Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Kai Bielenberg

Sichere Dateiablage für ad hoc-Teams
in heterogenen Netzen

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Kai Bielenberg

Sichere Dateiablage für ad hoc-Teams in heterogenen Netzen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Martin Hübner
Zweitgutachter : Prof. Dr. Kai von Luck

Abgegeben am 23.10.1013

Kai Bielenberg

Thema der Bachelorarbeit
Sichere Dateiablage für ad hoc-Teams in heterogenen Netzen
Stichworte
IT-Sicherheit, sichere Dateiablage, Cloud-Speicher, heterogene Netzwerke, ad hoc-Team, Dateiverschlüsselung, clientseitige Verschlüsselung, Wuala, OwnCloud, Cloudfogger
Kurzzusammenfassung
Diese Arbeit befasst sich mit der Analyse verschiedener Lösungen um Dateien sicher in einem ad hoc-Team, aus heterogenen Netzwerken heraus, austauschen zu können. Es werden insgesamt vier Lösungen verglichen. Zwei Lösungen nutzen direkten Datenaustausch. Die anderen beiden Lösungen nutzen einen zentralen Datenspeicher, der die Daten synchronisiert. Für die Vergleichbarkeit dieser Lösungen, wird anhand einer fachlichen und technischen Analyse ein ideales Zielsystem erstellt. Dieses Zielsystem dient der anschließenden Nutzwertanalyse als Bewertungsgrundlage. Abschließend werden die Bewertungen verglichen und die sicherheitstechnischen Probleme der einzelnen Lösungen aufgezeigt. Für die gefundenen Probleme werden anhand des Zielsystems Lösungsmöglichkeiten aufgezeigt.

Kai Bielenberg

Title of the paper
Secure data storage for ad hoc-teams in heterogeneous networks
Keywords
IT-security, secure data storage, cloud storage, heterogeneous networks, data encryption, client side encryption, Wuala, OwnCloud, Cloudfogger
Abstract
This paper deals with the problems of secure data storage in ad hoc-teams, which are working in heterogeneous networks. Four different solutions will be analysed. Two solutions are using direct data exchange to solve the problems. The other solutions are storing data in one central data storage. This storage is synchronized between the team members. A value benefit analysis will be used to compare these solutions. A value benefit analysis compares these solutions using the most important criterions for the assessment. These criterions need to be derived from the given scenario. Functional and non-functional criterions are analysed in this progress. Subsequently the security problems of the different approaches will be shown and possible solutions will be highlighted.

Inhaltsverzeichnis

Tabellenverzeichnis

1 Einleitung

Heutzutage ist es im privaten Bereich selbstverständlich neue Ereignisse sofort mit Freunden zu teilen und zu diskutieren. Die Weiterentwicklung der Smartphones und des mobilen Internets haben dies ermöglicht. Daher stellt sich die Frage, ob sich diese ständige Vernetzung auch auf die Arbeitswelt erweitern lässt. Ist es möglich spontan ein Team zu bilden und sofort gemeinsam an einem Projekt zu arbeiten? Im Mittelpunkt steht die Frage des Datenaustausches und der Datenspeicherung, wie schnell dieser und vor allem wie sicher dieser möglich ist. Wie wird es möglich neue oder aktualisierte Daten schnell und sicher an andere Teammitglieder zu verteilen? Als einfache Lösung wäre hier der Austausch der Daten per E-Mail oder USB-Stick zu nennen. Diese Art der Datenweitergabe ist einfach und selektiv, das heißt nur gewünschte Personen erhalten die Daten. Unter Studenten ist diese Art der Datenverbreitung keine Seltenheit. Schnell lassen sich die Grenzen dieser Verteilungsmöglichkeit feststellen. Wenn gleichzeitig an einem Dokument gearbeitet wird und jeder seine neue Version an alle verteilen möchte, kommt es zu Problemen. Wie werden die verschiedenen Versionen zusammengeführt? Überschneiden sich die Änderungen vielleicht sogar? Bei einem USB-Stick kommt erschwerend hinzu, dass die Übergabe der neuen Version physisch erfolgt, die entsprechenden Personen müssen also die Möglichkeit besitzen sich persönlich zu treffen. Es wird schnell klar, dass diese Art der Datenverarbeitung nur in sehr kleinen Teams mit guter Absprache funktionieren kann.
Zu erkennen ist, dass die Verteilung der Daten über Netzwerk erheblich komfortabler ist und dass dies ein Schritt in die richtige Richtung darstellt. Dies kann auch bedeuten, dass die Daten ebenfalls zentral in der Cloud gespeichert werden. Im Hinblick auf die Datenverteilung scheint dies die beste Variante zu sein, aber es ist zu klären, inwieweit die Anbieter solcher Speichermöglichkeiten die Daten vor fremden Zugriff schützen. Dies bezieht die Anbieter selbst mit ein. Folgende Fragen sind unter anderen im späteren Verlauf dieser Ausarbeitung zu klären:
Wie ist es möglich effizient und sicher in heterogenen Netzen teambasiert an gleichen Daten zu arbeiten? Wie wird die Sicherheit der Daten garantiert?
Wie wird der Zugriff auf die Daten geregelt, so dass nur das Team die Daten lesen und bearbeiten kann? Konkretere Anforderungen werden im Kapitel Analyse erarbeitet.

1.1 Szenario

An der HAW Hamburg gibt es verschiedene Probleme, an denen verschiedene Departments zusammen arbeiten müssen. Eine Problemstellung, die im Folgenden genauer betrachtet wird, sieht wie folgt aus:
Wenn sich ein Student an der HAW Hamburg einschreiben möchte und nicht angenommen wird, hat er die Möglichkeit die HAW Hamburg zu verklagen um eine Annahme zu erreichen. Dies sind so genannte Studienplatzklagen.
Dokumente, die diese Verfahren betreffen, müssen oft von verschiedenen Personen aus verschiedenen Departments eingesehen oder bearbeitet werden. Für die Bearbeitung dieser Verfahren werden für eine begrenzte Zeit Teams gebildet. Diese Teams kümmern sich um die Bearbeitung der Studienplatzklagen. Hier ist sicherzustellen, dass die Daten vor fremder Einsicht und Zugriff geschützt werden und dass das Team jederzeit Zugriff auf die Dokumente hat. Die Teammitglieder setzten sich, wie oben beschrieben, aus Mitarbeitern verschiedener Departments zusammen und nutzen daher verschiedene, voneinander getrennte heterogene lokale Netzwerke. Eine einfache Freigabe der Dateien im lokalen Netz oder die Nutzung eines lokalen Servers ist daher nicht möglich.
Diese Zusammenarbeit ist zum jetzigen Zeitpunkt an der HAW-Hamburg nicht wie gewünscht durchführbar.

1.2 Zielsetzung

Gesucht wird eine Lösung, die die oben genannten Probleme auf eine möglichst einfache Weise löst. Die Probleme lassen sich in drei Teilgruppen aufteilen:

1.) Eine spontane Teambildung, also so genannte ad hoc Teambildung, muss ohne großen Aufwand möglich sein. Erschwerend sind die vorhandenen heterogenen Netzwerke, die teilweise komplett getrennt voneinander aufgebaut sind.
2.) Die Dateien müssen so übertragen und abgelegt werden, dass sie vor fremden Zugriffen geschützt sind. Natürlich muss die sichere Dateiablage unter Berücksichtigung von Punkt 1, also des gemeinsamen Zugriffs, umgesetzt werden.
3.) Es muss möglich sein von verschiedenen heterogenen Netzen aus zusammen zu arbeiten.

Anhand dieser Problemgruppen wird analysiert, wie das gewünschte Zielsystem auszusehen hat. Besonders wichtig ist in diesem Zusammenhang, dass die Umsetzungen dieser Ziele einfach vom Nutzer umzusetzen ist. Die Kosten der Lösung müssen ebenfalls betrachtet werden.
Dies sind unter anderem Qualitätskriterien für Softwareprodukte, die ebenfalls in die Bewertung mit eingehen werden. Um eine möglichst gute Lösung finden zu können, werden Produkte verschiedener Kategorien bewertet, um sie dann einheitlich vergleichen zu können. Die Produkte der einen Kategorie verfolgen als Ansatz den direkten Datenaustausch,

Produkte der anderen Kategorie nutzen einen zentralen Datenspeicher für die Datenverteilung.

1.3 Abgrenzung

Diese Arbeit beschäftigt sich mit dem Thema der sicheren Datenspeicherung und der Möglichkeit diese Daten in Teams zu nutzen. Es wird angestrebt eine Lösungsmöglichkeit zu finden, die am besten auf die im Szenario genannten Anforderungen passt. Für eventuell fehlende Aspekte wird soweit wie möglich eine Lösungsmöglichkeit skizziert. Im Hinblick auf den Bereich der IT-Sicherheit und besonders des Datenschutzes gibt es viele juristische Kriterien, die ebenfalls eine wichtige Rolle spielen, aber in dieser Arbeit nicht weiter behandelt werden. Das Augenmerk liegt hauptsächlich im technischen Bereich der IT-Sicherheit und somit auf der korrekten Umsetzung der Anforderungen im Hinblick auf die Datensicherheit und dem Schutz vor Fremdeinwirkung.

1.4 Aufteilung und Organisation der Arbeit

Die Arbeit wird in folgende Teile unterteilt:
Nachdem das Thema der Arbeit einleitend beschrieben und umrissen wurde, sind verschiedene Problemstellungen aufgezeigt worden.
Im 2. Kapitel werden die nötigen Grundlagen zum Verständnis dieser Arbeit aufgezeigt. Hierdurch wird sichergestellt, dass ein einheitliches Verständnis für die in der Arbeit genutzten Begriffe und Verfahren vorherrscht.
Das 3. Kapitel widmet sich der Anforderungsanalyse. Kern dieses Kapitels ist die Erstellung eines Zielsystems, damit die unterschiedlichen Problemlösungen anhand eines einheitlichen Bewertungsmusters eingeordnet werden können. Hierfür wird das Problem in Ober- und Unterziele aufgeteilt, die dann je nach Wichtigkeit mit einer unterschiedlichen Gewichtung in die Bewertung mit eingehen.
Im 4. Kapitel werden die in dieser Arbeit behandelten Lösungsansätze vorgestellt. Diese werden anhand des Zielsystems im Stile einer Nutzwertanalyse bewertet. Dies ermöglicht eine Einordnung der Lösungsansätze anhand des größten Nutzens für den Anwender.
Die nächsten Kapitel beinhalten ein Fazit und einen Ausblick. Das Fazit zeigt zusammenfassend die Erkenntnisse aus den Bewertungen der Lösungsmöglichkeiten. Anhand dieser gewonnenen Erkenntnisse kann ein Ausblick auf die Zukunft dieses Themas aufgezeigt werden.

2 Grundlagen

Diese Kapitel dienen dazu den Leser über vorausgesetzte Grundlagen zu informieren, die zum Verständnis dieser Arbeit notwendig sind. So ist sichergestellt, dass ein einheitliches Verständnis der verwendeten Begriffe und Verfahren vorherrscht.

2.1 Kerckhoffs-Prinzip

Das Kerckhoffs'sche Prinzip wurde 1883 von Auguste Kerckhoffs in seinem Artikel La cryptographie militaire[1] formuliert und ist ein Grundsatz der Modernen Kryptographie. Im Allgemeinen besagt dieses Prinzip, dass die Sicherheit eines Verschlüsselungsverfahrens nicht auf der Geheimhaltung der Ver- oder Entschlüsselungsfunktion beruhen darf. Die Sicherheit der verwendeten kryptographischen Primitive darf nur von der Geheimhaltung des Schlüssels abhängen.
Das Kerckhoffs'sche Prinzip beschreibt das zweite der folgenden sechs Grundsätze zur Konstruktion eines sicheren Verschlüsselungsverfahrens, welche von Auguste Kerckhoffs im oben genannten Artikel verfasst wurden:

1.) Das System muss im Wesentlichen unentzifferbar sein.
2.) Das System darf nicht auf Geheimhaltung beruhen.
3.) Das System muss leicht übermittelbar sein und der Schlüssel muss ohne schriftliche Aufzeichnung merkbar sein.
4.) Das System sollte mit telegraphischer Kommunikation kompatibel sein.
5.) Das System muss transportierbar sein und dessen Nutzung darf nicht mehr als eine Person erfordern.
6.) Das System muss einfach anwendbar sein.

Wichtig sind in diesem Zusammenhang die ersten beiden Thesen. Die anderen
Thesen beschreiben die Nutzbarkeit, Übertragbarkeit und einfache Anwendbarkeit des Systems. Im heutigen Computerzeitalter, in dem die Kryptographie von den Rechnern erledigt wird, stellen diese Punkte oftmals keine großen Probleme mehr dar.
These 1 und 2 sagen aus, dass das System nicht entzifferbar sein darf und nicht auf Geheimhaltung beruhen darf. Dies hat viele Vorteile. Es ermöglicht, dass ein Algorithmus veröffentlicht und auf dessen Sicherheit geprüft werden kann, ohne dass dies später Sicher-

[1] [1] http://www.petitcolas.net/fabien/kerckhoffs/

heitsprobleme verursacht. Außerdem ist es in einem gegebenen System wesentlich einfacher einen Schlüssel zu tauschen, als neue Ver- und Entschlüsselungsfunktionen zu verwenden.
Daher ist das Kerckhoffs'sche Prinzip ein Grundsatz der modernen Kryptographie.

2.2 Kryptographische Verfahren

Im folgenden Abschnitt werden grundsätzliche Kryptographische Verfahren erläutert, so dass ein Überblick über die aktuell geltenden Prinzipien gegeben wird.
Um diese einheitlich zu beschreiben, werden in der Kryptographie bekannte Synonyme für die Kommunikationspartner verwendet. Es wird angenommen, dass **Alice** eine Nachricht an **Bob** senden will. **Eve** versucht diese Nachricht abzufangen und den Inhalt der Nachricht zu ermitteln. Die Kommunikationspartner werden hier zum leichteren Verständnis als Menschen dargestellt. In der Realität können dies aber auch verschiedene Rechner, Server oder sogar Komponenten innerhalb eines Rechners sein.

2.2.1 Verschlüsselungsverfahren

Verschlüsselungsverfahren beschreiben wie ein Klartext in Chiffretext und zurück überführt werden kann. Im Wesentlichen wird zwischen **symmetrischen**, **asymmetrischen** und **hybriden** Verfahren unterschieden, die wiederum verschiedene Ausprägungen haben können.

2.2.1.1. Symmetrische Verfahren

Symmetrische Verfahren zeichnen sich dadurch aus, dass zur Ver- und Entschlüsselung der gleiche Schlüssel verwendet wird. Diese Verfahren zeichnen sich durch ihre Einfachheit und Effizienz beim Verschlüsseln großer Datenmengen aus. Problematisch ist die Verwendung des gemeinsamen geheimen Schlüssels. Bei der Übergabe des Schlüssels muss gewährleistet sein, dass dieser so übertragen oder übergeben wird, dass niemand außer dem Kommunikationspartner Einsicht erhält. Dies ist oft schwieriger als es auf den ersten Blick erscheint. Selbst wenn es möglich ist den Schlüssel persönlich zu übergeben, gibt es Risiken. Es wird als unsicher erachtet mehrmals den gleichen Schlüssel zu verwenden, da nie sichergestellt werden kann, ob dieser wirklich noch geheim ist. Daher wird in der Praxis für jede Kommunikation ein neuer Schlüssel erstellt. Es wäre sehr schwer diesen jeweils persönlich zu übergeben. Ein Schlüsselaustausch zwischen zwei Personen ist bereits effizient realisiert worden. Hier kommt das Diffie-Hellmann-Schlüsselaustauschprotokoll[2] zum Einsatz. Dies wird hier nur der Vollständigkeit halber erwähnt und nicht genauer erläutert. Eine weitere Möglichkeit ist die Verwendung von hybriden[3] Verfahren.

[2] [2, p. 449] Eckert, Claudia: IT-Sicherheit, Oldenbourg 2013
[3] Siehe Kapitel 2.2.1.3 Hybride Verfahren.

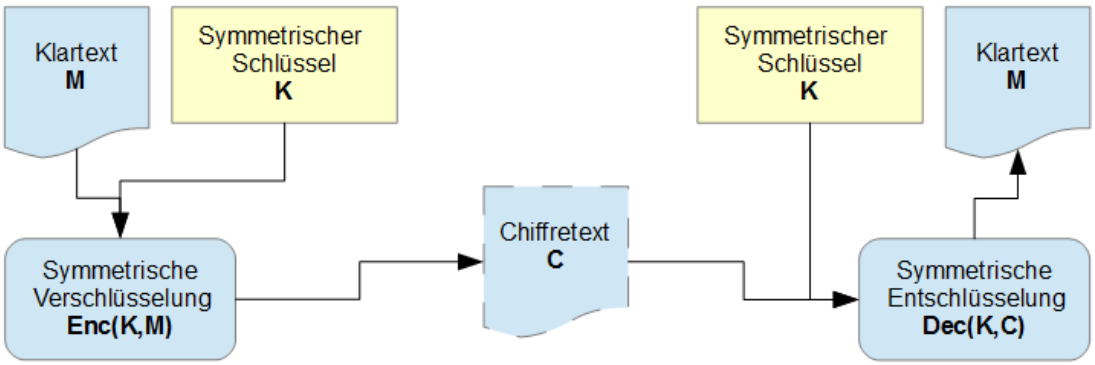

Abbildung 1: Symmetrische Verschlüsselung

In *Abbildung 1: Symmetrische Verschlüsselung* ist der Ablauf eines normalen symmetrischen Verschlüsselungsverfahrens zu sehen.

1.) Bob und Alice vereinbaren einen geheimen Schlüssel **K**
2.) Alice erstellt den Chiffretext **C** mithilfe der Verschlüsselungsfunktion **Enc(K,M)**, so dass **C = Enc(K,M).**
3.) Bob entschlüsselt **C** mit der Entschlüsselungsfunktion **Dec(K,C)**. Es gilt **M = Dec(K,C)** oder auch **M = Dec(K,Enc(K,M)).**

2.2.1.2. Asymmetrische Verfahren

Asymmetrische Verschlüsselungsverfahren beruhen auf der Nutzung eines Schlüsselpaares. Die beiden Schlüssel werden öffentlicher und privater Schlüssel genannt. Die Ver- und Entschlüsselung beruht auf der Nutzung der beiden Schlüssel in abwechselnder Reihenfolge. Je nach Reihenfolge ergeben sich unterschiedliche Anwendungsmöglichkeiten.

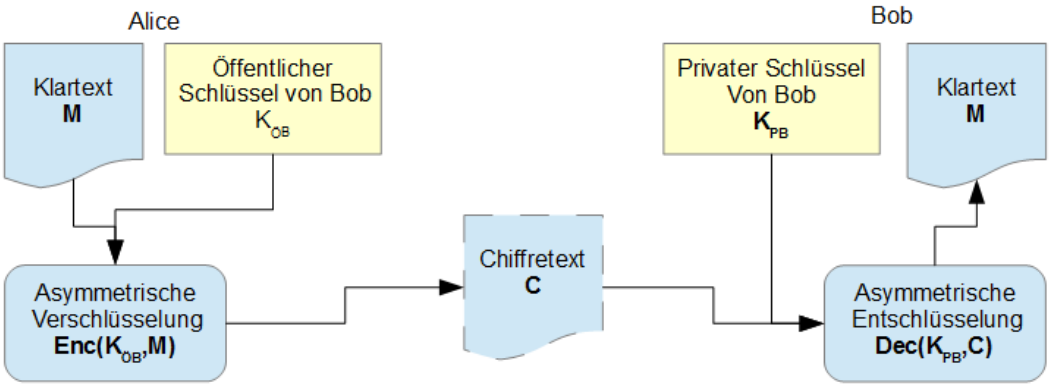

Abbildung 2: Asymmetrische Verschlüsselung 1

In *Abbildung 2: Asymmetrische Verschlüsselung 1* wird der typische Ablauf einer asymmetrischen Verschlüsselung gezeigt.

