MAIL VOM RECHTSANWALT? HERAUSFORDERUNGEN SICHERER MANDANTENKOMMUNIKATION

Dominik Leibenger[1], Stephan Ory[2], Christoph Sorge[1]

[1]Universität des Saarlandes, juris-Stiftungsprofessur für Rechtsinformatik und CISPA, 66123 Saarbrücken, DE
{dominik.leibenger,christoph.sorge}@uni-saarland.de, http://www.legalinf.de
[2]Rechtsanwalt, Kanzlei Ory, 66346 Püttlingen, DE
stephan.ory@ory.de, http://www.ory.de
Die Autoren sind außerdem Gesellschafter der SOLE Software GmbH, die Werkzeuge für sichere Kommunikation entwickelt.

Schlagworte: ***Sichere Mandantenkommunikation, Ende-zu-Ende-Verschlüsselung***

Abstract: *Auch in Kanzleien ist die Nutzung elektronischer Kommunikationswege aus dem Alltag nicht wegzudenken. Ihre derzeitige Ausprägung droht jedoch, die Verschwiegenheitspflicht von Anwälten zu gefährden. Zwar existieren sichere Verschlüsselungslösungen seit Jahrzenten; praktisch kommunizieren Anwalt und Mandant jedoch oft unverschlüsselt via Mail. Der Beitrag analysiert den State of the Art der Mandantenkommunikation aus rechtlicher und technischer Sicht, diskutiert Lösungen und präsentiert ein System zur Ende-zu-Ende-verschlüsselten Kommunikation, das ohne technische Vorkehrungen auf Mandantenseite auskommt.*

1. Hintergrund

Der vertrauliche Umgang mit Informationen der Mandanten ist selbstverständliche Basis des Anwaltsberufs. Die Verschwiegenheitspflicht ist im Berufsrecht (§ 2 BORA, § 43 BRAO) beschrieben, das Offenbaren von Mandantengeheimnissen wird durch § 203 StGB unter Strafe gestellt: Wer als Rechtsanwalt ohne Einwilligung Informationen preisgibt, die er von Mandanten erhält, riskiert eine Geldstrafe oder Freiheitsstrafe bis zu einem Jahr. Schon das Vorliegen eines Mandatsverhältnisses ist vertraulich. Es ist umstritten, wäre aber folgerichtig, auch eine unsichere Art der Kommunikation zwischen Anwälten und Mandanten nach diesen Grundsätzen zu beurteilen – unabhängig von Sanktionen sollte ein bestmöglicher Schutz der Kommunikationsinhalte für jeden Anwalt selbstverständlich sein. Obwohl Verfahren zur sicheren elektronischen Kommunikation schon lange existieren, werden sie bislang in der Anwaltschaft kaum eingesetzt (vgl. für einen Überblick [Sorge 2016]). Mag sich durch Einwilligung des Mandanten eine berufsrechtliche Lösung finden lassen, ist dieser Zustand doch zumindest unbefriedigend. Der vorliegende Beitrag stellt daher eine technische Lösung vor, die eine einfache und dennoch abhörsichere Mandantenkommunikation ermöglicht.

2. Elektronische Kommunikation: State of the Art

Geht es um elektronische Kommunikation, gilt als Mittel der Wahl oft die E-Mail. Zur sicheren Mandantenkommunikation ist sie jedoch ohne weitere Schutzmaßnahmen gänzlich ungeeignet: Nicht nur sind Kommunikationsbeziehungen – und damit auch Mandatsverhältnisse – für die Mailserverbetreiber des Senders und Empfängers sowie etwaige Zwischenstationen sichtbar, i. d. R. liegen auch Inhaltsdaten im Klartext vor: Während der Übertragung könnten bösartige Infrastrukturbetreiber und staatliche Stellen Inhalte ausspähen. Im Anschluss bleiben die Daten oft über Monate auf dem Mailserver des Empfängers gespeichert, wo sie zusätzlich Gefahren durch Datendiebstahl – etwa aufgrund von Hackerangriffen – ausgesetzt sind.

Eine gängige Alternative ist die Verwendung einer Webplattform zum Nachrichtenaustausch: Der Mandant erhält Zugangsdaten zu einer Webseite, die Zugriff auf ein zentral verwaltetes Postfach ermöglicht. Daten liegen dort im Klartext vor; Transportverschlüsselung (HTTPS/TLS [Dierks/Rescorla 2008]) verhindert aber

ein Ausspähen auf dem Übertragungsweg. Sichere Kommunikation ist so möglich, solange der einwandfreie Betrieb der Plattform sichergestellt ist. Die theoretische Gefahr von Datendiebstahl durch Angriffe auf die technische Infrastruktur bleibt jedoch bestehen: Eine kanzleieigene Infrastruktur führt zu hohen Administrationskosten; ein externer Dienstleister birgt zusätzliche Gefahren und rechtliche Hürden, da i. d. R. vom Personenbezug der Daten auszugehen ist. Insbesondere erfordert die Ausgestaltung als Auftragsdatenverarbeitung nicht nur eine schriftliche Auftragserteilung, die den Anforderungen aus § 11 Abs. 2 Satz 2 Nr. 1 bis 10 entspricht. Darüber hinaus sind auch Kontrollpflichten aus § 11 Abs. 2 Satz 3-4 BDSG zu erfüllen.

