Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

CA	Certificate Authority
DANE	DNS-based Authentication of Named Entities
DHE	Diffie-Hellmann-Schlüsselaustausch
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
ECDH	Elliptic Curve Diffie-Hellman
EmiG	E-Mail made in Germany
HTTP	Hypertext Transfer Protokoll
HTTPS	Hypertext Transfer Protokoll Secure
ICANN	Internet Corporation for Assigned Names and Numbers
IP	Internet Protocol
IPsec	Internet Protokoll Security
KSK	Key Sgning Key
MITM	Man in the Middle
MAC	Message Authentication Code
OSI	Open System Interconnection
PGP	Pretty Good Privacy
PFS	Perfect Foreward Secrecy
PKI	Public Key Infrastructure
RSA	RSA(Rivest, Shamir und Adleman)-Kryptosystem
SMTP	Simple Mail Transfer Protokoll
SSH	Secure Shell
SSL	Secure Socket Layer
S/MIME	Secure/Multipurpose Internet Mail Extensions
TCP	Transmission Control Protocol
TLD	Top-Level-Domain
TLS	Transport Layer Security
TLSA	Transport Layer Security Authentication
TLS/SSL	Transport Layer Security/Secure Socket Layer
TCR	Trusted Community Representatives
URL	Unified Ressource Locator
ZSK	Zone Signing Key

1 Sicherheitsniveaus

Das nachfolgende Kapitel geht auf verschiedene Sicherheitsbedürfnisse eines Nutzers ein. Gerade im Hinblick auf den Grad der Verschlüsselung bei der E-Mail Kommunikation ist es wichtig sich darüber bewusst zu werden, wie sensibel die Information ist, die man versenden möchte. Denn jede Verschlüsselung ist mit einem bestimmten Aufwand verbunden und folglich ist abzuwägen, welcher Verschlüsselungsaufwand dem Nutzer die zu versendende Information wert ist. Beispielsweise ist aus Sicht der Autoren der potentielle Schaden gering, wenn eine E-Card zu den Weihnachtsfeiertagen an den nicht rechtmäßigen Empfänger gerät, sodass für die Versendung einer solchen Information der Grad der Verschlüsselung niedrig und somit der Aufwand niedrig ausfällt. Dahingegen ist der potentielle Schaden größer, wenn es sich bei der versendeten Information um beispielsweise die eigenen Kontodaten handelt, was wiederum bedeutet, dass der Nutzer bereit ist einen höheren Aufwand zu betreiben, um diese Inhalte auf eine sichere Art und Weise via E-Mail zu übermitteln.
Das Sicherheitsbedürfnis einer jeden Person kann unterschiedlich stark ausgeprägt sein. Daher ist es den Autoren nicht möglich, eine allgemein gültige Auflistung aller möglichen Szenarien der E-Mail Kommunikation bereitzustellen, aus welcher die Teilnehmer der Zielgruppe lediglich das richtige Szenario heraussuchen müssen und dadurch den optimalen Grad der Verschlüsselung erhalten. Stattdessen wird in Abbildung 1.1 eine Übersicht präsentiert, die es dem Nutzer erlaubt auf Basis seines eigenen Sicherheitsbedürfnisses und mit Hilfe festgelegter Kriterien für die Übermittlung einer ganz bestimmten Information ein geeignetes Sicherheitsniveau zu ermitteln. In den nachfolgenden dieser Arbeit werden verschiedene Möglichkeiten der Verschlüsselung von E-Mail Kommunikation vorgestellt und jeweils passenden Sicherheitsniveaus zugeordnet. Dabei erfolgt diese Zuordnung mit dem Ziel, einen optimalen Ausgleich zwischen Notwendigkeit und Aufwand der Verschlüsselung von E-Mails zu erhalten, sodass der Nutzer nach der Ermittlung eines geeigneten Sicherheitsniveaus eine aus Sicht der Autoren geeignete Verschlüsselungsmethode ermitteln kann.

Merkmale	Streng Vertraulich	Vertraulich	Privat	Öffentlich
Wert der Information und Schutzwürdigkeit	Äußerst hoch	Hoch	Gering	Von keinem besonderem Wert und kein Schutz notwendig
Potentieller Schaden	Äußerst hoch	Hoch	Gering	Nicht vorhanden
Personen-bezogene Daten	Ja	Ja	Teilweise	Nein
Einfluss auf Privatsphäre	Äußerst hoch	Hoch	Gering	Nicht vorhanden
Autorisierter Personenkreis	Sender + Empfänger; evtl. zusätzlich engste Verwandte	Verwandtschaft, direkte Kollegen	Freunde + Bekannte	Jeder
Auswirkung bei Integritäts-verletzung	Äußerst hoch	Hoch	Gering	Nicht vorhanden
Analogie: Übermittlung auf Postweg	Einschreiben oder Verzicht auf Postweg und persönliche Übergabe	Doppelte Kuvertierung oder Einschreiben	Standard Brief	Postkarte
Häufigkeit des Niveaus	Sehr selten	Selten	Häufig	Oft