1.) Bob teilt Alice seinen öffentlichen Schlüssel $\mathbf{K_{ÖB}}$ mit.
2.) Alice verschlüsselt ihre Nachricht **M** mit dem öffentlichen Schlüssel von Bob $\mathbf{K_{ÖB}}$, so dass gilt $\mathbf{C = Enc(K_{ÖB}, M)}$.
3.) Bob entschlüsselt die erhaltene Nachricht **C** mit Hilfe seines privaten Schlüssels $\mathbf{K_{PB}}$. Also gilt $\mathbf{M = Dec(K_{PB},C)}$, beziehungsweise $\mathbf{M = Dec(K_{PB}, Enc(K_{ÖB}, M))}$.

Wird das Verfahren in diese Richtung angewandt, handelt es sich um ein normales Verschlüsselungsverfahren. Der Vorteil gegenüber den symmetrischen Verfahren ist, dass es kein Schlüsselaustauschproblem gibt. Der öffentliche Schlüssel kann einfach an den Kommunikationspartner übertragen werden. Leider sind diese Verfahren sehr ineffizient, so dass sie sich nicht zum Verschlüsseln großer Datenmengen eignen.

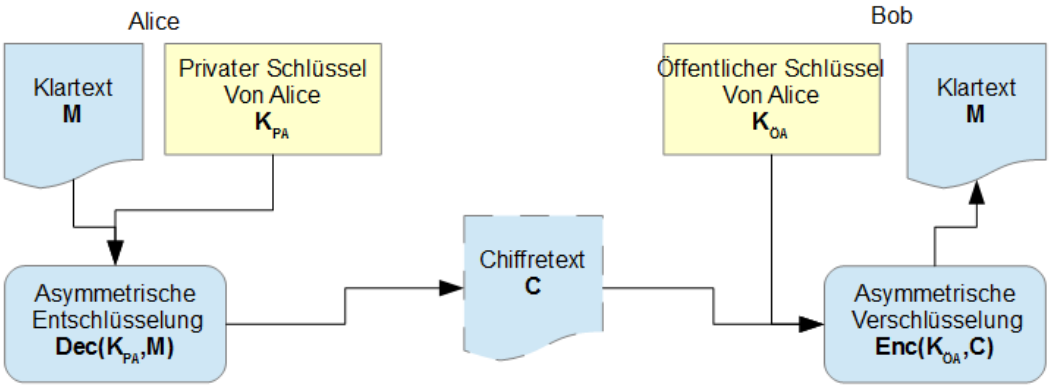

Abbildung 3: Asymmetrische Verschlüsselung 2

In *Abbildung 3: Asymmetrische Verschlüsselung 2* wird die Verschlüsselung in anderer Reihenfolge gezeigt.

1.) Alice teilt Bob ihren öffentlichen Schlüssel $\mathbf{K_{ÖA}}$ mit.
2.) Alice wendet die Entschlüsselungsfunktion mit ihrem privaten Schlüssel $\mathbf{K_{PA}}$ und ihrer Nachricht **M** an, um den Chiffretext $\mathbf{C = Dec(K_{PA},M)}$ zu erhalten.
3.) Bob wendet die Verschlüsselungsfunktion zusammen mit dem öffentlichen Schlüssel von Alice $\mathbf{K_{ÖA}}$ auf den Chiffretext an und erhält die Nachricht **M**. Es gilt $\mathbf{M = Enc(K_{ÖA},C)}$ oder auch $\mathbf{M = Enc(K_{ÖA}, Dec(K_{PA},M))}$

Mit dieser Variante der asymmetrischen Verfahren lässt sich der Ersteller einer Nachricht verifizieren. Dies kann als Grundlage von Authentifikationsverfahren und Signaturverfahren[4] verwendet werden.

2.2.1.3. Hybride Verfahren

In den vorigen Kapiteln wurde festgestellt, dass sowohl symmetrische Verfahren als auch asymmetrische Verfahren Probleme aufweisen, die nicht ohne weiteres gelöst werden können. Hybride Verfahren versuchen durch Kombination dieser beiden Arten die Fehler auszugleichen. Symmetrische Verschlüsselungsverfahren sind effizient, haben jedoch das Problem des sicheren Schlüsselaustausches. Asymmetrische Verfahren sind ineffizient bei großen Datenmengen, lösen aber das Schlüsselproblem.
Mit einer Kombination aus beiden Verfahren lassen sich diese Probleme ausgleichen.

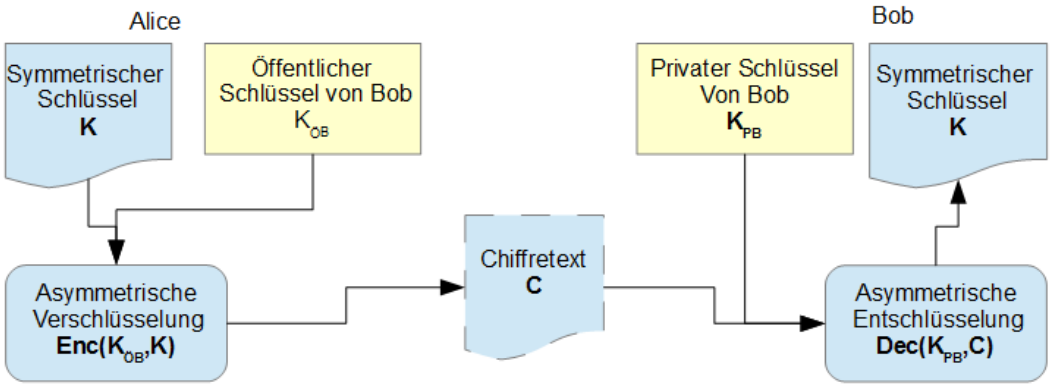

Abbildung 4: Schlüsselaustausch mittels asymmetrischer Verschlüsselung

In *Abbildung 4: Schlüsselaustausch mittels asymmetrischer Verschlüsselung* ist zu sehen wie ein geheimer symmetrischer Schlüssel **K** mittels asymmetrischer Verschlüsselung sicher ausgetauscht werden kann. Die weiteren Daten werden dann, mit dem symmetrischen Schlüssel **K** verschlüsselt, übertragen. Idealerweise wird der Schlüssel nach jeder Kommunikation neu erzeugt und ausgetauscht, um die Sicherheit zu erhöhen.

2.2.1 Kryptographische Hashverfahren[5]

Hashverfahren werden in vielen Bereichen der Informatik eingesetzt und dienen dazu, den Zugriff auf Objekte effizient zu realisieren. Eine Hashfunktion bildet eine variable Datenmenge, nachfolgend Urbild genannt, auf einen Block fester länge ab. Der so entstandene Block nennt sich Bild- oder Adressbereich. Die Bezeichnung Adressbereich stammt daher,

[4] Siehe Kapitel 1.1.1 Elektronische Signaturverfahren
[5] [3, p. 381 ff.] Eckert, Claudia: IT-Sicherheit, Oldenbourg 2013

dass diese Blöcke häufig zur Adressierung der Urbilder verwendet werden. Des Weiteren ist eine Hashfunktion eine nicht injektive Abbildung[6].
Da der Bildbereich von Hashfunktionen meist erheblich kleiner als der Urbildbereich ist und aufgrund der Tatsache, dass Hashfunktionen nicht injetiv sind, kann es zu so genannten **Kollisionen** kommen. Dies ist der Fall, wenn zwei verschiede Urbilder auf denselben Bildblock abgebildet werden. In den normalen Anwendungsgebieten, wie der Adressierung von Objekten, ist dies nicht problematisch. Wird die Hashfunktion allerdings in der Kryptographie verwendet, ist dies problematischer.
Daher wurde die Klasse der kryptographischen Hashfunktionen erstellt. Diese haben bestimmte Eigenschaften, die eine Kollision sehr unwahrscheinlich machen. Kryptographische Hashfunktionen dienen der eindeutigen Identifikation von Objekten und zur Erkennung von Manipulationen an diesen. Daher gilt es Kollisionen möglichst unwahrscheinlich zu machen.
Es wird zwischen **schwach kollisionsresistenten** und **stark kollisionsresistenten Hashfunktionen** unterschieden.
Eine **schwach kollisionsresistente Hashfunktion H** muss folgende Eigenschaften erfüllen:

1.) **H** ist eine **Einwegfunktion**. Der Hashwert **h** einer Nachricht **M** ist mit **h = H(M)** effizient zu berechnen. Umgekehrt ist die Nachricht **M** aus einem gegebenen Hashwert **h** mit $\mathbf{M = H^{-1}(h)}$ nicht effizient bestimmbar.
2.) Aus einem gegebenen Hashwert $\mathbf{h = H(M_1)}$ ist eine Nachricht $\mathbf{M_2}$, für die $\mathbf{h = H(M_2)}$ gilt, nicht effizient zu erstellen. Bei einem gegebenen Hashwert kann also nur schwer eine zweite Nachricht gefunden werden, die den gleichen Hashwert erzeugt.

Laut der Definition ist es so gut wie unmöglich bei einer gegebenen Nachricht $\mathbf{M_1}$ eine entsprechende Nachricht $\mathbf{M_2}$ zu konstruieren, so dass die Hashwerte übereinstimmen. Es wird allerdings keine Aussage darüber getroffen wie kollisionsresistent die Hashfunktion ist, wenn beide Nachrichten, also ein Nachrichtenpaar, erzeugt werden.
In der Definition von einer **stark kollisionsresistenten Hashfunktion H** ist dieser Fall hingegen abgedeckt. Laut Definition besitzt **H** folgende Eigenschaften:

1.) **H** ist eine schwach kollisionsresistente Hashfunktion.
2.) Es gibt kein effizientes Verfahren um ein Nachrichtenpaar $\mathbf{(M_1,M_2)}$ zu erzeugen, so dass folgendes gilt $\mathbf{H(M_1) = H(M_2)}$. Also ist es so gut wie unmöglich Nachrichtenpaare mit denselben Hashwerten zu erzeugen.

[6] f: A -> B ist injektiv, wenn gilt $\forall x, y \in X : (x \neq y \Rightarrow f(x) \neq f(y))$

2.2.2 Elektronische Signaturverfahren[7]

Elektronische Signaturverfahren dienen dazu, eine signierte Nachricht verbindlich an den Verfasser zu binden. Nach der Signatur ist also der Urheber zweifelsfrei identifizierbar und die Unveränderbarkeit der Nachricht ist gewährleistet. Wie bereits beschrieben, können einige asymmetrische Verschlüsselungsalgorithmen als digitale Signaturen verwendet werden. Es gibt aber auch dedizierte Algorithmen, die ausschließlich zum Signieren geeignet sind. Beide Varianten nutzen allerdings Schlüsselpaare, die aus einem öffentlichen und einem privaten Schlüssel beruhen. Im folgenden Abschnitt wird das Protokoll zum Erstellen einer digitalen Signatur erläutert.

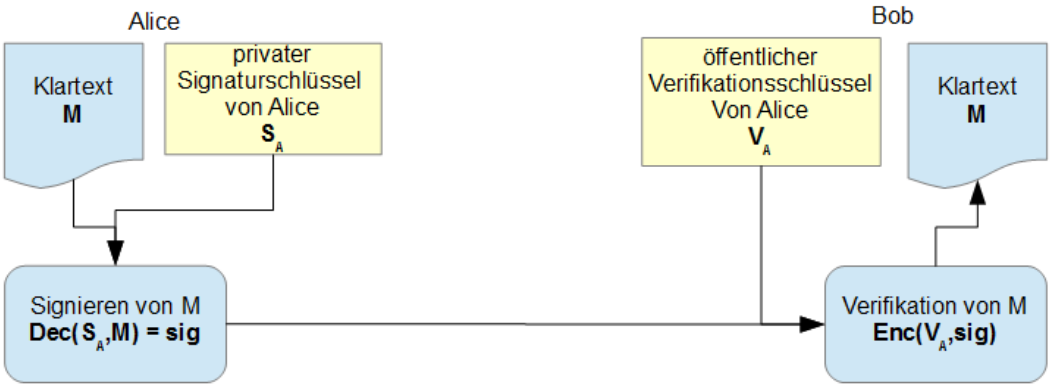

Abbildung 5: Signatur

In *Abbildung 5: Signatur* ist das Protokoll einer digitalen Signatur abgebildet. Sie läuft in folgenden Schritten ab:

1.) Alice generiert ein Schlüsselpaar **(S_A,V_A)** mit **S_A** = privater Signaturschlüssel von Alice und **V_A** = öffentlicher Verifikationsschlüssel von Alice.
2.) Alice hinterlegt **V_A** in einer öffentlichen Datenbank oder sendet ihn direkt zu Bob.
3.) Alice signiert/verschlüsselt eine Nachricht **M** unter Verwendung von **S_A** und erhält die Signatur **sig**. Es gilt: **sig = Dec(S_A, M)**.
4.) Alice sendet **sig** an Bob.
5.) Bob verifiziert die Signatur **sig** mit Hilfe von Alices Verifikationsschlüssel **V_A** und erhält die Nachricht **M**. Es gilt: **M = Enc(V_A, sig)**.

[7] [3, p. 400 ff.] Eckert, Claudia: IT-Sicherheit, Oldenbourg 2013

An digitale Signaturen werden folgende Anforderungen gestellt und es gilt zu prüfen, ob das vorgestellte Protokoll diesen entspricht:

1.) Zweifelsfreie Identität
2.) Keine Wiederverwendbarkeit
3.) Unveränderbarkeit
4.) Verbindlichkeit

Die **zweifelsfreie Identität** ist gewährleistet, wenn der zur Verifikation der Signatur verwendete öffentliche Schlüssel eindeutig einer juristischen Person[8] zuzuordnen ist. Damit dies der Fall ist, wurden Möglichkeiten geschaffen einen Schlüssel einer Person zuzuordnen. Die am meisten genutzte Möglichkeit ist die Nutzung von Zertifikaten. Zertifikaten binden einen Nutzer an ein Schlüsselpaar. Ausgestellt werden diese von einer Vertrauensinstanz. Diese Instanz wird von allen Nutzern als vertrauenswürdig angesehen und daher gelten die ausgestellten Zertifikate ebenfalls als vertrauenswürdig.
Eine Signatur ist nur für die unterzeichnete Nachricht gültig und darf **nicht wiederverwendbar** sein. Da die Signatur das Ergebnis einer Verschlüsselungsoperation ist, ist diese als Funktionswert abhängig von der signierten Nachricht. Daher ist bei Anwendung eines korrekten Verfahrens die Wiederverwendbarkeit ausgeschlossen.
Ein signiertes Dokument muss **unveränderbar** sein, damit der Inhalt des Dokuments nach der Signatur nicht mehr abstreitbar ist. Die Verifikation der Signatur ist nur möglich wenn das Dokument nicht geändert wurde, da die Signatur als Funktionswert vom Inhalt des Dokumentes abhängig ist.
Eine Signatur muss **verbindlich** sein. Wenn der private Schlüssel des Senders nicht kompromittiert wurde, ist die Signatur nicht abstreitbar. Denn die Signatur kann nur vom Besitzer des privaten Schlüssels erstellt worden sein und der private Schlüssel ist nur diesem bekannt.
Wird die Signatur eines Dokumentes an dieses angehängt und dann zusammen mit dem Dokument verschlüsselt, kann diese beim Entschlüsseln der Nachricht ebenfalls zur Verifikation der Nachricht verwendet werden.

2.3 Angriffstechniken

In einem IT-System gibt es viele verschiedene Angriffsvektoren. Erwähnenswert sind Sicherheitslücken in den Betriebssystemen, Programmen oder Netzwerken. In diesem Dokument werden nur Angriffsarten beschrieben, die direkt mit der Problemstellung zu tun haben und die durch die zur Lösung verwendete Software entstehen können. Die anderen Angriffsarten bzw. Sicherheitsmängel sind durch die Problemlösung nicht beeinflussbar und

[8] [4] http://wirtschaftslexikon.gabler.de/Definition/juristische-person.html „Eine juristische Person ist eine Personenvereinigung oder ein Zweckvermögen mit vom Gesetz anerkannter rechtlicher Selbstständigkeit. Die juristische Person ist Träger von Rechten und Pflichten, hat Vermögen, kann als Erbe eingesetzt werden, in eigenem Namen klagen und verklagt werden.“

daher hier nicht relevant. Zuerst wird eine grobe Einteilung der Angriffsarten aufgezeigt, dann werden verschiedene Methoden vorgestellt.
Angriffe lassen sich in **passive** und **aktive** Angriffe einteilen. Diese unterscheiden sich wie folgt:
Passive Angriffe zeichnen sich dadurch aus, dass keine Daten manipuliert werden. Der Angreifer schaltet sich lediglich zwischen zwei Kommunikationspartner und hört mit. Diese Angriffe sind schwerer aufzudecken als aktive Angriffe, da hier keine Daten manipuliert werden. Sie gefährden die Vertraulichkeit und die Integrität der Daten. Schützen kann man sich vor diesen Angriffen, indem die Daten niemals unverschlüsselt übertragen werden.
Aktive Angriffe werden dadurch beschrieben, dass gezielt Daten manipuliert werden, um dem Angreifer Vorteile zu verschaffen. Ziel ist es oft, auf dem angegriffenen System mehr Rechte zu erhalten, um hier schadhaften Code auszuführen oder die entsprechenden Schlüssel zum Entschlüsseln der Kommunikationsdaten zu erlangen. Diese Angriffsart gefährdet die Vertraulichkeit, Integrität und Authentizität der Daten. Die Verschlüsselung der Daten ist hier nicht ausreichend, es muss ebenfalls gewährleistet sein, dass die Daten auf dem Kommunikationsweg nicht geändert wurden. Hierfür eignen sich Hash- und Signierverfahren.
Oft werden passive und aktive Angriffe zusammen eingesetzt. Es wird zum Beispiel versucht aktiv die Kommunikationswege umzuleiten, um dann passiv zuhören zu können.
Im Folgenden werden Angriffstechniken aus diesen Bereichen vorgestellt:

1.) Manipulation der Kommunikation
2.) Erschleichen von Zugangsberechtigungen
3.) Ausnutzen von Schwachstellen in Verschlüsselungstechniken

Zwei typische Angriffstechniken zur **Manipulation der Kommunikation** sind das so genannte **Spoofing** und die **Man-in-the-middle** Attacke.
Spoofing bezieht sich auf das Fälschen der Identität. Es wird versucht sich so zu maskieren, dass das Opfer denke es kommuniziere mit dem richtigen Partner. Dies kann in verschiedensten Bereichen der Kommunikation geschehen. Beim IP-Spoofing wird die Senderadresse in IP-Paketen gefälscht. Anders kann beim DNS-Spoofing ein Eintrag in einem DNS Server modifiziert werden. Durch das Angeben einer falschen Mailadresse als Absender kann Mail-Spoofing betrieben werden.
Eine spezielle Art des Spoofing ist die **Man-in-the-middle** Attacke. Hier setzt sich der Angreifer zwischen zwei Kommunikationspartner und spielt beiden vor der jeweils andere zu sein. So können die Kommunikationsdaten der beiden Partner beliebig manipuliert werden.