Einen wirksamen Schutz hinsichtlich der Inhaltsdaten bietet der Einsatz von Ende-zu-Ende-Verschlüsselung: Kommunikationsinhalte zwischen Kanzlei und Mandanten werden schon auf den Geräten der Kanzlei / der Mandanten so verschlüsselt, dass sie nur von der Gegenseite entschlüsselt werden können. Verschlüsselte E-Mails sind vor Ausspähen auf dem Übertragungsweg und vor späterem Datendiebstahl über den speichernden Mailserver geschützt. Die technischen Verfahren (S/MIME [Ramsdell/Turner 2010], PGP/MIME [Elkins et al. 2001]) existieren (in ersten Versionen) seit Mitte der 90er und gelten als sicher. Erlaubt man die Offenlegung von Kommunikationsbeziehungen, sind sie zur sicheren Mandantenkommunikation geeignet.

Dennoch konnte sich bislang keines der Verfahren in der Breite durchsetzen. Möchte ein Anwalt verschlüsselte Mails mit seinen Mandanten austauschen, genügt es nicht, dies auf Kanzleiseite einzurichten – auch jeder Mandant müsste technische Vorkehrungen treffen. Ist ein Mandant dazu nicht bereit, wird in der Praxis oft auf Papierform ausgewichen oder es findet eine Einigung auf unverschlüsselte Kommunikation statt.

3. Idee und Ziel

In diesem Beitrag untersuchen wir, wie praktisch sichere Kommunikation zwischen Anwalt und Mandant erreicht werden kann, ohne zusätzliche technische Anforderungen – vor allem auf Mandantenseite – einzuführen. Die Kernidee ist, Anwälten und Mandanten einen Austausch von Nachrichten / Dokumenten über eine neu entwickelte, einfach bedienbare Webplattform zu ermöglichen, die im Gegensatz zu existierenden Lösungen sichere *Ende-zu-Ende-Verschlüsselung* integriert: Sämtliche Daten werden schon im Browser so verschlüsselt, dass nur die Kanzlei und berechtigte Mandanten sie entschlüsseln können. Betreiber der technischen Infrastruktur – und damit auch etwaige Angreifer nach erfolgreichem Datendiebstahl – haben weder Zugriff auf Kommunikationsinhalte noch auf Metadaten wie Namen / E-Mail-Adressen von Mandanten. So kann auch ein externer Dienstleister beauftragt werden, ohne die Vertraulichkeit ausgetauschter Informationen zu gefährden. Nach dem Stand der Technik verschlüsselte Daten sind für denjenigen nicht als personenbezogen anzusehen, der weder Zugriff auf den Schlüssel hat noch diesen ohne unverhältnismäßigen Aufwand erlangen kann (dazu ausführlich [Borges 2016, Rn. 35 ff]). Damit entfällt die Anwendbarkeit des Datenschutzrechts und folglich die Notwendigkeit von Vereinbarungen zur Auftragsdatenverarbeitung.

3.1. Anforderungen

An die Entwicklung einer praktisch einsetzbaren, Ende-zu-Ende-verschlüsselten Webplattform zur Mandantenkommunikation ergeben sich verschiedene Anforderungen, die sich gegenseitig beeinflussen.

Eine wichtige Anforderung ist die *Nutzbarkeit*: Kanzleimitarbeiter und Mandanten sollen auf die Plattform zugreifen können, ohne spezielle Software oder Browser-Erweiterungen installieren zu müssen. Die Plattform soll betriebssystemunabhängig auf allen gängigen Browsern laufen – einschließlich mobiler Endgeräte.

Neben Nutzbarkeit ist auch die *Sicherheit* des Systems essentiell: Technische Maßnahmen sollen sicherstellen, dass die Vertraulichkeit von Kommunikationsbeziehungen und -inhalten auch dann gewährleistet ist, wenn der Betreiber der technischen Infrastruktur Opfer eines erfolgreichen Datendiebstahls wird. Um das zu erreichen, sollen Klartextdaten ausschließlich im Browser der Kanzlei bzw. berechtigter Mandanten verarbeitet werden. Vor jeder Übertragung an die Webplattform sollen diese so verschlüsselt werden, dass nur berechtigte Nutzer sie entschlüsseln können.

Gleichzeitig sollen die Sicherheitsanforderungen die Nutzbarkeit der Software so wenig wie möglich beeinträchtigen; vom Nutzer gewohnte Abläufe hinsichtlich des Umgangs mit existierenden, unverschlüsselten Plattformen sollen anwendbar bleiben. Nötige kryptographische Maßnahmen sollen nach Möglichkeit automatisch und für die Nutzer *transparent* – also unbemerkt – durchgeführt werden. Das dem System zugrundeliegende Sicherheitsmodell muss dabei sorgfältig so definiert werden, dass seine *Funktionalität* möglichst wenig eingeschränkt wird: So soll ein Infrastrukturbetreiber bspw. zwar grundsätzlich keinen Zugriff auf E-Mail-Adressen von Mandanten haben; zum Zwecke des Versands einer Benachrichtigungsmail – etwa um auf neu eingegangene Dokumente hinzuweisen – kann diese Information aber im Einzelfall unerlässlich sein.