Abbildung 1.1: Sicherheitsniveaus

Die Abbildung 1.1 stellt im Tabellenkopf vier verschiedene Sicherheitsniveaus dar: *Streng Vertraulich, Vertraulich, Privat und Öffentlich*. Dabei nimmt das Sicherheitsbedürfnis sowie der Verschlüsselungsaufwand von Streng Vertraulich hin zu Öffentlich ab. In der ersten Spalte sind verschiedene Merkmale aufgelistet, welche es dem Nutzer ermöglichen sollen, für eine ganz bestimmte Information ein geeignetes Sicherheitsniveau zu bestimmen. Alle weiteren Felder enthalten die Ausprägung des Merkmals innerhalb des jeweiligen Sicherheitsniveaus.
Der Wert der Information und deren Schutzwürdigkeit beschreiben, wie wertvoll die zu versendende E-Mail für einen nicht rechtmäßigen Empfänger ist und welche Notwendigkeit des Schutzes daraus folgt. Der potentielle Schaden stellt das Ausmaß dar, welches eintritt für den Fall, dass die E-Mail durch einen unberechtigten Dritte gelesen wird. Dieser Schaden kann verschiedener Art sein. Zum Beispiel können daraus rechtliche Konsequenzen erfolgen, es kann zu einem finanziellen Verlust führen oder mit einer Schädigung des Images des Senders einhergehen [Vgl.][]Reinhausen GmbH, S. 6. Aus dem potentiellen Schaden lässt sich außerdem gut der Einfluss auf die eigene Privatsphäre ableiten. Das Merkmal *Personenbezogene Daten* besagt, dass die zu versendende Information personenbezogene Daten enthält und daher grundsätzlich das Sicherheitsniveau *Vertraulich* zu wählen ist[Vgl.][]TSE. In Abhängigkeit von der Ausprägung der anderen Merkmale, kann als resultierendes Sicherheitsniveau auch *Streng Vertraulich* oder *Privat* ermittelt werden. Der Autorisierte Personenkreis ist eine weitere Eigenschaft anhand derer der Nutzer ein passendes Sicherheitsniveau bestimmen kann. Grundsätzlich gilt: je wertvoller die Information, desto geringer ist der Personenkreis, der Einblick in die zu versendende E-Mail erhalten darf [Vgl.][]TSE. Daraus folgt, dass eine streng vertrauliche Nachricht ausschließlich

zwischen dem Sender und dem Empfänger ausgetauscht wird und in der Regel keine weitere Person über den Inhalt erfahren darf. Eine Ausnahme an dieser Stelle sind allenfalls engste Verwandte. Dahingegen ist für eine Nachricht, deren Inhalt prinzipiell jedermann erfahren darf, das Sicherheitsniveau *Öffentlich* zu wählen[Vgl.][]Reinhausen GmbH, S. 10. Die *Auswirkung bei Integritätsverletzung* beschreibt das eingetretene Ausmaß, wenn die E-Mail in die Hände eines Angreifers gelangt ist. Eine weitere Möglichkeit ein geeignetes Sicherheitsniveau zu ermitteln ist die Überlegung, wie der Inhalt der E-Mail auf dem Postweg versandt werden würde. Für eine streng vertrauliche Information würde ein Einschreiben gewählt werden oder gänzlich auf den Postweg verzichtet und stattdessen die Nachricht persönlich überbracht werden. Dahingegen ist für Information, die ohne Bedenken auf einer Postkarte übermittelt werden können, das Sicherheitsniveau *Öffentlich* zu wählen Das letzte Merkmal beschreibt das Aufkommen der einzelnen Sicherheitsniveaus. Streng vertrauliche Informationen sind sehr selten und am häufigsten werden öffentliche Nachrichten ausgetauscht [Vgl.][]TSE.
Anhand des nachfolgenden Beispiels soll der Umgang mit der Abbildung 1.1 verdeutlicht werden. Hierbei ist zu erwähnen, dass das resultierende Sicherheitsniveau auf Basis des beschriebenen Sicherheitsbedürfnisses ermittelt wurde und für die gleiche Information bei anderen Nutzern unterschiedlich ausfallen kann:
Herr Meier war vor kurzem in einen Auffahrunfall verwickelt, den ein unachtsamer Autofahrer verursacht hatte. Daraufhin brachte Herr Meier sein Fahrzeug in die Werkstatt und ließ ein Gutachten des Schadens erstellen. Dieses Gutachten möchte er nun zusammen mit seinen Kontodaten via E-Mail an die Versicherung des Unfallverursachers senden. Der Wert dieser Information ist hoch, denn einerseits sind die Kontodaten in der Nachricht vorhanden. Andererseits ist in dem Gutachten Herr Meiers Anschrift angegeben und es lässt sich aus den Fotos und der Schadenshöhe des Gutachtens ableiten, dass Herr Meier einen Luxuswagen besitzt. Daraus wiederum lassen Rückschlüsse auf seine finanzielle Situation schließen. Der potentielle Schaden, der sich daraus ergibt, ist hoch bis äußerst hoch. Denn bei ausreichender krimineller Energie können nicht nur die Kontodaten missbraucht werden, sondern mit Hilfe der Anschrift kann der Wohnsitz von Herrn Meier ausgekundschaftet und beispielsweise bei seiner Abwesenheit in sein Anwesen eingebrochen werden. Die Anschrift stellt personenbezogene Daten dar und ermöglicht einen hohen Einfluss auf die Privatsphäre von Herrn Meier. Der Autorisierte Personenkreis für diese Nachricht beschränkt sich auf den Sender (Herr Meier), den Empfänger (die Versicherung) sowie auf seine Frau und auf die Werkstatt, die das Gutachten erstellt hat. Seinen Freunden und Kollegen hat Herr Meier zwar auch von dem Unfall erzählt. Allerdings wissen diese keine genaueren Details hinsichtlich des Schadens und dessen Höhe. Hätte die Versicherung den E-Mail Service nicht angegeben, so würde Herr Meier die Unterlagen des Gutachtens mit einem Standard-Brief versenden. Die Kontodaten würde er mittels Einschreiben verschicken.
Somit kann als Resultat für dieses Beispiel unter dem beschriebenen Sicherheitsbedürfnis ein Sicherheitsniveau von *Vertraulich* ermittelt werden. Welches Verschlüsselungsverfahren konkret für dieses Sicherheitsniveau anzuwenden ist, wird in den nachfolgenden Kapiteln beschrieben.