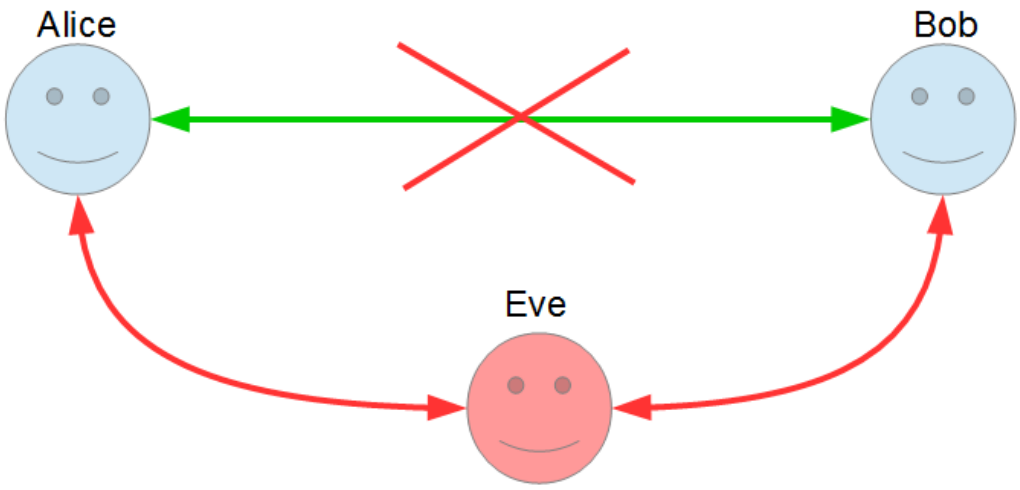

Abbildung 6: Man-in-the-middle

In *Abbildung 6: Man-in-the-middle* wird ein typischer Man-in-the-middle Angriff gezeigt. Zuerst lenkt Eve die Kommunikation zwischen Alice und Bob auf sich um. Dann kann Eve die Daten beliebig manipulieren. Angenommen dies geschieht bei einem Schlüsselaustausch für eine asynchrone Verschlüsselung. Eve schickt Alice und Bob einfach ihren eigenen öffentlichen Schlüssel. Alice und Bob wissen davon nichts und nutzen diesen zur Verschlüsselung der Daten. Eve kann diese nun beim Empfang entschlüsseln und beliebig manipulieren, um sie dann an den anderen Partner weiter zu senden.
Um dies zu verhindern müssen sich die Kommunikationspartner hinreichend authentifizieren, damit gesichert ist, mit wem tatsächlich kommuniziert wird. Die Daten sollten ebenfalls so gesichert sein, dass eine Manipulation der Daten nicht unbemerkt möglich ist. Hierfür eignen sich Hash- und Signier-Mechanismen[9].
Eine Man-in-the-middle Attacke kann die Vorbereitung für einen **Replay-Angriff** sein. Hier versucht **Eve** durch das wiederholte Senden von abgefangenen Paketen Vorteile zu erlangen. Zum Beispiel können Zugangsinformationen abgefangen, kopiert und weitergeleitet werden. Eve kann nun zu einem späteren Zeitpunkt diese Daten erneut senden und erhält Zugriff auf das gewünschte System.
Das **Erschleichen von Zugangsberechtigungen** ist ebenfalls ein großes Problem. Bekannte Techniken in diesem Bereich sind das **Social-Engineering** und die **Brute-Force** Methode.
Social-Engineering ist ein sehr effektiver Angriff. Diese Methode nutzt eine schlecht zu beeinflussende Sicherheitslücke im System aus, den Menschen. Der Angreifer versucht durch direkten Kontakt zum Kommunikationspartner die Sicherheitskennung herauszubekommen. Dies geschieht oft durch das Vorspielen falscher Tatsachen und das Ausnutzen der Unwissenheit der Opfer. Der Angreifer ruft zum Beispiel bei einem Opfer an und gibt an der Administrator zu sein. Er behauptet dann, unter einem Vorwand das Passwort des Opfers

[9] Siehe Kapitel 2.2.1 Hashverfahren und Kapitel 1.1.1 Signaturverfahren

zu benötigen. Ist dieses schlecht geschult oder leichtgläubig genug, erhält der Angreifer die sicherheitsrelevanten Informationen. Wie oben beschrieben, ist der beste Schutz eine gezielte Schulung. Entsprechende Hinweise in der Software, dass niemals Passwörter oder andere sicherheitsrelevante Informationen herausgegeben werden dürfen, sind zusätzliche Hilfen.

Brute-Force[10] beschreibt eine Methode, die sich auf das simple Ausprobieren aller Möglichkeiten stützt. Wird ein Passwort für den Zugang zu einem System genutzt, kann ein Angreifer versuchen alle möglichen Schlüssel auszuprobieren, um Zugriff auf das System zu erhalten oder Daten zu entschlüsseln. Diese Methode lässt sich verfeinern indem sogenannte Wörterbücher verwendet werden. Es wird versucht anhand oft verwendeter Wörter und Zahlenkombinationen das Passwort zu erraten. Generell ist die Brute-Force Methode immer anwendbar. Ein sinnvoller Schutz ist es eine hohe Anzahl von zu ratenden Möglichkeiten zu erzwingen oder systemseitig Latenzzeiten nach falschen Passworteingaben zu implementieren. Wird zum Beispiel eine bestimmte Mindestpasswortlänge erzwungen, ist das Erraten dieses Passwortes[11] schwieriger.

Wird eine unsichere Verschlüsselungsmethode verwendet, ist **das Ausnutzen von Schwachstellen** ein weiteres Sicherheitsproblem. Im Wesentlichen gibt es drei Angriffsarten auf Kryptographische Funktionen. Eine sichere kryptographische Funktion muss diesen Angriffsarten standhalten. Dies sind die folgenden Möglichkeiten:

1.) Ciphertext-Only-Attacke
2.) Known-Plaintext-Attacke
3.) Chosen-Plaintext-Attacke

Die **Ciphertext-Only-Attacke** beschreibt die Möglichkeit nur aus dem Chiffretext den Klartext zu erhalten. Ist zum Beispiel der Schlüsselraum zu klein, lassen sich einfach alle Schlüssel durchprobieren und man erhält nur aus dem Chiffretext den Klartext.

Die **Known-Plaintext-Attacke** beschreibt Verfahren, die aus mehreren Klartext-/Chiffretext-Paaren den Schlüssel bestimmen können.

Die **Chosen-Plaintext-Attacke** ähnelt der Known-Plaintext-Attacke, nur dass hier der Angreifer in der Lage ist, die Klartext/Chiffretext-Paare selber zu erzeugen. Durch geschickte Auswahl der Paare soll der Schlüssel herausgefunden werden.

Zusammenfassend ist es wichtig, dass der Schlüsselraum möglichst groß ist und dass die kryptographischen Funktionen nicht deterministisch sind. Dies bedeutet, dass aus Klartext/Chiffretext-Paaren nicht der Schlüssel bestimmbar ist.

[10] Übersetzt: „rohe Gewalt“

[11] Weitere Möglichkeiten zur Erhöhung der Passwortsicherheit sind in *Tabelle 6: Sichere Passwörter* zu finden.

2.4 Sichere Kommunikationsprotokolle

In diesem Kapitel werden vornehmlich zwei Protokolle vorgestellt, die momentan für eine sichere Datenübertragung verwendet werden. Diese Protokolle müssen laut dem Bundesamt für Sicherheit in der Informationstechnik[12] folgende Anforderungen erfüllen:

- **Vertraulichkeit der übertragenen Daten:**
  Selbst wenn die Daten abgehört werden, darf kein Rückschluss auf den Dateninhalt möglich sein. Dies kann durch entsprechende Verschlüsselung[13] der Daten geschehen.
- **Integrität der übertragenen Daten:**
  Zufällige Datenänderungen während der Übertragung müssen erkannt und behoben werden. Absichtliche Manipulationen der Daten müssen ebenfalls erkannt werden. Dies lässt sich durch entsprechende Signatur- und Verschlüsselungsverfahren erreichen.
- **Verfügbarkeit der Datenübertragung:**
  Ein hinreichend redundant ausgelegter Kommunikationsweg stellt die Verfügbarkeit der Datenübertragung sicher. Empfohlen wird die Nutzung des öffentlichen Kommunikationsnetzes, da dieses viele redundante Kommunikationswege bietet. Über die redundante Auslegung der Anbindung zu diesem Netz kann je nach Relevanz der Datenübertragung nachgedacht werden.
- **Authentizität der Daten:**
  Bei Kommunikation zwischen zwei Nutzern muss die Identität der beiden Nutzer sichergestellt werden, so dass sich niemand anderes als einer der Kommunikationspartner ausgeben kann. Dies lässt sich durch entsprechende Authentifikationsverfahren gewährleisten.
- **Nachvollziehbarkeit der Datenübertragung:**
  Um eine Nachvollziehbarkeit der Datenübertragung zu erlangen, kann die Übertragung protokolliert werden. So kann später festgestellt werden wer Daten versendet hat, wer diese empfangen hat und wann dies geschehen ist.
- **Sicherstellen des Datenempfanges:**
  Für einige Übertragungen kann es relevant sein, dass die Daten sicher und korrekt angekommen sind. In diesem Fall kann dies durch entsprechende Quittungsmechanismen sichergestellt werden.

Wie stark die entsprechenden Mechanismen zum Erreichen dieser Anforderungen sein müssen, richtet sich nach den Anforderungen und dem Schutzbedarf der zu kommunizierenden Daten. Die **Verfügbarkeit der Datenübertragung** ist durch entsprechende redun-

12

[5]https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m05/m05051.html

13 Siehe Kapitel 2.2 Kryptographische Verfahren

dante Hardware zu gewährleisten. Die **Nachvollziehbarkeit der Datenübertragung** kann durch entsprechendes Loggen der Datenübertragung geschehen. Die **Sicherung des Datenempfanges** kann durch ein entsprechendes Übertragungsprotokoll wie zum Beispiel TCP[14] gewährleistet werden. Die übrigen Anforderungen können von entsprechenden Protokollen erfüllt werden.
In den folgenden Abschnitten werden zwei Protokolle, die die restlichen Anforderungen erfüllen, vorgestellt. Diese nennen sich TLS und IPSec.

2.4.1 Transport Layer Security[15]

Transport Layer Security, im Folgenden TLS genannt, ist eine Weiterentwicklung von SSL. SSL bedeutet Secure Socket Layer. Zu beachten ist, wie TLS dafür sorgt, dass die oben genannten Anforderungen erfüllt werden. Hierzu wird das Handshake Protokoll[16] von TLS gezeigt und erläutert.

Tabelle 1: TLS Handshake-Protokoll

Nr.	Client Server	Nachricht
1	→	ClientHello
2	←	ServerHello Certificate (optional) ServerKeyExchange (optional) CertificateRequest (optional) ServerHelloDone
3	→	Certificate (optional) ClientKeyExchange CertificateVerify (optional) ChangeCipherSpec Finished
4	←	ChangeCipherSpec Finished
5	↔	Anwendungsdaten

Die Schritte des Handshake-Protokolls in *Tabelle 1: TLS Handshake-Protokoll* werden genauer erläutert um zu verstehen welche Informationen genau ausgetauscht werden und wie dadurch die Anforderungen für sichere Kommunikation erfüllt werden.

[14] Transmission Control Protocol
[15] [3, p. 801 ff.] Eckert, Claudia: IT-Sicherheit, Oldenbourg 2013
[16] Handshake-Protokoll ist der Teil des Kommunikationsprotokolls in dem die Verbindung aufgebaut wird.

1. **ClientHello:**
   Die ClientHello Nachricht ist eine Klartextnachricht, die folgende Informationen enthält:
   Die enthaltene Datenstruktur $\mathbf{R_C}$ besteht aus einem 32-Bit Zeitstempel und einer 28-Byte Zufallszahl. Zusätzlich sind ein **Sitzungsidentifikator** und eine **Cipher-Suite** enthalten. Die Cipher-Suite enthält die Kombinationen aus Verschlüsselungs- und Hashverfahren, die der Client unterstützt. Die Cipher-Suites sind in TLS vordefiniert, so dass nicht jede Kombination aus Verschlüsselungs- und Hashverfahren möglich ist. Wenn der Sitzungsidentifikator eine Sitzungsnummer enthält, soll eine alte oder bestehende Sitzung wiederaufgenommen werden. Ist dies nicht der Fall, wird eine neue Sitzung angelegt. Die Zufallszahl in $\mathbf{R_C}$ dient als Schutz vor Replay Attacken[17] beim Schlüsselaustausch.
2. **ServerHello:**
   Das ServerHello ist dem ClientHello sehr ähnlich. Es enthält ebenfalls eine Datenstruktur $\mathbf{R_S}$ mit Zeitstempel und Zufallszahl, eine Liste mit **Cipher-Suites** und einem **Sitzungsidentifikator**. Die Cipher-Suite besteht aus den Verschlüsselungs- und Hashverfahren, die vom Client und vom Server unterstützt werden. Sie wird vorher mit der Cipher-Suite des Clients abgeglichen. Die Suite an erster Stelle der Liste hat die höchste Priorität und wird später verwendet. Soll eine bestehende Sitzung fortgesetzt werden, ist also der Sitzungsidentifikator des Clients ungleich Null, sucht der Server diesen in seinem Sitzungscache. Wird der Sitzungsidentifikator gefunden, antwortet der Server mit dem gleichen Identifikator. Ansonsten wird hier Null zurück gesendet. Dies sagt dem Client, dass eine neue Sitzung aufgebaut werden muss.
   Certificate (optional) und CertificateRequest (optional):
   Sollen Client und/oder Server authentifiziert werden, müssen Client und Server Zertifikate austauschen. Die Form der Zertifikate ist in der Certificate und CertificateRequest Nachricht enthalten.
   ServerExchangeKey (optional):
   Werden keine Zertifikate ausgetauscht schickt der Server dem Client einen temporären öffentlichen Schlüssel[18] für den späteren Austausch des symmetrischen Schlüssels[19].
   ServerHelloDone:
   Mit dieser Nachricht signalisiert der Server dem Client, dass die ServerHello Nachricht abgeschlossen ist.
3. **Certificate (optional) und CertificateVerify (optional)**
   Hat der Server ein Zertifikat angefordert, wird dies gesendet. Ebenfalls wird das optional angeforderte Zertifikat vom Server überprüft.
   ClientKeyExchange

[17] Siehe Kapitel 2.3 Angriffstechniken
[18] Siehe Kapitel 2.2.1.2 Asymmetrische Verfahren
[19] Siehe Kapitel 2.2.1.1 Symmetrische Verfahren

Diese Nachricht enthält ein 48-Byte **Pre-Master Secret**, welches mit dem öffentlichen Schlüssel des Servers verschlüsselt ist.
Client und Server können nun aus dem Pre-Master Secret und den Zufallszahlen aus R_C und R_S das 48-byte **Master Secret** errechnen. Aus dem Master Secret lassen sich die Schlüssel zum späteren Verschlüsseln der Kommunikation ableiten.
ChangeCipherSpec und Finished
Diese Nachricht zeigt an, dass ab jetzt die ausgehandelte Cipher-Suite zur Kommunikation verwendet wird.

4. **ChangeCipherSpec und Finished**
   Der Server bestätigt, dass ab jetzt mit der aushandelten Cipher-Suite verschlüsselt kommuniziert wird.
5. **Anwendungsdaten**
   Verschlüsselter Austausch der Anwendungsdaten.

Die Verschlüsselung dient dem Erhalt der **Vertraulichkeit** der Daten. Die **Integrität** der Daten wird dadurch gewährleistet, dass die zu sendende Nachricht gehasht und dieser Hashwert zusammen mit der Nachricht verschlüsselt wird. Die verwendete Hashfunktion muss kryptographisch[20] sicher sein. So kann der Empfänger später anhand des Hashwertes die Integrität der Daten prüfen. Die **Authentizität** der Daten wird durch den Austausch der Zertifikate sichergestellt, da Client und Server durch die Zertifikate authentifiziert werden. Wenn Client und Server authentifiziert sind, nennt sich dies beidseitige Authentifikation. So können alle Anforderungen an eine sichere Kommunikation erfüllt werden.

[20] Siehe Kapitel 2.2.1 Kryptographische Hashverfahren

3 Anforderungsanalyse

Folgendes Kapitel befasst sich mit der Analyse der in der Einleitung beschriebenen Problemstellung. Um die späteren Lösungsmöglichkeiten möglichst objektiv bewerten zu können, wird eine Nutzwertanalyse vorgenommen. Die einzelnen Teilaspekte werden genauer herausgearbeitet und Probleme aufgezeigt. Das herausgearbeitete Zielsystem dient als allgemeine Bewertungsgrundlage für spätere Lösungsmöglichkeiten. Dies geschieht unter der Berücksichtigung der drei wesentlichen Schutzziele der IT-Sicherheit, der Authentizität, der Datenintegrität und der Informationsvertraulichkeit[21].
Zuerst muss das beschriebene Thema in Teilaspekte unterteilt werden, die dann separat bewertet werden können. Diese Teilaspekte stellen die Oberziele der Nutzwertanalyse dar.
Das Thema beschreibt eine sichere Dateiablage für ad hoc-Teams in heterogenen Netzwerken. Hieraus ergeben sich die Oberziele sichere Dateiablage, ad hoc-Teambildung und Unterstützung heterogener Netzwerke. Neben diesen drei Oberzielen gibt es noch einen weiteren Gesichtspunkt, der nicht außer Acht gelassen werden darf. Die Qualität des Zielsystems spielt ebenfalls eine wichtige Rolle. Daher wird die Qualität der Software ebenfalls in die Bewertung mit einfließen. Abschließend lassen sich folgende vier Oberziele definieren:

1.) Sichere Dateiablage
2.) Ad hoc-Teambildung
3.) Heterogene Netzwerke
4.) Softwarequalität

Die Oberziele werden mit unterschiedlichen Gewichtungen in die Bewertung eingehen. Die Summe der einzelnen Gewichtungen wird immer 100 ergeben, sodass Parallelen zur Prozentrechnung gegeben sind. Dies ermöglicht eine leicht verständliche und intuitive Einteilung der Gewichtungen. Das Hauptaugenmerk in dieser Arbeit beruht auf der IT-Sicherheit. Somit wird die sichere Dateiablage stärker gewichtet als die ad hoc-Teambildung und die heterogenen Netzwerke.

[21] [3, p. 7] Eckert, Claudia: IT-Sicherheit, Oldenbourg 2013

Tabelle 2: Gewichtung der Oberziele

Oberziel	Gewichtung
Sichere Dateiablage	40
Ad hoc-Teambildung	25
Heterogene Netzwerke	25
Softwarequalität	10

In *Tabelle 2: Gewichtung der Oberziele* ist die Einstufung der Oberziele zu sehen. Das Oberziel der sicheren Dateiablage geht fast doppelt so stark in die Bewertung mit ein, wie die ad hoc-Teambildung und die Unterstützung heterogener Netzwerke. Da die Softwarequalität zwar wichtig ist, aber nicht so speziell wie die anderen Oberziele, wird hier ein geringeres Gewichtung gewählt. So kann diese Gewichtung die Bewertung zwar beeinflussen, aber nicht maßgeblich entscheiden.