3.2. Sicherheitsmodell

Zur Definition eines geeigneten Sicherheitsmodells ist zunächst zu klären, welche Daten im entwickelten System anfallen können. *Benutzer* der Plattform sind Mitarbeiter der Kanzlei und ihre Mandanten. Für jeden Nutzer werden Zugangsdaten (Benutzername, Passwort) zur Authentifizierung benötigt. E-Mail-Benachrichtigungen setzen zudem die E-Mail-Adressen und – zwecks persönlicher Anrede – auch Namen, Titel und Geschlechter der Nutzer voraus. *Akten* organisieren zwischen Kanzlei und Mandanten fallspezifisch ausgetauschte Informationen. Jede Akte wird durch ein Aktenzeichen identifiziert, kann durch ein optionales Rubrum beschrieben werden und ist zugeordnet zu einer Menge zugriffsberechtigter Benutzer. *Dokumente* bzw. *Nachrichten* schließlich sind die Kommunikationsinhalte, die zwischen Kanzlei und Mandanten ausgetauscht werden. Jedes Dokument / jede Nachricht ist einer Akte zugeordnet und besteht aus einem Namen, einer optionalen Kurzbeschreibung und dem eigentlichen Inhalt. Neben den aufgezählten *Daten* (Kommunikationsinhalte) und *Metadaten* (Daten über Nutzer und Akten sowie beschreibende Daten zu Kommunikationsinhalten wie Dateinamen) fallen außerdem *Zugriffsdaten* an, die sich aus der Nutzung des Systems ergeben, aber unabhängig von konkreten Daten und Metadaten sind – etwa Zugriffs- und Änderungszeitpunkte.

Für die Festlegung der Sicherheitsgarantien unterscheiden wir zwischen vier Parteien, die mit dem System interagieren: Ein *primärer Infrastrukturbetreiber* kümmert sich um die Einrichtung und den Betrieb der Webplattform, ein *sekundärer Infrastrukturbetreiber*[1] übernimmt den Versand optionaler E-Mail-Benachrichtigungen; *Kanzleimitarbeiter* und *Mandanten* schließlich verwenden die Plattform zur Verwaltung von Akten und zur Kommunikation untereinander. Es wird vorausgesetzt, dass Kanzleimitarbeiter vollständig und die Infrastrukturbetreiber eingeschränkt vertrauenswürdig sind.

Der *primäre Infrastrukturbetreiber* ist „*honest but curious*“, d. h. er erbringt seinen Dienst (aufgrund finanzieller Anreize) ordnungsgemäß, indem er nicht von vereinbarten Protokollen abweicht und keine manipulierten Webseiten ausliefert, aber ihm wird unterstellt, dass er neugierig ist und daher versucht, unbefugt an vertrauliche Informationen zu gelangen. Dies ist eine in der IT-Sicherheit gängige Annahme und entspricht dem Szenario eines *passiven Angreifers*, der unbemerkt Zugriff auf Systeme des Anbieters erhalten und erfolgreich Daten entwenden kann. Unser Sicherheitsmodell garantiert, dass der primäre Infrastrukturbetreiber unter diesen Voraussetzungen lediglich Namen und E-Mail-Adressen von Kanzleimitarbeitern (nicht aber von Mandanten) erhalten und daten- und metadatenunabhängige Zugriffsdaten sehen darf, die sich durch die Nutzung der Infrastruktur ergeben. Er darf keinen Zugriff auf andere Daten oder Metadaten erhalten.

Da der *sekundäre Infrastrukturbetreiber* Klartext-Informationen über zu versendende Benachrichtigungen erhalten muss, lässt sich ein möglicher Datendiebstahl im Falle eines erfolgreichen Angriffs nicht vollständig ausschließen. Ein Angreifer könnte in diesem Fall Zugriff auf Namen und E-Mail-Adressen von Mandanten erhalten, an die während oder kurz vor Beginn des Angriffs Benachrichtigungen versendet wurden. Zugriff auf weitergehende Daten/Metadaten hingegen ist selbst bei aktiven Angriffen, bei denen der Angreifer die Funktion der Infrastruktur manipuliert, ausgeschlossen.

---

[1] Wir unterscheiden zwischen primärem und sekundärem Infrastrukturbetreiber, da für beide geringfügig unterschiedliche Anforderungen gelten. In der Praxis kann ein einzelner Anbieter auch beide Rollen einnehmen.

Kanzleimitarbeiter und Mandanten sind die einzigen Parteien, die Zugriff auf weitere Daten / Metadaten erhalten: *Kanzleimitarbeiter* erhalten vollständigen Zugriff auf alle Daten, Metadaten und Zugriffsdaten – ausgenommen ist nur das Lesen anderer Nutzer eigenständig festgelegter Passwörter. Entsprechend wird vorausgesetzt, dass Kanzleimitarbeiter nur über abgesicherte, funktionierende Systeme auf die Plattform zugreifen und ihre Zugangsdaten nicht in unbefugte Hände geraten. *Mandanten* erhalten Zugriff auf Akten, für die ihnen zuvor Zugriff durch einen Kanzleimitarbeiter eingeräumt wurde. Für die Einhaltung der Sicherheitsgarantien setzen wir lediglich ein sicheres, *symmetrisches* Verschlüsselungsverfahrens voraus.