2 Potenzielle Gefahren und Schwachstellen der E-Mail-Kommunikation

2.1 Informationsgewinnung durch Provider

warum ist E-Mail oft Kostenlos? Welchen Sinn hat das für die Provider? Welche Bots lassen die Provider über unsere Kommunikation laufen? Wie mächtig sind Meta-Daten?
Überleitung... Neben den weniger gerichteten (Kein Angriff?) Maßnahmen der Informationsgewinnung aus E-Mail-Kommunikation gibt es auch klassische Angriffsmethoden Dritter.

2.2 Angriffsarten

2.2.1 MITM - Man in the Middle

2.2.2 Umleitungsangriff

2.3 Schwachstellen

2.3.1 Zertifikatsaussteller

AUTHENTIZITÄT 200 Aussteller (z.T. keine eigenen Private Keys der User) CAs können nachlässig werden. Angreifern können somit gültiges Zertifikat für einen Host erstellen dessen Besucher das Ziel sind.

2.3.2 Zertifikatsprüfung

AUTHENTIZITÄT Ideal wäre: Alle Serverzertifikate der gesamten Komunikationskette zu prüfen. In der Praxis jedoch kein sog. İdentifiziertes TLS"(Kommunikationspartner eideutig festgestellt) In der Praxis arbeiten Server jedoch nach Öpportunistischem TLS", dass bedeutet das lediglich wichtig ist, dass die Nachricht verschlüsselt übertragen wird, egal von wem.

3 Kryptographie

Die Kryptographie ist eine Wissenschaftslehre, die sich mit den Verfahren der Ver- und Entschlüsselung von Informationen sowie deren Anwendung befasst. Meistens werden dazu geheime Schlüssel oder Schlüsselpaare verwendet. Damit lassen sich mithilfe mathematischen Berechnungsverfahren, sogenannten Algorithmen, Nachrichten verschlüsseln. Heutige Verschlüsselungsverfahren basieren entweder auf einem symmetrischen oder asymmetrischen Algorithmus.

3.1 Grundlagen

3.1.1 Symmetrische Verschlüsselung

Bei der symmetrischen Verschlüsselung wird sowohl zur Ver- und Entschlüsselung derselbe Schlüssel verwendet, der auch Shared Secret genannt wird. Mithilfe dieses Schlüssels und einem symmetrischen Verschlüsselungsalgorithmus wird die Nachricht des Absenders verschlüsselt. In diesem Zusammenhang wird der "lesbare Text einer Nachricht [...] Klartext [..] genannt"[S. 21]Ertel2012. Aus dem Shared Secret und dem Klartext wird durch eine mathematische Vorschrift ein Geheimtext erzeugt. Dieser verschlüsselte Text kann ausschließlich mit dem gleichen Schlüssel entschlüsselt werden, mit dem er verschlüsselt wurde. Diese Tatsache wirkt sich im elektronischen Briefverkehr nachteilig auf die Anwendung der symmetrischen Verschlüsselung aus. Da beide Kommunikationspartner denselben Schlüssel benötigen, muss dieser zuvor ausgehandelt und übertragen werden. Die Übertragung dieses Schlüssels stellt ein Sicherheitsrisiko dar. Wird die Übertragung der Schlüssels abgehört, können Dritte mit seiner Hilfe die verschlüsselten Nachrichten mitlesen.

3.1.2 Asymmetrische Verschlüsselung

Wie bei der symmetrischen Verschlüsselung kommt auch bei der asymmetrischen Verschlüsselung, die auch Public-Key-Kryptography genannt wird, ein kryptographischer Algorithmus zum Einsatz. Hierbei wird jedoch wird statt eines gemeinsamen Schlüssels ein Schlüsselpaar verwendet. Dieser besteht aus einem öffentlichen und einem privaten Schlüssel, die mathematisch zusammenhängen. Jeder, der verschlüsselte Nachrichten empfangen möchte, verfügt über ein solches Schlüsselpaar. Der private Schlüssel wird niemals bekanntgegeben, wohingegen der öffentliche Schlüssel jedem zugänglich gemacht werden kann. Obwohl die Schlüsseln zusammenhängen, kann aus der Kenntnis des öffentlichen Schlüssels nicht auf den privaten Schlüssel geschlossen werden [S. 177]Schmeh2013. Soll eine Nachricht durch ein asymmetrisches Verschlüsselungsverfahren verschlüsselt verschickt werden, wird zunächst der öffentliche Schlüssel des Empfängers benötigt. Zusammen mit der Nachricht wird der Geheimtext erzeugt und an den Empfänger geschickt. Zum Entschlüsseln der Nachricht wird der private Schlüssel des Empfängers verwendet, der zum bei der Verschlüsselung genutztem öffentlichen Schlüssel gehört.