3.1 Anforderungen

In diesem Kapitel werden die oben genannten Oberziele genau definiert und es wird eine Bewertungsgrundlage, in Form eines Zielsystems, für Lösungsvorschläge geschaffen. So wird eine einheitliche Bewertung möglich. Inhalt der Bewertungsgrundlage sind die funktionalen und nicht funktionalen Anforderungen, sowie eine Gewichtung der Anforderungen.

3.1.1 Sichere Dateiablage

In diesem Abschnitt werden die Unterziele der sicheren Dateiablage definiert und gewichtet.
Eine sichere Dateiablage ist eine Kernanforderung für die Problemlösung.
Der Begriff der Sicherheit lässt sich auf zwei Arten definieren:

1. Angriffssicherheit
2. Betriebssicherheit

Angriffssicherheit entspricht der Sicherheit vor Fremdeinwirkung. Die Daten dürfen nur von Teammitgliedern im Klartext einsehbar sein und Änderungen von fremden Personen dürfen nicht unbemerkt geschehen. Werden Daten unautorisiert geändert, muss dies festgestellt und verhindert werden. Solche Änderungen dürfen nicht festgeschrieben werden.
Die **Betriebssicherheit** beinhaltet Mechanismen, die die Daten vor äußeren Umständen schützen, zum Beispiel einem Hardwaredefekt oder bei Bedienfehlern. Ein nicht gewolltes Löschen der Daten oder eine defekte Festplatte sollte nicht zum Verlust aller Daten führen. Dies geschieht im Allgemeinen durch Backupmechanismen.

Für die sichere Dateiablage werden folgende Schutzmaßnahmen in Bezug auf die Angriffssicherheit gefordert:

- Einsicht der Daten nur durch Teammitglieder möglich
- Änderungen sind Mitgliedern zuzuordnen

In Bezug auf die Betriebssicherheit gibt es folgende Forderungen:

- Schutz vor Hardwaredefekten durch Backups
- Rücksprung zu früherem Datenstand möglich

Hier wird gezeigt, dass die Anforderungen an die allgemeine Sicherheit stark mit den oben genannten Schutzzielen zusammenhängen. Die Authentizität wird gewährleistet, da nur Operationen durch Teammitglieder möglich sein dürfen. Die Datenintegrität wird dadurch erfüllt, dass Änderungen an den Daten nicht unautorisiert und unbemerkt geschehen dürfen. Die Informationsvertraulichkeit wird sichergestellt, da Daten nur von Teammitgliedern einsehbar sein dürfen.
Die sichere Dateiablage ist als komplette Einheit zu sehen. Es wird nicht nur gefordert, die Daten sicher zu speichern, auch der Übertragungsweg soll entsprechend gesichert sein. Eine Verschlüsselung der Daten zum frühestmöglichen Zeitpunkt wird angestrebt. Die Daten müssen nach dieser Maßgabe vor der Übertragung, also beim Client, verschlüsselt werden. Die Datenverschlüsselung alleine reicht nicht aus. Es muss sichergestellt werden, dass die Daten von einem Teammitglied mit den nötigen Rechten stammen und dass die Daten auch auf den richtigen Server übertragen werden. Eine beidseitige Authentifizierung ist also erforderlich. Ansonsten wäre die Anwendung einer Man-in-the-middle Attacke ohne weiteres möglich[22].
Die Übertragung der Daten sollte mithilfe eines anerkannten öffentlichen Protokolls geschehen, um eventuelle Sicherheitsprobleme so gut wie möglich zu vermeiden. Außerdem würde die Nutzung eines geheimen Protokolls gegen das Kerckhoffs-Prinzip[23] verstoßen.
Zusammengefasst soll die Datenübertragung folgende Merkmale aufweisen:

- Offenes anerkanntes Protokoll
- Unbefugte Änderungen werden erkannt und nicht festgeschrieben
- Beidseitige Authentifikation bei Datenübertragung
- Verschlüsselung clientseitig

Zu beachten ist, dass die Forderung nach Verwendung von öffentlichen anerkannten Protokollen ebenfalls für das Oberziel heterogener Netzwerke gilt. Da Unterziele unabhängig sein

[22] Siehe Kapitel 2.3 Angriffstechniken
[23] Siehe 2.1 Kerckhoffs-Prinzip

müssen, wird dieses nur in der Kategorie der heterogenen Netzwerke aufgeführt. Folgende Anforderungen sind an die sichere Dateiablage und die Übertragung der Daten zu stellen:

Tabelle 3: Anforderungen an eine sichere Dateiablage

Anforderungen für eine sichere Dateiablage
Einsicht der Daten nur durch Teammitglieder möglich
Änderungen sind Mitgliedern zuzuordnen
Schutz vor Hardwaredefekten
Rücksprung zu früherem Datenstand möglich

Tabelle 4: Anforderungen an eine sichere Dateiübertragung

Anforderungen für eine sichere Datenübertragung
Unbefugte Änderungen werden erkannt und nicht festgeschrieben
Beidseitige Authentifikation bei Datenübertragung
Verschlüsselung clientseitig

In *Tabelle 3: Anforderungen an eine sichere Dateiablage* und *Tabelle 4: Anforderungen an eine sichere Dateiübertragung* sind die Unterziele des Oberziels der sicheren Dateiablage zu sehen. Den Unterzielen werden unterschiedliche Gewichtungen zugeordnet, um deren unterschiedlich starken Nutzen in der Bewertung widerzuspiegeln. Wie oben beschrieben, wird die Summe der Gewichtungen 100 betragen.

Tabelle 5: Unterziele der sicheren Dateiablage

Unterziele für eine sichere Dateiablage	**Gewichtung**
Einsicht der Daten nur durch Teammitglieder möglich	20
Änderungen sind Mitgliedern zuzuordnen	10
Schutz vor Hardwaredefekten	10
Rücksprung zu früherem Datenstand möglich	10
Unbefugte Änderungen werden erkannt und nicht festgeschrieben	15
Beidseitige Authentifikation bei Datenübertragung	10
Verschlüsselung clientseitig	25

In *Tabelle 5: Unterziele der sicheren Dateiablage* ist zu sehen, dass durch die Gewichtung das Hauptaugenmerk auf der clientseitigen Verschlüsselung und dem Schutz des Dateninhaltes liegt.

3.1.2 Ad hoc-Teambildung

In diesem Absatz werden die Unterziele der ad hoc-Teambildung bestimmt und gewichtet. Ad hoc bedeutet umgangssprachlich „spontan“ oder „auf einen Schlag“. Diese Begriffe zeigen, dass es ohne großen Aufwand möglich sein muss ein Team zu bilden.

Im Allgemeinen ist der Sinn eines ad hoc-Teams eine einmalige Zusammenarbeit zur Erfüllung eines bestimmten Ziels oder bestimmter Ziele. Nach der Erfüllung dieser Ziele zerfällt das Team wieder und jeder Teilnehmer geht seiner ursprünglichen Tätigkeit nach. Es ist daher sinnvoll, wenn die Teambildung nicht von anderen Personen abhängt. Es ist unpraktisch einen Antrag beim Administrator stellen zu müssen, um dann später für das Team freigeschaltet zu werden. Sinnvoller ist ein Verantwortlicher oder Gründer in jedem Team, der die nötigen Rechte an Teammitglieder verteilen kann.
Die Rechteverwaltung im Team wird vom Teamgründer, oder von einer von ihm autorisierten Person, ausgeführt. Eine Nutzerverwaltung muss also Teil der Problemlösung sein.
Das Schutzziel der Authentizität fordert, dass die Authentizität der Nutzer geprüft werden muss, damit diese Zugang zu den geschützten Dokumenten erhalten. Eine hinreichende Authentifizierung der Nutzer ist gefordert. Hierfür gibt es verschiedene Möglichkeiten. Sehr verbreitet ist die Authentifizierung durch etwas, dass der Nutzer weiß (z.B. Passwörter) oder etwas, dass der Nutzer besitzt (z.B. Smartcards, TAN-Listen). Diese beiden Verfahren lassen sich auch kombinieren, wie es bei Banksystemen häufig der Fall ist. Am gebräuchlichsten ist derzeit das Passwortverfahren.
Wird ein Passwortverfahren angewendet, muss sichergestellt sein, dass das Passwort eine hinreichende Sicherheit aufweist. Laut Bundesamt für Sicherheit in der Informationstechnik sollte ein Passwort folgenden Anforderungen genügen[24]:

Tabelle 6: Sichere Passwörter

Kriterien für die Wahl eines sicheren Passwortes
Mindestens 8 Zeichen Länge
Verwendung von Groß- und Kleinbuchstaben sowie Sonderzeichen und Ziffern
Es sollte nicht im Wörterbuch vorkommen
Es sollte nicht aus Varianten oder Wiederholung von Tastaturmustern bestehen

Das Schutzziel der Datenintegrität fordert, dass es Nutzern nicht möglich ist, die zu schützenden Daten unautorisiert und unbemerkt zu ändern. Änderungen an den Daten des Teams müssen daher dem entsprechenden Teammitglied zuzuordnen sein.

Tabelle 7: Anforderungen an die ad hoc-Teambildung

Anforderungen an die ad hoc-Teambildung
Rechteverwaltung vom Teamgründer oder autorisierter Person
Erstellung eines Teams allein vom Teamgründer abhängig
Sichere Nutzeridentifikation und Authentifikation
Zuordnung der Datenänderung zu dem entsprechenden Teammitglied

Die ausgearbeiteten Anforderungen können nicht direkt in Unterziele umgewandelt und gewichtet werden. Es wird gefordert, dass Datenänderungen einem Teammitglied zugeord-

[24] [7]https://www.bsi-fuer-buerger.de/BSIFB/DE/MeinPC/Passwoerter/passwoerter_node.html

net werden müssen. Dieses Unterziel ist bereits in der sicheren Dateiablage aufgetaucht. Um eine falsche Gewichtung und somit Bewertung zu vermeiden, wird dieses Unterziel in diesem Bereich nicht in die Bewertung mit eingehen.
Im Bereich der ad hoc-Teambildung wird keines der Unterziele wichtiger als eines der anderen eingestuft. Sie sind alle gleich wichtig, daher erhalten diese die gleiche Gewichtung.

Tabelle 8: Unterziele für die ad hoc-Teambildung

Unterziele für die ad hoc-Teambildung	Gewichtung
Rechteverwaltung vom Teamgründer oder autorisierter Person	33
Erstellung eines Teams allein vom Teamgründer abhängig	33
Sichere Nutzeridentifikation und Authentifikation	33

In *Tabelle 8: Unterziele für die ad hoc-Teambildung* ist die Gewichtung der einzelnen Unterziele zu sehen.

3.1.3 Heterogene Netzwerke

In diesem Abschnitt werden die Unterziele des Oberziels Heterogene Netzwerke herausgearbeitet und gewichtet.
Gewünscht ist, eine sichere Kommunikation zwischen den Teammitgliedern, die den Schutzzielen der Vertraulichkeit, der Integrität und der Authentizität genügt.
Hierbei handelt es sich um verschiedene Netzwerke, die teilweise nicht direkt miteinander verbunden sind. Direkt bedeutet zum Beispiel über Netzwerkbrücken. Diese Netzwerke sind oft für eine Vielzahl verschiedener Personen zugänglich, so dass diese aus Gründen der Sicherheit stark abgeschottet sind. Dies bedeutet, es stehen oftmals nur wenige Ports und damit Möglichkeiten für eine Kommunikation zu den anderen Netzwerken offen. Im beschriebenen Szenario ist dies ebenfalls der Fall. Es gibt mehrere Netzwerke die nur über das Internet miteinander verbunden und stark abgeschottet sind. Es ist also nur möglich über Standardnetzwerkprotokolle, die den meisten Netzen zur Verfügung stehen sollten, zu kommunizieren. Selbsterstellte Kommunikationsprotokolle, die eventuell nicht öffentlich einsehbar sind, werden generell als nicht sicher eingestuft.
Die Vergangenheit hat gezeigt, dass neue Protokolle sehr oft Sicherheitsmängel haben. Ein nicht öffentlich einsehbares Protokoll, ist ein Verstoß gegen das Kerckhoffs-Prinzip[25] und wird ebenfalls als nicht sicher eingestuft. Aus den beschriebenen Problemen ergeben sich folgende Anforderungen:

Tabelle 9: Anforderungen für heterogene Netzwerke

Anforderungen für den Teilaspekt heterogene Netzwerke
Kommunikation mittels anerkannter Netzwerkprotokolle
Protokoll muss öffentlich einsehbar sein

[25] [3, p. 302] Eckert, Claudia: IT-Sicherheit, Oldenbourg 2013

Die in *Tabelle 9: Anforderungen für heterogene Netzwerke* aufgezeigten Anforderungen können nicht direkt für die Nutzwertanalyse verwendet werden. Es wird verlangt, dass die Unterziele nicht voneinander abhängen, da sonst eine ungewollte doppelte Gewichtung entstehen würde. Jedes anerkannte Netzwerkprotokoll ist öffentlich einsehbar und ein nicht öffentliches Protokoll wird generell als nicht sicher angesehen, daher werden die beiden Anforderungen zu einem Unterziel zusammengefasst.

Tabelle 10: Unterziele für heterogene Netzwerke

Unterziele für heterogene Netzwerke	Gewichtung
Kommunikation mittels anerkannter Netzwerkprotokolle	100

Tabelle 10: Unterziele für heterogene Netzwerke zeigt das Unterziel für die Nutzung heterogener Netzwerke. Um eine Kommunikation in verschiedenen Netzwerken zu gewährleisten, wird die Verwendung von anerkannten Netzwerkprotokollen gefordert.

3.1.4 Qualitätsanforderungen

Zusätzlich zu den Anforderungen, die sich direkt aus dem Szenario ableiten lassen, gibt es noch weitere wichtige Gesichtspunkte. Hierbei handelt es sich um so genannte nicht funktionale Anforderungen. Diese müssen bei der Lösungssuche und Bewertung ebenfalls berücksichtigt werden. In der ISO/IEC 9126 Norm[26] werden grundlegend sechs Qualitätsmerkmale definiert:

1. Funktionalität
2. Zuverlässigkeit
3. Benutzbarkeit
4. Effizienz
5. Wartbarkeit
6. Übertragbarkeit

Diese Qualitätsmerkmale dienen der Klassifikation der Softwarequalität. Die Begriffe können je nach Anwendung unterschiedlich ausgelegt werden. Ein Systementwickler hat andere Ansprüche an die Softwarequalität als der Kunde. Daher werden die Begriffe auf das Szenario angepasst, um ein einheitliches Verständnis zu gewährleisten.
Die **Funktionalität** der Software gibt an, inwieweit die aus der Analyse hervorgegangenen Anforderungen korrekt erfüllt werden. Eine Software, die nur die Hälfte der Anforderungen erfüllt ist also nicht funktional.
Die **Zuverlässigkeit** ist eine Kennzahl für die Fehlertoleranz und Wiederherstellbarkeit im Fehlerfalle. Wenn eine Software bei Falscheingaben durch den Nutzer abbricht, ist diese nicht zuverlässig.

[26] [8, p. 468 ff.]Balzert, Heide: Lehrbuch Der Softwaretechnik: Basiskonzepte Und Requirements Engineering 2009

Benutzbarkeit gibt Aufschluss über die Verständlichkeit und Erlernbarkeit der Software. Eine Software kann gut implementiert sein, aber nicht bedienbar. Die Bedienbarkeit der Software sollte soweit wie möglich intuitiv geschehen. Es sollte zum Beispiel darauf geachtet werden bekannte Symbole zu verwenden.
Die **Effizienz** beschreibt ein Verhältnis zwischen Leistung und eingesetzten Betriebsmitteln. Eine Software die alle Anforderungen schnell und gut erfüllt, aber einen unverhältnismäßigen Hardwareaufwand benötigt, ist nicht sehr effizient.
Die **Wartbarkeit** einer Software lässt sich beschreiben, indem analysiert wird, wie leicht Änderungen an den Anforderungen in der Software umzusetzen sind. Wenn zum Beispiel eine Schaltfläche in der Software eine andere Farbe erhalten soll und mehrere Tage programmieraufwand nötig sind, dann ist diese Software nicht gut wartbar.
Die **Übertragbarkeit** zeigt auf, wie leicht sich die Software auf eine andere Umgebung übertragen lässt.
Neben diesen Anforderungen an die Qualität der Software, spielen die **Kosten** für eine Softwarelösung eine wichtige, wenn nicht sogar die wichtigste, Rolle. Es ist daher essentiell diese ebenfalls in das Bewertungsschema mit aufzunehmen.
Folgende Anforderungen an die Qualität der Software lassen sich ableiten:

Tabelle 11: Anforderungen an die Softwarequalität

Qualitätsanforderungen
Funktionalität
Zuverlässigkeit
Benutzbarkeit
Effizienz
Wartbarkeit
Übertragbarkeit
Geringe Kosten

Diese herausgearbeiteten Qualitätsanforderungen lassen sich direkt in Unterziele für das Oberziel der Softwarequalität umwandeln und gewichten.

Tabelle 12: Unterziele für Softwarequalität

Unterziele für Softwarequalität	**Gewichtung**
Funktionalität	10
Zuverlässigkeit	10
Benutzbarkeit	10
Effizienz	10
Wartbarkeit	10
Übertragbarkeit	10
Geringe Kosten	40

In *Tabelle 12: Unterziele für Softwarequalität* wird gezeigt, dass möglichst geringe Kosten am wichtigsten in diesem Bereich sind. Die restlichen Unterziele erhalten eine gleichwertige Einstufung, da eine qualitativ hochwertige Software alle diese Kriterien erfüllen muss.

3.2 Technische Spezifikation der Anforderungen

Nachdem die Oberziele und Unterziele erstellt worden sind und somit das Zielsystem feststeht, muss festgelegt werden, wie die einzelnen Unterziele zu erfüllen sind.
Um die verschiedenen Anforderungen bewerten zu können, muss aufgezeigt werden, wann diese als erfüllt gelten. Daher werden im folgenden Abschnitt die nicht technischen Anforderungen auf ein technisches Niveau gebracht, um diese dann klar bewerten zu können. Eine Einteilung der Kriterien in Muss- und Kann-Kriterien ist ebenfalls sinnvoll, da nicht alle Kriterien gleich wichtig für die Aufgabenstellung sind.