4. Technisches Konzept

Betrieb und Nutzung einer Webplattform erfolgen – vereinfacht dargestellt – wie folgt: Ein Nutzer öffnet einen Browser und steuert eine URL an. Der Browser ermittelt die IP-Adresse des zuständigen Webservers und fordert die unter der URL hinterlegten Daten bei ihm an. Der Webserver führt eine URL-spezifische Webanwendung aus, die HTML-Code generiert, der die dem Nutzer anzuzeigende Webseite kodiert, und an den Browser zurücksendet. Der Browser interpretiert diesen Code, lädt ggf. Daten (Grafiken, JavaScripts, …) über weitere URLs nach und erzeugt die Darstellung. Dateneingabe durch den Nutzer wird traditionell durch in den HTML-Code eingebettete Formulare ermöglicht: Beim Absenden eines Formulars wird eine Ziel-URL aufgerufen und Formularinhalte dabei an die Webanwendung übermittelt und von ihr verarbeitet.

Um die zuvor beschriebenen Sicherheitsgarantien zu erreichen, nehmen wir eine wesentliche Änderung an diesem Konzept vor: Wann immer Formulareingaben an die Webanwendung übermittelt werden, überführen wir diese zuvor in eine verschlüsselte und für den Webserver nicht entschlüsselbare Repräsentation. Der Webserver liefert den HTML-Code zur Darstellung der Seite wie gehabt aus; eingebundene, von Nutzern eingegebene Daten liegen hier aber ebenfalls nur verschlüsselt vor. Eine in den ausgelieferten HTML-Code eingebundene JavaScript-Bibliothek, die bei jedem Seitenaufruf geladen und im Browser ausgeführt wird, bereitet die Daten schließlich auf, indem sie sie (vom Nutzer unbemerkt) vor ihrer Darstellung entschlüsselt.

Da die Ver-/Entschlüsselung durch Code durchgeführt wird, der von der Webanwendung an den Browser ausgeliefert wird, ist ein Betrieb durch den eingeschränkt vertrauenswürdigen primären Infrastrukturbetreiber (siehe Abschnitt 3.2.) essentiell. Um Manipulation auch auf dem Übertragungsweg auszuschließen, erfolgt sämtliche Kommunikation zwischen Webanwendung und Browser ausschließlich authentifiziert über TLS.

Über das grundlegende Konzept hinaus müssen jedoch zahlreiche Detailfragen geklärt werden, um eine vollständig Ende-zu-Ende-verschlüsselte und funktionale Kommunikationsplattform zu entwickeln: Um Daten ver-/entschlüsseln zu können, benötigen Nutzer Zugriff auf passende Schlüssel, die nicht vom Infrastrukturbetreiber bereitgestellt werden dürfen. Sollen Objekte (Akten, Nachrichten) von mehreren Nutzern zugreifbar sein, muss ein Schlüsselaustausch stattfinden. Zudem werden Klartextdaten innerhalb einer Webanwendung typischerweise nicht nur gespeichert und dargestellt, sondern auch verarbeitet. Serverseitige Verarbeitung wird durch sichere Verschlüsselung erschwert/verhindert, weshalb zusätzliche Maßnahmen erforderlich sind. Diese Kernherausforderungen werden auf den folgenden Seiten beleuchtet und Lösungen vorgestellt.

4.1. Nutzer-Login

Möchte ein Nutzer auf eine Webplattform zugreifen, muss er sich authentifizieren. In gängigen Systemen erfolgt dies mit Benutzername und Passwort, die an die Webanwendung übertragen werden. Diese verwendet den Namen, um den Nutzer in ihrer Datenbank zu finden, und gleicht anschließend eingegebenes und gespeichertes Passwort ab. Im Erfolgsfall wird ein Sitzungsschlüssel übermittelt, der die Plattformnutzung ohne erneute Passworteingabe ermöglicht. Passwörter werden dabei im Klartext übertragen; Klartext-Speicherung ist aber nicht nötig. Stattdessen wird i. d. R. ein über ein Passwortableitungsverfahren hergeleiteter Hashwert gespeichert, etwa via Password-Based Key Derivation Function 2 (PBKDF2) [Kaliski 2000]:

PasswortHash = PBKDF2(Passwort, Salt, Iterationsanzahl)

Das *Passwort* wird mit einem zufälligen Wert (*Salt*) verkettet und darüber ein kryptographischer Hashwert berechnet, der eindeutig für die Eingabedaten ist, aus dem sich das Passwort aber nicht effizient rekonstruieren lässt. Um die Berechnung (und damit auch Brute-force-Angriffe, bei denen Hashwerte für eine Vielzahl möglicher Passwörter berechnet werden) zu verlangsamen, wird die Funktion mehrfach (*Iterationsanzahl*) auf den entstehenden Wert angewendet. Salt und Iterationsanzahl werden im Klartext gespeichert, um eine Überprüfung des Passworts beim Login zu ermöglichen. So kann ein Angreifer nach Datendiebstahl zwar Benutzernamen sehen und auf bekannte Passwörter testen, nicht aber beliebige Passwörter zurückrechnen.

Soll der Nutzer anhand seiner Zugangsdaten Zugriff auf vertrauliche Daten erhalten, ist dieser Ansatz nicht ausreichend: Zum einen könnte der Server übermittelte Zugangsdaten speichern, um selbst auf vertrauliche Daten zuzugreifen; zum anderen könnte bereits die Klartext-Übertragung und -Speicherung des Benutzernamens die Vertraulichkeit verletzen – etwa, wenn als Benutzername eine E-Mail-Adresse verwendet wird.