Mit diesem Verfahren wurde das Sicherheitsrisiko der symmetrischen Verschlüsselung gelöst, da der öffentliche Schlüssel zum Verschlüsseln jedem bekannt sein darf. Zur Entschlüsselung wird der dazugehörige private Schlüssel benötigt, der im Besitz des Empfängers ist und niemals veröffentlicht wird. Ein Sicherheitsrisiko ergibt sich jedoch aus der Tatsache, dass ein Dritter die Übertragung des öffentlichen Schlüssels abfangen kann und dem Absender stattdessen seinen öffentlichen Schlüssel überträgt. Damit ist er in der Lage, alle vom Absender verschlüsselten Nachrichten ohne dessen Kenntnis zu entschlüsseln. Daher muss "bei der Verwendung eines fremden Schlüssels [..] möglichst immer die Authentizität des Schlüssels geprüft bzw. sichergestellt werden."[S. 90]Ertel2012

3.1.3 Digitale Signaturen

Asymmetrische Verschlüsselungsverfahren ermöglichen, die menschliche Unterschrift in der digitalen Welt abzubilden. Diese Funktionalität wird mit digitalen Signaturen umgesetzt. Damit eine digitale Signatur den Anforderungen einer menschlichen Unterschrift erfüllt, müssen einige Bedingungen eingehalten werden: [S. 202]Schmeh2013

- Sie darf nicht zu fälschen sein.
- Ihre Echtzeit muss überprüfbar sein
- Sie darf nicht unbemerkt von einem Dokument zum anderen übertragen werden können.
- Das dazugehörende Dokument darf nicht unbemerkt verändert werden können.

Diese Voraussetzungen dienen dazu, die Verbindlichkeit des Absenders sowie die Integrität der Nachricht zu gewährleisten. Beim Signieren verschlüsselt der Absender seine Nachricht mit seinem privaten Schlüssel. Die resultierende Nachricht ist die digitale Signatur. In der Regel wird beim Signieren aufgrund des Rechenaufwandes für lange Nachrichten nicht die gesamte Nachricht verschlüsselt, sondern ein sogenannter Hashwert. Hashwerte sind eine Zeichenfolge mit einer bestimmten Länge, die durch eine mathematische Einweg-Hashfunktion generiert werden. Diese Funktionen haben einen Eingabeparameter und berechnen daraus einen Hashwert. Aus der Kenntnis des Hashwertes oder der Hashfunktion lässt sich der Eingabeparameter nicht ableiten, wodurch die Integrität einer signierten Nachricht gewährleistet ist. Die signierte Nachricht wird im nächsten Schritt an den Empfänger geschickt. Dieser kann nun die Verbindlichkeit der Nachricht überprüfen, indem er die Signatur mit dem öffentlichen Schlüssel des Absenders entschlüsselt. Dazu berechnet er den Hashwert der erhaltenen Nachricht und vergleicht diesen mit dem zuvor entschlüsselten Hashwert des Absenders. Diese Überprüfung wird dabei auch Verifizierung genannt. Stimmen beide überein, kann er sicher sein, dass die Nachricht von dem Absender stammt, da nur mit dessen privatem Schlüssel die Nachricht verschlüsselt werden konnte. Zusätzlich ist damit garantiert, dass die Nachricht vollständig und ungeändert beim Absender angekommen ist.
Durch die Anwendung des asymmetrischen Verschlüsselungsverfahrens beim Signieren bleibt die Authentizität der öffentlichen Schlüssels weiterhin ein Sicherheitsrisiko. Das Risiko besteht darin, dass "man einem öffentlichen Schlüssel nicht ansieht, wem er gehört"[S. 506]Schmeh2013.

3.1.4 Zertifikate

Bei den bisher genannten Verfahren wurde davon ausgegangen, dass der öffentliche Schlüssel wirklich dem beabsichtigten Kommunikationspartner gehört. Die Authentizität des öffentlichen

Schlüssels ist durch Szenarien wie dem Man-In-The-Middle-Angriff nicht immer garantiert. Um die Authentizität des öffentlichen Schlüssels sicherzustellen, werden Zertifikate verwendet.
Ein Zertifikat ist ein elektronisches Dokument, das einer Person zugeordnet werden kann. Dieses Dokument enthält neben den persönlichen und weiteren Informationen des Inhabers dessen öffentlichen Schlüssel. Außerdem enthält ein Zertifikat eine Signatur über all den genannten Angaben. Das Signieren wird dabei von einer vertrauenswürdigen Instanz durchgeführt, die auch Certificate Authority (CA) oder Zertifizierungsstelle genannt wird.
Zum Versenden einer verschlüsselten Nachricht wird zunächst das Zertifikat vom Empfänger besorgt. Der Absender überprüft die Authentizität des öffentlichen Schlüssels des Empfängers, indem er die Signatur unter Verwendung des öffentlichen Schlüsses des CAs] verifiziert. Mit der Verifizierung ist gewährleistet, dass der öffentliche Schlüssel auf dem Zertifikat dem Zertifikatsinhaber gehört.
Das Sicherheitsrisiko bezüglich der Authentizität des öffentlichen Schlüssels der Zertifizierungsstelle wird gelöst, indem ein Zertifikat über den seinen öffentlichen Schlüssel erstellt wird, das von der CA selbst signiert wurde. Dieses Zertifikat wird als self-signed bezeichnet.

