgesicherten Scanverfahrens nach dem Stand der Technik mit akkreditierter Signatur und / oder akkreditiertem Zeitstempel durch qualifiziertes eigenes Personal oder einen geeigneten externen Dienstleister.

    c. Sicherstellung des uneingeschränkten Fortbestands des Versicherungsschutzes.

10. Dokumentation und Handlungsanweisungen hinsichtlich der Verfahren, des Einsatzes der Signatur und weitergehender Regelungen (Verantwortlichkeiten, Datenschutz, Aktenstruktur etc.) in einer Archivordnung.

5 Diskussion und Ausblick

Die Rechtsgrundlagen bezüglich der Aufbewahrungsfristen und Signaturnotwendigkeit im Gesundheitswesen sind sehr umfangreich und heterogen. Die daraus entstandene Unübersichtlichkeit und Unsicherheit bei den Anwendern hat dazu beigetragen, dass in der Praxis bei der papierbasierten Patientendokumentation erheblich mehr Unterschriften geleistet werden als notwendig. Eine Vereinheitlichung der Gesetzesgrundlagen kann mehr Übersichtlichkeit schaffen.

Bei der Umstellung der medizinischen Dokumentation auf originär elektronische Dokumentation und Archivierung wird eine direkte und unbewertete Übersetzung der auf Papierdokumenten geleisteten Unterschriften in qualifizierte elektronische Signaturen, welche hohe organisatorische, ökonomische und technische Anforderungen stellen, für die äquivalenten elektronischen Dokumente als kritisch erachtet. Die Sicherstellung der Authentizität von elektronischen Dokumenten ist z.B. auch durch geeignete Authentifizierungsverfahren oder fortgeschrittene elektronische Signaturen erreichbar. Erfahrungen zeigen, dass das Personal im Gesundheitswesen selbst nach der Einarbeitung in die Thematik der elektronischen Signatur das aktuell praktizierte Unterschriftsverhalten meist eins zu eins in qualifizierte elektronische Signaturen umsetzen möchte. Eine Sensibilisierung, dass das zu verwendende Verfahren individuell anhand der rechtlichen Grundlagen und des jeweiligen Beweisinteresses geprüft werden sollte, ist notwendig. Eine praxisnahe und systematische Betrachtung, unter umfassender Berücksichtigung der rechtlichen Werke, existierte diesbezüglich bisher nicht. Es wird angenommen, dass die erarbeitet Systematik der „Empfehlungen für den Einsatz elektronischer Signaturen und Zeitstempel in Versorgungseinrichtungen des Gesundheitswesens“ zur Sensibilisierung bezüglich elektronischer Signaturen und Zeitstempel im Gesundheitswesen beiträgt und Grundlage zum Abbau von Unsicherheiten ist. Die systematische Untersuchung und die Modellierung der konkreten Ausgestaltung und Anwendung elektronischer Signaturen in klinischen Prozessen, aufbauend auf der ermittelten Systematik, stehen noch aus. Dabei ist die Einbindung medizinischen Fachpersonals aber auch insbesondere von Herstellern essentiell. Die Empfehlungen [Se10] sollten in weiteren Arbeiten bezüglich ihrer technischen Umsetzung in klinischen Prozessen beschrieben und evaluiert werden. Die in Zukunft durchzuführenden Prozessanalysen und entwickelten Empfehlung müssen unter anderem Hersteller dabei unterstützen elektronische Signaturen in die klinischen Anwendungssysteme einzubetten. Durch eine tiefe Integration elektronischer Signaturen in die Prozesse des Gesundheitswesens können Akzeptanzprobleme überwunden werden.

Zahlreiche Kliniken und Hersteller zeigen großes Interesse an der Nutzung von elektronischen Signaturen und dem damit verbundenen Einsatz von elektronischen Dokumentenmanagement- und Archivierungssystemen im Gesundheitswesen. Verschiedene rechtskonforme Lösungen sind bereits jetzt am Markt verfügbar. Unsicherheiten bestehen jedoch bei der Ablösung der auf Papier beziehungsweise Mikrofilm basierenden Langzeitaufbewahrung der Dokumentation der Patientenbehandlung. Besonders rechtlichte Risiken spielen eine große Rolle. Die Empfehlungen [Se10] diskutiert diese und sollte entsprechende Unsicherheiten abbauen.

Aktuell nutzen nur wenige Kliniken in Form von Pilotprojekten die Vorteile von elektronischen Signaturen. Es wird die Auffassung vertreten, dass Ausbau der Verwendung von elektronischen Signaturen und Zeitstempeln im Gesundheitswesen notwendig ist und mittelfristig möglichst flächendeckend in den Institutionen durchgeführt werden sollte. Jedoch wird eine weitreichende Umstellung auf ausschließlich elektronische Dokumentmanagement- und Archivierungssysteme innerhalb der nächsten 15 Jahre im deutschen Gesundheitswesen nicht als realistisch betrachtet. Daher sind alle Mitglieder und Kooperationspartner des CCESigG ermutigt, entsprechende Aktivitäten und Arbeitsgruppen einzuleiten und zu unterstützen um praxiserprobte Ergebnisse gemeinsam zusammenzuführen. Ferner ist das CCESigG bestrebt, die Basis der Mitglieder und Kooperationspartner umfangreich auszubauen damit alle Beteiligten von der neutralen Plattform profitieren und weitere fundierte Ergebnisse erarbeitet werden können.

Literaturverzeichnis

[Bs10] Bundesamt für Sicherheit in der Informationstechnik (BSI), IT Security Guidelines IT-Grundschutz in brief: siehe https://www.bsi.bund.de/cae/servlet/contentblob/475854/publicationFile/28013/guidelines_pdf.pdf. Letzte Einsicht am 13.04.2010; S. 8.

[Cc10] Competence Center für die Elektronische Signatur im Gesundheitswesen (CCESigG) e.V., siehe www.ccesigg.de. Letzte Einsicht am 14.4.2010.

[HR08] Hackel, S.; Roßnagel, A.: Langfristige Aufbewahrung elektronischer Dokumente. In: Klumpp, D; Kubicek, H.; Roßnagel, A.; Schulz, W. (Hrsg.): Informationelles Vertrauen für die Informationsgesellschaft. Springer, Berlin, Heidelberg, 2008; S. 199-207.

[Hü10] Hübner, U.; Sellemann, B.; Egbert, N.; Liebe JD.; Flemming D.; Frey, A.: IT-Report Gesundheitswesen – Schwerpunkt Vernetzte Versorgung: siehe http://l4asrv-2.wi.fh-osnabrueck.de/joomla/index.php?option=com_remository&Itemid=13&func=fileinfo&id=25. Letzte Einsicht am 14.04.2010; S. 46.

[Le06] Leiner, F.; Gaus, W.; Haux, R.; Knaup-Gregori, P.; Pfeiffer, KP.: Medizinische Dokumentation - Grundlagen einer qualitätsgesicherten integrierten Krankenversorgung. Lehrbuch und Leitfaden., Schattauer, Stuttgart, 2006 (5. Auflage).

[Le08] Lehmann, CU.; Altuwaijri, MM.; Li, YC; Ball, MJ; Haux, R.: Translational research in medical informatics or from theory to practice. A call for an applied informatics journal.: Methods Inf Med., 2008; 47: S. 1-3.

[RS06] Roßnagel, A.; Schmücker, P.: Beweiskräftige elektronische Archivierung - Bieten elektronische Signaturen Rechtssicherheit?. Economica, Heidelberg, 2006.

[Ro07] Roßnagel, A.; Fischer-Dieskau, S.; Jandt, S.; Knopp, M.: Langfristige Aufbewahrung elektronischer Dokumente – Anforderungen und Trends. Nomos, Baden-Baden, 2007; S. 47.

[Ro09] Roßnagel, A.; Schmidt, A.; Wilke, D. (Hrsg.): Rechtssichere Transformation signierter Dokumente – Anforderung, Konzepte und Umsetzung. Nomos, Baden-Baden, 2009.

[SDH08] Schmücker, P.; Dujat, C.; Häber, A.: Leitfaden für das rechnerunterstützte Dokumentenmanagement und die digitale Archivierung von Patientenunterlagen im Gesundheitswesen. 2. Ausgabe, Darmstadt: GIT Verlag 2008, S. 6.

[Se10] Seidel, C.; Kosock, H.; Brandner, A.; Balfanz, J.; Schmücker, P.: Empfehlungen für den Einsatz elektronischer Signaturen und Zeitstempel in Versorgungseinrichtungen des Gesundheitswesens. Shaker Verlag GmbH, Aachen, 2010.

OpeneGK – Benutzerfreundliche und sichere Authentisierung für Mehrwertdienste im Gesundheitswesen

Daniel Eske[1] · Detlef Hühnlein[1] · Sachar Paulus[2]
Johannes Schmölz[1,3] · Tobias Wich[1,3] · Thomas Wieland[3]

[1] ecsec GmbH, Sudetenstr. 16, 96247 Michelau,
{daniel.eske,detlef.huehnlein,johannes.schmoelz,tobias.wich}@ecsec.de

[2] paulus.consult, Am Mühlrain 21/2, 69151 Neckargemünd
sachar.paulus@paulus-consult.de

[3] Hochschule Coburg, Friedrich-Streib-Str. 2, 96450 Coburg
thomas.wieland@hs-coburg.de

Abstract: Dieser Beitrag zeigt, wie die elektronische Gesundheitskarte (eGK) in Verbindung mit dem OpenID-Protokoll bei web-basierten Mehrwertdiensten im Gesundheitswesen zur sicheren, datenschutz- und benutzerfreundlichen Registrierung und Authentisierung genutzt werden kann. Außerdem verspricht die Kombination mit dem weit verbreiteten OpenID-Protokoll eine schnellere Akzeptanz und Verbreitung der eGK-basierten Authentisierung im Internet.

1 Einleitung

Im Gesundheitswesen gibt es eine steigende Zahl von Web-basierten Diensten. Waren diese herkömmlicherweise eher Informationssammlungen und Nachschlagewerke, die zum Teil einfache Foren beinhalteten, so finden sich inzwischen mehr und mehr Lösungen, die stärker personalisierte Inhalte anbieten und sehr individuell auf die Bedürfnisse der einzelnen Nutzer eingehen – sei es zur gegenseitigen Selbsthilfe oder zur gezielten professionellen Beratung. Einige Anbieter machen bereits das Anlegen von elektronischen Gesundheitsakten im Internet möglich[1].

Anders als bei den meisten Anwendungen im Internet, wie etwa sozialen Netzwerken, geht es bei Web-Angeboten im Gesundheitsweisen um besonders sensible Informationen, nämlich Informationen über den Gesundheitszustand des Nutzers. Daher werden hohe Anforderungen an den Datenschutz an derartige Anwendungen gestellt: Jeder Nutzer muss allein darüber entscheiden können, an wen er welche Auskünfte weitergibt, und er muss sich zudem sicher sein können, dass diese Angaben nicht missbraucht oder weitergegeben werden. Diese Anforderungen bedeuten für die Entwicklung wie auch den operativen Betrieb eines Web-Dienstes, dass ein besonders hohes Maß an Sicherheit erreicht werden muss.

[1] Siehe z.B. http://www.gesundheitsakte.de, http://www.onmeda.de/ratgeber/gesundheitswesen/gesundheitsakte.html, https://www.lifesensor.com/de/de/ oder http://www.econmed.de/produkte/gesundheitsakte.html

Leider sind viele Anwendungen im Internet nicht in der Lage, diese Anforderungen zu erfüllen. Dies beginnt oft bereits bei der Authentisierung, die meist ausschließlich über Benutzername und Passwort abgewickelt wird. Der Anbieter kann sich nicht sicher sein, dass der Benutzer auch derjenige ist, der er vorgibt zu sein. Gleichzeitig hat der Nutzer keine Garantie, dass nicht Unbefugte, etwa durch Ausspähen (Phishing, Pharming) oder Erraten des Passworts, Zugriff auf seine Daten erlangen.

Damit personalisierte medizinische Mehrwertdienste im Internet Akzeptanz in der Bevölkerung finden, muss ein hoher Sicherheitsstandard angesetzt werden, gleichzeitig aber auch ein hoher Bedienungskomfort sichergestellt werden. Eine Möglichkeit dazu bietet die starke Authentisierung mit einem physikalischen kryptographischen Sicherheitstoken, z.B. einer Chipkarte. Dann benötigt ein Angreifer neben dem Wissen auch noch den Besitz des Authentizitätsnachweises, was das Risiko deutlich senken würde. Eine solche Chipkarte müsste aber den Benutzern in einfacher und weit verbreiteter Form zugänglich gemacht und genutzt werden, um bei Anbietern wie bei Benutzern wirklich akzeptiert und damit marktfähig zu werden. Spezialisierte, Dienste-spezifische Kartenlösungen einzelner Anbieter[2] können dies voraussichtlich nicht erreichen.

Die Deutsche elektronische Gesundheitskarte erfüllt genau diese Anforderungen. Sie soll an alle gesetzlich Versicherten ausgegeben werden, unterstützt eine starke Authentisierung und stellt damit im Prinzip ein attraktives Authentisierungswerkzeug für Web-basierte Anwendungen im Gesundheitswesen dar.