Um dieses Problem zu umgehen, verwenden wir die Funktion PBKDF2 wie in Abb. 1 dargestellt zusätzlich bereits im Browser des Nutzers beim Ausfüllen des Login-Formulars, um aus den Zugangsdaten zwei pseudonyme Werte abzuleiten, die an ihrer Stelle übermittelt werden. So wird nicht nur sichergestellt, dass Angreifer in der Datenbank gespeicherte Passwörter nicht zurückrechnen können – dies gilt sowohl für das vom Nutzer eingegebene als auch das pseudonyme Passwort –; auch die Prüfung einzelner Benutzernamen auf Existenz ist einem Angreifer nicht möglich, solange er nicht auch Kenntnis des zugehörigen Passworts hat.

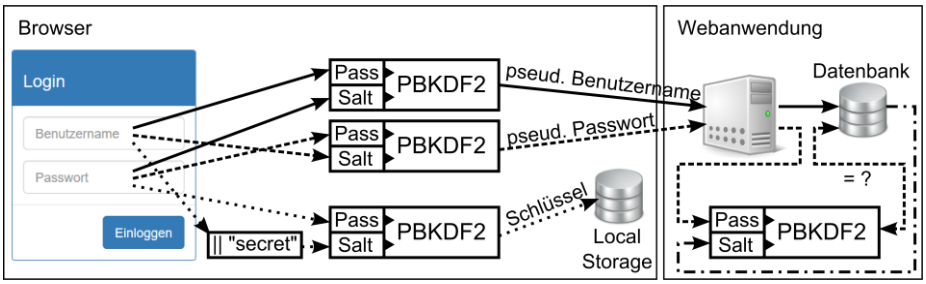

Abbildung 1: Verwendung der Zugangsdaten zur pseudonymen Authentifizierung und Schlüsselherleitung

Ein dritter aus den Zugangsdaten hergeleiteter Wert – der *Zugriffsschlüssel* – schließlich wird niemals übertragen, sondern verbleibt im Browser des Nutzers. Er dient als Anker, um vor der Webanwendung geheim zu haltende Informationen zu verschlüsseln. Um Ver-/Entschlüsselung während der Sitzung ohne erneute Passworteingabe zu ermöglichen, wird er im *Local Storage* des Browsers zwischengespeichert.

4.2. Vertrauliche Formulare

Ein Formular, über das ein Nutzer Daten eingeben kann, wird vom Server als Teil des HTML-Codes ausgeliefert und enthält Eingabefelder, deren Werte nach Bestätigung durch den Nutzer an den Server übertragen werden. Formulare, die der Erfassung eines neuen Datensatzes dienen, enthalten dabei i. d. R. leere Eingabefelder; zur Bearbeitung bereits gespeicherter Datensätze oder für Korrekturen unvollständiger/inkorrekter Eingaben kann auch ein vorausgefülltes Formular vom Server bereitgestellt werden. Um *Ende-zu-Ende-Verschlüsselung* in diese Abläufe zu integrieren, ordnen wir jedem Formularfeld einen kryptographischen Schlüssel zu; mehrere Felder können den gleichen Schlüssel haben. Bei Auslieferung des Formulars werden die Schlüssel genutzt, um vorausgefüllte Feldinhalte lokal – mithilfe der JavaScript-Bibliothek – im Browser zu entschlüsseln; analog werden vor Absenden durch den Nutzer sämtliche Feldinhalte verschlüsselt.

Legt die Kanzlei ein Objekt (z. B. eine Akte) an, so wird innerhalb des Formulars zunächst ein zufälliger Schlüssel generiert, der zur Verschlüsselung aller Eingabefelder verwendet wird. Um später Zugriff auf diesen Schlüssel zu ermöglichen, wird er mit einem *Zugriffsschlüssel* verschlüsselt in einem versteckten Formularfeld abgelegt und nach Absenden zusammen mit den anderen verschlüsselten Feldinhalten vom Server gespeichert. Bei vorausgefüllten Formularen wird dieser verschlüsselte Schlüssel – wieder als Wert eines unsichtbaren Formularfelds – vom Server übermittelt, um eine Entschlüsselung der restlichen vorausgefüll-

ten Formularfelder sowie eine spätere Neuverschlüsselung geänderter Formularfelder (bei erneutem Absenden) zu ermöglichen. Die Nutzung des Formulars unterscheidet sich dabei aus Nutzerperspektive nicht von herkömmlichen, unverschlüsselten Formularen – die Schlüsselgenerierung sowie Ver-/Entschlüsselung erfolgen im Hintergrund. Detailanpassungen wie eine technische Entkopplung von Eingabefeldern und Formularen stellen darüber hinaus sicher, dass auch bei falsch konfigurierten Browsern oder deaktiviertem JavaScript nicht versehentlich unverschlüsselte Informationen an den Server übermittelt werden können.

4.3. Schlüsselmanagement

Wie beschrieben ordnen wir jedem Objekt einen zufällig generierten Schlüssel zu, den wir zur Verschlüsselung seiner Metadaten verwenden, und machen ihn zugreifbar, indem wir ihn mit einem Nutzern bekannten Zugriffsschlüssel verschlüsselt ablegen. Der aus den Zugangsdaten eines Nutzers abgeleitete Zugriffsschlüssel (vgl. Abschnitt 4.1.) wird dabei nicht direkt genutzt, sondern wir verwenden eine Hierarchie aus sog. *Cryptographic Links* [Grolimund et al. 2006], die in Abb. 2 dargestellt ist. Um eine einfache Passwortänderung zu gewährleisten, verschlüsselt der aus den Zugangsdaten eines Nutzers abgeleitete Zugriffsschlüssel nur einen langlebigen Nutzerschlüssel, der im Browser der Kanzlei generiert wird.[2] Dieser Schlüssel dient als Zugriffsschlüssel für andere Schlüssel wie Aktenschlüssel, die ihrerseits Zugriff auf Schlüssel (und damit auch Metadaten) ihrer enthaltenen Dokumente ermöglichen. Kanzleinutzer erhalten Zugriff auf die Nutzerschlüssel ihrer Mandanten, um auf deren Metadaten (Namen, E-Mail-Adressen) zugreifen zu können.