3.2 Web of Trust

Das Web of Trust ist ein Vertrauensmodell, bei dem sich die Nutzer gegenseitig vertrauen und somit ein netzartiges Modell entstehen lässt. Die Grundidee ist, dass die Nutzer dieses Modells gegenseitig ihre öffentlichen Schlüsseln signieren. Im Gegensatz zum hierarchischen Verfahren gibt es keine zentrale Zertifizierungsstelle. Die Funktionsweise des Web of Trust wird anhand eines Beispiels erläutert. [S. 120]Ertel2012: Teilnehmer C möchte Teilnehmer B eine verschlüsselte Nachricht schicken. Dazu besorgt er sich zunächst das Zertifikat von Teilnehmer B. Dieser wurde zuvor von Teilnehmer A signiert. Da Teilnehmer C Teilnehmer A vertraut, beschafft sich Teilnehmer C den öffentlichen Schlüssel von Teilnehmer A und verifiziert mit das Zertifikat von Teilnehmer B. Ist die Verifizierung erfolgreich, so kann Teilnehmer C den öffentlichen Schlüssel von Teilnehmer B vertrauen.
Bei diesem Modell wird zwischen zwei Arten des Vertrauens unterschieden. Einerseits existiert das Vertrauen in eine Person bzw. dessen Signatur. Andererseits besteht ein Vertrauen in einen signierten Schlüssel eines Dritten, der von einer vertrauensvollen Person signiert wurde. Beide Arten des Vertrauens können unabhängig voneinander existieren.

4 Schlüsselaustausch

Für die sichere Kommunikation ist ein sicherer Schlüsselaustausch die Ausgangsbasis um Angriffe zu verhindern. “Hierbei ist der geschützte Austausch geheimer Schlüssel symmetrischer Kryptosysteme von Interesse.”[S. 437]Eckert2013

4.1 Perfect Forward Secrecy - PFS

Beim klassischen Schlüsselaustausch werden die Sitzungsschlüssel durch den Public-Key innerhalb des Server-Zertifikat übertragen.[Vgl.][]Boeck2013 Dies geschieht mittels RSA(Rivest, Shamir und Adleman)-Kryptosystem (RSA)-Verfahren. Verschlüsselte Kommunikation ist jedoch nur solang die Schlüssel geheim bleiben sicher. Die Gefahr beim klassischen RSA-Public Key Infrastructure (PKI)-Verfahren ist dass vergangene Kommunikation nachträglich zu jedem Zeitpunkt entschlüsselt werden kann, sobald Angreifer in Besitz des Private-Key sind.

Sinnvoller ist es die Sitzungsschlüssel zum Einen nicht mehr zu übertragen und zum Anderen unabhängig voneinander ständig neu zu generieren und bei Terminierung zu löschen. Realisiert wird dies durch die Protokoll-Eigenschaft Forward Secrecy, die im Kryptographischen Fachjargon auch Perfect Foreward Secrecy (PFS)[Vgl.][]Boeck2013 genannt wird. Die “Anforderung an die Verfahren und Protokolle zur Schlüsselerneuerung besteht darin dafür zu sorgen, dass die Kenntnis eines Schlüssels, also dessen Aufdeckung, nicht dazu führt, dass damit auch vorherige und nachfolgende Schlüssel direkt aufdeckbar sind.”[S. 439]Eckert2013 Wie notwendig PFS geworden ist zeigen die jüngsten Ereignisse im Zusammenhang mit dem OpenSSL-Bug "Heartbleed", mit dem es sehr einfach war Private Schlüssel von Server auszulesen.[Vgl.][]Zhu2014

Das Diffie-Hellmann-Schlüsselaustausch (DHE) Verfahren ermöglicht dabei als Basis die Aushandlung eines Sitzungsschlüssels bei dem die Kommunikationspartner verschiedene Nachrichten senden und sich auf einen Sitzungsschlüssel einigen können, ohne diesen je übertragen zu haben. Dieser Schlüssel ist auch nur für die aktuelle Verbindung gültig und wird anschließend gelöscht. Der Public-Key des Servers wird weiterhin übertragen, jedoch nur um den Schlüsselaustausch zu signieren. “Abgeschlossene Sitzungen können somit im Nachhinein nicht mehr entschlüsselt werden.”[Vgl.][]Schulz2014 Die Verschlüsselungsverfahren Transport Layer Security/Secure Socket Layer (TLS/SSL) und IPsec beherrschen bereits PFS.

Aufgezeichnete verschlüsselte Daten können somit bei Besitz des privaten Schlüssels nicht entschlüsselt werden. Zudem wird einfaches Belauschen einer aktiven Verbindung deutlich erschwert, denn es müsste die gesamte Kommunikation mit einem gezieltem Man in the Middle (MITM)-Angriff2.2.1 manipuliert werden. Für diese Problematik gibt es wiederum moderne Ansätze wie DANE (Vgl. Kapitel 6.2), die in Kombination mit PFS aktuell bei der Verschlüsselung von Verbindungen höchsten Sicherheitsansprüchen entsprechen, indem zusätzlich die Authentizität der Kommunikation gewährleistet wird.