3.2.1 Sichere Dateiablage

In diesem Abschnitt werden die in *Tabelle 5: Unterziele der sicheren Dateiablage* aufgeführten Kriterien genauer betrachtet, um diese in konkrete technische Anforderungen umzuwandeln. Des Weiteren wird eine Einteilung in Muss- und Kann-Kriterien vorgenommen.
Als Erstes wird gefordert, dass eine **Einsicht der Daten nur durch Teammitglieder** geschehen darf. Zuerst bedeutet dies, dass die Daten vor fremden Zugriff geschützt werden müssen. Es bedeutet aber auch, dass ein Teammitglied entsprechend als dieses authentifiziert werden muss. Da die Authentifizierung der Teammitglieder im Kapitel 3.2.2 genauer betrachtet wird, wird in diesem Zusammenhang der Schutz vor fremden Zugriff beleuchtet. Im Kontext der sicheren Dateiablage ist die Frage zu klären, wie die Daten so abgelegt werden, dass niemand außer dem Team die Daten einsehen kann. Dies schließt den Besitzer des Datenspeichers mit ein, der im Zeitalter der Vernetzung nicht zwangsläufig Teil des Teams ist. Die Verschlüsselung der Daten muss zum frühestmöglichen Zeitpunkt geschehen. Für das beschriebene Szenario bedeutet dies eine Verschlüsselung der Daten, bevor diese übertragen werden. Die Bundesnetzagentur[27] bringt jedes Jahr einen Algorithmenkatalog[28] heraus, in dem entsprechende Algorithmen für Signier-, Hash- und Verschlüsselungsverfahren[29] empfohlen werden. Anhand dieses Kataloges werden die Anforderungen für die Datenverschlüsselung auf dem Speichermedium festgelegt. RSA und DSA[29] Verfahren werden mit entsprechenden Schlüssellängen als sicher eingestuft. Die empfohlenen Schlüssellängen werden für den Zeitraum bis Ende 2019 in der folgenden Tabelle dargestellt:

[27] [22] http://www.bundesnetzagentur.de
[28] [9] Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung, Bundesnetzagentur
[29] Siehe Kapitel 2.2 Kryptographische Verfahren

Tabelle 13: Schlüssellängen

Verfahren	Parameter	Schlüssellänge in Bit
RSA	n	(mindestens) 1976 (empfohlen) 2048
DSA	p/q	2048/256
AES	n	256

Aus *Tabelle 13: Schlüssellängen* ergibt sich, dass für die Dateiverschlüsselung ein RSA oder DSA Verfahren unter Berücksichtigung der in der Tabelle angegebenen Schlüssellängen zu verwenden ist. Die Verschlüsselung soll automatisch für jede Datei erfolgen. Für jede Datei soll ein eigener Schlüssel zur Verschlüsselung verwendet werden. Dies garantiert maximale Sicherheit, auch wenn einzelne Schlüssel bekannt werden sollten. Die einzelnen Schlüssel für die Daten sollten von der Software verwaltet werden, so dass diese automatisch einem korrekt authentifizierten und autorisierten Nutzer zur Verfügung gestellt werden, um angefragte Daten zu entschlüsseln. Dies ist ein Muss-Kriterium.
Weiterhin müssen **Datenänderungen dem Verursacher** zugeordnet werden können. Hierfür muss der Verursacher authentifiziert[30] sein. Um die Datenänderungen zu verfolgen, ist ein System erforderlich, welches den Nutzer, die Änderung der Daten und den Zeitpunkt notiert. Dies kann durch eine einfache Log Datei oder besser durch ein Log System geschehen, welches ein Durchsuchen der aufgezeichneten Daten erlaubt und diese sinnvoll grafisch darstellt. Dieses Änderungslogging ist ein Kann-Kriterium.
Ein **Schutz vor Hardwaredefekten** wird gefordert. In Hinblick auf die Dateiablage bedeutet dies, dass die Daten nicht nur von einem Speichermedium abhängig sein dürfen. Eine hinreichende Backupstrategie ist hierfür erforderlich. Die Sicherung der Daten sollte auf unabhängigen Datenträgern erfolgen. Eine Backupstrategie muss allerdings nicht von der Software alleine abhängen. Viele Serverbetreiber haben bereits ein Backup der Daten auf Betriebssystemebene eingerichtet. Eine Datensicherung direkt in der Software hat den Vorteil, dass Daten mit weniger Aufwand wiederhergestellt werden können. Aufgrund der einfachen Integrierbarkeit in bestehende Backuplösungen wird dies als Kann-Kriterium gewertet.
Zusätzlich zum Schutz vor Hardwaredefekten soll ein **Rücksprung zu früheren Datenständen möglich** sein. Alle gängigen Backuplösungen, die den Ansprüchen zum Schutz der Hardware genügen, bieten diese Möglichkeit an. Wenn dies nicht möglich wäre, wäre ein Backup wenig sinnvoll. Eine integrierte Lösung ist hier besser, da die Wiederherstellung ohne großen Aufwand erfolgen kann. Da dies ebenfalls ohne Probleme außerhalb der Software zu lösen ist, ist dies ein Kann-Kriterium.
Unbefugte Änderungen an den Daten bei der Datenübertragung werden erkannt und nicht festgeschrieben. Dies wird in der Regel von einem geeigneten Datenübertragungsprotokoll übernommen. Die oben genannten Protokolle IPSec und TLS/SSL[31] verfügen über Schutzmechanismen, die die Datenintegrität sicherstellen. Werden diese Protokolle nicht

[30] Siehe Kapitel 3.2.2 Ad hoc-Teambildung
[31] Siehe Kapitel 2.4.1 Transport Layer Security

genutzt, können geeignete Signaturverfahren[32] genutzt werden um Daten auf ihre Integrität und Authentizität zu Prüfen. Diese Forderung ist sicherheitskritisch und daher ein Muss-Kriterium.
Eine **beidseitige Authentifikation bei Datenübertragung** ist gefordert. Diese wird durch geeignete Übertragungsprotokolle wie TLS und IPSec bereits unterstützt. Diese Maßnahme dient dazu einen Man-in-the-middle[33]“ Angriff vorzubeugen. Hierbei handelt es sich ebenfalls um ein Muss-Kriterium.
Die **Clientseitige Verschlüsselung** entspricht einer frühestmöglichen Verschlüsselung der Daten, so wird eine hohe Vertraulichkeit der Daten gewährleistet. Im Wesentlichen entsprechen die Anforderungen an die Clientseitige Verschlüsselung den Anforderungen, die an die Einsicht der Daten am Anfang dieses Kapitels gestellt werden. Empfohlene Algorithmen mit entsprechenden Schlüssellängen sind *Tabelle 13: Schlüssellängen* zu entnehmen. Dies ist ein Muss-Kriterium.
In *Tabelle 14: Technische Anforderungen für eine sichere Dateiablage* sind alle Informationen noch einmal übersichtlich zusammengefasst.

Tabelle 14: Technische Anforderungen für eine sichere Dateiablage

Anforderungen für eine sichere Dateiablage	**Technische Umsetzung**	**Muss - Kriterium**
Einsicht der Daten nur durch Teammitglieder möglich	Verschlüsselung mittels DSA oder RSA mit geeigneten Schlüssellängen[34].	Ja
Änderungen sind Mitgliedern zuzuordnen	Geeignetes Log System, welches die Änderungen der Nutzer speichert und lesbar darstellt.	Nein
Schutz vor Hardwaredefekten	Backup auf anderen Speichermedien	Nein
Rücksprung zu früherem Datenstand möglich	Speicherung verschiedener Datenversionen mit Rücksprungmöglichkeit	Nein
Unbefugte Änderungen werden erkannt und nicht festgeschrieben	Verwendung von TLS oder IPSec. Ansonsten Daten signiert übertragen, so dass eine Überprüfung der Daten möglich ist.	Ja
Beidseitige Authentifikation bei Datenübertragung	Sowohl Client als auch Server müssen sich authentifizieren. In IPSec oder TLS bereits genutzt.	Ja
Verschlüsselung clientseitig	Verschlüsselung der Daten vor der Übertragung zum Server	Ja

[32] Siehe Kapitel 1.1.1 Signaturverfahren
[33] Siehe Kapitel 2.3 Angriffsverfahren

3.2.2 Ad hoc-Teambildung

In diesem Abschnitt werden die in *Tabelle 8: Unterziele für die ad hoc-Teambildung* aufgeführten Kriterien genauer betrachtet und in konkrete technische Anforderungen überführt. Anschließend wird eine Einteilung in Muss- und Kann-Anforderungen vorgenommen, um eine bessere Bewertung zu ermöglichen.

Es wird gefordert, dass **die Rechteverwaltung vom Teamgründer oder einer autorisierten Person ausführbar ist**. Die Lösung muss daher über einstellbare Zugriffsrechte verfügen und diese müssen zentral von einer oder mehreren Personen ausführbar sein. Sinnvoll ist in dieser Problemstellung eine rollenbasierte Rechtevergabe. Es wäre möglich die Rolle des Teamleaders und die Rolle des Teammitglieds einzuführen, um dann den Rollen gezielt die richtigen Rechte zu geben. Hier gilt der Leitsatz: So wenig Rechte wie möglich, so viele Rechte wie nötig. Dies ist ein Kann-Kriterium

Das Einrichten des Teams soll allein vom Teamgründer abhängen. Es soll nicht nötig sein andere Personen (zum Beispiel die Administratoren der Rechner oder Netzwerke) zu konsultieren, um ein Team zu erstellen. Die oben aufgeführte Rechtevergabe soll durch die Software und nicht durch das System gelöst werden. Dies ist ein Kann-Kriterium.

Die **sichere Nutzeridentifikation und Authentifikation** soll durch die Software erzwungen werden. Eine gute Lösung ist die Zwei-Faktor-Authentifikation. Diese nutzt neben der bekannten Nutzername/Passwort Authentifikation ein weiteres Element. Dies kann, wie bei Banksystemen heutzutage üblich, ein Code sein, der auf das Mobiltelefon gesendet wird. Möglich ist dies auch über spezielle Hardware. Ein Beispiel ist in diesem Zusammenhang ein spezieller USB-Stick, der eingesteckt sein muss, um sich anzumelden. Damit dieses Verfahren sicherer als eine normale Authentifikation über Passwort ist, muss sichergestellt sein, dass die beiden Authentifikationsstufen getrennt voneinander sind. Ist zum Beispiel ein mobiler Login über das Telefon möglich und zur Authentifikation wird eine per SMS gesendete TAN Nummer verwendet, ist die Methode unsicher. Wird das Telefon gestohlen muss nur noch das Passwort überwunden werden und das Verfahren ist genauso sicher wie eine normale Authentifikation über ein Passwort.

Daher ist es wichtig, dass sichere Passwörter genutzt werden. Die Kriterien für sichere Passwörter sind in *Tabelle 6: Sichere* Passwörter nachzulesen.

Eine Zwei-Faktor-Authentifikation ist, wenn zusätzliche Hardware benötigt wird, oftmals kostenintensiv. Daher wird diese als Kann-Kriterium angesehen. Als besonders sicherheitsrelevant wird die Nutzung eines sicheren Passwortes nach den oben genannten Kriterien angesehen. Oft wird für die Accounterstellung eine E-Mail Adresse benötigt. Wenn dies der Fall ist, sollte diese authentifiziert werden. Normalerweise geschieht dies über einen Link, der an die eingegebene E-Mail Adresse gesendet wird. Der Nutzer beweist durch Aufrufen des Links, dass er Zugriff auf diese E-Mail Adresse hat. Dies ist ein Muss-Kriterium.

Tabelle 15: Technische Anforderungen an die ad hoc-Teambildung

Anforderungen an die ad hoc-Teambildung	Technische Umsetzung	Muss-Kriterium
Rechteverwaltung vom Teamgründer oder autorisierter Person	Rollenbasierte Rechtevergabe	Nein
Erstellung eines Teams allein vom Teamgründer abhängig	Rechtevergabe und Teamerstellung in der Software und nicht im System einstellbar	Nein
Sichere Nutzeridentifikation und Authentifikation	Verwendung sicherer Passwörter oder einer zwei-Faktor-Authentifikation	Ja

Tabelle 15: Technische Anforderungen an die ad hoc-Teambildung zeigt übersichtlich die Anforderungen an die ad hoc-Teambildung.

3.2.3 Heterogene Netzwerke

Im folgenden Abschnitt werden die in *Tabelle 10: Unterziele für heterogene Netzwerke* genannten Forderungen betrachtet und in konkrete technische Anforderungen überführt.
Gefordert wird eine **Netzwerkkommunikation durch anerkannte Netzwerkprotokolle.** Dies ist wichtig, da nur durch Erfahrung geprüfte Protokolle maximale Sicherheit in der Übertragung garantieren. Des Weiteren wird die Kompatibilität der Software durch Beachtung von Standards gefördert. Das Prinzip von Kerckhoffs[35] fordert, dass die Sicherheit einer Verschlüsselung nur von der Geheimhaltung des Schlüssels und nicht von der Geheimhaltung des Verfahrens abhängen darf. Dies lässt sich auf Netzwerkprotokolle erweitern. Ein Protokoll muss sicher sein, auch wenn dieses öffentlich bekannt ist, nur dann kann es als sicher gelten. Daraus wird gefolgert, dass die Veröffentlichung eines Protokolls nur zu deren Sicherheit beitragen kann. Momentan wird standartmäßig für sichere Kommunikation TLS/SSL verwendet. Dies ist unter anderem dem Maßnamenkatalog des BSI zur Sicherung der Kommunikation[36] zu entnehmen. Diese Protokolle werden unter der Betrachtung der Schutzziele als sicher angesehen.

[35] Siehe Kapitel 2.1 Kerckhoffs-Prinzip

[36] [10]https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m05/m05066.html

Tabelle 16: Technische Anforderungen an heterogene Netzwerke

Anforderungen für das Oberziel heterogene Netzwerke	Technische Umsetzung	Muss-Kriterium
Kommunikation mittels anerkannter Netzwerkprotokolle	Nutzung von Standartprotokollen zur Kommunikation wie z.B. IPSec, SSL/TLS	Ja

Tabelle 16: Technische Anforderungen an heterogene Netzwerke fasst die Anforderungen an die Verwendung heterogener Netzwerke zusammen.

3.2.4 Qualitätsanforderungen

Die technische Spezifikation der in *Tabelle 12: Unterziele für Softwarequalität* genannten Unterziele gestaltet sich anders, als in den vorherigen Kapiteln. Die genannten Gesichtspunkte der Softwarequalität können auf vielen, sehr unterschiedlichen, Wegen erreicht werden. Daher werden in diesem Bereich nicht direkte technische Angaben zur Erfüllung der Unterziele vorgegeben, sondern objektivere Kriterien vorgegeben. Da es um die Bewertung der Softwarequalität und nicht um die Bewertung von geforderten Funktionen geht, sind alle Unterziele in diesem Kapitel Kann-Kriterien.

Das Unterziel mit der höchsten Gewichtung sind die **Kosten**. Je höher die Kosten ausfallen, desto schlechter die Note. Bewertet werden hier nur Kosten, die durch die Nutzung des Produktes entstehen, also Anschaffungskosten und Betriebskosten. Oft stehen verschiedene Preismodelle zur Auswahl, hier wird das günstigste passende Preismodell gewählt.

Als Nächstes wird die **Funktionalität** der Software bewertet. Für dieses Unterziel wird bewertet inwieweit die vom Hersteller versprochenen Leistungen funktionieren. Eine gute Note kann in diesem Punkt erreicht werden, wenn es keine Unterschiede zwischen beworbenen und tatsächlichen Funktionen gibt. Sollten versprochene Funktionen fehlen wird eine entsprechend schlechtere Note vergeben.

In Hinblick auf die **Zuverlässigkeit** der Software wird geprüft, ob bei der Nutzung der Software ungewollte Fehler auftreten. Um dies zu testen, wird ein bestimmter Anwendungsablauf ausprobiert. Dieser sieht wie folgt aus:

Tabelle 17: Anwendungsfall Qualitätstest

Name	Qualitätstest 1
Beschreibung	Standard Ablaufprotokoll zum Testen der Softwarequalität
Vorbedingung	-
Nachbedingung	Nutzer 1 und Nutzer 2 haben den selben Datenbestand
Ablauf	1.) Einrichten der Software 2.) Erstellen von zwei Accounts 3.) Nutzer 1 stellt Nutzer 2 eine Datei zur Verfügung 4.) Nutzer 2 bearbeitet diese 5.) Nutzer 1 überprüft die Änderungen
Hinweise	Sollte ein Punkt nicht ausgeführt werden können oder müssen, wird er übersprungen.

Es wird ein Anwendungsfall genommen, der im angegebenen Szenario häufig vorkommen wird und daher auf jeden Fall fehlerfrei funktionieren muss.
Die **Benutzbarkeit** der Software gibt an, wie gut die Software bedienbar ist. Es wird angenommen, dass eine gut bedienbare Software die gewünschten Aufgaben schnell und mit möglichst wenig Aufwand ausführt. Um ein Maß zur Bewertung der äußerst subjektiven Benutzbarkeit der Software zu finden, wird gemessen wie viel Zeit zum Erfüllen des Qualitätstests gebraucht wird. Je weniger Zeit benötigt wird, desto besser die Note. Sollten Probleme mit der Bedienung der Software auftreten, wird dies ebenfalls negativ in die Note einbezogen. Dies wird im Test entsprechend vermerkt. Es wird auch darauf geachtet, ob das „look and feel“ der Software einheitlich ist. Zum Beispiel ist eine Software leichter zu bedienen, die ein einheitliches Design hat. Schaltflächen mit einheitlichen Funktionen sollten zum Beispiel immer an derselben Stelle der Benutzeroberfläche zu finden sein.
Die **Effizienz** der Software kann auf verschiedene Arten bestimmt werden. In diesem Szenario wird hierfür die benötigte Hardware als Vergleichsoption gewählt. Es wird vorausgesetzt, dass jedes Teammitglied einen PC zur Verfügung hat. Darüber hinaus wird alles an zusätzlich benötigter Hardware negativ in die Benotung mit eingehen. Ein weiterer negativer Aspekt wird sein, ob zum Beispiel ein Server gebraucht wird und ob dieser selber betrieben werden kann oder nicht.
Als Messlatte für die **Wartbarkeit** der Software wird bewertet wie einfach es ist neue Softwareversionen einzupflegen. Muss der Nutzer hier selber Hand anlegen oder geschieht dies automatisch? Je einfacher dieser Prozess ist, desto besser die Note. Falls die Software vom Hersteller nicht weiter gepflegt wird, geht dies mit einer Bemerkung ebenfalls schlecht in die Benotung mit ein.
Für die **Übertragbarkeit** der Software wird als Maß die Kompatibilität zu verschiedenen Systemen gewählt. Es wird also bewertet, welche unterschiedlichen Systeme von der Software unterstützt werden. Versionen für die gängigen PC Betriebssysteme sind gefordert. Dies beinhaltet Windows, Linux und Mac. Eine Unterstützung von mobilen Geräten wirkt sich zusätzlich positiv auf die Bewertung aus, führt aber zu keinen Abzügen, wenn diese

nicht vorhanden sind. Es wird davon ausgegangen, dass die Teammitglieder hauptsächlich an einem PC oder Laptop arbeiten. Je besser die Kompatibilität, desto besser die Benotung.
Aus den oben gewonnenen Erkenntnissen lässt sich zusammenfassend folgende Bewertungstabelle erstellen:

Tabelle 18: Spezifikation der Softwarequalität

Anforderungen an die Qualität	**Umsetzung**	**Muss-Kriterium**
Kosten	Möglichst geringe Kosten	Nein
Funktionalität	Überprüfen, ob angegebene Funktionen mit getesteten Funktionen übereinstimmen	Nein
Zuverlässigkeit	Fehlerfreies Ausführen vom Anwendungsfall Qualitätstest möglich	Nein
Benutzbarkeit	Messung der Zeit, die zum Ausführen vom Anwendungsfall Qualitätstest gebraucht wird. Überprüfen des „look and feel“	Nein
Effizienz	Vergleich der benötigten zusätzlichen Hardware	Nein
Wartbarkeit	Überprüfen wie Softwareupdates eingepflegt werden und ob Software vom Hersteller gepflegt wird	Nein
Übertragbarkeit	Bietet die Software Versionen für die gängigen verschiedenen Systemplattformen	Nein

3.3 Fazit Anforderungsanalyse

In den Kapiteln 3.1 Anforderungen und 3.2 Technische Spezifikation der Anforderungen wurde ein Zielsystem für eine Nutzwertanalyse festgelegt. Des Weiteren wurde spezifiziert in welcher Form die einzelnen Unterziele erfüllt werden müssen und mit welcher Gewichtung diese dann in die Bewertung eingehen. Jedes Unterkapitel enthält abschließend eine Tabelle mit den entsprechenden Informationen. Diese werden hier noch einmal zusammengefasst dargestellt, so dass klar ist, was getestet wird.
In *Tabelle 19: Zielsystem* sind alle Kriterien enthalten, die zur Bewertung der Lösungsmöglichkeiten genutzt werden. Auf dieser Basis wird im folgenden Kapitel die Bewertung vorgenommen.