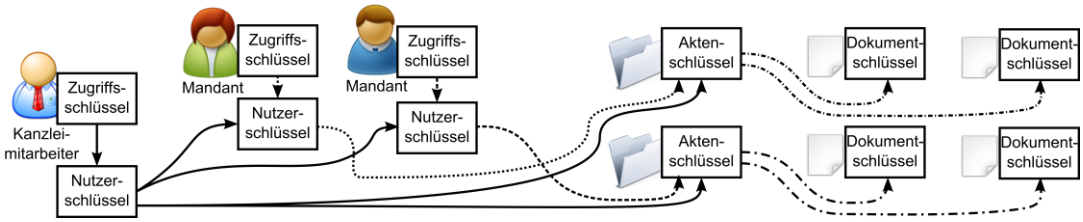

Abbildung 2: Hierarchie aus Zugriffsschlüsseln (aus Zugangsdaten abgeleitet) und zufällig erzeugten Schlüsseln

4.4. E-Mail-Benachrichtigungen

Da Namen und E-Mail-Adressen von Kanzleinutzern nicht als vertraulich eingestuft werden, können allgemeine E-Mail-Benachrichtigungen – etwa der Hinweis auf ein neues Dokument – an Kanzleinutzer wie in gängigen Webanwendungen realisiert werden, indem zu versendende Mails vom primären an den sekundären Infrastrukturbetreiber (bzw. an einen Mailserver) kommuniziert werden. Für Benachrichtigungen, die vertrauliche Informationen – etwa E-Mail-Adressen zu informierender Mandanten – enthalten, gilt dies indes nicht. Um diese im Rahmen unseres Sicherheitsmodells zu ermöglichen, ist eine direkte Kommunikation zwischen zugriffsberechtigtem Kanzleinutzer und sekundärem Infrastrukturbetreiber nötig. Zu versendende Benachrichtigungen werden dazu vom Server zum Browser eines eingeloggten Kanzleinutzers[3] kommuniziert, der die E-Mail – analog zu anderen Datensätzen – lokal erzeugt und mit einem zufällig generierten Schlüssel verschlüsselt zurücksendet, um sie zu speichern. Zwecks Versand wird der E-Mail-Schlüssel an den sekundären Infrastrukturbetreiber übermittelt, der diesen ausschließlich temporär im Arbeitsspeicher vorhält. Periodisch fordert er geplante E-Mails vom primären Infrastrukturbetreiber an, um sie zu entschlüsseln und zu versenden. Sind primärer / sekundärer Infrastrukturbetreiber physisch getrennt, ist eine Einsichtnahme des Webservers in E-Mail-Inhalte ausgeschlossen; werden Webserver und E-Mail-Dienst auf gemeinsamer Hardware betrieben, bleibt sichergestellt, dass vertrauliche Inhalte nicht persistent gespeichert werden.

---

[2] Nach Einrichtung der Plattform durch einen externen Dienstleister erhält die Kanzlei von ihm vorläufige Zugangsdaten. Die Generierung ihres Nutzerschlüssels erfolgt im Rahmen einer obligatorischen Passwortänderung bei Erstanmeldung. Bei Mandanten erfolgt die Schlüsselgenerierung während der Anlegung durch den Kanzleimitarbeiter.

[3] Dies führt zu einer Einschränkung, die für Anwaltskanzleien akzeptabel scheint: Hat eine Akte mehrere Mandanten, kann nach Dokumentupload durch einen Mandanten nur die Kanzlei sofort benachrichtigt werden. Andere Mandanten erhalten die Benachrichtigung, sobald die Kanzlei reagiert oder sich aus anderen Gründen eingeloggt hat.

5. Software

Wir haben die beschriebenen Konzepte zum Umgang mit Ende-zu-Ende-verschlüsselten Daten in Webanwendungen als Erweiterung für das verbreitete Framework *Django* [Django Software Foundation 2016] implementiert und damit eine Kommunikationsplattform entwickelt. Kryptographische Operationen werden basierend auf der JavaScript-Bibliothek *forge* [Digital Bazaar, Inc. 2017], die nötige Algorithmen und einen kryptographisch sicheren Zufallsgenerator auch auf älteren Browsern nachrüstet, transparent durchgeführt. Beim ersten Einloggen wird die Kanzlei aufgefordert, Zugangsdaten zu wählen; unbemerkt wird dabei ihr Nutzerschlüssel generiert. Im Anschluss kann sie Akten und Mandanten verwalten. Das Anlegen von Akte und Mandant ist in Abb. 3 zu sehen: Die Kanzlei gibt die Daten in ein Formular ein; nach Absenden werden ihr zufällig generierte Zugangsdaten angezeigt, die sie per Briefpost an den Mandanten senden kann.