Nachteile gibt es lediglich bei der Verwendung des bereits überholten, und seit Jahren als geknackt bekannte DHE-Verfahren, denn dabei verzögert sich zusätzlich der Verbindungsaufbau.

Die Schlüssellänge ist Minimum 1024 Bit, und längere Schlüssel mit 2048 oder 4096 Bit sind dabei nicht sicherer.[Vgl.][]Boeck2013 Der moderne Nachfolger mit elliptischen Kurven Elliptic Curve Diffie-Hellman (ECDH) gilt aktuell als sicher und benötigt dabei weniger als 1024 Bit und verzögert den Verbindungsaufbau nur unweigerlich.

Obwohl es Forward Secrecy bereits seit 1999 im Transport Layer Security (TLS) Standard 1.0[Vgl.][]Boeck2013 vorgesehen ist und somit essenzieller Bestandteil von Verschlüsselung ist, hat sich PFS noch nicht als Standard durchgesetzt.SSLLabs Dies liegt zum einen an den Webservern. Mit einem Apache Webserver ist nur eine Moduluslänge von 1024 Bit vorgesehen. Beim Einsatz von DHE würden Provider damit ihre Server daher unsicher betreiben. Zum Anderen sind es auf Client-Seite die Browser die DHE bzw. ECDH lange Zeit ignoriert haben. Der Internet Explorer verschlüsselte nur nach DSS, wobei der de-facto Standard für Verschlüsselung bereits RSA war. Opera unterstützte lediglich das überholte DHE-Verfahren und Safari priorisiert Forward Secrecy niedrig und bevorzugt bei gegebener Option sogar die unverschlüsselte Kommunikation. Lediglich Firefox und Chrome unterstützen PFS in vollem Umfang.

Für die E-Mail Kommunikation ist PFS essenziell für die Befriedigung hoher Sicherheitsbedürfnisse. Um die dazugehörigen Sicherheitsniveaus abzudecken müssen E-Mail-Server Forward Secrecy unterstützen. Im August 2013 war PFS nur sporadisch verbreitet in der E-Mail-Kommunikationslandschaft.[Vgl.][]Schulz2014 Mittlerweile wird von vielen E-Mail-Providern auch PFS angewandt. Die Umsetzung erfolgt jedoch noch zu zögerlich, wenn man die Tatsache betrachtet, dass PFS im Zusammenhang mit Heartbleed der letzte Funken Hoffnung für betroffene Nutzer war, das zumindest vergangene Kommunikation nicht entschlüsselt werden kann.[Vgl.][]Zhu2014

4.2 OpenPGP - Open Pretty Good Privacy

Ein anderer wichtiger Austausch von Schlüsseln, die für hohe Sicherheitsniveaus entscheidend sind, nämlich für die Ende-zu-Ende-Verschlüsselung, ist der mittels OpenPGP. Die größte Schwierigkeit beim Austausch von Schlüsselinformationen für Ende-zu-Ende-Verschlüsselung ist der für Privatanwender unbequeme Weg die öffentlichen Schlüssel auszutauschen. Dies muss noch manuell erfolgen und ist daher noch nicht Standard heutiger E-Mail Kommunikation. Es gibt zwar die Möglichkeit private Schlüssel durch Dienste[1] zu verteilen, jedoch ist diese Methode umstritten. Sie bietet zwar Ende-zu-Ende Verschlüsselung, jedoch ist der eigentliche Sinn des privaten Schlüssels verfehlt, wenn der Dienst diesen für den Nutzer bestimmt und auf den eigenen Server zwischenspeichert. Der Nutzer hat keinen Einfluss auf den privaten Schlüssel und ist damit nicht Eigentümer dieses Schlüssels, der in Folge dessen kein private Key ist. Selbst vereinzelte CA's gehen mit solcher Methodik vorKaps2014, und untergraben damit die Sicherheit des ganzen CA-Modell.

Für den Austausch von privaten Schlüsseln für Ende-zu-Ende Verschlüsselung ist heute OpenPGP eine gute Werkzeugumgebung für gängige E-Mail Clients, und die Verschlüsselung und den Schlüsselaustausch so bequem wie möglich zu gestalten.

[1] iMessage von Apple für Instant Messaging

5 Transportwegverschlüsselung

Das Secure Socket Layer (SSL)-Protokoll wurde zunächst durch die Firma Netscape entwickelt, um die Kommunikation über Hypertext Transfer Protokoll (HTTP)-Verbindungen abzusichern.[Vgl.][S. 796]Eckert2013 SSL kann auf der Sitzungs- und Präsentationsschicht des Open System Interconnection (OSI)-Referenzmodells angesiedelt werden und setzt meist auf dem Transmission Control Protocol (TCP) auf. Es hat die Aufgabe den darüber liegenden Schichten die Möglichkeit für eine authentifizierte, integritätsgeschützte und verschlüsselte Kommunikation zu geben.[Vgl.][S. 799 ff.]Eckert2013 Die Version SSL 3.0 hat sich mittlerweile als de facto Standard im Internet durchgesetzt und wird von allen gängigen Browsern unterstützt.

Das TLS-Protokoll kann als Weiterentwicklung von SSL 3.0 angesehen werden und liegt aktuell in der Version 1.2 vor. Da beide Protokolle in ihren Kernkonzepten übereinstimmen werden sie häufig synonym verwandt. Da TLS jedoch eine Weiterentwicklung von SSL ist, werden dort einige Erweiterungen eingeführt sowie unsichere Verfahren zur Berechnung von Message Authentication Code (MAC)-Werten durch neuere Varianten ersetzt.