Tabelle 19: Zielsystem

Ober-/Unterziel	Gewichtung	Technische Umsetzung	Muss-Kriterium
Sichere Dateiablage	**40**		
Einsicht der Daten nur durch Teammitglieder möglich	20	Verschlüsselung mittels DSA oder RSA mit geeigneten Schlüssellängen[37]	Ja
Änderungen sind Mitgliedern zuzuordnen	10	Geeignetes Log System, welches die Änderungen der Nutzer speichert und lesbar darstellt	Nein
Schutz vor Hardwaredefekten	10	Backup auf anderen Speichermedien	Nein
Rücksprung zu früherem Datenstand möglich	10	Speicherung verschiedener Datenversionen mit Rücksprungmöglichkeit	Nein
Unbefugte Änderungen werden erkannt und nicht festgeschrieben	15	Verwendung von TLS oder IPSec. Ansonsten Daten signiert übertragen, so dass eine Überprüfung der Daten möglich ist	Ja
Beidseitige Authentifikation bei Datenübertragung	10	Sowohl Client als auch Server müssen sich authentifizieren. In TLS/SSL bereits genutzt	Ja
Verschlüsselung clientseitig	25	Verschlüsselung der Daten vor der Übertragung zum Server	Ja
Ad hoc-Teambildung	**25**		
Rechteverwaltung vom Teamgründer oder autorisierter Person	33	Rollenbasierte Rechtevergabe	Nein
Erstellung eines Teams allein vom Teamgründer abhängig	33	Rechtevergabe und Teamerstellung in der Software und nicht im System einstellbar	Nein
Sichere Nutzeridentifikation und Authentifikation	33	Verwendung sicherer Passwörter oder einer zwei-Faktor-Authentifikation	Ja
Heterogene Netzwerke	**25**		
Kommunikation mittels anerkannter Netzwerkprotokolle	100	Nutzung von Standartprotokollen zur Kommunikation wie z.B. IPSec, SSL/TLS	Nein
Qualität	**10**		
Kosten	40	Möglichst geringe Kosten	Nein

[37] Siehe Tabelle 13: Schlüssellängen

Funktionalität	10	Überprüfen ob angegebene Funktionen mit getesteten Funktionen übereinstimmen	Nein
Zuverlässigkeit	10	Fehlerfreies Ausführen vom Anwendungsfall Qualitätstest	Nein
Benutzbarkeit	10	Messung der Zeit, die zum Ausführen vom Anwendungsfall Qualitätstest gebraucht wird. Überprüfen des „look and feel“	Nein
Effizienz	10	Vergleich der benötigten zusätzlichen Hardware	Nein
Wartbarkeit	10	Überprüfen wie Softwareupdates eingepflegt werden und ob Software vom Hersteller gepflegt wird	Nein
Übertragbarkeit	10	Unterstützung gängiger Systemplattformen	Nein

4 Lösungsmöglichkeiten

In diesem Kapitel werden Lösungsansätze für die eingangs genannten Probleme des Szenarios aufgezeigt und bewertet. Die es viele sehr unterschiedliche Lösungsansätze für Teamarbeit und Dateiablage gibt, werden die Lösungen in Kategorien präsentiert. Für jede Kategorie werden dann bekannte Stellvertreter gewählt, so dass eine gute Abschätzung über die besten Lösungsansätze abgegeben werden kann.

4.1 Vorstellung der ausgewählten Lösungen

In diesem Abschnitt werden Lösungsmöglichkeiten für das beschriebene Szenario vorgestellt. Die Lösungsmöglichkeiten sind sehr unterschiedlich, lassen sich aber in zwei grundlegende Kategorien aufteilen. Auf der einen Seite gibt es Lösungen, bei denen die Daten direkt übergeben werden. Auf der anderen Seite gib es Lösungen, die die Daten zentral Speichern und dann verteilen. Aus jeder Kategorie werden dann repräsentative Vertreter ausgewählt und später mit dem beschriebenen Testverfahren ausgewertet.

Die schnellste und einfachste Möglichkeit in einem Team Daten auszutauschen scheint auf den ersten Blick die **direkte Datenübergabe** zu sein. Der typische Vertreter dieser Kategorie ist der USB-Stick. Nutzer 1 speichert seine Arbeitsergebnisse ab und übergibt diese dann per USB-Stick an Nutzer 2. Dieses System ist soweit einfach und unkompliziert. Da die Daten persönlich übergeben werden, ist auch sichergestellt, dass nur die entsprechenden Teammitglieder Zugriff auf diese haben. Diese Art der Datenübergabe ist auch über ein Netzwerk möglich. Zum Beispiel lassen sich Daten per E-Mail im Team verteilen. Nachteile bei diesen Verfahren sind natürlich die komplizierte Datensynchronisation und der unkomfortable Datenaustausch. Beide Möglichkeiten klingen ähnlich, sind aber in Bezug auf die Sicherheit durchaus unterschiedlich. Daher werden beide Verfahren bewertet.

Besser in dieser Hinsicht sind Methoden die eine **zentrale Datenspeicherung** ermöglichen. Hier werden alle Daten zentral an einem Punkt abgelegt. Jedes Teammitglied hat Zugriff zum Datenspeicherort und die Daten werden nach dem bearbeiten wieder dort abgelegt. Dies bietet Vorteile beim Datenaustausch. Schwieriger ist es zu garantieren, dass nur Teammitglieder Einsicht in die Daten erhalten. Es kann auch zu Problemen führen, wenn zeitgleich an der gleichen Datei gearbeitet wird. Bekannte Lösungen in diesem Bereich sind Cloud Speicherdienste wie Dropbox[38]. Dienste aus der Cloud, die die Zusammenarbeit erleichtern, gibt es mittlerweile viele. In diesem Fokus werden hauptsächlich die Datenspei-

[38] [23] https://www.dropbox.com/

cherdienste getestet, da das Szenario eine sichere Datenspeicherung verlangt. Besonders interessant ist in diesem Zusammenhang der Vergleich von kommerziellen und Open Source[39] Produkten. Daher werden Vertreter aus beiden Bereichen gewählt.
Ähnlich wie Dropbox ist Wuala[40] ein Cloud Speicher Dienst. Dieser Dienst wirbt speziell mit der Datensicherheit und einer clientseitigen Verschlüsselung. Daher wird dieser Dienst bewertet. Als Open Source alternative wird OwnCloud[41] bewertet. Da OwnCloud noch keine clientseitige Verschlüsselung bietet, wird in Kombination mit OwnCloud zusätzlich Cloudfogger[42] verwendet. Cloudfogger bietet kostenlose clientseitige Verschlüsselung für Cloud Speicher Dienste an.
Neben den aufgezeigten Cloud Lösungen existieren noch viele weitere. Gerade Business Lösungen, die auf internen Servern betrieben werden, haben oft die Eigenschaft, dass sie auf bestimmte Plattformen zugeschnitten sind. Microsoft SharePoint[43] ist ein gutes Produkt in diesem Bereich, welches viele der geforderten Funktionen bietet. Leider ist Microsoft SharePoint darauf ausgelegt mit anderen Microsoft Produkten zusammen zu arbeiten. Im Szenario sind jedoch heterogene Netzwerke mit verschiedenen Plattformen beschrieben. Aus diesem Grund eignen sich diese Produkte nicht für dieses Szenario.
Alle Arten bieten Vor- und Nachteile in diesem Szenario. Daher werden diese im Hinblick auf das geforderte Zielsystem getestet und bewertet. So ist eine Einordung der einzelnen Lösungen möglich.
Zusammenfassend werden folgende Lösungsansätze bewertet:

- **Direkte Dateiübergabe**
  - USB-Stick
  - E-Mail
- **Zentrale Datenspeicherung**
  - Wuala
  - OwnCloud/Cloudfogger

4.2 Bewertung der Lösungen

Die in Kapitel 4.1 vorgestellten Lösungen werden in diesem Kapitel anhand des Zielsystems bewertet und eingeordnet.
Die im *Kapitel 3.1 Anforderungen* und Kapitel *3.2 Technische Spezifikation der Anforderungen* herausgearbeiteten gewünschten Softwarefunktionen bilden die Grundlage für diese Bewertung. Je nach Erfüllungsgrad werden dann den Unterzielen Punktbewertungen zugeteilt. Als Bewertungsskala wird das Schulnotensystem von 1-6 verwendet. Wobei 1 die bes-

[39] Für Erklärung siehe [24] http://opensource.org/osd-annotated
[40] [25] http://www.wuala.com/de/
[41] [26] http://owncloud.org/
[42] [27] http://www.cloudfogger.com/de/
[43] [28] http://office.microsoft.com/de-de/sharepoint/

te und 6 die schlechteste Bewertung ist. Dieses ist intuitiv anzuwenden und für jeden leicht verständlich. Zur Einordnung der einzelnen Noten folgt eine kurze Erläuterung:

- 1 sehr gut, die Erwartungen werden mehr als erfüllt.
- 2 gut, das Kriterium ist erfüllt.
- 3 befriedigend, das Kriterium ist zwar erfüllt, aber nicht zukunftssicher.
- 4 schlecht, es gibt Sicherheitsmängel.
- 5 sehr schlecht, es gibt gravierende Sicherheitsmängel.
- 6 Ungenügend, das Kriterium wird gar nicht erfüllt.

Diese sechs Bewertungsstufen reichen für eine hinreichende Unterscheidung der Lösungsmöglichkeiten aus. Kann-Kriterien werden zur Unterscheidung von den Muss-Kriterien kursiv dargestellt. Muss-Kriterien gelten ab einer Benotung von 3 als erfüllt.
Zuerst werden die Lösungen der Kategorie **direkte Dateiübergabe** bewertet. Dies beinhaltet den Datenaustausch per USB-Stick und per E-Mail. Danach werden die Lösungsansäte mit **zentraler Datenspeicherung** bewertet. Wie oben beschrieben sind dies Wuala und OwnCloud/Cloudfogger.

4.2.1 USB-Stick

Der **USB-Stick** ist gerade unter Studenten, die oft in kleinen ad hoc-Teams zusammenarbeiten, ein durchaus beliebtes Mittel zum Datenaustausch. Im diesem Kapitel wird bewertet, inwieweit dies eine optimale Lösung für das beschriebene Szenario ist.
Zuerst wird geprüft wie gut sich diese Lösung für die **sichere Dateiablage** eignet:

1. Die Einsicht der Daten darf nur durch Teammitglieder möglich sein:
   Ein Vorteil des USB-Sticks ist, dass er direkt von einer Person an die nächste übergeben wird. Daher ist die Sicherheit in diesem Bereich zumindest teilweise gegeben. Sollte der USB-Stick allerdings gestohlen werden oder er geht verloren, hat der Dieb oder Finder Einsicht in die Klartext Daten. Dies wäre nur zu unterbinden wenn ein entsprechendes System eine Verschlüsselung der gespeicherten Daten erzwingt. Dies ist hier nicht der Fall und wird als schlecht eingestuft.
2. Änderungen an Daten sind Teammitgliedern zuzuordnen:
   Ein USB-Stick bzw. die zugrunde liegenden Filesysteme bieten kein ausreichendes Log, um diese Informationen zu erlangen. Lediglich das Änderungsdatum könnte auf Aufschluss darüber geben, wer welche Dateien geändert hat. Dies ist nicht ausreichend und wird als sehr schlecht eingestuft.
3. Es soll ein Schutz vor Hardwaredefekten bestehen:
   Bei der Nutzung eines USB-Sticks zum Datenaustausch sind die Daten des USB-Sticks verloren, sollte dieser defekt sein. Es ist möglich, dass einer der Nutzer ein komplettes aktuelles Backup der Daten hat, aber dies wird keinesfalls erzwungen. Daher wird dies als sehr schlecht bewertet.
4. Ein Rücksprung zu einem früheren Datenbestand soll möglich sein:

Auch hier ist der Nutzer selbständig dafür verantwortlich inwieweit alte Versionen der Daten aufgehoben werden. Dies ist sehr schlecht.

5. Unbefugte Änderungen sollen erkannt und nicht festgeschrieben werden:
   Sollte der USB-Stick außerhalb des Teams verwendet werden, können unbemerkt Daten geändert und gesichert werden. Hier ist kein Schutz. Dies ist ungenügend.
6. Es soll eine beidseitige Authentifikation bei der Datenübertragung stattfinden:
   Da kein Netzwerk verwendet wird und der USB-Stick direkt übergeben wird, findet durch die direkte Übergabe des Datenspeichers eine direkte beidseitige Authentifikation statt. Dies ist eine sehr gute Authentifikation.
7. Eine Clientseitige Verschlüsselung soll stattfinden:
   Es wird in keiner Weise eine Verschlüsselung erzwungen und sie ist dem Nutzer selbst überlassen. Daher ist dies eine ungenügende Lösung.

Tabelle 20: Bewertung sichere Dateiablage bei USB-Sticks

Ober-/Unterziel	**Gewichtung**	**Anmerkungen**	**Note**
Sichere Dateiablage	**40**		
Einsicht der Daten nur durch Teammitglieder möglich	20	Verschlüsselung mittels DSA oder RSA mit geeigneten Schlüssellängen[44].	4
Änderungen sind Mitgliedern zuzuordnen	*10*	*Kein Log der Dateiänderungen*	*5*
Schutz vor Hardwaredefekten	*10*	*Backup auf anderen Speichermedien nicht erzwungen*	*5*
Rücksprung zu früherem Datenstand möglich	*10*	*Speicherung verschiedener Datenversionen mit Rücksprungmöglichkeit nicht gegeben*	*5*
Unbefugte Änderungen werden erkannt und nicht festgeschrieben	15	Kein Schutz vor unbefugten Änderungen	6
Beidseitige Authentifikation bei Datenübertragung	10	Ist durch den physischen Datenaustausch gegeben	1
Verschlüsselung clientseitig	25	Keine Dateiverschlüsselung erzwungen	6
Zwischennote			**4,8**

In Hinblick auf das Oberziel der **sicheren Dateiablage** ergeben sich, wie in *Tabelle 20: Bewertung sichere Dateiablage bei USB-Sticks* zu sehen, erhebliche Mängel. Zusätzlich sind zwei Muss-Kriterien, nämlich *Einsicht in Daten nur durch Teammitglieder möglich* und *Ver-*

[44] Siehe Tabelle 13: Schlüssellängen

schlüsselung clientseitig. Dies zeigt, dass der USB-Stick Lösungsansatz in diesem Bereich nicht gut abschneidet.
Die Bildung eines **ad hoc-Teams** ist hingegen sehr einfach und komfortabel. Die Mitglieder müssen sich lediglich absprechen und als Team zusammenfinden. Es ergeben sich folgende Teilbewertungen:

1. Die Rechteverwaltung soll vom Teamgründer oder einer autorisierten Person ausgehen:
   Generell bestimmt der Teamgründer die Mitglieder des Teams und kann daher bestimmen, wer den USB-Stick erhalten darf und wer nicht. Die Verwaltung geht, wie gefordert, von dem Teamgründer aus. Abzüge gibt es, da die Rechte nur durch Absprachen und nicht durch das Produkt gesichert sind. Da davon ausgegangen wird, dass die Teammitglieder sich gegenseitig vertrauen können, gilt dies
   als befriedigend.
2. Die Erstellung des Teams soll allein vom Teamgründer abhängen:
   Der Teamgründer erstellt das Team durch Absprachen und ist hier in keiner Weise von anderen Personen außerhalb des Teams abhängig. Das Unterziel wird gut erfüllt.
3. Die Verwendung einer sicheren Nutzeridentifikation und Authentifikation ist gefordert:
   Da der USB-Stick direkt von einem zum anderen Teammitglied übergeben wird, findet eine sichere Authentifikation statt. Dies ist eine gute Lösung.

Tabelle 21: Bewertung ad hoc-Teambildung bei USB-Sticks

Ober-/Unterziel	**Gewichtung**	**Anmerkungen**	**Note**
Ad hoc-Teambildung	**25**		
Rechteverwaltung vom Teamgründer oder autorisierter Person	*33*	*Nur durch Absprachen im Team möglich. Team wird als vertrauenswürdig angesehen*	*3*
Erstellung eines Teams allein vom Teamgründer abhängig	*33*	*Siehe oben*	*2*
Sichere Nutzeridentifikation und Authentifikation	33	Gegeben durch direkte Übergabe des USB-Sticks	2
Zwischennote			**2,3**

Tabelle 21: Bewertung ad hoc-Teambildung bei USB-Sticks zeigt, dass der USB-Stick zur Nutzung in ad hoc-Teams gut geeignet ist. Zu beachten ist, dass der Fokus auf der Sicherheit der Lösung liegt. Der Austausch des Speichers innerhalb des Teams kann kompliziert sein.

Die Verwendung von USB-Sticks ist in **heterogenen Netzwerken** ohne Weiteres möglich, da die Datenübergabe direkt und ohne Netzwerk stattfindet.

1. Die Kommunikation soll mittels anerkannter Netzwerkprotokolle stattfinden:
   Es werden keine Netzwerke zur Datenübertragung verwendet, daher treten in diesem Bereich keine Sicherheitsrisiken auf. Dies ist gut. Abzüge gibt es für den Umstand, dass die Daten physisch übergeben werden müssen. Insgesamt ist dies befriedigend.

Tabelle 22: Bewertung heterogener Netzwerke bei USB-Sticks

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Heterogene Netzwerke	**25**		
Kommunikation mittels anerkannter Netzwerkprotokolle	100	Keine Netzwerknutzung. Abzüge durch unkomfortablen Datenaustausch	3
Zwischennote			**3**

In *Tabelle 22: Bewertung heterogener Netzwerke bei USB-Sticks* ist zu sehen, dass heterogene Netzwerke keine Einschränkungen für die Nutzung eines USB-Sticks mitbringen. Dies liegt daran, dass ein USB-Stick ohne Netzwerk auskommt.
Im folgenden Abschnitt wird die **Qualität** der USB-Stick Lösung bewertet. Zu beachten ist, dass der USB-Stick keine direkte Softwarelösung ist. Trotzdem lassen sich die Kriterien anwenden, da diese bei Bedarf passend ausgelegt werden können. Die Bewertung setzt sich wie folgt zusammen:

1. Die Kosten setzten sich wie folgt zusammen:
   Die Anschaffung einer oder mehrere USB-Sticks ist notwendig. Die Kosten für ein aktuelles Modell mit genügend Speicherplatz[45] liegen bei unter 10,00€ pro Stück. Dies ist sehr günstig und wird als befriedigend angesehen.
2. Die Funktionalität ist voll gegeben. USB-Sticks sind in der Lage Daten zu speichern und diese können später wieder ausgelesen werden. Die Funktionalität ist daher gut.
3. Die Zuverlässigkeit wird mit gut bewertet. Der Anwendungsfall[46] ließ sich ohne Probleme ausführen.
4. Die Bewertung der Benutzbarkeit ist abhängig von der Entfernung der Teammitglieder. Zur Bewertung wird die Zeit zum Ausführen des

[45] 16GB Speicherkapazität
[46] Siehe Tabelle 17: Anwendungsfall Qualitätstest

Qualitätstest Anwendungsfalls[47] gemessen. Da die Datenübertragung in diesem Fall physisch erfolgt, müssen Nutzer 1 und Nutzer 2 sich Treffen, um den USB-Stick zu übergeben. Befinden sich beide im gleichen Gebäude kann dies schnell geschehen. Sind sie jedoch in unterschiedlichen Städten, dauert dies sehr lange. Dies ist im beschriebenen Szenario nicht unwahrscheinlich. Die Zeit zum Ausführen des Anwendungsfalles kann also zwischen 2 Minuten und eventuell mehreren Stunden oder Tagen variieren. Aufgrund der Variabilität der Übertragungszeit ist die Benutzbarkeit sehr schlecht.