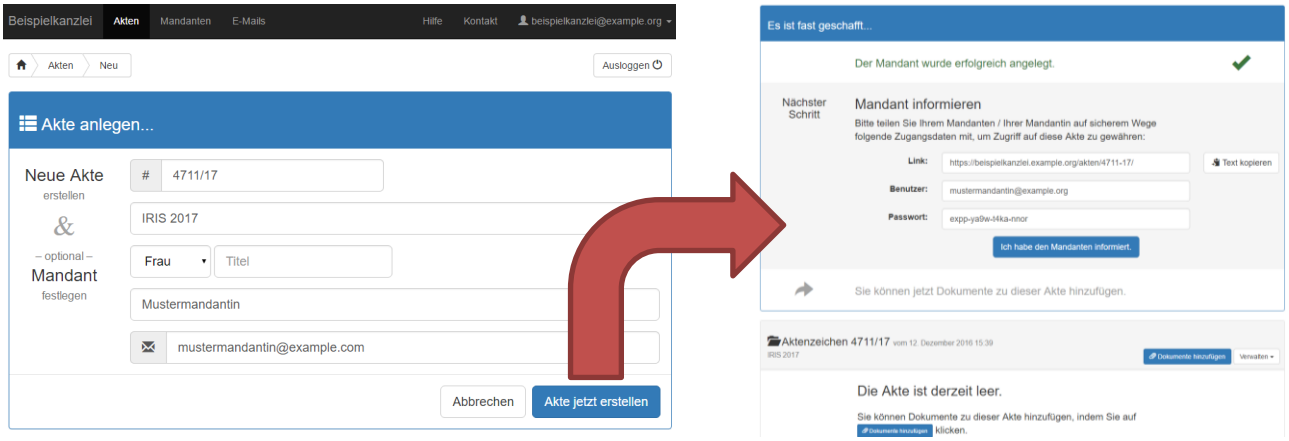

Abbildung 3: Anlegen von Akten und Mandanten (links) und automatisch generierte Zugangsdaten (rechts)

Um Dokumente auszutauschen, müssen Kanzlei bzw. Mandant lediglich die Akte aufrufen und Dateien in ihr Browserfenster ziehen. Die Dokumente werden daraufhin verschlüsselt und hochgeladen (Abb. 4 oben); beim Herunterladen erfolgt die Entschlüsselung im Browser. Falls gewünscht, werden nach Hochladen E-Mail-Benachrichtigungen für zugriffsberechtigte Nutzer generiert (Abb. 4 unten). Der Versand erfolgt dabei verzögert: Benachrichtigungen über mehrere Dokumente werden zusammengefasst; beim Löschen von Dokumenten werden geplante Benachrichtigungen storniert. Primärer und sekundärer Infrastrukturbetreiber fallen in der hier vorgestellten Implementierung zusammen.

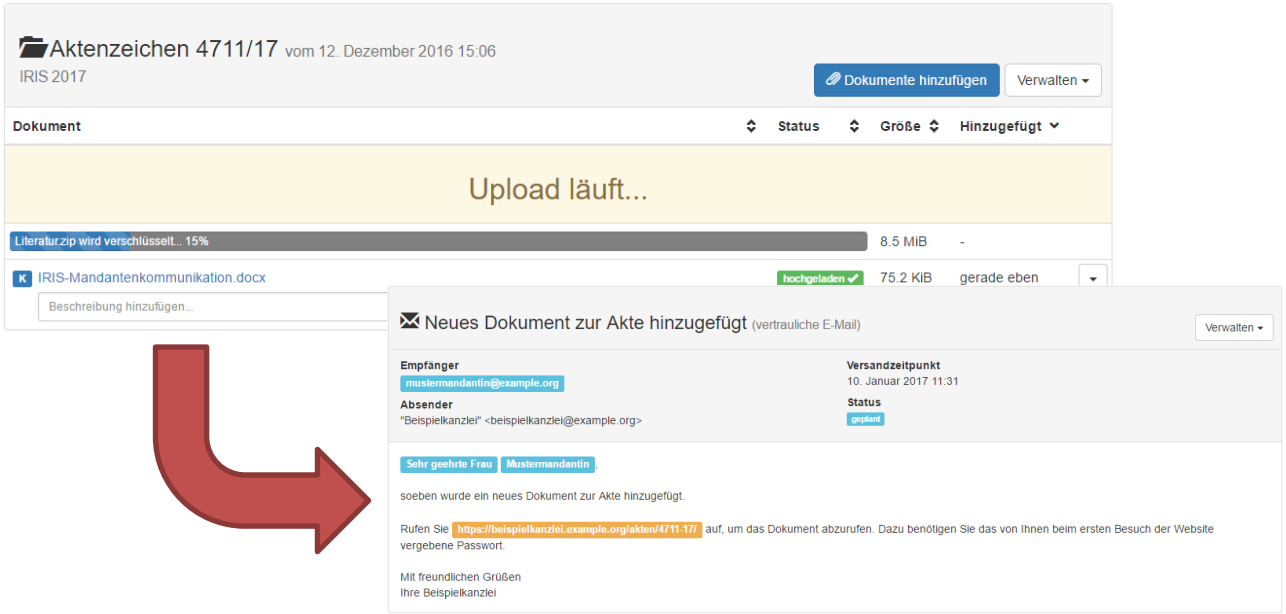

Abbildung 4: Laufender Dokumentupload (oben) und automatisch generierte Benachrichtigungs-E-Mail (unten)