Allerdings sind die an die eGK geknüpften Vorgaben und Sicherheitsanforderungen vergleichsweise komplex, so dass vermutlich nur wenige Anbieter den Aufwand investieren werden, eine starke Authentisierung auf Basis der eGK innerhalb ihrer Anwendung zu realisieren. Wir schlagen daher eine Variante des OpenID-Protokolls[3] vor, in der die eGK zur sicheren, datenschutz- und benutzerfreundlichen Registrierung und Authentisierung genutzt wird. Hierdurch wird der komplexe Zugriff auf die eGK an den zentralen und speziell gesicherten OpeneGK-Dienst delegiert, den der einzelne Mehrwertanbieter nur noch für sich nutzbar machen muss. Für den Anwender ergibt sich damit eine einfache Registrierung bei einem neuen Web-basierten Mehrwertdienst und ein benutzerfreundliches Single Sign-On (SSO).

Durch die Verwendung des OpenID-Protokolls ist bereits vor dem Rollout der eGK eine (schwache) Authentisierung gegenüber dem OpeneGK-Dienst oder einem anderen OpenID-Provider möglich; nach dem flächendeckenden Rollout der elektronischen Gesundheitskarte kann das System dann für verschieden starke Stufen der Authentisierung genutzt werden. Damit kann für jeden Anwendungsfall die optimale Mischung aus Sicherheit und Benutzerfreundlichkeit gewählt werden.

Im Folgenden wollen wir zunächst (vgl. Abschnitt 2) das Konzept von OpenID vorstellen und einige relevante technische Aspekte der eGK hervorheben. In Abschnitt 3 stellen wir dann die Idee von OpeneGK vor und gehen näher auf die beteiligten Systemkomponenten (Bürgerclient, Mehrwertdienst und OpeneGK-Dienst) ein. Schließlich diskutieren wir in Abschnitt 4 die aus diesem Ansatz erwachsenden Konsequenzen und geben in Abschnitt

[2] Siehe beispielsweise `http://www.vita-x.de`.
[3] Siehe `http://openid.net/` und [Raep09].

5 einen Ausblick auf mögliche zukünftige Entwicklungen.

2 Grundlagen

In diesem Abschnitt werden die in diesem Papier benötigten Grundlagen zusammengetragen. Hierbei geht Abschnitt 2.1 auf das OpenID-Protokoll und Abschnitt 2.2 auf die relevanten Aspekte der elektronischen Gesundheitskarte ein.

2.1 OpenID

OpenID wurde ursprünglich entwickelt, um einen einfachen Login-Mechanismus für LiveJournal[4]-basierte Weblogs zu realisieren und zählt heute neben der Security Assertion Markup Language (SAML) [SAML(v2.0)] und dem Identity Metasystem Interoperability Profile [ID-MI(v1.0)] zu den vielversprechendsten Ansätzen für das Web-basierte Single Sign-On, die derzeit im Rahmen der Kantara-Initiative[5] harmonisiert werden sollen.

Beispielsweise wird OpenID von Google, Yahoo, MySpace und AOL, sowie etlichen weiteren namhaften Organisationen unterstützt [OpenID-Pro]. Seit kurzem zählt beispielsweise auch NTT docomo, der größte japanische Mobilfunkanbieter mit mehr als 55 Millionen Kunden [Saki10] zu den Unterstützern von OpenID. Bereits im Jahr 2009 wurde die Grenze von einer Milliarde OpenID-Benutzerkonten überschritten [Kiss09, Wagn09]. Selbst wenn vermutlich nicht all diese Konten aktiv genutzt werden, so verdeutlicht diese Zahl das große Potenzial von OpenID. Frühere Versuche Single Sign-On-Lösungen im Internet zu etablieren scheiterten aus verschiedenen Gründen oder konnten sich nur in Teilgebieten durchsetzen. So fand das im Jahr 1999 von Microsoft eingeführte .NET Passport aufgrund der zentralen Speicherung der Profildaten in Zusammenhang mit der Monopolstellung Microsofts und dessen Lizenzpolitik nur geringe Akzeptanz unter Anbietern und Anwendern und wurde schließlich zur „Windows Live ID“ weiter entwickelt, die inzwischen auch OpenID[6] unterstützt.

Die von der Liberty Alliance[7] erarbeiteten Konzepte für das Single Sign-On erreichten in der Praxis nur eine vergleichsweise geringe Verbreitung, sie bildeten aber den Grundstein für die Entwicklung und Standardisierung von SAML. Während sich SAML im Bereich der öffentlichen Verwaltung (E-Government) und im geschäftlichen Umfeld (E-Business) aufgrund der Möglichkeit, sehr spezifische Policies zu unterstützen, langsam verbreitet [HRZ10], steht einer schnellen Akzeptanz im Consumer-Bereich insbesondere die vergleichsweise große Komplexität von SAML entgegen. Während SAML erst noch entsprechende Profile für den typischen Anwendungsfall, bei dem man eine Authentisierung mit

[4] Siehe `http://www.livejournal.com/`.
[5] Siehe `http://kantarainitiative.org`.
[6] Siehe `http://winliveid.spaces.live.com/Blog/cns!AEE1BB0D86E23AAC!1745.entry`.
[7] Siehe `http://www.projectliberty.org/`.

einer bestimmten Qualitätsstufe und einige Attribute des Benutzers anfordert, erfordert [EHS10], unterstützen die einfach gestalteten OpenID-Spezifikationen und zahlreichen frei verfügbaren Implementierungen den üblichen Consumer-Anwendungsfall bereits seit geraumer Zeit und es ist deshalb zu erwarten, dass sich OpenID für die Authentisierung im Internet weiter verbreiten und möglicher Weise auch durchsetzen wird [HRZ10]. SAML hingegen wird bei B2B- und B2A-Prozessen, bei denen komplexer Policies und die Weitergabe von Berechtigungsinformationen erforderlich sind, vermutlich erste Wahl bleiben.

Wie in Abbildung 1 ersichtlich, besteht das OpenID-System aus einem User (U), der mit seinem User Agent (UA) (z.B. seinem Web-Browser) und mit Unterstützung des OpenID-Providers (OP) auf einen von der Relying Party (RP) angebotenen Dienst zugreifen möchte.

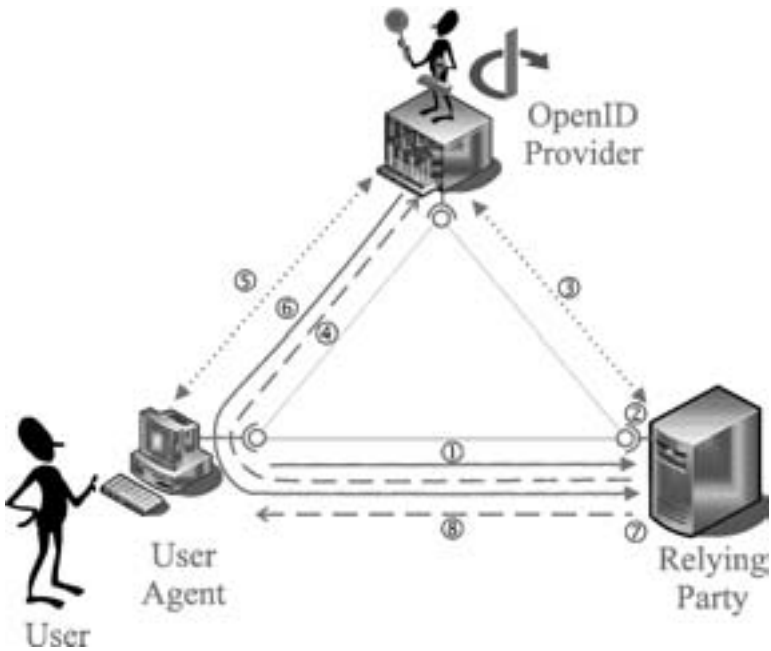

Abbildung 1: Authentisierung und Registrierung mit OpenID

Bevor der User den von der Relying Party angebotenen Dienst nutzen kann sind folgende Schritte nötig (vgl. [OpenID-Auth(v2.0), Section 3]):

1. $UA \rightarrow RP$: Der UA möchte auf eine Ressource zugreifen und kontaktiert daher die RP, die diese Ressource anbietet.
2. RP: Die RP löst die User ID mittels Yadis [Yadis(v1.0)], XRI und XRDS oder einer anderen geeigneten Discovery-Methode auf [RCT08]. Im einfachsten Fall ist hier aber nichts zu tun, da der Benutzer seinen OpenID-Identifier in Form einer URL bereitstellen kann.

3. $RP \rightarrow OP$ (optional): In diesem optionalen Schritt kann die RP (z.B. durch eine Diffie-Hellman-Schlüsselvereinbarung) mit dem OP einen geheimen Schlüssel $K_{RP,OP}$ vereinbaren, der später zur Prüfung der Authentizität des vom OP ausgestellten Authentisierungstoken $T = MAC(m, K_{RP,OP})$ genutzt wird. Sofern bereits eine entsprechende Sicherheitsbeziehung vorhanden ist oder – was aus Sicherheitsgründen *nicht* empfehlenswert ist [Lind09] – die Prüfung des Authentisierungstoken in Schritt 7 an den OP delegiert werden soll, kann dieser Schritt entfallen.

4. $RP \rightarrow OP$: Die RP leitet den UA zum OP um.

5. $OP \leftrightarrow U(A)$: Der UA muss sich am OP in geeigneter Weise authentisieren.

6. $OP \rightarrow RP$: Nach erfolgreicher Authentisierung erzeugt der OP ein entsprechendes Authentisierungstoken T, das durch eine Umleitung des UA zum RP gelangt.

7. RP: Die RP prüft das in der Nachricht enthaltene Authentisierungstoken $T = MAC(m, K_{RP,OP})$. Sofern noch keine Sicherheitsbeziehung zwischen RP und OP (vgl. Schritt 3) existiert und somit der Schlüssel $K_{RP,OP}$ der RP gar nicht bekannt ist, kann die RP die Prüfung von T an den OP delegieren. Wie in [Lind09] erläutert, sollte diese Variante aber aus offensichtlichen Sicherheitsgründen nicht genutzt werden.

8. $RP \rightarrow UA$: Der UA erhält schließlich Zugriff auf die gewünschte Ressource.

Erweiterungen. Für das grundlegende OpenID-Protokoll existieren derzeit die folgenden Erweiterungen:

- *Simple Registration Extension (SRE)*
  Die Simple Registration Extension [OpenID-SRE(v1.0)] bietet einen sehr „leichtgewichtigen“Austausch von Profildaten eines OP-Users zu einem RP. Die Intention der Erweiterung ist ein einfacher und schneller Datenaustausch von insgesamt neun gebräuchlichen Profilattributen (`nickname`, `fullname`, `email` etc.) für die Erstellung eines Benutzerkontos bei einer RP.

- *Provider Authentication Policy Extension (PAPE)*
  Mit der Provider Authentication Policy Extension [OpenID-PAPE(v1.0)] kann eine RP festlegen, wie sich ein Nutzer bei seinem OpenID-Provider zu authentisieren hat. Hierbei kann festgelegt werden, wie lange die Authentisierung maximal zurückliegen darf (`max_auth_age`), welche grundsätzliche Policy[8] vom OpenID-Provider zur Authentisierung genutzt werden soll (`preferred_auth_policies`) und ggf. welcher „Assurance Level“ (siehe z.B. [NIST-800-63]) mit der Authentisierung erreicht werden soll.

[8] Neben den drei in [OpenID-PAPE(v1.0)] definierten Authentication Policies (d.h. `phishing-resistant`, `multi-factor`, `multi-factor-physical`) können eigene Policies definiert werden (vgl. Abschnitt 3.4).

- *Attribute Exchange Erweiterung (AX)*
  Die Attribute Exchange Erweiterung [OpenID-AX(v1.0)] dient zum Austausch von Attributen zwischen RP und OP. Anders als bei der oben erläuterten Simple Registration Erweiterung können hier beliebige Attribute mit der `Fetch`-Operation beim OP gelesen und mit der `Store`-Operation gespeichert werden. Auch wenn die Spezifikation [OpenID-AX(v1.0)] hierzu keine Vorgaben macht, müssen bei der Unterstützung dieser Erweiterung zwingend Aspekte des Datenschutzes berücksichtigt und entsprechende Berechtigungskonzepte umgesetzt werden.

Außerdem befinden sich derzeit verschiedene Erweiterungen für OpenID in der Entwicklung. Zu ihnen gehören das OpenID Data Transport Protocol (DTP), das die Spezifikationen für OpenID Service Key Discovery [OpenID-SKD(D01)] und für OpenID DTP Messages [OpenID-DTPM(D03)] enthält, sowie Version 1.1 der OpenID Simple Registration Extension [OpenID-SRE(v1.1)] und ein OpenID Artifact Binding [OpenID-AB], das ein höheres Maß an Sicherheit verspricht. Seit Januar 2009 liegt auch ein Entwurf für eine Erweiterung namens OpenID OAuth Extension [OpenID-OAE] vor, die beschreibt wie man OpenID und OAuth [RFC5849] in Einklang bringt. In [Adid08] wurde vorgeschlagen, statt einer URL eine E-Mail-Adresse als Identifier zu verwenden, [CDS+08] enthält eine OpenID-Erweiterung für die Rechteverwaltung und Delegation und in [TaWa09] wurde schließlich gezeigt wie OpenID mit mobilen Endgeräten genutzt werden kann.

Sicherheit. Die Entwicklung von sicheren Browser-basierten Web-Applikationen und entsprechenden Single Sign-On Protokollen ist eine herausfordernde Aufgabe (vgl. [Slem01, Gros03, SAML-SecP(v2.0), GrPf06, GLS08, EHS09]). Deshalb ist es auch nicht verwunderlich, dass eine naive Realisierung des OpenID-Protokolls anfällig gegen Phishing, Man-in-the-Middle-Angriffe, Replay-Attacken, Cross-Site Request Forgery (CSRF) und Cross-Site Scripting (XSS) ist (vgl. [HySe08, Lind09, TsTs07]). Darüber hinaus wurde in [SKS10] ein Angriff gegen die Attribute Exchange Erweiterung [OpenID-AX(v1.0)] vorgestellt, die ausnutzt, dass die Anfragen beim OpenID-Protokoll nicht signiert werden.