Beide Protokolle bestehen aus mehreren Schichten bzw. Unterprotokollen wobei das Record- und das Handshakeprotokoll von besonderer Bedeutung sind. Das Record-Protokoll ist für die Fragmentierung, Authentifizierung mittels MAC und Verschlüsselung der zu übertragenden Daten zuständig. Mittels des Handshakeprotokolls werden Sitzungen zwischen den Kommunikationspartnern hergestellt. Dies bedeutet, dass die Kommunikationspartner durch den Austausch von Zertifikaten authentifiziert werden können und alle Informationen, die zur Berechnung des Shared Secret für die symmetrische Verschlüsselung der Daten benötigt werden, ausgetauscht werden. Abbildung 5.1 verdeutlicht den schematischen Ablauf eines solchen Sitzungsaufbaus unter Verwendung von RSA für den Schlüsselaustausch.

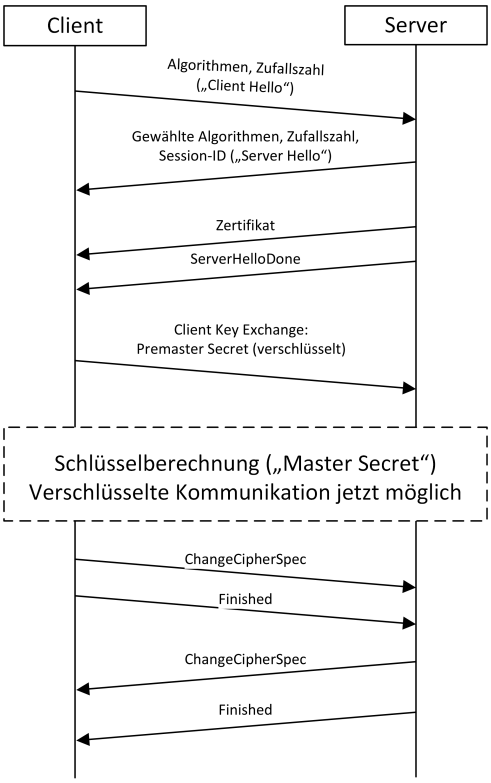

Abbildung 5.1: Handshake-Protokoll mit RSA[2]

Durch die flexible Gestaltung des Handshake-Protokolls wird gleichzeitig auch die Komplexität von TLS/SSL stark erhöht. Dies hat zur Folge, dass durch die hohe Komplexität nicht mit Si-

[2][In Anlehnung an][S. 170]Sorge2013

cherheit alle Schwachstellen beim Design des Protokolls entfernt werden konnten. Außerdem ist zu bedenken, dass TLS/SSL aufgrund seiner weiten Verbreitung ein lohnendes Ziel für Angriffe ist. Die Sicherheit des Protokolls hängt dabei stark von den genutzten kryptologischen Methoden ab. Außerdem ist zu beachten, dass es sich, aufgrund der Ansiedlung des Protokolls unterhalb der Anwendungsschicht, nur um eine Verschlüsselung der transportierten Nutzdaten auf dem Transportweg handelt. Die Daten werden am Kommunikationsendpunkt entschlüsselt und anschließend an die entsprechende Anwendung weitergereicht.

Auch die Authentifizierung der Kommunikationspartner mittels Zertifikaten weißt dieselben Schwachstellen durch die Vertrauensbeziehung zu bekannten CAs, die unzureichend gesichert sind, auf.

6 DNS - Domain Name System

Das Domain Name System (DNS) ist ein wichtiger Dienst im Internet, da er dafür zuständig ist, gut merkbare Domainnamen in Internet Protocol (IP)-Adressen, und umgekehrt, aufzulösen. Um diese Auflösung bewerkstelligen zu können ist DNS in einer Baumstruktur aufgebaut. Wie in der Abbildung[3] veranschaulicht, ist auf der obersten Ebene der Knoten "rootßu finden. Dies ist sozusagen die Wurzel des DNS und hier finden sich auch die Root-Server des DNS Auf der nächsten Ebene kommen die Top-Level-Domain (TLD), anschließend folgen die Second-Level-Domains und zum Aufbau einer gängigen Unified Ressource Locator (URL) für HTTP fehlt noch eine weitere Ebene, die den Hostnamen enthält. Soll ein Domainnamen aufgelöst werden, so fragt der Host zunächst einen Root-Server. Dieser sendet ihm die Adresse des für die entsprechende TLD zuständigen Nameservers mit, an die der Client eine erneute Anfrage stellt. Auch der Nameserver der TLD verweist den Client an für die Second-Level-Domain zuständigen Nameserver. Auf diese Weise kann der Client den dargestellten Baum traversieren, bis er die gewünschte Information erhält. Die für diese Auskünfte benötigten Datensätze werden in sogenannten DNS-Records abgelegt. Das Problem ist hierbei, dass allein mit den DNS-Records nicht die Authentizität des Absenders und damit auch nicht die Integrität der Daten sicher gestellt werden kann. Dies wird beim sogenannten DNS Cache Poisoning ausgenutzt und der Cache eines DNS-Servers manipuliert. Zumeist werden dabei mittels gefälschter Pakete an einen DNS-Server falsche Zuordnungen von Domainnamen auf IP-Adressen hinterlegt, um ein Opfer auf den Server des Angreifers umzuleiten.