6. Diskussion

Die vorgestellte Lösung ermöglicht Kanzleien, einfach und sicher mit ihren Mandanten zu kommunizieren: Über eine Webplattform werden Dokumente/Nachrichten verschlüsselt so ausgetauscht, dass Inhalte und Kommunikationsbeziehungen vertraulich bleiben; optionale Benachrichtigungen senken das Sicherheitsniveau auf das verschlüsselter E-Mails zugunsten effizienterer Abläufe. Die Lösung funktioniert ohne Softwareinstallation in allen gängigen Browsern. Insbesondere können so jene Mandanten sicher erreicht werden, mit denen sonst nur über unsichere Kanäle kommuniziert werden kann. Die Lösung erfüllt das Sicherheitsmodell aus Abschnitt 3.2. und löst damit die Problematik der Auftragsdatenverarbeitung; die Beschränkung auf Browserkryptographie bedeutet aber – jedenfalls derzeit – auch Einschränkungen für die Praxis.

Dies betrifft einerseits *Skalierbarkeit*: Für größere Datenmengen sind Funktionen zum Sortieren / Filtern / Suchen essentiell. In gängigen Plattformen wird dies serverseitig realisiert; bei Einsatz von Ende-zu-Ende-Verschlüsselung müssen jedoch ungefilterte Daten an den Browser übermittelt werden. Zwar genügt eine einmalige Übertragung, wenn Daten im *Local Storage* zwischengespeichert werden; viele Browser haben jedoch ein Limit von 5 MB, was ausreichend für ca. 5000 Akten/Mandanten ist. Verfahren aus dem aktiven Forschungsfeld *Searchable Encryption* [Hahn 2014] könnten ermöglichen, diese Beschränkung zu umgehen.

Andererseits ist auch das erzielbare *Sicherheitsniveau* begrenzt: Der Programmcode für die Verschlüsselung wird bei jedem Seitenaufruf vom Server abgerufen. Wird er manipuliert, geht die Garantie der Vertraulichkeit verloren. Durch Einsatz von TLS für sämtliche Kommunikation sowie Verzicht auf sonst übliche Einbindung von Code aus externen Quellen kann Manipulation durch Dritte verhindert und einiger gängiger, auf früherem Stand der Technik basierender Kritik an Browserkryptographie (vgl. [Ptacek 2011]) begegnet werden. Der ordnungsgemäße Betrieb der Infrastruktur ist für praktische Sicherheit jedoch essentiell. Dies gilt zwar auch bei installierbarer Software, etwa bzgl. der Update-Infrastruktur; aufgrund höherer Lebensdauer einzelner Versionen im Gegensatz zu Seitenaufrufen scheint unbemerkte Manipulation dort jedoch unwahrscheinlicher. Als Gegenmaßnahme könnte das vorgeschlagene System um installierbare Software (bspw. Erweiterungen für E-Mail- und Kanzleisoftware) erweitert werden. Kanzleimitarbeiter und sicherheitsbewusste Mandanten könnten so auch vor aktiven Angriffen geschützt werden; andere Mandanten würden mit den beschriebenen Einschränkungen weiterhin sicher erreicht. Die Software könnte auch zusätzliche Sicherheitsmaßnahmen umsetzen, die Klartextzugriff erfordern – etwa eine Anbindung an Virenschutzlösungen.

7. Literatur

BORGES, GEORG, Kapitel 3. Datenschutzrechtliche Aspekte des Cloud Computing § 6 (Einführung). In: Borges, Georg/Meents, Jan Geert (Hrsg.): Cloud Computing, C.H. Beck, München 2016.

DIERKS, TIM/RESCORLA, ERIC, The Transport Layer Security (TLS) Protocol. RFC 5246, 2008.

DIGITAL BAZAAR, INC., GitHub - digitalbazaar/forge: A native implementation of TLS in Javascript and tools to write crypto-based and network-heavy webapps, https://github.com/digitalbazaar/forge (aufgerufen am 10.01.2017).

DJANGO SOFTWARE FOUNDATION, Django, https://www.djangoproject.com/ (aufgerufen am 14.12.2016).

ELKINS, MICHAEL/DEL TORTO, DAVE/LEVIEN, RAPH/ROESSLER, THOMAS, MIME Security with OpenPGP. RFC 3156, 2001.

GROLIMUND, DOMINIK/MEISSER, LUZIUS/SCHMID, STEFAN/WATTENHOFER, ROGER, Cryptree: A Folder Tree Structure for Cryptographic File Systems. In: Proceedings of SRDS'06, 2006, S. 189-198.

HAHN, FLORIAN/KERSCHBAUM, FLORIAN, Searchable Encryption with Secure and Efficient Updates. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14), 2014, S. 310-320.

KALISKI, BURT, PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898, 2000.

PTACEK, THOMAS, Javascript Cryptography Considered Harmful, https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2011/august/javascript-cryptography-considered-harmful/ (aufgerufen am 10.01.2017), 2011.

RAMSDELL, BLAKE/TURNER, SEAN, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. RFC 5751, 2010.

SORGE, CHRISTOPH, Sicherheit der Kommunikation zwischen Rechtsanwalt und Mandant. In: NJW (Sonderbeilage: Das besondere elektronische Anwaltspostfach und elektronischer Rechtsverkehr), Nr. 38, 2016, S. 20-22.