Sofern der Benutzer jedoch ein starkes Authentisierungsverfahren verwendet, das TLS-Protokoll genutzt wird, die hierfür und für die Vereinbarung von $K_{RP,OP}$ und die Erstellung und Prüfung von $T = MAC(m, K_{RP,OP})$ genutzten Schlüssel authentisch sind und schließlich die Nachricht m, deren Integrität und Authentizität durch T geschützt wird, die sicherheitsrelevanten Daten[9] und alle Attribute umfasst, ist kein spezifischer Angriff gegen OpenID bekannt; ein formaler Sicherheitsbeweis im Stile von [GPS05, Gaje08, ACC+08] steht hierfür aber noch aus.

2.2 Elektronische Gesundheitskarte

Die elektronische Gesundheitskarte (eGK) ist eine in [eGK-1(v2.2.2), eGK-2(v2.2.1)] und [eGK-3(v2.2.0)] spezifizierte Chipkarte, die ein wesentliches Element der Sicherheitsar-

[9] Gemäß [OpenID-Auth(v2.0), Section 10.1] muss sich die „Signatur“ mindestens auf `op_endpoint`, `return_to`, `response_nonce`, `assoc_handle` sowie ggf. `claimed_id` und `identity` beziehen.

chitektur [FGHL07] der geplanten Telematikinfrastruktur für das deutsche Gesundheitswesen bildet. Technisch gesehen ist diese Chipkarte, deren Sicherheit durch eine Common Criteria Zertifizierung gemäß [BSI-PP-0020(v2.6)] nachgewiesen werden muss, insbesondere in der Lage die ausgefeilten in § 291a Abs. 4-5 [SGBV] definierten Zugriffsregeln für die darauf gespeicherten Patientendaten durchzusetzen und verschiedene kryptographische Operationen auszuführen.

Insbesondere enthält die eGK die folgenden Schlüssel, die grundsätzlich zur Authentisierung genutzt werden könnten:

- *PrK.eGK.AUT_CVC*
  Mit diesem privaten RSA-Schlüssel wird im Rahmen des in [eGK-1(v2.2.2), Abschnitt 16.2] spezifizierten, gegenseitigen Authentisierungsprotokolls, das typischer Weise mit einem Heilberufsausweis (HBA) [HBA-1(v2.3.2), HBA-2(v2.3.2)] oder einer Secure Module Card (SMC) [HBA-3(v2.3.2)] als Gegenstelle durchgeführt wird, die Echtheit der eGK nachgewiesen.

  Dieses zur gegenseitigen Authentisierung vorgesehene Protokoll besteht aus zwei weitgehend unabhängigen Teilprotokollen zur einseitigen Authentisierung im Stile des „Two-pass unilateral authentication protocol“ gemäß [ISO9798-3]. Da immer – also insbesondere auch ohne eine PIN-Eingabe und ohne die vorherige Authentisierung der Gegenseite – auf den privaten Schlüssel PrK.eGK.AUT_CVC zugegriffen werden kann (vgl. [eGK-2(v2.2.1), Abschnitt 6.2.9]), kann mit diesem Schlüssel ein besonders benutzerfreundliches, einseitiges Authentisierungsprotokoll realisiert werden, bei dem der Benutzer nur die eGK einstecken aber keine PIN eingeben muss.

  Dieses Authentisierungsprotokoll besteht aus folgenden Schritten:

  1. Erzeugen einer Zufallszahl r.
  2. Bilden eines APDU-Stapels, durch den
     - die CV-Zertifikate C.CA_eGK.CS und C.eGK.AUT_CVC von der eGK gelesen werden und
     - die Zufallszahl r mit dem privaten Schlüssel PrK.eGK.AUT_CVC signiert wird.
  3. Übermitteln des APDU-Stapels zur eGK mit der `Transmit`-Funktion aus [BSI-TR-03112(v1.1)], wodurch man in `TransmitResponse` die CV-Zertifikate und eine gemäß [ISO9796-2] DS1 gebildete Signatur s erhält.
  4. Prüfen der CV-Zertifikate gegen den Wurzelschlüssel der gematik.
  5. Prüfen der Signatur s über die Zufallszahl r.

- *PrK.CH.AUT und PrK.CH.AUTN*
  Die beiden privaten RSA-Schlüssel PrK.CH.AUT (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.6]) und PrK.CH.AUTN (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.7]) können beispielsweise nach Eingabe der PIN.home für die Authentisierung gemäß [ISO9798-3] genutzt werden. Da für diese beiden Schlüssel auch entsprechende X.509-Zertifikate

(vgl. [eGK-2(v2.2.1), Abschnitte 6.4.1-6.4.2] und [gemX.509-eGK(v1.5.9)]) zur Verfügung stehen, könnte damit auch eine TLS-Client-Authentisierung gemäß [RFC5246] durchgeführt werden. Außerdem kann mit dem Online Certificate Status Protocol (OCSP) gemäß [RFC2560] der Sperrstatus dieser Zertifikate bzw. der eGK ermittelt werden.

- *PrK.CH.ENC und PrK.CH.ENCV*
  Die beiden privaten RSA-Schlüssel PrK.CH.ENC (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.8]) und PrK.CH.ENCV (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.9]) können beispielsweise nach Eingabe der PIN.home zur Entschlüsselung von Daten genutzt werden, wodurch ein Authentisierungsprotokoll gemäß [NeSc78, Lowe96] realisiert werden könnte.

Außerdem sind auf der eGK weitere geheime Schlüssel (SK.CMS,SK.VSD und SK.VSDCMS, siehe [eGK-2(v2.2.1), Abschnitt 6.2.11-6.2.13]) vorhanden, die jedoch nur zur Authentisierung gegenüber dem Kartenmanagementsystem bzw. dem Versichertenstammdatendienst der Krankenkasse genutzt werden können.

Schließlich kann auf der eGK ein privater Signaturschlüssel Prk.CH.QES (vgl. [eGK-2(v2.2.1), Abschnitt 7.1.3 und 7.7.7]) vorhanden sein, der aber aus nahe liegenden Gründen nicht zur Authentisierung genutzt werden sollte.

3 OpeneGK

3.1 Überblick

Bei der in diesem Papier vorgeschlagenen Kombination der elektronischen Gesundheitskarte mit dem OpenID-Protokoll ist der Bürger, wie in Abbildung 2 dargestellt, mit einer elektronischen Gesundheitskarte (siehe Abschnitt 2.2), einem entsprechenden Kartenterminal und einem Bürgerclient (siehe Abschnitt 3.2) ausgestattet. Um sich bei einem Mehrwertdienst (siehe Abschnitt 3.3) zu registrieren und authentisieren läuft das in Abschnitt 2.1 beschriebene Protokoll ab, bei dem der OpeneGK-Dienst (siehe Abschnitt 3.4) in Schritt (5) auf die elektronische Gesundheitskarte zugreift.

3.2 Bürgerclient

Der Bürgerclient ist eine Chipkarten-Middleware, die derzeit im Auftrag des Bundesinnenministeriums entwickelt wird [Heise091116] und alle Chipkarten der eCard-Strategie [eCard-PM, Kowa07] – also insbesondere auch die elektronische Gesundheitskarte – unterstützt. Der Bürgerclient setzt das in [BSI-TR-03112(v1.1)] spezifizierte eCard-API-Framework um, das wiederum auf einer Vielzahl von internationalen Standards basiert [HuBa08].

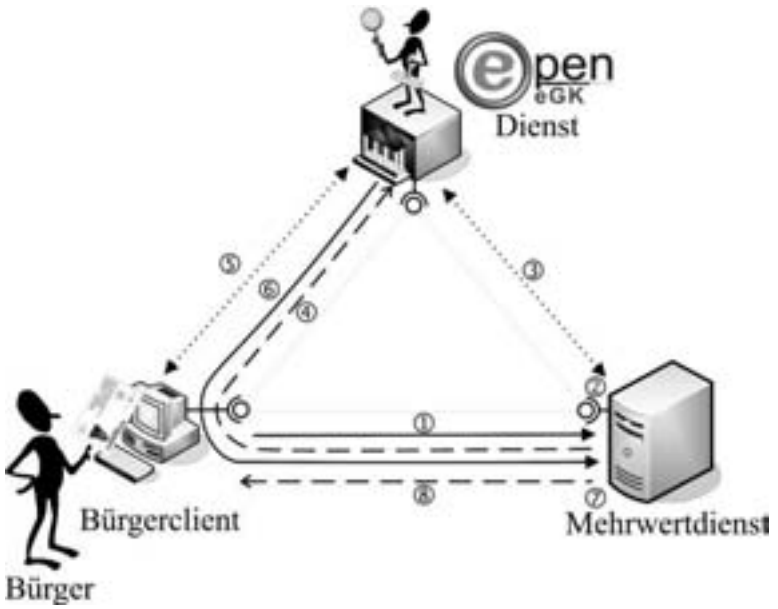

Abbildung 2: Authentisierung und Registrierung mit OpeneGK

3.3 Mehrwertdienst

Der kritische Faktor für die schnelle Verbreitung (vgl. [HRZ10]) eines Protokolls wie OpenID [OpenID-Auth(v2.0)] ist vor allem dessen Einfachheit verglichen mit den konkurrierenden Protokollen wie beispielsweise SAML [SAML(v2.0)]. Dabei geht es nicht nur um den Protokollablauf selbst, sondern vor allem auch um die breite Verfügbarkeit und leichte Integrierbarkeit der erforderlichen Programmbibliotheken. Da für alle relevanten Programmiersprachen bereits entsprechende OpenID-Bibliotheken existieren[10], könnte der OpeneGK-Dienst beispielsweise unter Verwendung einer dieser Bibliotheken in einen Mehrwertdienst integriert werden. Allerdings operieren diese Bibliotheken zumeist auf einem vergleichsweise niedrigen Abstraktionsniveau und der Entwickler des Mehrwertdienstes bzw. ein Integrator müsste sowohl die genauen Protokollabläufe als auch die damit verbundenen Sicherheitsaspekte (vgl. [HySe08, Lind09, TsTs07]) kennen, um eine zuverlässige Benutzerauthentifizierung sicher zu stellen.

Um die sichere Anbindung des OpeneGK-Dienstes zu erleichtern soll im Folgenden ein Schnittstellenentwurf für eine erweiterte Bibliothek vorgestellt werden, die mit Version 1.1 und 2.0 der OpenID Spezifikation [OpenID-Auth(v1.1), OpenID-Auth(v2.0)] und deren Erweiterungen [OpenID-SRE(v1.0), OpenID-PAPE(v1.0), OpenID-AX(v1.0)] kom-

[10] Siehe `http://openid.net/developers/libraries`.

patibel ist, eine besonders einfache Integration ermöglicht, sinnvolle Voreinstellungen für die sichere Authentisierung mit der eGK mitbringt und schließlich die verschiedenen Policies und speziellen Features des OpeneGK-Dienstes, wie z.B. die Erzeugung von Mehrwertdienst-spezifischen Pseudonymen (vgl. Abschnitt 3.4), unterstützt.

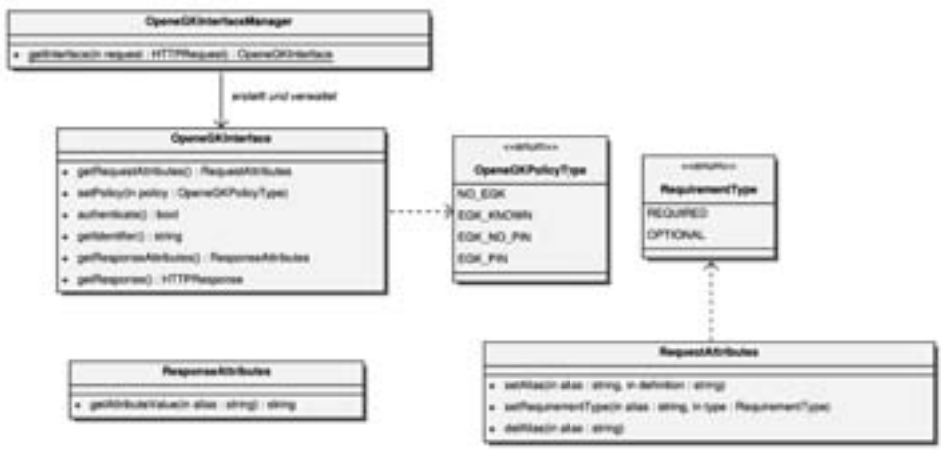

Abbildung 3: UML-Design der OpeneGK-Schnittstelle

Abbildung 3 zeigt ein UML Klassendiagramm mit allen Klassen der OpeneGK-Schnittstelle. Die gesamte Protokolllogik wird von `OpeneGKInterface` zur Verfügung gestellt. Wie in Abschnitt 2.1 erläutert, sind für eine Registrierung oder Authentisierung beim OpenID-Protokoll zwei HTTP Nachrichten zwischen dem Mehrwertdienst (Relying Party) und dem Bürgerclient (User Agent) nötig. Um die HTTP-Sitzungsverwaltung zu erleichtern kümmert sich die Klasse `OpeneGKInterfaceManager` um die Erstellung und Zuordnung der `OpeneGKInterface`-Instanzen zu den HTTP-Sitzungen.