6.1 DNSSEC - Domain Name System Security Extensions

Bei Domain Name System Security Extensions (DNSSEC) handelt es sich um eine Sicherheitserweiterung des DNS. Sie beruht auf der Einführung weiterer DNS-Records. Unter anderem gibt es einen Recordtyp, der einen öffentlichen Schlüssel enthält und einen weiteren, der die Signatur eines Ressource Records enthält. Ein autoritativer Nameserver kann eine Zone an einen anderen Nameserver delegieren und hält dann einen Verweis auf den entsprechenden Nameserver vor. Dadurch gibt es pro Zone des DNS maximal einen Zone Signing Key (ZSK) mit dem die Ressource Records dieser Zone signiert werden, dies ist jedoch nicht zwingend notwendig. Mithilfe dieser Schlüssel und Signaturen kann nun eine Zertifikatskette aufgebaut werden. Ein Schwachpunkt dabei ist, dass mindestens ein öffentlicher Schlüssel, der als Key Sgning Key (KSK) fungiert und den ZSK einer Zone signiert, auf einem sicheren Weg zum Client kommen muss. Daher wird dieser Schlüssel auch als Secure Entry Point bezeichnet. Ein weiterer Kritikpunkt an DNSSEC ist die Komplexität des Systems. Diese ist jedoch der benötigten Kompatibilität mit bestehenden Systemen geschuldet und momentan alternativlos.[Vgl.][S. 195]Sorge2013 DNSSEC ist im Prinzip eine eigene PKI deren Hauptschlüssel

[3] Abb. in Anlehnung an Sorge S. 180 einbinden

Root DNSSEC Key die Non-Profit-Organisation Internet Corporation for Assigned Names and Numbers (ICANN) verwaltet.Koetter2014. Der "Hauptschlüssel und damit die DNSSEC-PKI [kann] als vertrauenswürdig [angesehen werden]."Koetter2014 Entscheidender Grund hierfür sind die 21 Trusted Community Representatives (TCR) die an der Erstellung des root key und der Signierungsprozesse partizipieren.

6.2 DANE - DNS-based Authentication of Named Entities

Die Basis für die Applikation DNS-based Authentication of Named Entities (DANE) stellt DNSSEC und soll dabei die Schwachstellen von TLS beim Verschlüsseln des Datentransport entfernen. Denn die Authentizität der verwendeten Zertifikate kann nicht immer gewährleistet werden, und somit besteht die Gefahr der kompromittierten Daten durch MITM-Angriffe oder DNS-Cache-Poisoning. Die Schwachstellen der Zertifikatsprüfung und -aussteller (Vgl. Kapitel 2.3.1 und 2.3.2) werden mittels DANE eliminiert, denn es werden damit nicht mehr mehrere CA-Stellen benötigt. Lediglich das DNS der Empfängerdomain benennt das gültige Zertifikat. Die somit veröffentlichte Prüfsumme aus dem Server-Zertifikat des Ziels kann durch den sendenden Server zutreffend identifiziert werden.

Die Aufgabe von DANE besteht darin TLS-Zertifikate über das DNS automatisch zu verteilen und zu prüfen. Dabei ist DANE nicht auf E-Mail Protokolle beschränkt sondern ist vielmehr für sämtlichen verschlüsselten Datenverkehr einsetzbar. Serverbetreiber tragen den Hash (Fingerabdruck) vom eigenen Public Key in die DNS-Zone ein damit das eigene Zertifikat prüfbar wird. Ein Sender erhält bei einer DNSSEC gesicherten DNS-Anfrage den passenden Transport Layer Security Authentication (TLSA)-Record. Für die Zertifikatsprüfung wird anschließend aus dem empfangenen Publik Key des TLS-Verbindungsaufbau auf dem Sender der Hash berechnet. Ist der Fingerabdruck dem Hash aus der DNS-Abfrage identisch ist die Verbindung vertrauensvoll. Ein durchgängig verfizierter Transport ist allerdings nur möglich wenn alle beteiligten Mail-Server TLSA-Records vorhalten. Wenn eine Prüfstelle keinen besitzt muss “müssen [die Server] auf ungeprüftes TLS zurückfallen oder gar unverschlüsselt kommunizieren”[S. 197]Koetter2014 Bisher ist DNSSEC noch kein Durchbruch gelungen. Aber die Situation für identifizierte, sichere E-Mail-Kommunikation ist dank DANE deutlich besser, denn “die Verwendung [...] ist u.a. schon bei Internet Protokoll Security (IPsec), Secure Shell (SSH), Pretty Good Privacy (PGP) und Secure/Multipurpose Internet Mail Extensions (S/MIME) angedacht,”[S. 196]Koetter2014 und für Hypertext Transfer Protokoll Secure (HTTPS) bereits 2012 standardisiert worden. Die Simple Mail Transfer Protokoll (SMTP)-Standardisierung ist in der Finalisierungsphase. Wenn sich DANE durchsetzt ist die Kommunikation nicht nur unter E-Mail-Verbünden wie E-Mail made in Germany (EmiG) gewährleistet, sondern auch mit der restlichen Welt. Bisher scheitert DANE allerdings an der Unterstützung der Browser, von denen kein gängiger DANE ohne Add-on umsetzt.

Literaturverzeichnis