5. Die Effizienz der USB-Stick Lösung ist befriedigend. Es wird nur sehr wenig zusätzliche Hardware benötigt, da nur die USB-Sticks angeschafft werden müssen.
6. Die Wartbarkeit ist gut. Es handelt sich bei den USB-Sticks um eine Lösung, die mit Standartsoftware der Betriebssysteme auskommt. Benötigte Treiber und eventuelle Updates werden automatisch vom Betriebssystem installiert. Der Nutzer kann den USB-Stick nach Einstecken in den PC direkt verwenden.
7. Die Übertragbarkeit ist ebenfalls gut. USB-Sticks werden von allen gängigen Betriebssystemen automatisch erkannt. Eine mobile Unterstützung ist nur teilweise gegeben. Durch spezielle Adapter ist es teilweise möglich USB-Sticks an Mobiltelefonen zu nutzen.

Tabelle 23: Bewertung Qualität bei USB-Sticks

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Qualität	**10**		
Kosten	*40*	*10,00€ pro Stück*	3
Funktionalität	*10*	*Alle erwarteten Funktionen erfüllt*	2
Zuverlässigkeit	*10*	*Fehlerfreies Ausführen vom Anwendungsfall Qualitätstest möglich*	2
Benutzbarkeit	*10*	*Sehr variable Ausführungszeit*	5
Effizienz	*10*	*Vergleich der benötigten zusätzlichen Hardware*	3
Wartbarkeit	*10*	*Installation und Updates erfolgt automatisch vom jeweiligen Betriebssystem*	2
Übertragbarkeit	*10*	*Alle gängigen Plattformen werden unterstützt*	2
Zwischennote			**2,8**

Tabelle 23: Bewertung Qualität bei USB-Sticks zeigt, dass der USB-Stick im Bereich eine befriedigende Bewertung erhält. Auffällig ist die schlechte Benotung in Bezug auf die Benutz-

[47] Siehe Tabelle 17: Anwendungsfall Qualitätstest

barkeit des USB-Sticks. Dies zeigt wie kompliziert die Teamarbeit mit einem USB-Stick sein kann.
Nachdem alle Unterziele bewertet wurden, können die Zwischenergebnisse zur Entwertung zusammengefasst werden. Diese endgültige Bewertung ist in der folgenden Tabelle zu sehen.

Tabelle 24: Gesamtbewertung USB-Stick

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Sichere Dateiablage	40		4,8
Ad hoc-Teambildung	25		2,3
Heterogene Netzwerke	25		3
Qualität	10		2,8
Gesamtnote			**3,5**

In *Tabelle 24: Gesamtbewertung USB-Stick* ist zu sehen das die USB-Stick Lösung im Bereich der sicheren Dateiablage sehr schlecht abschneidet. Hier wurden mehrere Muss-Kriterien nicht eingehalten. Die anderen Bereiche sind nur teilweise befriedigend. Zusammengefasst ist die Lösung nicht mehr befriedigend. Es ist anschaulich, dass diese Lösung nicht optimal ist und eine bessere Lösung gefunden werden muss.

4.2.2 E-Mail

In diesem Kapitel wird beleuchtet wie die im Szenario beschriebenen Probleme mit einfachem E-Mail Austausch gelöst werden können. Dies ist eine Variante, die im privaten Leben gern und viel genutzt wird. Daher wird diese Lösung genauer untersucht.
Zuerst wird das Oberziel der **sicheren Dateiablage** mit den entsprechenden Unterzielen bewertet:

1. Die Einsicht der Daten darf nur durch Teammitglieder erfolgen:
   Wird eine E-Mail von einem zum anderen Empfänger gesendet, passiert diese den Server des E-Mail Providers. Eine E-Mail ist standartmäßig nicht verschlüsselt. Es kann nicht sichergestellt werden, dass zum Beispiel der Provider Einsicht in die Daten erhält. Selbst eine oft genutzte verschlüsselte Übertragung per SSL schützt nicht vor der Einsicht durch den Provider. Da die Daten nur auf dem Übertragungsweg zum E-Mail Provider und vom E-Mail Provider zum Empfänger verschlüsselt sind. Es gibt die Möglichkeit die E-Mails zusätzlich zu verschlüsseln. Die Konfiguration der Verschlüsselung ist umständlich. Nur wenn das Zertifikat des Empfängers vorhanden ist, können E-Mails verschlüsselt werden. Es wird davon ausgegangen, dass die Zertifikate des Teams bekannt sind. Nur ist dies nicht bei jedem Kontakt des Teammitglieds der Fall. Das führt dazu, dass die Verschlüsselung für verschiedene E-Mails aktiviert und deaktiviert wird. Daher ist die Möglichkeit, dass die Verschlüsselung

vergessen wird, sehr hoch. Wegen dieser Sicherheitsmängel wird dieser Punkt als schlecht bewertet.

2. Dateiänderungen sollen Teammitgliedern zuzuordnen sein:
   Änderungen können in der E-Mail beschrieben werden und sind so dem Sender zuzuordnen. Dies ist aber nicht erzwungen und den Teammitgliedern selbst überlassen. Diese Lösung beruht allein auf den Teammitgliedern. Daher ist dies eine schlechte Lösung.
3. Ein Schutz vor Hardwaredefekten soll gegeben sein:
   Es ist kein Schutz vor Defekten in Form eines geeigneten Datenbackups vorhanden. Eventuell existieren Teilbackups in den E-Mail Postfächern der Nutzer, dies ist aber nicht sichergestellt. Diese Lösung wird mit schlecht bewertet.
4. Ein Rücksprung zu einem früheren Datenstand soll möglich sein:
   Wenn eine angekommene E-Mail nach dem Speichern der Daten nicht gelöscht wird, ist zumindest ein Rücksprung auf diesen Datenstand möglich. Dies ist nicht gesichert und dem Nutzer überlassen, daher ist es eine schlechte Lösung.
5. Unbefugte Änderungen sollen erkannt und nicht festgeschrieben werden:
   Sollte die E-Mail auf dem Versandweg zum Beispiel vom Provider geändert werden, besteht keine Möglichkeit dies festzustellen. Diese Lösung ist als ungenügend anzusehen.
6. Bei Datenübertragung ist eine beidseitige Authentifikation erwünscht:
   Bei Verwendung entsprechender Protokolle wie SSL/TLS ist dies gegeben. Viele E-Mail Provider unterstützen oder erzwingen sogar die Nutzung dieser Protokolle. Abzüge gibt es dafür, die Protokolle nicht erzwungen werde. Daher ist diese Lösung befriedigend.
7. Eine clientseitige Verschlüsselung wird gefordert:
   Bei heutigen E-Mail Programmen ist es möglich die Daten verschlüsselt zu versenden. Benötigt wird das Zertifikat des Empfängers, um die Daten mit diesem zu verschlüsseln. Zusätzlich können die Daten mit dem eigenen Zertifikat signiert werden. So kann der Empfänger auch verifizieren, dass die Nachricht von dem richtigen Sender kommt. Soll allerdings eine E-Mail an die ganze Gruppe versendet werden, muss dies entweder einzeln geschehen, oder es muss ein Gruppen Zertifikat eingerichtet werden. Das Handling der Verschlüsselung ist daher in diesem Bereich nicht einfach. Dies gibt Abzüge in der Bewertung. Insgesamt wird dieser Punkt mit befriedigend bewertet.

In *Tabelle 25: Bewertung sichere Dateiablage bei E-Mail Austausch* ist die Zusammenfassung der Bewertung der sicheren Dateiablage zu sehen. Ein Muss-Ziel, *Einsicht der Daten nur durch Teammitglieder*, wird nicht eingehalten, da das Handling der Verschlüsselung umständlich ist. Auffällig ist die Zwischennote von 4,7 die für ein schlechtes Gesamtergebnis spricht.

Tabelle 25: Bewertung sichere Dateiablage bei E-Mail Austausch

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Sichere Dateiablage	**40**		
Einsicht der Daten nur durch Teammitglieder möglich	20	Verschlüsselung der Daten nicht erzwungen	4
Änderungen sind Mitgliedern zuzuordnen	*10*	*Durch Eintrag in der E-Mail möglich. Abzüge dafür, dass dies nicht automatisch geregelt ist*	*4*
Schutz vor Hardwaredefekten	*10*	*Backup auf anderen Speichermedien*	*4*
Rücksprung zu früherem Datenstand möglich	*10*	*Speicherung verschiedener Datenversionen mit Rücksprungmöglichkeit*	*4*
Unbefugte Änderungen werden erkannt und nicht festgeschrieben	15	Selbst wenn zur Übertragung TLS/SSL genutzt wird ist eine Änderung der Daten durch den Provider nicht feststellbar	6
Beidseitige Authentifikation bei Datenübertragung	10	Sowohl Client als auch Server müssen sich Authentifizieren. In TLS/SSL bereits genutzt	3
Verschlüsselung clientseitig	25	Verschlüsselung der Daten vor der Übertragung zum Server	3
Zwischennote			**4**

Im weiteren Verlauf wird bewertet wie gut sich der E-Mail Verkehr für die **ad hoc-Teambildung** eignet:

1. Die Rechteverwaltung soll vom Teamgründer oder einer autorisierten Person ausgehen:
   Der Teamleader kann sein Team per Mail kontaktieren oder in einem persönlichen Treffen zusammenstellen. Die Rechte können lediglich per Absprache festgelegt werden. Da das Team als vertrauenswürdig angesehen wird, ist dies im Hinblick auf das Szenario eine befriedigende Lösung.
2. Die Teamerstellung ist allein vom Teamleader abhängig:
   Dies ist hier der Fall, da der Teamleader, wie oben beschrieben, sein Team einfach kontaktieren und so zusammensetzen kann. Es wird davon ausgegangen, dass jedes Teammitglied bereits einen E-Mail Account besitzt. Trotzdem ist dies eine leichte Abhängigkeit zu demjenigen der die E-Mail Accounts verwaltet. Daher gibt es einen Abzug. Insgesamt ist diese Lösung befriedigend.
3. Eine sichere Nutzeridentifikation und Authentifikation ist gefordert:

Es wird angenommen, dass der Empfänger hinreichend durch seine E-Mail Adresse identifiziert ist. Aber es ist ohne weiteres möglich eine E-Mail mit gefälschtem Absender zu versenden[48]. Daher kann der Empfänger nicht sicher sein, dass die E-Mail von dem Sender und nicht von einer dritten Person stammt. Dies kann zwar durch Verwendung einer Signatur[49] unterbunden werden, aber diese werden in der Regel nicht benutzt und müssen zusätzlich eingerichtet werden. Da diese Lücke durch Zusatzsoftware unterbunden werden kann, dies aber umständlich und nicht erzwungen ist, wird dieser Punkt mit schlecht bewertet.

Tabelle 26: Bewertung ad hoc-Teambildung bei E-Mail Austausch

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Ad hoc-Teambildung	**25**		
Rechteverwaltung vom Teamgründer oder autorisierter Person	*33*	*Nur durch Absprachen im Team möglich. Team wird als vertrauenswürdig angesehen*	*3*
Erstellung eines Teams allein vom Teamgründer abhängig	*33*	*Abzüge durch leichte Abhängigkeit vom Verwalter der E-Mail Accounts*	*3*
Sichere Nutzeridentifikation und Authentifikation	33	Wenn keine Signatur verwendet wird, kann dem Empfänger nicht vertraut werden	4
Zwischennote			**3,3**

In *Tabelle 26: Bewertung ad hoc-Teambildung bei E-Mail Austausch* wird die Zusammenfassung der ad hoc-Teambewertung gezeigt. Generell ist das Zwischenergebnis besser als das der sicheren Dateiablage, aber es ist mit 3,3 auch noch weit von einer guten Bewertung entfernt. Dies liegt auch an dem nicht erfüllten Muss-Ziel *sichere Nutzeridentifikation und Authentifikation.* Dies spricht nicht für die Verwendung von E-Mail bei Zusammenarbeit im ad hoc-Team.
Im nächsten Abschnitt wird die Verwendung von E-Mail in **heterogenen Netzwerken** bewertet:

1. Die Kommunikation soll mittels anerkannter Netzwerkprotokolle erfolgen:
   E-Mail ist ein sehr verbreiteter Standard und ist normalerweise in allen Netzen verfügbar. Alle gängigen E-Mail Provider bieten die Möglichkeit der Übertragung per TLS/SSL Protokoll. Eine Absprache im Team kann dafür sorgen, dass diese Einstel-

[48] [11] https://www.bsi-fuer-buerger.de/BSIFB/DE/GefahrenImNetz/GefaelschteAbsenderadressen/gefaelschteabsenderadressen_node.html
[49] Siehe Kapitel 1.1.1 Elektronische Signaturverfahren

lungen verwendet werden. Da trotzdem die ungesicherte Übertragung per SMTP[50] möglich ist, wird dieser Punkt nur mit befriedigend bewertet.

Tabelle 27: Bewertung heterogene Netzwerke bei E-Mail Austausch

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Heterogene Netzwerke	**25**		
Kommunikation mittels anerkannter Netzwerkprotokolle	100	E-Mail normalerweise in allen Netzen verfügbar. Abzüge da die Verwendung von TLS/SSL nicht erzwungen werden kann	3
Zwischennote			**3**

Tabelle 27: Bewertung heterogene Netzwerke bei E-Mail Austausch zeigt das heterogene Netzwerke befriedigend unterstützt werden.
Abschließend wird die **Qualität** der E-Mail Lösung bewertet:

1. Die Kosten werden als gut bewertet. E-Mail Adressen sind, falls noch nicht vorhanden, kostenfrei erhältlich.
2. Die Funktionalität der Software ist gut. Alle erwarteten Funktionen sind verfügbar.
3. Die Zuverlässigkeit der E-Mail Lösung ist gut. E-Mail Software ist schon lange am Markt und daher sehr ausgereift. Der Anwendungsfall[51] für den Qualitätstest lässt sich ohne Auffälligkeiten durchführen.
4. Für die Bewertung der Benutzbarkeit wird die Zeit zur Ausführung des Anwendungsfalls für den Qualitätstest gemessen. Wenn kein E-Mail Account vorhanden ist, lässt sich dieser in zehn Minuten einrichten. Für das Versenden der E-Mail wird eine Minute gebraucht. Im schlimmsten Fall müssen also zwei E-Mail Accounts eingerichtet werden, die E-Mail muss hin und zurück gesendet werden. Die Ausführungszeit beträgt also circa 17 Minuten. Sind die E-Mail Adressen bereits vorhanden reduziert sich die Zeit auf 2 Minuten. Allerdings wird der Datenaustausch zwischen mehreren Personen komplizierter. Es kann passieren, dass unterschiedliche Dateiversionen umhergeschickt und bearbeitet werden. Dies führt zu Abzügen in der Bewertung. Dieser Punkt wird insgesamt als schlecht bewertet.

[50] [2, p. 152 ff.] Eckert, Claudia: IT-Sicherheit, Oldenbourg 2013
[51] Siehe Tabelle 17: Anwendungsfall Qualitätstest

5. Die Effizienz der E-Mail Lösung wird als gut bewertet. Ist ein E-Mail Server vorhanden, kann dieser verwendet werden. Gibt es diesen nicht, können kostenfreie Dienste im Internet genutzt werden. Daher wird keine zusätzliche Hardware benötigt.
6. Die Wartbarkeit ist abhängig vom verwendeten E-Mail Programm. Wird das Betriebssystem eigene E-Mail Programm verwendet, wird ein entsprechendes Update automatisch eingespielt. Im beschriebenen Szenario wird davon ausgegangen, dass dies der Fall ist. Daher ist die Wartbarkeit gut.
7. Die Übertragbarkeit der E-Mail Lösung ist sehr gut. Eine E-Mail lässt sich von allen Plattformen die über Internet verfügen verschicken. Daher gibt es keine Einschränkungen in dieser Hinsicht. Zusätzlich werden alle internetfähigen mobilen Plattformen unterstützt.

Tabelle 28: Bewertung Qualität bei E-Mail Austausch

Ober-/Unterziel	**Gewichtung**	**Anmerkungen**	**Note**
Qualität	**10**		
Kosten	*40*	*Kostenfrei, beziehungsweise Nutzung des vorhandenen E-Mail Servers*	*2*
Funktionalität	*10*	*Alle erwarteten Funktionen erfüllt*	*2*
Zuverlässigkeit	*10*	*Fehlerfreies Ausführen vom Anwendungsfall Qualitätstest möglich*	*2*
Benutzbarkeit	*10*	*Problematischer je größer das Team*	*4*
Effizienz	*10*	*Es wird keine zusätzliche Hardware benötigt*	*2*
Wartbarkeit	*10*	*Installation und Updates erfolgt automatisch vom jeweiligen Betriebssystem*	*2*
Übertragbarkeit	*10*	*Alle gängigen Plattformen werden unterstützt*	*1*
Zwischennote			**2,1**

Tabelle 28: Bewertung Qualität bei E-Mail Austausch zeigt, dass die Qualität der E-Mail Lösung insgesamt als gut bewertet wird. Hier zeigt sich, dass es sinnvoll sein kann auf bestehende Technologien aufzubauen.
Abschließend folgt die Gesamtauswertung der Zwischennoten der einzelnen Oberziele.

Tabelle 29: Gesamtbewertung E-Mail

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Sichere Dateiablage	40		4
Ad hoc-Teambildung	25		3,3
Heterogene Netzwerke	25		3
Qualität	10		2,1
Gesamtnote			**3,4**

In *Tabelle 29: Gesamtbewertung E-Mail* ist zu sehen, dass das Verwenden einer E-Mail Lösung nicht optimal ist. Gerade die Oberziele sichere Dateiablage und ad hoc-Teambildung zeigen verbesserungswürdige Noten. Dies kann daran liegen, dass E-Mail anfangs nicht mit dem Hintergedanken an IT-Sicherheit entwickelt wurde.