Während des Authentisierungsprozesses stellt die OpeneGK-Schnittstelle Möglichkeiten bereit, um spezifische Einstellungen vorzunehmen und Ergebnisdaten abzufragen. Die Klasse `RequestAttributes` bietet beispielsweise die Möglichkeit die in einer Konfigurationsdatei definierten Standardwerte für die gewünschten Attribute [11] zu überschreiben. Mit der Funktion `setAlias` werden den Attributdefinitionen kurze Alias-Namen und URLs zugewiesen, um nach der Authentisierung bequem auf diese zugreifen zu können. Das OpenID-Protokoll sieht für angefragte Attribute vor, dass diese nicht zwingend in der Antwort des OpenID-Providers enthalten sein müssen. Mit den Werten aus `RequirementType` können bestimmte Attribute als `OPTIONAL` markiert werden, wodurch der Standardwert (`REQUIRED`) überschrieben wird.

Die gewünschte Authentisierungsmethode wird mit den vordefinierten Werten aus `OpeneGKPolicyType` gesteuert, die eine eGK-spezifische Authentication Policy im Sinne von [OpenID-PAPE(v1.0)] umsetzen. Ebenso wie bei den Attributen wird die benötigte Policy im Regelfall nur durch die Konfigurationsdatei bestimmt.

[11] Siehe [OpenID-SRE(v1.0)] und [OpenID-AX(v1.0)]

Nach einer erfolgreichen Authentisierung wird durch die Klasse `ResponseAttributes` eine komfortable Schnittstelle zum Ausgeben der Attribute bereitgestellt. Zu einem Alias wird entweder der empfangene Wert, oder ein sprachspezifisches leeres Symbol zurückgeliefert.

Nachdem die Möglichkeiten der Einflussnahme der Applikation auf den Authentisierungsprozess beschrieben wurde, bleibt es noch den Ablauf unter Verwendung der Schnittstelle zu beschreiben. In Anlehnung an die Schritte aus Abbildung 1 in Abschnitt 2.1 ergibt sich folgender Ablauf:

- Schritt (1)-(4):

  Der Prozess wird dadurch gestartet, dass der Mehrwertdienst eine Authentisierungsanfrage in Form einer HTTP GET Nachricht empfängt. An die HTTP Nachricht werden zwei Anforderungen gestellt. Zum Einen muss der Parameter `return_to`, der einen Verweis zu der ursprünglich angefragten Ressource darstellt, gesetzt sein. Zum Anderen muss in der Nachricht eine HTTP Sitzung zu finden sein anhand derer die richtige `OpeneGKInterface`-Instanz ausgewählt werden kann. Anders als bei einer gewöhnlichen OpenID-Authentisierung ist der Wert `openid.identity` optional. Fehlt er, so wird automatisch eine eGK-basierte Authentisierung angenommen und die Identität des Nutzers unter Verwendung der eGK ermittelt.

  In diesem Schritt sind folgende Funktionsaufrufe zu tätigen.

  a) `OpeneGKInterfaceManager.getInterface`
  Durch diesen Aufruf wird mit Informationen aus der HTTP Nachricht eine Instanz der Schnittstelle erstellt. Im Nachfolgenden werden Funktionen aus der Klasse `OpeneGKInterface` immer mit dieser Instanz ausgeführt.

  b) `getRequestAttributes` und `setPolicy` (optional)
  Nun können optional die konfigurierten Standardwerte für die anzufordernden Attribute (mit `getRequestAttributes`) oder die gewünschte Policy (mit `setPolicy`) (vgl. Abschnitt 3.4) überschrieben werden.

  c) `getHTTPResponse`
  Sofern noch keine Sicherheitsbeziehung zwischen dem Mehrwertdienst und dem OpeneGK-Dienst vorhanden ist, wird diese etabliert und danach die HTTP Antwort mit der OpenID-Anfrage erzeugt und zurückgegeben. Die Funktion stellt auch das vorläufige Ende des Prozesses dar, da nun der OpeneGK-Dienst für die Authentisierung in Schritt (5) zuständig ist und der Mehrwertdienst auf eine neue HTTP Anfrage wartet.

- Schritt (6)-(8):

  Die folgenden Schritte laufen ab, sobald der Mehrwertdienst eine HTTP GET Anfrage mit einer OpenID-Authentisierungsantwort empfängt:

  e) `OpeneGKInterfaceManager.getInterface`
  Statt wie im vorherigen Schritt eine neue Instanz zu erzeugen, wird hier die der HTTP-Sitzung zugehörige Instanz zurückgeliefert.

f) `authenticate`
In diesem Schritt wird die OpenID-Antwort ausgewertet, das Authentisierungstoken T geprüft und das Ergebnis der Authentifizierung zurückgeliefert.

g) `getIdentifier` (optional)
Im Fall einer erfolgreichen Authentifizierung kann, sofern nicht eine völlig anonyme Nutzung des Mehrwertdienstes vorgesehen ist, mit dieser Funktion das Dienst-spezifische Pseudonym des Benutzers zurückgeliefert werden.

h) `getResponseAttributes` (optional)
Aus der bereits oben beschriebenen Struktur können ggf. die zur Registrierung in der Applikation notwendigen Attribute extrahiert werden.

i) `getHTTPResponse`
Den Abschluss der OpenID-Kommunikation markiert die an den Bürgerclient geschickte HTTP-Antwort, wodurch eine Umleitung des Bürgerclients auf die in der ersten Anfrage gesendete `return_to`-URL erfolgt.

3.4 OpeneGK-Dienst

Der OpeneGK-Dienst ist einerseits ein OpenID-Provider, der über das in [OpenID-Auth(v2.0)] definierte Protokoll angesprochen werden kann. Andererseits umfasst der OpeneGK-Dienst ein „serverseitiges eCard-API-Framework“ [BSI-TR-03112(v1.1)] über das mit dem Bürgerclient kommuniziert und letztlich auf die elektronische Gesundheitskarte zugegriffen werden kann.

Der OpeneGK-Dienst unterstützt die folgenden Authentication Policies (vgl. Abbildung 3):

- `NO_EGK`
  In diesem Fall, der in Verbindung mit einem beliebigen OpenID-Provider genutzt werden kann, wurde die elektronische Gesundheitskarte weder zur erstmaligen Registrierung noch zur aktuellen Authentisierung genutzt. Durch diese Policy können Mehrwertdienste die OpeneGK-Schnittstelle und ggf. den OpeneGK-Dienst bereits nutzen obwohl die elektronische Gesundheitskarte noch gar nicht flächendeckend ausgerollt ist.

- `EGK_KNOWN`
  In diesem Fall wurde zwar die erstmalige Registrierung mit der elektronischen Gesundheitskarte durchgeführt aber die aktuelle Authentisierung erfolgte am OpeneGK-Dienst unter Verwendung eines alternativen Authentisierungsverfahrens. Dadurch kann der OpeneGK-Dienst zur Authentisierung auch dann genutzt werden, wenn gerade kein kontaktbehaftetes Chipkartenterminal für die direkte Nutzung der eGK zur Hand ist.

- `EGK_NO_PIN`
  Bei dieser Policy erfolgt die Authentisierung mit dem für die Card-2-Card-Authenti-

sierung vorgesehenen Schlüssel PrK.eGK.AUT_CVC. Wie in [eGK-2(v2.2.1), Abschnitt 6.2.9] spezifiziert, ist für die Nutzung dieses Schlüssels keine PIN-Eingabe erforderlich, so dass die eGK zur Authentisierung einfach gesteckt sein muss und eine besonders komfortable Authentisierung möglich wird.

- `EGK_PIN`
  Bei dieser Authentisierungsvariante wird nach Eingabe der PIN der private Schlüssel PrK.CH.AUTN (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.7]) zur Authentisierung genutzt. Da bei dieser Variante auch der Sperrstatus der eGK geprüft werden kann, bietet diese Authentication Policy das größte Maß an Sicherheit.

Ein weiterer Vorteil des OpeneGK-Dienstes im Vergleich zur naiven Nutzung von OpenID oder der elektronischen Gesundheitskarte liegt in der Möglichkeit Mehrwertdienstspezifische Pseudonyme zu konstruieren, die einen Benutzer am Mehrwertdienst selbst nach einem Krankenkassen- und Kartenwechsel hinweg eindeutig wiedererkennen lassen aber eine Dienst-übergreifende Verkettung der Pseudonyme und dadurch die Profilbildung unmöglich machen. Dies kann beim OpeneGK-Dienst beispielsweise dadurch erreicht werden, dass der im C.CH.AUT-Zertifikat (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.1] und [gemX.509-eGK(v1.5.0), Abschnitt 6]) enthaltene unveränderbare Teil der Krankenversicherungsnummer des Versicherten (vgl. [gemX.509-eGK(v1.5.0), Abschnitt 5.6]), die Domain des Mehrwertdienstes und ein OpeneGK-spezifisches Geheimnis konkateniert und daraus ein kryptographischer Hashwert gebildet wird. Eine standardmäßige Verwendung von solchen Pseudonymen wäre aufgrund der Einfachheit für alle Mehrwertdienste zu empfehlen.

4 Diskussion

Der OpenID-Ansatz ist zunächst primär für den Benutzer vorteilhaft: er muss sich nicht wieder neue Zugangsdaten merken und kann mit einem Konto sich bei verschiedenen Websites anmelden. Diese Idee ist nicht besonders neu (vgl. [Berr98, HiWi00]), wurde aber bei OpenID besonders neutral, flexibel und dabei für alle Seiten besonders einfach umgesetzt. Für den Service-Provider ergibt sich damit indessen lediglich ein Marketing-Effekt. Er wird für die Nutzer etwas attraktiver und kann so eventuell seine Kundenzahl erhöhen. Technisch bedeutet es für ihn die Auslagerung des Benutzermanagements, was eine gewisse Vereinfachung der Verwaltung darstellt.

Unser Vorschlag einer OpeneGK-Anmeldung bringt dem Benutzer bereits ähnliche Vorteile wie bei OpenID allein, beinhaltet jedoch für den Dienstanbieter einen echten Mehrwert. Er weiß dadurch, dass es sich bei den sich damit anmeldenden Bürgern um Personen handelt, deren Identität und persönliche Daten zuverlässig verifiziert wurden. Verschiedene Missbrauchsszenarien sind damit von vornherein verhindert; der Service-Provider kann solchen Nutzern sogar durch das Anbieten weitergehender Dienste mehr Vertrauen entgegen bringen.

Der Bürger hat aber gleichfalls zusätzliche Vorteile. Er ist beispielsweise nicht gezwungen

sich mit ein und demselben Benutzernamen bei allen angeschlossenen Sites anzumelden. Der OpeneGK-Ansatz erlaubt es, für jeden Service-Provider individuelle Pseudonyme zu verwenden, die auf Wunsch sogar automatisch generiert werden können. So bleibt die Privatsspähre des Benutzers auch dann gewahrt, wenn er den gleichen Authentisierungsmechanismus bei mehreren untereinander verbundenen Anbietern nutzt.

Angesichts der aktuellen politischen Diskussionen um die elektronische Gesundheitskarte steckt im OpeneGK-Vorschlag aber noch ein ganz pragmatischer Vorteil: Diese Technik setzt nur auf die Karte selbst auf und kann daher bereits starten und genutzt werden, bevor die geplante Infrastruktur der gematik verfügbar ist. Denn wann Letzteres soweit ist, ist Stand heute nicht abzuschätzen.

Bei der Konzeption von OpeneGK wurde besonderer Wert auf sichere Kommunikation und Datenschutz gelegt. Um dies an allen Stellen zu gewährleisten, ist natürlich eine korrekte Anbindung dieses Dienstes bei einem Service-Provider zu realisieren. Insbesondere ist ein Schlüsselaustausch zwischen OpeneGK-Provider und Service-Provider zwingend erforderlich.

Es sind nur wenige sensible medizinische Daten auf der eGK vorgesehen, überdies nur freiwillig. Für den OpeneGK-Dienst werden diese aber nicht einmal benötigt und daher auch nicht verwendet. Er sorgt nur für die Registrierung, Authentisierung und Identifikation; medizinische Angaben muss der Benutzer sofern gewünscht direkt mit dem Service-Provider austauschen. Daher ist die Hemmschwelle für die Anwender sicher als gering einzustufen, da sie kein Datenschutzrisiko eingehen.

5 Fazit

Gerade im medizinischen Bereich geht es oft um individuelle Daten, bei denen jeder von uns selbst darüber bestimmen will, welche er an wen weiter gibt. Durch die starke Authentisierung bei OpeneGK wird sicher gestellt, dass Unbefugte nicht einfach durch Ausspähen eines Passworts an diese Daten gelangen können. Für die Dienstanbieter heißt das, dass ihnen schrittweise mehr Vertrauen geschenkt werden wird, so dass die Kunden eher bereit sind, individuelle medizinische Daten bei ihnen zu hinterlegen. Der Datenschutz kann so signifikant erhöht werden. Und das alles kann mit einer Karte, die in absehbarer Zeit ein Großteil der bundesdeutschen Bevölkerung ohnehin bei sich tragen wird, umgesetzt werden.

Das Konzept, das OpenID-Protokoll um die Verwendung einer Chipkarte zur starken Authentisierung zu erweitern, ist im medizinischen Umfeld besonders attraktiv, aber keineswegs darauf beschränkt. Neben der eGK könnte auch eine andere staatliche Identitätskarte verwendet werden, wie sie in vielen europäischen Ländern geplant ist und bereits verwendet wird. In Deutschland ist dies der neue Personalausweis, der ab Ende 2010 ausgegeben wird. Diese Option macht den oben beschriebenen Ansatz noch flexibler und ermöglicht eine Vielzahl weiterer Anwendungsfälle.

Auf der anderen Seite hat OpeneGK den Vorteil, dass die Schnittstelle bewusst einfach und generisch gehalten ist. Dies erlaubt es prinzipiell, das Konzept auch für andere Fra-

meworks für Single-Sign-On wie SAML oder CardSpace [BSB08] nutzbar zu machen. Dies würde technisch zwar eine deutlich komplexere Realisierung bedeuten, kann für die Service-Provider aber weitgehend transparent bleiben.

Somit stellt OpeneGK einen sehr flexiblen Ansatz dar, der großes Potenzial zur Erweiterung in vielerlei Hinsicht in sich trägt.

Literatur

[ACC+08] A. ARMANDO, R. CARBONE, L. COMPAGNA, J. CUELLAR, und L. TOBARRA. *Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps*. ACM Workshop on Formal Methods in Security Engineering. http://www.ai-lab.it/armando/pub/fmse9-armando.pdf, 2008.

[Adid08] BEN ADIDAD. *EmID: Web Authentication by Email Address*. http://ben.adida.net/research/w2sp2008-emid.pdf, 2008.

[Berr98] PHILIPPE LE BERRE. *Authentication between servers*. European Patent Application EP 0 940 960 A1, March 1998.

[BSB08] V. BERTOCCI, G. SERACK, und C. BAKER. *Understanding Windows CardSpace - An Introduction to the Concepts and Challenges of Digital Identities* (Addison Wesley, 2008).

[BSI-PP-0020(v2.6)] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK. *Common Criteria Protection Profile–electronic Health Card (eHC)–elektronische Gesundheitskarte (eGK)*. BSI-PP-0020-V2-2007-MA02, Version 2.6, 29.07.2008. http://www.gematik.de/upload/gematik_eGK_Specification_Part2_e_V1%_1_1_516.pdf, 2008.

[BSI-TR-03112(v1.1)] FEDERAL OFFICE FOR INFORMATION SECURITY (BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK). *eCard-API-Framework*. Technical Directive (BSI-TR-03112), Version 1.1, Part 1-7. https://www.bsi.bund.de/ContentBSI/Publikationen/TechnischeRichtlinien/tr03112/index_htm.html, 2009.

[CDS+08] BRYANT CUTLER, DEVLIN DALEY, KENT SEAMONS, und PHIL WINDLEY. *SimplePermissions: an OpenID Extension for Delegation and Permissions Model Discovery*. http://www.eclab.byu.edu/simplepermissions_techreport.pdf, 2008.

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

[eGK-1(v2.2.2)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Die Spezifikation der elektronischen Gesundheitskarte - Teil 1: Spezifikation der elektrischen Schnittstelle*. Version 2.2.2 vom 16.09.2008. http://www.gematik.de/upload/gematik_eGK_Spezifikation_Teil1_V2_2_2_4411.pdf, 2008.

[eGK-2(v2.2.1)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Die Spezifikation der elektronischen Gesundheitskarte - Teil 2: Grundlegende Applikationen*. Version 2.2.1 vom 16.09.2008. `http://www.gematik.de/upload/gematik_eGK_Spezifikation_Teil2_V2_2_1_3805.pdf`, 2008.

[eGK-3(v2.2.0)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Die Spezifikation der elektronischen Gesundheitskarte - Teil 3: Äußere Gestaltung*. Version 2.2.0 vom 02.07.2008. `http://www.gematik.de/upload/gematik_eGK_Spezifikation_Teil3_V2_2_0_3806.pdf`, 2008.

[EHS09] JAN EICHHOLZ, DETLEF HÜHNLEIN, und JÖRG SCHWENK. *SAMLizing the European Citizen Card*. In *Proceedings of* BIOSIG 2009: Biometrics and Electronic Signatures, Band 155 von *Lecture Notes in Informatics (LNI)*, Seiten 105–117 (GI-Edition, 2009). `http://www.ecsec.de/pub/SAMLizing-ECC.pdf`.

[EHS10] JAN EICHHOLZ AND DETLEF HÜHNLEIN AND JOHANNES SCHMÖLZ. *A SAML-profile for electronic identity cards*. to appear, 2010.

[FGHL07] FLORIAN FANKHAUSER, THOMAS GRECHENIG, DETLEF HÜHNLEIN, und MANFRED LOHMAIER. *Die Basiskonzepte der Sicherheitsarchitektur bei der Einführung der eGK*. In PATRICK HORSTER (Herausgeber), *Tagungsband DACH Security 2007*, Seiten 326–337 (IT-Verlag, 2007). `http://www.ecsec.de/pub/2007_DACH_eGK-Sicherheitsarchitektur.pdf`.

[gemX.509-eGK(v1.5.0)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Festlegungen zu den X.509 Zertifikaten der Versicherten*. Version 1.5.0 vom 12.06.2008. `http://www.gematik.de/upload/gematik_PKI_X509_Zertifikate_des_Versicherten_eGK_V1.5.0_3854.pdf`, 2008.

[gemX.509-eGK(v1.5.9)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Festlegungen zu den X.509 Zertifikaten der Versicherten*. Version 1.5.9 vom 03.07.2009, 2009.

[Gaje08] SEBASTIAN GAJEK. *A Universally Composable Framework for the Analysis of Browser-Based Security Protocols*. In JOONSANG BAEK, FENG BAO, KEFEI CHEN, und XUEJIA LAI (Herausgeber), *Provable Security – Second International Conference, ProvSec 2008, Shanghai, China, October 30 - November 1*, Band 5324 von *Lecture Notes in Computer Science*, Seiten 283–297 (Springer, 2008).

[GLS08] JÖRG SCHWENK, LIJUN LIAO, und SEBASTAN GAJEK. *Stronger Bindings for SAML Assertions and SAML Artifacts*. In *Proceedings of the 5th ACM CCS Workshop on Secure Web Services (SWS'08)*, Seiten 11–20 (ACM Press, 2008).

[GPS05] THOMAS GROSS, BIRGIT PFITZMANN, und AHMAD-REZA SADEGHI. *Browser Model for Security Analysis of Browser-Based Protocols*. In *ESORICS: 10th European Symposium on Research in Computer Security*, Band 3679, Seiten 489–508 (Berlin, Germany, 2005). `http://eprint.iacr.org/2005/127.pdf`.

[Gros03] THOMAS GROSS. *Security Analysis of the SAML Single Sign-on Browser/Artifact Profile*. In *Annual Computer Security Applications Conference, December 8-12, 2003, Aladdin Resort & Casino Las Vegas, Nevada, USA* (2003). http://www.acsac.org/2003/papers/73.pdf.

[GrPf06] THOMAS GROSS und BIRGIT PFITZMANN. *SAML Artifact Information Flow Revisited*. In *In IEEE Workshop on Web Services Security (WSSS)*, Seiten 84–100 (IEEE, Berkeley, 2006). http://www.zurich.ibm.com/security/publications/2006/GrPf06.SAML-Artifacts.rz3643.pdf.

[HBA-1(v2.3.2)] BUNDESÄRZTEKAMMER ET. AL. *German Health Professional Card and Security Module Card – Part 1: Commands, Algorithms and Functions of the COS Platform*. Version 2.3.2, 05.08.2009. http://www.bundesaerztekammer.de/downloads/HPC-Spezifikation_2.3.2_-_COS_Teil_1_.pdf, 2009.

[HBA-2(v2.3.2)] BUNDESÄRZTEKAMMER ET. AL. *German Health Professional Card and Security Module Card – Part 2: HPC Applications and Functions*. Version 2.3.2, 05.08.2009. http://www.bundesaerztekammer.de/downloads/HPC-Spezifikation_2.3.2_-_HPC_Teil_2_.pdf, 2009.

[HBA-3(v2.3.2)] BUNDESÄRZTEKAMMER ET. AL. *German Health Professional Card and Security Module Card – Part 3: SMC Applications and Functions*. Version 2.3.2, 05.08.2009. http://www.bundesaerztekammer.de/downloads/HPC-Spezifikation_2.3.2_-_SMC_Teil_3_.pdf, 2009.

[Heise091116] HEISE. *Elektronischer Personalausweis: Bürger-Client auf dem Weg zum Nutzer*. Meldung vom 16.11.2009, 15:13 Uhr. http://tinyurl.com/yz4kzno, 2009.

[HiWi00] HEATHER MARIA HINTON und DAVID JOHN WINTERS. *Method and system for web-based cross-domain single-sign-on authentication*. World-wide patent, WO 02/39237 A2, November 2000.

[HRZ10] DETLEF HÜHNLEIN, HEIKO ROSSNAGEL, und JAN ZIBUSCHKA. *Diffusion of Federated Identity Management*. to appear, 2010.

[HuBa08] DETLEF HÜHNLEIN und MANUEL BACH. *Die Standards des eCard-API-Frameworks – Eine deutsche Richtlinie im Konzert internationaler Normen. Datenschutz und Datensicherheit (DuD)*, (6):379–384. http://www.ecsec.de/pub/2008_DuD_eCard.pdf, 2008.

[HySe08] HYUN-KYUNG-OH und SEUNG-HUN-JIN. *The security limitations of SSO in OpenID*. In *2008 10th International Conference on Advanced Communication Technology, Gangwon-Do, South Korea, 17-20 Feb. 2008*, Seiten 1608–1611 (IEEE, 2008). http://mnet.skku.ac.kr/data/2008data/ICACT2008/pdf/tech/08F-03.pdf.

[ID-MI(v1.0)] MICHAEL B. JONES und MICHAEL MCINTOSH. *Identity Metasystem Interoperability Version 1.0*. OASIS Standard. http://docs.oasis-open.org/imi/identity/v1.0/os/identity-1.0-spec-os.pdf, July 2009.

[ISO9796-2] *ISO-IEC 9796-2: Information Technology - Security Techniques - Digital Signature Schemes Giving Message Recovery - Part 2: Integer Factorization Based Mechanisms*. International Standard, Oktober 2002.

[ISO9798-3] *ISO-IEC 9798-3: Information Technology – Security Techniques – Entity Authentication – Part 3: Mechanisms using digital signature techniques*. International Standard, 1998.

[Kiss09] BRIAN KISSEL. *OpenID 2009 Year in Review*. December 16, 2009. `http://openid.net/2009/12/16/openid-2009-year-in-review/`.

[Kowa07] BERND KOWALSKI. *Die eCard-Strategie der Bundesregierung im Überblick*. In D. HÜHNLEIN A. BRÖMME, C. BUSCH (Herausgeber), *BIOSIG 2007: Biometrics and Electronic Signatures, Proceedings of the Special Interest Group on Biometrics and Electronic Signatures*, Band 108 von *LNI*, Seiten 87–96 (2007).

[Lind09] ALEXANDER LINDHOLM. *Security Evaluation of the OpenID Protocol*. Master-Thesis, Royal Institute of Technology, School of Computer Science and Communication, KTH CSC, Stockholm. `http://w3.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2009/rapporter09/lindholm_alexander_09076.pdf`, 2009.

[Lowe96] GAVIN LOWE. *Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR*. In TIZIANA MARGARIA und BERNHARD STEFFEN (Herausgeber), *Tools and Algorithms for Construction and Analysis of Systems, Second International Workshop, TACAS '96, Passau, Germany, March 27-29, 1996, Proceedings*, Band 1055 von *Lecture Notes in Computer Science*, Seiten 147–166 (Springer, 1996).

[NeSc78] ROGER NEEDHAM und MICHAEL D. SCHROEDER. *Using encryption for authentication in large networks of computers*. *Communications of the ACM*, Band 21(12), 1978.

[NIST-800-63] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Electronic Authentication Guideline*. NIST Special Publication 800-63 Version 1.0.2. `http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf`.

[OpenID-AB] N. SAKIMURA J. BRADLEY. *OpenID Artifact Binding 1.0*. Draft07, 14.05.2010. `http://www.sakimura.org/specs/ab/1.0/`.

[OpenID-AX(v1.0)] OPENID FOUNDATION. *OpenID Attribute Exchange 1.0*. Final, December 5, 2007. `http://openid.net/specs/openid-attribute-exchange-1_0.html`.

[OpenID-Auth(v1.1)] D. RECORDON und B. FITZPATRICK. *OpenID Authentication 1.1*. May 2006. `http://openid.net/specs/openid-authentication-1_1.html`.

[OpenID-Auth(v2.0)] OPENID FOUNDATION. *OpenID Authentication 2.0*. Final, December 5, 2007. `http://openid.net/specs/openid-authentication-2_0.html`.

[OpenID-DTPM(D03)] OPENID FOUNDATION. *OpenID DTP Messages 1.0 - Draft 03*. Draft, December 06, 2006. http://openid.net/specs/openid-dtp-messages-1_0-03.html.

[OpenID-OAE] D. BALFANZ, B. DE MEDEIROS, D. RECORDON, J. SMARR, und A. TOM. *OpenID OAuth Extension*. Draft, January 7, 2009. http://step2.googlecode.com/svn/spec/openid_oauth_extension/latest/openid_oauth_extension.html, 2009.

[OpenID-PAPE(v1.0)] OPENID FOUNDATION. *OpenID Provider Authentication Policy Extension 1.0*. December 30, 2008. http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html.

[OpenID-Pro] OPENID FOUNDATION. *Get an OpenID*. http://openid.net/get-an-openid, 2010.

[OpenID-SKD(D01)] OPENID FOUNDATION. *OpenID Service Key Discovery 1.0 - Draft 01*. Draft, December 06, 2006. http://openid.net/specs/openid-service-key-discovery-1_0-01.html.

[OpenID-SRE(v1.0)] OPENID FOUNDATION. *OpenID Simple Registration Extension 1.0*. June 30, 2006. http://openid.net/specs/openid-simple-registration-extension-1_0.html.

[OpenID-SRE(v1.1)] OPENID FOUNDATION. *OpenID Simple Registration Extension 1.1 - Draft 1*. Draft, December 06, 2006. http://openid.net/specs/openid-simple-registration-extension-1_1-01.html.

[Raep09] MARTIN RAEPPLE. *Netzweite Identitäten mit OpenID. Datenschutz und Datensicherheit (DuD)*, Band 33(3):174–177. http://www.springerlink.com/content/755180g7h7741187/, 2009.

[RCT08] DRUMMOND REED, LES CHASEN, und WILLIAM TAN. *OpenID identity discovery with XRI and XRDS*. In *IDtrust '08: Proceedings of the 7th symposium on Identity and trust on the Internet*, Seiten 19–25 (ACM, 2008). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.4747&rep=rep1&type=pdf.

[RFC2560] M. MYERS, R. ANKNEY, A. MALPANI, S. GALPERIN, und C. ADAMS. *X.509 Internet Public Key Infrastructure – Online Certificate Status Protocol - OCSP*. Request For Comments – RFC 2560. http://www.ietf.org/rfc/rfc2560.txt, 1999.

[RFC5246] T. DIERKS und E. RESCORLA. *The Transport Layer Security (TLS) Protocol Version 1.2*. Request For Comments – RFC 5246. http://www.ietf.org/rfc/rfc5246.txt, August 2008.

[RFC5849] E. HAMMER-LAHAV. *The OAuth 1.0 Protocol*. Request For Comments – RFC 5849. http://www.ietf.org/rfc/rfc5849.txt, April 2010.

[Saki10] NAT SAKIMURA. *NTT docomo is now an OpenID Provider*. March 9, 2010. http://openid.net/2010/03/09/ntt-docomo-is-now-an-openid-provider/.

[SAML-SecP(v2.0)] FREDERICK HIRSCH, ROB PHILPOTT, und EVE MALER. *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, 15.03.2005. http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf, 2005.

[SAML(v2.0)] SCOTT CANTOR, JOHN KEMP, ROB PHILPOTT, und EVE MALER. *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, 15.03.2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, 2005.

[SGBV] *Sozialgesetzbuch - Fünftes Buch (V) - Gesetzliche Krankenversicherung*. zuletzt geändert durch Art. 1 G v. 30.7.2009 I 2495. http://bundesrecht.juris.de/bundesrecht/sgb_5/, 2009.

[SKS10] PAVOL SOVIS, FLORIAN KOHLAR, und JÖRG SCHWENK. *Security Analysis of OpenID*. In *Proceedings of* Sicherheit 2010, Lecture Notes in Informatics (LNI) (GI-Edition, 2010).

[Slem01] M. SLEMKO. *Microsoft passport to trouble*. http://alive.znep.com/marcs/passport/, 2001.

[TaWa09] RYU WATANABE und TOSHIAKI TANAKA. *Federated Authentication Mechanism using Cellular Phone - Collaboration with OpenID*. In *ITNG '09: Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations*, Seiten 435–442 (IEEE Computer Society, 2009).

[TsTs07] EUGENE TSYRKLEVICH und VLAD TSYRKLEVICH. *Single Sign-On for the Internet: A Security Story*. http://www.orkspace.net/secdocs/Conferences/BlackHat/USA/2007/OpenID%20-%20%20Single%20Sign-On%20for%20the%20Internet-paper.pdf, 2007.

[Wagn09] OLIVER WAGNER. *Eine Milliarde OpenID Accounts*. December 17, 2009. http://www.agenturblog.de/2009-12/eine-milliarde-openid-accounts/.

[Yadis(v1.0)] JOAQUIN MILLER. *Yadis Specification - Version 1.0*. March 18, 2006. http://yadis.org/wiki/Yadis_1.0_%28HTML%29.

Meeting EHR Security Requirements: Authentication as a Security Service

Basel Katt[1] Thomas Trojer[1] Ruth Breu[1] Thomas Schabetsberger[2] Florian Wozak[2]

[1] Research Group Quality Engineering, University of Innsbruck, Austria.
[2] ITH-Icoserve, Innsbruck, Austria.

Abstract: Electronic Health Record (EHR) is a promising concept to collect and manage electronic health information of all citizens. Integration the Heathcare Enterprise (IHE) was one of the first initiatives that aims at standardizing the way healthcare systems exchanging information in a distributed environment. Based on EHR concepts and IHE profiles different approaches have been introduced in the industry and the literature to implement and apply solutions for different stakeholders in the healthcare domain (see e.g., `http://www.ith-icoserve.com/`). Due to the sensitivity of the data dealt with in these systems, security is a major concern that must be considered. In previous work we have presented a general architectural solution to apply the evolving *Security as a Service* (*SeAAS*) paradigm in distributed architectures for EHR in conformance to IHE–proposed profiles. While our architecture proposed is generic and covers all security requirements, we focus in this work on one security requirement, namely, authentication and show how it can be offered as a service while adhering to IHE profiles. [1]

1 Introduction

Information and communication technologies have been involved in most sectors of our lives, and healthcare is not an exception. Electronic Health Record (EHR) systems have been proposed and researched recently aiming at decreasing healthcare costs, increasing healthcare quality and reducing medical errors. IHE was one of the first initiatives started in 1998 with a main goal of building a framework that seamlessly enables the exchange of health information across multiple healthcare institutions and enterprises. While IHE does not create new standards, it proposes profiles that specify precisely how current standards can be used to reach it goals. Due to the sensitivity of the information that healthcare system are dealing with, security is one of the major concerns that must be tackled. Despite of the fact that IHE has recognized the importance of security by introducing few profiles that tackle different security requirements, however they are oversimplified, vague and do not consider architectural design [KTB+10].

IHE IT infrastructure profiles use the Service Oriented Architecture (SOA) paradigm in

[1] This work is paritally supported by the Austrian Federal Ministry of Economy as part of the Laura-Bassi —Living Security Models —project FFG 822740

its design, thus, IHE based systems can be featured as a highly heterogeneous and distributed. The current main practice to offer security functionalities in such highly dynamic and distributed environments is based on *end point security* concept. End point security is based on putting security functionality exclusively at end points, which means that each actor —functional component of the healthcare enterprise —in any domain must implement, maintain, and manage its own security related functions. Recent study [HMB09] shows that this methodology is inadequate and ineffecient in distributed and heterogenous systems. The proposed alternative to end point security is the *Security As A Service (SeAAS)* paradigm. SeAAS aims at extracting all security functionalities and mechanisms from end points in one domain and offer these functions as a central services for the whole end points in that domain. In [KTB+10] we proposed a general architectural solution to apply SeAAS concepts in IHE based healthcare systems that are based in theirdesign on the Cross-Document Sharing (XDS) profile [tHEI09a, tHEI09b]. We proposed to offer secuirty functionalities as services for each XDS affinity domain —a group of healthcare enterprises that have agreed to work together using a common set of policies and share a common infrastructure —without discussing the details of each security service. In this paper we move a head and discuss how authentication can be offered to an affinity domain based on the general architecture we proposed previously.

The rest of this paper is organized as follows. In Section 2 we discuss the concept of *Security as a Service*. In Section 3 we focus our study on the authentication service and present how brokered authentication can be offered as a service. Finally, we conclude and discuss future work in Section 4.

2 Security as a Service for Distributed EHR Systems

SeAAS, unlike end point security, provides security functionality centrally for endpoints within a common domain. In most scenarios nowadays endpoint security is applied. Setting all security mechanisms at the endpoint in such distributed and heterogeneous environment (like IHE-based infrastructures) increases dramatically the processing overhead applied on each endpoint. This yields the management and maintenance of these decentralized security mechanisms an exhausting tasks, and poses interoperability challenges [MHB09]. The benefits of our SeAAS (please refer to [KTB+10] for more details on the genera architecture) solution can be gained in the following issues:

- Performance: In critical systems like healthcare systems that deals with people's lives, performace is one of the key factors that should be considered by architects and designer. Security services involve performance costly functions that affects the perfromance of the whole system. An emperical study conducted in the context of Sectissimo project (http://www.sectisimmo.info) showed that SeAAS prototype performed better than end point security and was at least 1.2 time faster (More details can can be found in a paper to appear soon in the context mentioned project). Thus, the first advantage of SeAAS is performance.
- Maintainance and policy management: The maintainance and the managment of

security solutions for a dynamic distributed and heterogenous systems is comlex task. With security functions done at end points, security mechanisms are spread over the system infrastructure and involve all functional services. Thus, in order to keep the solutions updated with (i) the new functions or policy changes required due to the changing of security requirements, or (ii) upgrades of current solutions to cope with new security risks and threats, changes must be propogated to each end point. With a larg number of services and end points, maintenance and management tasks will be very inefficient and complex. Central solution for security services that provide central secuirty services eases the updating tasks as they are done once and do not require any propogation.

- Configurability: Our solution allows for two main types of configuration at two layers. First, at the upper layer, we have the composition policy that indicate which sercurity services to invoke and in which order. Second, the configuration of each security service in order to offer more than one security pattern, more about security patterns can be found in [DFLPW07, ESP07, RGFMP06]. Security patterns provide different solution for each security service based on different requirements. For example, authentication can be either direct authentication, borkered authentication, distributed authentication (fedirated identity), or centralized authentication [Erl09].

The concept of SeAAS bases on two modules, namely the *SeAAS engine* and *security services*. The SeAAS engine is responsible of orchestrating security functionality according to requests of secured endpoints. Deciding what are the needed secuirty requirements, i.e., secuirty services that must be invoked, and in which order these services must be invoked is done using a declarative policies called *Composition Policies*. Furthermore, security services can be classified into two types. First, *primitive security services* implement basic security or security related functionality, like (de-)encryption, signature, and time stamping. Second, *composed security services* utilize multiple primitive security services according to a general security requirement to be fulfilled. Based on this concept we introduced in [KTB+10] a SeAAS architecture to an IHE-based healthcare system. Security services to realize e.g., *authorization*, *non-repudiation*, *monitoring* and *authentication* are briefly mentioned.

Based on IHE XDS profile each affinity domain contains of four main actors: one document registry, one or more document repositories, one patient ID service,a gateway, a document consumer, and a document source, more details about XDS profile can be found in [tHEI09a]. Assuming an affinity domain with three document respositories, Figure 1 shows how this domain can be extended with the SeAAS components. Upon receiving a request from another domain by the gateway ①, the gateway forwards this request to the SeAAS engine ②. Based on the composition policy that corresponds the received request ③, the SeAAS engine invokes the required composed security services ④ in the order mentioned in the policy (policy can be defined as WS-BPEL [OAS], or WS-Policy). While composed security services are executed, primitive security services can be invoked ⑤. Finally, after all required security services are executed and the security requirement are fulfilled, the SeAAS engine returns the final decision to the gateway ②. Upon a positive decision the gateway forwards the functional request to the corresponding actor ⑥.

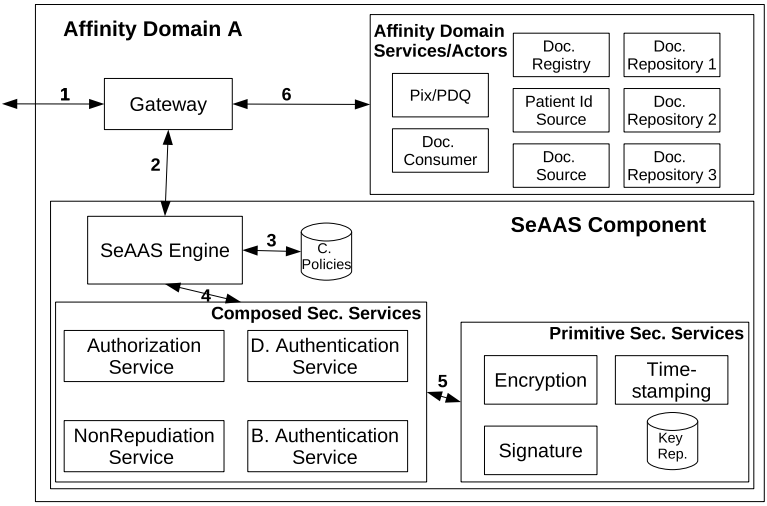

Figure 1: SeAAS architecture in an affinity domain.

It can be noticed that for any composed security service, the SeAAS component can offer more than one security pattern, (mechanism or solution). For example, Figure 1 shows that two authentication can be offered either as *Direct (D.)* or *Brokered (B.)* authentication. We show these two service for authentication protocol in separate components, however, in reality we will have one service that can be configured to act as direct of brokered authentication service.

3 Authentication Security Service

Authentication service is the service that aims at the verification of identiy. Authentication can be considered from two perspectives, node and user authentication, the first authenticate the node, using Transport Layer Secuirty (TLS) for example, and the latter verifies the authentication of users.

Transaction ITI-19 [tHEI09a] suggests the mutual authentication of nodes. Authentication of nodes is useful to provide trusted channels for specific transactions using functionality or data provided by multiple nodes. Nodes are therefore provided with trusted certificates and validation of those is covered by credentials validation services. On the other hand, transactions ITI-2,ITI-2, and ITI-4 [tHEI09a] suggest user authentication based on a challenge and response mechanism to verify the identity of an individual communication with the enterprise. *Kerberos* protocol was suggested to be used [NT94]. Kerberos user/password authentication is available for users within a protected domain, beside more sophisticated means of identity provisioning like smart cards or biometrics available to protected domains and external ones.

In this work we focus only on user authentication to be offered as a service. Two main drawbacks can be identifed in the IHE profiles related to authentication (cf. Section 2). First, it only proposes one authentication pattern and technology. Different health care institution apply different authentication services and protocols, based on different authentication patterns. For example brokered authentication with kerberos, X509 PKI, or STS (Security Token Service) options, or distributed authentication. Proposing only one solution oversimplifies the problem and decreases the viability of this service. Second, each IHE actor must implement and take care of the authentication mechanism by its own, which dereases the overall performance of the system. Applying SeAAS allows (i) offering multiple authentication patterns for the authentication service due to the configurability feature, and (ii) remove the security functionality form the end points and apply then in a dedicated services for authentication mechanism, thus enhancing the performance.

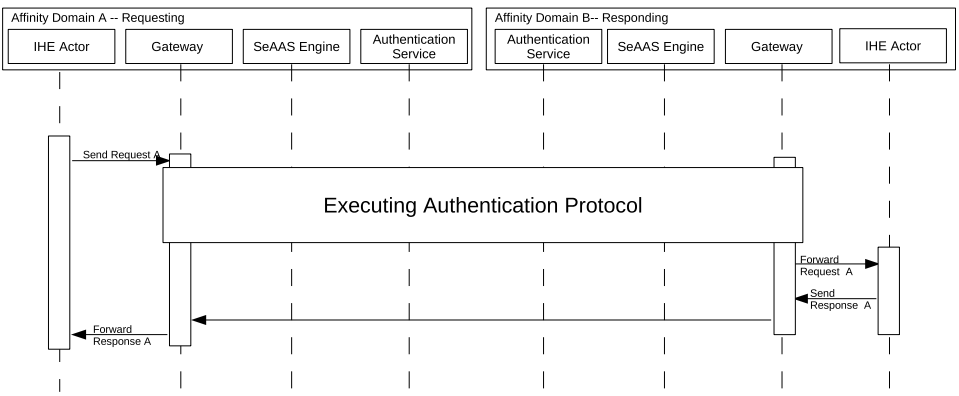

Figure 2: Sequence diagram shows how authentication functionality is moved from the IHE actors' endpoints and executed by the SeAAS components.

Figure 2 shows a how the execution protocol is moved from the IHE actors (document repository, registry etc.) to the SeAAS engine and authentication service. After validating the identity of the user the result is sent to the gateway. If the user is authenticated, the gateway forward the rquest further to the ITH actor, other wise send back an error message to the requesting gateway. The figure does not show the details of the authentication service, which might support different authentication patterns. The selection of the suitable one to execute is done by the SeAAS engine based on the *composition policy*. In the following we discuss the authentication service using a general borker authentication pattern. Please note that authroization that must be checked after authentication is out of scope of this work and is not shows in Figure 2. Furthermore, we assume that the response that is sent back to the requesting domain does not need any authentication check, which is the normal situation. That is why the response message is sent directly from the gateway of the responding domain to the gateway of the requesting domain.

3.1 Brokered Authentication

Brokered authentication pattern is used when the both the service consumer and the service provider do not trust each other and the consumer require an access to multiple services. An authentication broker in this case is responsible for authenticating the consumer and issuing a secuirty token to the consumer. This security token is used by the consumer to the access the service.

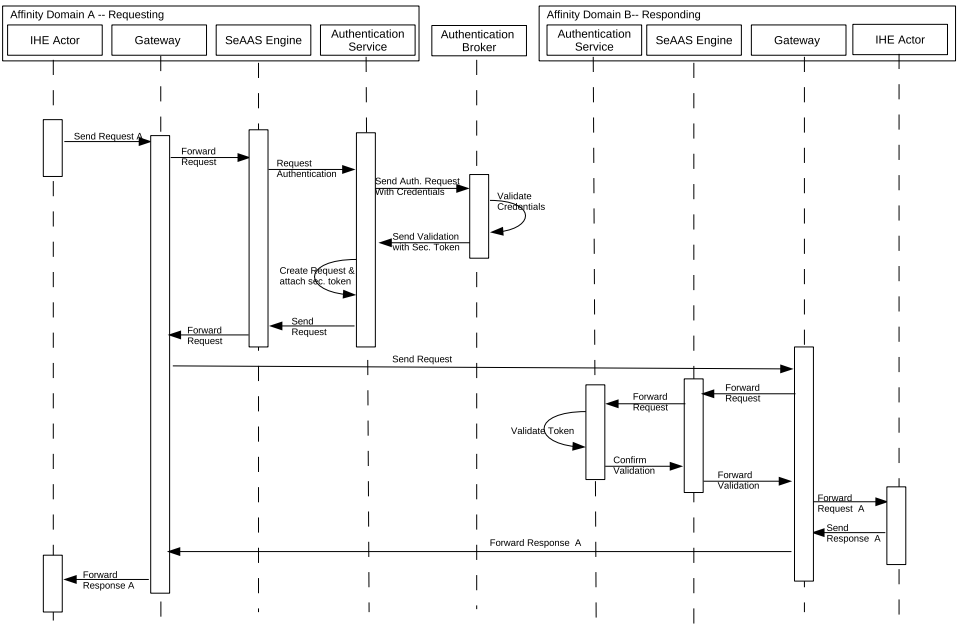

Figure 3: Brokered authentication service.

Figure 3 shows the execution sequence when brokered authentication is used to authenticate a user in an affinity domain A that is trying to get an access to a service in an affinity domain B. After the gateway of the requesting domain receives the request from the IHE actor, it forword the request to the SeAAS engine, which in turn forward the request to the required security services, authentication in our case. Authentication service send an authentication request to the authentication broker with the credentials of the requesting user. After the authentication broker validate the credentials it sends the validation with a security token to the authentication service. In the case of Kerberos, this will be the service ticket. Using the received token and the original request, the authentication service creates a new request with the security token attached to it. This request is forwarded to the SeAAS engine, which will send it forward to the requesting gateway. At this stage the request is created with the required security attachment and ready to be sent to the domain B. The requesting gateway send this request to the responding gateway, which in turn forwards the request to the SeAAS engine at domain B. After checking the required security service that need to be invoked, in our case only the authentication service, it forwards this

request to the authentication service. The authentication service validates the token that is sent with the request and confirms the validation to the SeAAS engins, which in turn forward the validation to the gateway in domain B. Upon a positibe validation, the gateway at the responding domain forwards the request to the IHE actor. The actor proccess the request and send back a response to the requesting IHE actor through the gateways in both domains.

4 Conclusion and Future Work

In this work we present an architectural solution for applying the evolving *SeAAS* paradigm to secure healthcare systems focusing on one security measure, namely, user authentication. SeAAS methodology overcomes the shortcomings of the current widely adapted *endpoint security* solutions with respect to management, maintainability and performance. In the future we plan to tackle other security requirements and develop a proof of concept prototype of the SeAAS framework.

References

[DFLPW07] N. Delessy, E.B. Fernandez, M.M. Larrondo-Petrie, and J. Wu. Patterns for access control in distributed systems. In *Proceedings of the 14th Conference on Pattern Languages of Programs*, pages 1–11. ACM, 2007.

[Erl09] T. Erl. *SOA Design Patterns*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2009.

[ESP07] R. Erber, C. Schlager, and G. Pernul. Patterns for Authentication and Authorisation Infrastructures. 2007.

[HMB09] M. Hafner, M. Memon, and R. Breu. SeAAS-A Reference Architecture for Security Services in SOA. *Journal of Universal Computer Science*, 15(15):2916–2936, 2009.

[KTB+10] B. Katt, T. Trojer, R. Breu, T. Schabetsberger, and F. Wozak. Meeting EHR Security Requirements: SeAAS Approach. In *EFMI STC 2010. Accepted*, June 2010.

[MHB09] M. Memon, M. Hafner, and R. Breu. Security As A Service: A Reference Architecture for SOA. In *7th International Workshop on Security in Information Systems (WOSIS 2009)*, Milan, Italy, May 2009. Springer, Springer.

[NT94] B.C. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, September 1994.

[OAS] OASIS. Web Services Business Process Execution Language (WSBPEL) TC. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel.

[RGFMP06] D.G. Rosado, C. Gutierrez, E. Fernandez-Medina, and M. Piattini. Security patterns and requirements for internet-based applications. *Internet Research*, 16(5):519–536, 2006.

[tHEI09a] Integrating the Healthcare Enterprise (IHE). *IT Infrastructure (ITI) Technical Framework, Volume 1, Integration Profiles*. IHE, August 2009.

[tHEI09b] Integrating the Healthcare Enterprise (IHE). *IT Infrastructure (ITI) Technical Framework, Volume 2a, Transactions ITI–1 through ITI–28*. IHE, August 2009.

perspeGKtive 2010

Kurzbeitrag zum Workshop

Grundzüge eines Sicherheitskonzepts für Arztpraxen mit Hilfe von Attack Trees und unter Berücksichtigung der Gesundheitstelematik

Raffael Rittmeier, Dr. Karsten Sohr

Fachbereich Mathematik und Informatik
Universität Bremen
Bibliothekstr. 1
28359 Bremen
raffael@informatik.uni-bremen.de
sohr@tzi.de

Abstract: Ziel dieser Arbeit ist es, den Schutz sensibler Patientendaten zu verbessern. Es wird eine Vorgehensweise zur Erstellung eines Sicherheitskonzepts für Arztpraxen skizziert. Dabei werden die entsprechenden Standards des Bundesamts für Sicherheit in der Informationstechnik (BSI) durch Bedrohungsbäume (*attack trees*) erweitert. Damit können z. B. Bedrohungen analysiert werden, die durch die Einführung neuer Technologien entstehen können. Ein spezieller IT-Grundschutzbaustein für Arztpraxen und eine freiwillige Zertifizierung werden diskutiert.

1 Einleitung

Arztpraxen setzen heute zunehmend auf digitalisierte Arbeitsprozesse. So sind Ärzte ab 2010/11 verpflichtet ihre Abrechnung leitungsgebunden elektronisch zu übertragen, statt wie bisher quartalsweise einen Datenträger bei der Kassenärztlichen Vereinigung einzureichen [KBV08]. Mit der neuen elektronischen Gesundheitskarte (eGK) sollen die Beteiligten des Gesundheitswesens über das Internet vernetzt werden. Unter Protest der Ärzteschaft macht die aktuelle Gesetzgebung die Onlineanbindung für Arztpraxen verpflichtend [AEZ10]. Durch die zunehmende Digitalisierung und Vernetzung wird die Informationssicherheit in Arztpraxen zunehmend wichtiger.

Ärzte speichern und verarbeiten medizinische Daten ihrer Patienten und damit nach Bundesdatenschutzgesetz (BDSG) besonders schützenswerte personenbezogene Informationen. Diese Daten müssen unter allen Umständen vertraulich behandelt werden. Bei der elektronischen Speicherung und Verarbeitung von Patientendaten müssen auch die Computersysteme besonders geschützt sein, um unbefugten Zugriff zu verhindern. Durch die Verpflichtung zur Online-Verbindung entstehen jedoch zusätzliche Risiken, z. B. durch Fremdzugriff.

2 Ziele

Am Beispiel von Arztpraxen wird in diesem Beitrag eine Vorgehensweise vorgestellt, um die Informationssicherheit im Gesundheitswesen zu untersuchen und zu verbessern. Bedrohungen und Risiken werden erfasst und analysiert. Anschließend werden Maßnahmen zur Behandlung der Risiken vorgeschlagen. Daraus ergeben sich die Grundzüge eines Sicherheitskonzepts für Arztpraxen, bei dem der Schutz von Patientendaten und deren Vertraulichkeit im Vordergrund stehen.

Die am IT-Grundschutz des Bundesamts für Sicherheit in der Informationstechnik (BSI) ausgerichtete Vorgehensweise wird durch Bedrohungsbäume *(attack trees*) erweitert. Diese Darstellung ermöglicht es, die Ärzte und ihre Angestellten besser für den Datenschutz und die Informationssicherheit der Patientendaten zu sensibilisieren.

3 Methoden

Grundlage für diesen Beitrag ist die Untersuchung von drei Bremer Arztpraxen, in denen jeweils bis zu 400 Patienten pro Tag behandelt werden und die ihre Dokumentation weitgehend elektronisch führen [Ri09]. In Gesprächen mit den für die EDV verantwortlichen Ärzten wurden die Geschäftsprozesse und die Infrastruktur der Arztpraxen erfasst. Die Daten wurden durch die Infrastrukturanalyse und die Schutzbedarfsermittlung nach dem IT-Grundschutz des BSI aufbereitet [BSI08a]. Die vom BSI empfohlene Vorgehensweise zur Erstellung einer Bedrohungs- und Risikoanalyse wurde durch Bedrohungsbäume erweitert [BSI08b]. In der erweiterten Risikoanalyse wurden die relevanten Bedrohungen bewertet [Ec09]. Anschließend wurde ein Maßnahmenkatalog zusammengestellt, um die Risiken zu mindern.

Bezüglich der Einführung der Telematikinfrastruktur (TI) wurden u.a. mögliche Bedrohungen ermittelt, die durch die Integration des Konnektors in das Praxisnetz entstehen können. Als Quellen dienten v. a. die Spezifikationen der gematik.

3.1 Verwendung von Bedrohungsbäumen

Bruce Schneier hat die Bedrohungsbäume im Bereich der IT-Sicherheit 1999 eingeführt [Sc99]. Diese sind von der Fehlerbaumanalyse (*fault tree analysis*) aus dem „Safety"-Bereich abgeleitet. Dort werden Baumstrukturen z. B. im Rahmen von Sicherheitsüberprüfungen kerntechnischer Anlagen eingesetzt [Ha81].

Bedrohungsbäume zeigen Abhängigkeiten zwischen verschiedenen Ereignissen bzw. Angriffsschritten auf. Gängig sind die grafische Darstellung sowie die Textform. In dieser Arbeit wird Letztere gewählt, da sie für detaillierte Bäume übersichtlicher ist [Sc04].

Ein Baum besteht aus einer Menge von Knoten. Ein Knoten kann dargestellt werden als:

- Menge von Knoten, die alle erfüllt werden müssen, damit ein Angriff erfolgreich ist. In diesem Fall spricht man von einer UND-Verknüpfung.
- Menge von Knoten, von denen mindestens ein Knoten erfüllt werden muss, damit der Angriff erfolgreich ist. In diesem Fall spricht man von einer ODER-Verknüpfung.
- Knoten, dem keine weiteren Knoten mehr folgen (Blätter).

Das Angriffsziel befindet sich im obersten Knoten, der sogenannten Wurzel.

3.2 Beispiele

Im Folgenden werden zwei Beispiele für Bedrohungsbäume angegeben. Diese behandeln Angriffe auf ein Praxisverwaltungssystem (PVS).

Beispiel 1:

Angriffsziel: Unbefugter Zugriff auf vertrauliche Patientendaten im PVS

1 Inbesitznahme von Patientendaten, die auf dem PVS-Server gespeichert sind (ODER)

1.1 Physikalischer Angriff auf den PVS-Server (UND)

1.1.1 Zutritt zur Praxis (ODER)

1.1.1.1 Zutritt während der Sprechzeiten

1.1.1.2 Zutritt außerhalb der Sprechzeiten

1.1.2 Zutritt zum Serverraum

1.1.3 Zugriff auf PVS-Server und gespeicherte Patientendaten

1.2 Entfernter Angriff auf den PVS-Server

Erfolgreiche Angriffsszenarien von Beispiel 1 sind demnach: {1.1.1.1, 1.1.2, 1.1.3}, {1.1.1.2, 1.1.2, 1.1.3}, {1.2}.

Der Vorteil von Bedrohungsbäumen ist, dass sie wiederverwendbar sind und Teilbäume unabhängig voneinander betrachtet werden können. In Beispiel 2 wird der entfernte Angriff auf den PVS-Server (Subziel 1.2) genauer analysiert.

Beispiel 2:

Subziel 1.2: Entfernter Angriff auf den PVS-Server (ODER)

1.2.1 Anschluss eines eigenen Systems in der Praxis (UND)

1.2.1.1. Physikalischer Anschluss an das Praxisnetz

1.2.1.2. Gültige Netzkonfiguration einstellen

1.2.1.3. IP-Adresse des PVS-Servers beschaffen

1.2.1.4. Erfolgreicher Angriff auf Betriebssystem oder PVS-Software und Auslesen von Patientendaten

1.2.1.5. {1.1.1}

1.2.2 Lokale Übernahme eines PVS-Clients (UND)

1.2.2.1. Überwinden der Betriebssystem-Authentifizierung

1.2.2.2. Überwinden der PVS-Authentifizierung und Einsehen bzw. Export von Patientendaten

1.2.2.3. {1.1.1}

1.2.3 Entfernter Angriff über ISDN

1.2.4 Entfernter Angriff über DSL

Um erfolgreich zu sein, benötigen die Angriffsschritte 1.2.1 und 1.2.2 die Erfüllung des bereits aufgeführten Angriffsschritts 1.1.1. Um Redundanz zu vermeiden, wird lediglich auf den entsprechenden Teilbaum verwiesen. Unter Beibehaltung der Übersichtlichkeit können die Angriffsziele so bis zur gewünschten Tiefe verfeinert werden. Anschließend werden die Bedrohungen bewertet und Maßnahmen vorgeschlagen.

4 Ergebnisse

Die bekannten Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit gelten auch für den Schutz der Patientendaten. Schwerpunkt dieser Arbeit war die Sicherstellung der Vertraulichkeit der Daten. Das PVS verarbeitet die sensiblen Daten, die zusätzlich in Archiven und auf externen Speichermedien gesichert werden. Eine physikalische Trennung zwischen Internet und PVS minimiert die Risiken eines Internetanschlusses. Ist die strikte Trennung nicht möglich, müssen organisatorische und technische Schutzmaßnahmen die Vertraulichkeit der Patientendaten sicherstellen, z. B. durch verhaltensregelnde Richtlinien und den Einsatz von Firewalls und Intrusion Detection Systems (IDS). Weitere Bedrohungen entstehen z. B. durch Diebstahl oder Verlust von Speichermedien. Starke Verschlüsselungsmechanismen stehen dem entgegen.

Die geplante Telematikinfrastruktur soll die Akteure des Gesundheitswesens miteinander vernetzen. Verschiedene Telematikanwendungen werden ihre Daten in einem vernetzten System speichern. Der Versichertenstammdatendienst (VSDD) verarbeitet z. B. die verwaltungsrelevanten Daten der Patienten. Später sollen der elektronische Arztbrief und die elektronische Patientenakte (ePA) folgen.

Für Arztpraxen ist der Konnektor ein entscheidender Bestandteil. Er soll das Praxisnetz sicher mit der Telematikinfrastruktur verbinden und vor Bedrohungen aus dem Internet schützen. Werden die ausführlichen Vorgaben aus Spezifikationen und Common Criteria-Schutzprofil (*protection profile*) bei der Implementierung befolgt, wird die Vertraulichkeit der Patientendaten bei der Datenverbindung gewährleistet [BSI07]. Die Technologie ist jedoch komplex und eine fehlerfreie Implementierung kann nicht garantiert werden. Hinzu kommt, dass der korrekte Einsatz in der Arztpraxis gewährleistet werden muss. Durch die zusätzliche Technik erhöht sich der administrative Aufwand. Ohne einen IT-Dienstleister werden die Inbetriebnahme und die Wartung Probleme für die Arztpraxen darstellen. Verbesserungsbedarf besteht ebenfalls bei der Anbindung der PVS. Eine verschlüsselte Übertragung soll zwar ermöglicht werden, ist jedoch nicht vorgeschrieben [Ge08]. Die Technologie der Telematikinfrastruktur mag zwar einen entsprechenden Sicherheitsstandard aufweisen; sie gilt jedoch nicht zwingend für die Systeme beim Leistungserbringer wie z. B. bei dem PVS oder dem Praxisnetz.

Eine weitere Bedrohung ergibt sich durch die gesetzlich vorgesehene Datenwiederherstellung. Bei Verlust der elektronischen Gesundheitskarte oder beim Wechsel der Verschlüsselungsstärke soll der geheime Schlüssel automatisch rekonstruiert werden, um die sensiblen Patientendaten für eine neue Karte zu verschlüsseln. Die Sicherheit asymmetrischer Verschlüsselung basiert jedoch auf der Geheimhaltung des privaten Schlüssels. Wenn dieser durch Dritte rekonstruiert werden kann, ergibt sich ein entsprechendes Sicherheitsproblem.

5 Diskussion

Gesetzliche Vorgaben und die Einführung von Telematiksystemen fordern einen Internetanschluss der Arztpraxen. Dabei muss der unbefugte Zugriff auf Patientendaten unbedingt unterbunden werden. Ein spezieller IT-Grundschutzbaustein des BSI wäre sinnvoll, denn die bestehenden Empfehlungen genügen nicht und werden in der Praxis nur selten beachtet. Maßnahmen zur Verbesserung der Informationssicherheit müssen geeignet sein, um die optimierten Abläufe in den Arztpraxen nicht zu behindern. Des Weiteren müssen die Maßnahmen konkret genug sein, damit sie umgesetzt und überprüft werden können. Hier bieten sich Grundschutzbausteine an, gegen die bei einer Zertifizierung evaluiert werden könnte. Basierend auf der Evaluierung könnte dann auch durch die Bundesärztekammer bzw. Kassenärztliche Bundesvereinigung ein Siegel für Informationssicherheit und Datenschutz vergeben werden. Eine Zertifizierung wird sicherlich nur auf freiwilliger Basis möglich sein.

Attack Trees ergänzen die Bedrohungs- und Risikoanalyse sinnvoll. Risiken können übersichtlich dargestellt werden und die Sensibilisierung für Sicherheitsprobleme wird erleichtert. Der Aufwand lässt sich reduzieren, da die Baumstruktur das Wiederverwenden einzelner Teilbäume ermöglicht.

Organisatorische Maßnahmen, wie z. B. Schulungen, dürfen nicht vernachlässigt werden. Die Einhaltung des BDSG in seiner aktuellen Fassung muss stärker überprüft werden. Dazu gehören nach § 42a BDSG auch die konsequente Meldung von Datenlecks und die Benachrichtigung der von Datenverlusten Betroffenen.

Die Einführung der elektronischen Gesundheitskarte ist eine Möglichkeit, Patientendaten besser zu schützen. Es wird jedoch nicht ausreichen, wenn die neuen Telematiksysteme sicher konzipiert werden. Die bestehenden Infrastrukturen in den Arztpraxen müssen ebenso kritisch untersucht und dürfen nicht weiter ignoriert werden. Es wäre wünschenswert, wenn die Betreibergesellschaft klare Sicherheitsvorgaben für die PVS veröffentlichen würde. Da die gematik dies nicht als ihren Aufgabenbereich versteht, sind die zuständigen Ärztekammern, Kassenärztlichen Vereinigungen und Datenschutzaufsichtsbehörden in der Pflicht, Mechanismen zu entwerfen, um die Sicherheit der sensiblen Daten im Gesundheitswesen zu verbessern. Die in diesem Beitrag vorgestellte Vorgehensweise könnte als Basis genutzt werden, um ein Auditverfahren speziell für Arztpraxen zu entwickeln. So könnten der Datenschutz und die Informationssicherheit im Gesundheitswesen verbessert werden.

Literaturverzeichnis

[AEZ10] Online-Stammdatenabgleich kommt jetzt ins Gesetz. In: Ärzte Zeitung online, 2010, http://www.aerztezeitung.de/praxis_wirtschaft/gesundheitskarte/article/607574/online-stammdatenabgleich-kommt-jetzt-gesetz.html (Abruf am 09.08.10).

[BSI07] Bundesamt für Sicherheit in der Informationstechnik: Common Criteria Schutzprofil (Protection Profile) für einen Konnektor im elektronischen Gesundheitswesen. Bonn, 2007, https://www.bsi.bund.de/cae/servlet/contentblob/480298/publicationFile/29312/PP0033b_pdf.pdf (Abruf am 09.08.2010).

[BSI08a] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 100-2, IT-Grundschutz-Vorgehensweise. Bonn, 2008, https://www.bsi.bund.de/cae/servlet/contentblob/471452/publicationFile/30758/standard_1002.pdf (Abruf am 09.08.2010).

[BSI08b] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 100-3, Risikoanalyse auf der Basis von IT-Grundschutz. Bonn, 2008, https://www.bsi.bund.de/cae/servlet/contentblob/471454/publicationFile/30757/standard_1003.pdf (Abruf am 09.08.10).

[Ge08] Gematik: Konnektorspezifikation 2.10.0. Berlin, 2008, http://www.gematik.de/cms/media/dokumente/release_2_3_4/release_2_3_4_dezkomponenten/gematik_KON_Konnektor_Spezifikation_V2100.pdf (Abruf am 09.08.2010), S. 136.

[Ec09] Eckert, C.: IT-Sicherheit: Konzepte-Verfahren-Protokolle. 6. Auflage, Oldenbourg Wissenschaftsverlag, München, 2009.

[Ha81] Haasl, D.F., et. al.: Fault tree handbook. Office of Nuclear Regulatory Research, Nuclear Regulatory Commission, Washington, DC (USA). 1981 (NUREG-0492). – Technical Report.

[KBV08] Kassenärztliche Bundesvereinigung: Änderungen der Richtlinien für den Einsatz von IT-Systemen in der Arztpraxis. In: Dtsch Arztebl, 105, 2008, Nr. 12, S. A–650 / B–570 / C–558.

[Ri09] Rittmeier, R.: Grundzüge eines Sicherheitskonzepts für Arztpraxen unter Berücksichtigung der Gesundheitstelematik. Diplomarbeit, Universität Bremen, 2009.

[Sc99] Schneier, B.: Attack Trees. In: Dr. Dobb's Journal, 24, 1999, Nr. 12, S. 21–29.

[Sc04] Schneier, B.: Secrets and lies: digital security in a networked world. Wiley, Indianapolis, Ind., USA, 2004, S. 324.

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Busi-ness 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Erfahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT: Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik – Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwick-lung und -wartung im Fachbereich Wirtschaftsinformatik

P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics

P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I^2CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf,
Ulrike Lechner, Phayung Meesad,
Herwig Unger (Eds.)
10th International Conference on
Innovative Internet Community Systems
(I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair,
Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on
Electronic Voting 2010
co-organized by the Council of Europe,
Gesellschaft für Informatik and
E-Voting.CC

P-168 Ira Diethelm, Christina Dörge,
Claudia Hildebrandt,
Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer
Forschungsmethoden und Perspektiven
der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek
Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung
der Fachgruppe E-Learning
der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski,
Martin Juhrisch (Hrsg.)
Modellierung betrieblicher
Informationssysteme (MobIS 2010)
Modellgestütztes Management

P-174 Arslan Brömme, Torsten Eymann,
Detlef Hühnlein, Heiko Roßnagel,
Paul Schmücker (Hrsg.)
perspeGKtive 2010
Workshop „Innovative und sichere
Informationstechnologie für das
Gesundheitswesen von morgen“

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de