4.2.3 Wuala

Wuala ist ein deutscher Cloud Speicher Dienst, der aktiv mit der Sicherheit der Daten wirbt. In der Studie *On the Security of Cloud Storage Services*[52] des Fraunhofer Instituts wurde Wuala untersucht und hatte im Vergleich zu anderen Diensten wenige Sicherheitsmängel. Wuala hat darauf reagiert und einige der Bedenken[53] behoben. Der folgende Test wird zeigen ob dies zutrifft.
Zuerst wird das Oberziel **sichere Dateiablage** bewertet:

1. Die Einsicht der Daten darf nur durch Teammitglieder möglich sein:
   Generell bietet Wuala ein Freigabesystem für Dateien an. Es kann für jede Datei oder jeden Ordner eingestellt werden, für wen diese Freigegeben werden sollen.
   Die Daten werden automatisch vom Client verschlüsselt und dann auf dem Server abgelegt. Wuala übernimmt das Schlüsselmanagement[54], so dass der Nutzer nichts konfigurieren muss. Dies wird als gut bewertet.
2. Änderungen sollen Teammitgliedern zugeordnet werden können:
   Wuala ist in der Lage Änderungen der Nutzer zu speichern und diese den Nutzern zuzuordnen. Dies erfolgt automatisch ohne weitere Konfiguration. Dies wird als gut bewertet.
3. Es soll ein Schutz vor Hardwaredefekten vorliegen:
   Die Daten werden auf jeden PC und beim Anbieter Wuala im entsprechenden Cloud Storage gespeichert. Wuala versichert, dass die Daten redundant auf mehreren Servern gespeichert werden. Die Daten liegen also bei Wuala in gesicherter Form

[52] [13] On the Security of Cloud Storage Services
https://www.sit.fraunhofer.de/de/angebote/projekte/cloud-studie/
[53] [14] http://wualablog.blogspot.de/2012/05/fraunhofer-study-on-cloud-storage.html
[54] [15] Cryptree: A Folder Tree Structure for Cryptographic File Systems

vor und sind zusätzlich auf jedem verbundenen PC gespeichert. Dies ist ein guter Schutz gegen Hardwaredefekte. Tritt ein defekt am PC auf bevor die Daten synchronisiert sind, gehen diese verloren. Daher wird dieser Teil insgesamt mit befriedigend bewertet.

4. Ein Rücksprung zu einem früheren Datenstand soll möglich sein:
   Wuala speichert bis zu zehn alte Dateiversionen, bis diese vom Nutzer gelöscht werden. Die alten Versionen sind einfach im Wuala Client wiederherstellbar.
   Dies ist eine gute Lösung.
5. Unbefugte Änderungen sollen erkannt und nicht festgeschrieben werden:
   Wuala nutzt zur Datenübertragung kein öffentliches Protokoll wie TLS/SSL. Daher kann nicht sichergestellt werden, ob die Daten auf dem Übertragungsweg geändert werden. Die Daten werden verschlüsselt übertragen, somit würde die Änderung der verschlüsselten Daten spätestens beim Entschlüsseln auffallen. Da durch das Geheimhalten des Übertragungsprotokolls grundlegende kryptographische Prinzipien[55] verletzt werden, wird dieser Punkt mit schlecht bewertet.
6. Eine beidseitige Authentifikation ist bei Datenübertragung gefordert:
   Da Wuala kein öffentliches Protokoll zur Datenübertragung verwendet, muss sich der Nutzer auf die Aussagen von Wuala verlassen. Dies ist ein Verstoß gegen das Kerckhoffs´sche Prinzip. Daher wird dieser Punkt mit sehr schlecht bewertet.
7. Eine clientseitige Verschlüsselung ist gefordert:
   Wuala verschlüsselt die Daten automatisch auf dem Client. Hierbei werden AES-256 und RSA-2048 verwendet. Dies entspricht den Empfehlungen des Bundesamtes für Sicherheit in der Informationstechnik[56]. Die verwendeten Methoden sind fest eingestellt und nicht konfigurierbar. Bei eventuellen Änderungen ist der Nutzer von Wuala abhängig. Da die gängigen Sicherheitsempfehlungen eingehalten werden, wird dieser Punkt trotzdem mit gut bewertet.

Tabelle 30: Bewertung sichere Dateiablage Wuala zeigt, dass Wuala im Bereich sichere Dateiablage gute Ansätze verfolgt. Trotzdem werden zwei Muss-Kriterien verletzt. Da Wuala auf ein geheimes Übertragungsprotokoll setzt, statt eine gute getestete Variante zu verwenden, fällt Wuala im Bereich *unbefugte Änderungen werden erkannt und nicht festgeschrieben* und *beidseitige Authentifikation bei Datenübertragung*, durch. Dies beeinflusst die Note negativ und verhindert eine gute Bewertung.

[55] Siehe Kapitel 2.1 Kerckhoffs-Prinzip
[56] Siehe Tabelle 13: Schlüssellängen

Tabelle 30: Bewertung sichere Dateiablage Wuala

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Sichere Dateiablage	**40**		
Einsicht der Daten nur durch Teammitglieder möglich	20	Automatische Datenverschlüsselung auf Clientseite	2
Änderungen sind Mitgliedern zuzuordnen	*10*	*Gute Log Funktion, die alle Änderungen dem jeweiligen Mitglied zuordnet*	*2*
Schutz vor Hardwaredefekten	*10*	*Backup auf anderen Speichermedien*	*3*
Rücksprung zu früherem Datenstand möglich	*10*	*Speicherung verschiedener Datenversionen mit Rücksprungmöglichkeit*	*2*
Unbefugte Änderungen werden erkannt und nicht festgeschrieben	15	Verwendung eines geheimen Übertragungsprotokolls	4
Beidseitige Authentifikation bei Datenübertragung	10	Geheimes Übertragungsprotokoll, daher nicht sicher	5
Verschlüsselung clientseitig	25	Verschlüsselung der Daten vor der Übertragung zum Server	2
Zwischennote			**2,7**

Das Oberziel **ad hoc-Teambildung** ist wie folgt bewertet worden:

1. Die Rechteverwaltung soll vom Teamgründer oder einer autorisierten Person ausgehen:
   In Wuala ist es möglich Gruppen zu erstellen. Innerhalb dieser Gruppen kann der Gruppenersteller allen Mitgliedern spezifische Rechte zuweisen. Die Rollen können umbenannt und konfiguriert werden. Komplett neue Rollen können nicht hinzugefügt werden und eigene Rechte pro Teammitglied können auch nicht eingestellt werden. Dies gibt Abzüge in der Bewertung. Insgesamt wird dieser Punkt mit befriedigend bewertet.
2. Die Erstellung eines Teams soll allein vom Teamgründer abhängen:
   Bei Wuala kann jeder Nutzer eigenständig eine Gruppe gründen und dann Mitglieder einladen. Haben diese noch keinen Wuala Account wird eine E-Mail mit einer Einladung verschickt. Zusammengefasst wird die Lösung mit gut bewertet.
3. Eine sichere Nutzeridentifikation und Authentifikation ist gefordert:
   Wuala fordert lediglich ein Passwort mit mindestens sechs Zeichen. Es wird nicht überprüft ob das Passwort sicher[57] ist. Die Zahlenkombination 123456 wird als gül-

[57] Siehe Tabelle 6: Sichere Passwörter

tiges Passwort akzeptiert. Es wird nur darauf hingewiesen, dass ein längeres Passwort sinnvoller wäre. Des Weiteren ist die Authentifikation der angegebenen E-Mail Adresse nicht erzwungen. Der Nutzer wird lediglich durch einen Hinweis dazu aufgefordert. Trotz nicht authentifizierter E-Mail Adresse kann der Nutzer zu Gruppen eingeladen werden. Der Gruppenleiter erhält keinen Hinweis darüber, dass der eingeladene Nutzer eine nicht authentifizierte E-Mail Adresse angegeben hat. Dies führt zu Abzügen in der Bewertung. Insgesamt wird dieser Punkt mit schlecht bewertet.

Tabelle 31: Bewertung ad hoc-Teambildung Wuala

Ober-/Unterziel	**Gewichtung**	**Anmerkungen**	**Note**
Ad hoc-Teambildung	**25**		
Rechteverwaltung vom Teamgründer oder autorisierter Person	*33*	*Rollenverteilung möglich* *Keine Rechte pro Nutzer einstellbar*	*3*
Erstellung eines Teams allein vom Teamgründer abhängig	*33*	*Teamerstellung autonom möglich*	*2*
Sichere Nutzeridentifikation und Authentifikation	33	Nur mindestens sechs Zeichen für Passwort. Keine E-Mail Authentifikation	4
Zwischennote			**3**

Die *Tabelle 31: Bewertung ad hoc-Teambildung Wuala* zeigt, dass Wuala im Bereich ad hoc-Teambildung befriedigend abschneidet. Wuala kann auch in diesem Bereich ein Muss-Kriterium nicht erfüllen. Da Wuala keine hinreichend sicheren Passwörter erzwingt, folgt eine schlechte Note. Eine bessere Note wäre möglich gewesen, wenn Wuala mehr Wert auf die Authentifikation und Identifikation der Nutzer legen würde.
Im folgenden Abschnitt wird die Nutzung **heterogener Netzwerke** bewertet:

1. Die Kommunikation im Netzwerk soll mit anerkannten Netzwerkprotokollen geschehen:
   Wuala verwendet zur Kommunikation ein eigenes Protokoll und bietet keine Einsicht auf dieses. Dies ist ein Verstoß gegen das Kerckhoffs'sche Prinzip und daher wird dieser Punkt mit sehr schlecht bewertet.

Tabelle 32: Bewertung heterogene Netzwerke Wuala

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Heterogene Netzwerke	**25**		
Kommunikation mittels anerkannter Netzwerkprotokolle	100	Verwendung eines geheimen Übertragungsprotokolles	5
Zwischennote			**5**

Die sehr schlechte Zwischennote für das Oberziel heterogene Netzwerke in *Tabelle 32: Bewertung heterogene Netzwerke Wuala* zeigt, dass Wuala das Muss-Kriterium in diesem Bereich nicht erfüllen kann.
Im nächsten Abschnitt wird die **Qualität** von Wuala bewertet:

1. Die Kosten für Wuala sehen wie folgt aus:
   Wuala bietet bis zu 5GB Speicher umsonst. Wenn mehr Speicher gebraucht wird, muss dieser bezahlt werden. Das günstigste Angebot ist 20GB für 2,99€/Monat. Für Firmen gibt es ein extra Angebot ab 389,00€/Jahr für 100GB und 5 Nutzer. Wird das Firmenangebot genutzt, gibt es komfortablere Versions- und Gruppenverwaltungsfunktionen. Für das beschriebene Szenario sollte das freie Angebot ausreichen. Falls doch mehr Speicher benötigt wird ist dieser mit 2,99€/Monat günstig zu erwerben. Für ein Team mit fünf Mitgliedern würden die Kosten circa 15,00€/Monat betragen. Dies wird mit befriedigend bewertet.
2. Die Funktionalität von Wuala wird wie folgt bewertet:
   Allgemein erfüllt Wuala die erwarteten Funktionen. Wuala wirbt aktiv mit Sicherheit, hat aber eine schlechte Nutzerauthentifikation und verstößt bei der Kommunikation zwischen Client und Server gegen das Kerckhoffs'sche Prinzip. Da die Daten schon auf Clientseite verschlüsselt werden, ist dies nicht so gravierend wie es sein könnte. Im beschriebenen Szenario kann davon ausgegangen werden, dass der Teamleiter die Nutzernamen der Teammitglieder über Telefon oder deren HAW E-Mailadresse empfangen kann. Daher wird dieser Punkt insgesamt noch mit befriedigend bewertet.
3. Die Zuverlässigkeit von Wuala ist gut.
   Der Anwendungsfall Qualitätstest[58] lässt sich problemlos ausführen.
4. Die Benutzbarkeit von Wuala wird wie folgt bewertet:
   Die Software ist leicht einzurichten. Sie muss lediglich heruntergeladen und installiert werden. Dies ist in fünf Minuten erledigt. Eine Gruppe ist intuitiv mit wenigen Aktionen eingerichtet. Eine Datei lässt sich durch Verschieben in den entsprechenden Ordner mit der Gruppe teilen. Die Datei kann direkt in diesem Ordner bearbei-

[58] Siehe Tabelle 17: Anwendungsfall Qualitätstest

tet werden. Die Verschlüsselung und Synchronisation mit dem Server erfolgt automatisch. Das Anlegen eines Accounts erfolgt schnell und einfach direkt über den Desktop Client. Das Anlegen eines Accounts dauert nicht mehr als fünf Minuten. Das Teilen einer neuen Dateiversion erfolgt automatisch. Hierfür wird eine Minute berechnet. Der Anwendungsfall für den Qualitätstest[59] kann innerhalb von 17 Minuten ausgeführt werden. Auch bei größeren Teams ist durch die zentrale Datenhaltung die Organisation des Datenaustausches einfach und komfortabel. Insgesamt wird dieses Unterziel mit gut bewertet.

5. Die Effizienz von Wuala ist gut.
   Zur Nutzung der Lösung wird nur der entsprechende Client benötigt. Die Synchronisation der Daten erfolgt schnell und effizient.
6. Die Wartbarkeit wird wie folgt bewertet:
   Der Wuala Client informiert selbstständig über neue Updates. Diese können dann einfach vom Nutzer ausgeführt werden. Dieser Punkt wird mit gut bewertet.
7. Die Übertragbarkeit von Wuala hat folgende Bewertung erhalten:
   Wuala bietet für die gängigen Betriebssysteme (Windows, Linux und Mac) einen Client an. Ebenfalls wird ein Client für mobile Geräte, die IOS oder Android nutzen, bereitgestellt. Ein Webzugriff ist nur mit einem explizitem geheimen Link zu der Datei möglich. Insgesamt ergibt dies eine gute Bewertung.

Tabelle 33: Bewertung Qualität Wuala zeigt, dass Wuala eine befriedigende Qualität aufweist. Die Kosten und Funktionalität der Software ziehen die Note etwas nach unten.
Nachdem alle Oberziele bewertet wurden, ist eine **Gesamtauswertung** möglich.

[59] Siehe Tabelle 17: Anwendungsfall Qualitätstest

Tabelle 33: Bewertung Qualität Wuala

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Qualität	**10**		
Kosten	*40*	*5Gb kostenfrei, 20Gb für 2,99€/Monat*	*3*
Funktionalität	*10*	*Werbung mit Sicherheit, aber Lücken bei der Authentifikation von Nutzer und E-Mail Adressen*	*3*
Zuverlässigkeit	*10*	*Fehlerfreies Ausführen vom Anwendungsfall Qualitätstest möglich*	*2*
Benutzbarkeit	*10*	*Einfache und intuitive Bedienung*	*2*
Effizienz	*10*	*Vergleich der benötigten zusätzlichen Hardware*	*2*
Wartbarkeit	*10*	*Client informiert automatisch über Updates*	*2*
Übertragbarkeit	*10*	*Alle gängigen Plattformen werden unterstützt, eingeschränkter Webzugriff*	*2*
Zwischennote			**2,5**

Tabelle 34: Gesamtbewertung Wuala

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Sichere Dateiablage	40		2,7
Ad hoc-Teambildung	25		3
Heterogene Netzwerke	25		5
Qualität	10		2,5
Gesamtnote			**3,3**

Tabelle 34: Gesamtbewertung Wuala zeigt, dass Wuala eine befriedigende Auswertung hat. Das Oberziel heterogene Netzwerke wirkt sich negativ auf die Endnote aus. Wuala erreicht hier eine schlechte Bewertung, da gegen Grundprinzipien der IT-Sicherheit verstoßen wird. Es ist nicht nachvollziehbar, warum für die Netzwerkkommunikation kein sicheres Standartprotokoll wie TLS/SSL verwendet wird.

4.2.4 OwnCloud/Cloudfogger

OwnCloud ist eine Open Source alternative zu Diensten wie Wuala und Dropbox. Ein großer Vorteil ist, dass der Server zum Speichern der Daten selber aufgesetzt werden kann. Dies

kann Vorteile beim Übertragen der Dateien bringen. Im diesem Szenario wird davon ausgegangen, dass die HAW-Hamburg die nötigen Ressourcen besitzt, um ohne Mehraufwand den Server zu betreiben. Da der Source Code öffentlich ist, kann das Produkt wenn nötig komplett an die eigenen Bedürfnisse angepasst werden. Dies setzt voraus, dass das nötige Wissen vorhanden ist. Um die clientseitige Verschlüsselung nicht selbst einrichten zu müssen, wird ein kostenloses Produkt namens Cloudfogger verwendet. Für Cloudfogger ist ein gesonderter Account notwendig. Die Daten werden zum Verschlüsseln und für die Freigabe der Dateien im Team benutzt. Inwieweit sich dieser zusätzliche Konfigurationsaufwand auf das Team auswirkt, wird die Bewertung zeigen.
Im folgenden Abschnitt wird die Bewertung für das Oberziel sichere Dateiablage vorgenommen:

1. Die Einsicht der Daten soll nur durch Teammitglieder möglich sein:
   Da zusätzlich Cloudfogger verwendet wird, werden die Daten automatisch vor dem Transfer zum Server verschlüsselt. So ist gesichert, dass die Daten nur von Teammitgliedern eingesehen werden können. Dies ist eine gute Lösung.
2. Änderungen an den Daten sollen Teammitgliedern zuzuordnen sein:
   Das Log System von OwnCloud zeigt nur die Uhrzeit der Datenänderung an. Es wird nicht gespeichert, wer diese Änderung vorgenommen hat. So wird es schwierig bei der Teamarbeit nachzuvollziehen, welcher Nutzer welche Daten geändert hat. Dies führt zu einer schlechten Bewertung.
3. Ein Schutz vor Hardwaredefekten soll vorhanden sein:
   OwnCloud bietet den Vorteil, dass die Daten auf allen Clients synchronisiert werden. Dies bietet eine gute Datensicherheit. Diese allein reicht jedoch nicht aus, da nicht sichergestellt werden kann, ob ein Client den aktuellen Stand der Daten besitzt. Der Server muss daher eine Datensicherung besitzen, die die Anforderungen erfüllt. In der HAW-Hamburg ist dies der Fall. Die HAW-Hamburg nutz eine virtuelle Serverstruktur. Zusätzlich werden regelmäßig Datensicherungen der virtuellen Server ausgeführt. Dies ist eine gute Lösung.
4. Ein Rücksprung zu einem früheren Datenstand soll möglich sein:
   OwnCloud hält für jede Datei alte Dateiversionen vor, so dass ein Rücksprung zu alten Versionen jederzeit möglich ist. Es erfolgt eine gestaffelte Speicherung alter Versionen. Für die ersten zehn Sekunden wird jeweils alle zwei Sekunden eine Version gespeichert. Für die letzte Minute wird alle zehn Sekunden eine Version gesichert. Für die letzte Stunde wird eine Version jeder Minute vorgehalten. Diese Staffelung erstreckt sich bis auf eine tägliche Sicherung für eine Version. Dies ermöglicht, dass für kurzfristige Änderungen sehr granular Versionen vorgehalten werden. Je weiter die Änderung zurück liegt desto gröber die Version. Insgesamt wird Speicherplatz gespart ohne auf eine gute Versionierung zu verzichten. Dies ist eine gute Lösung.
5. Unbefugte Änderungen werden erkannt und nicht festgeschrieben:

OwnCloud nutzt TLS/SSL zur Kommunikation mit dem Server. Dieses Protokoll sieht die Überprüfung der Datenintegrität vor, so dass Änderungen während der Kommunikation erkannt und nicht festgeschrieben werden. Dies ist eine gute Lösung.

6. Bei Datenübertragung soll eine beidseitige Authentifikation erfolgen:
   OwnCloud nutzt TLS/SSL zur Datenübertragung. Dieses Protokoll unterstützt die beidseitige Authentifikation. Zusätzlich authentifiziert sich der Client mit seinen Nutzerdaten beim Server. Diese Lösung ist gut.
7. Die Verschlüsselung der Daten soll clientseitig erfolgen:
   Durch die zusätzliche Nutzung von Cloudfogger60 wird eine clientseitige Verschlüsselung erzwungen. Cloudfogger nutz hierfür AES-256 für die Verschlüsselung der Daten. Jede Datei wird mit einem eigenen Schlüssel verschlüsselt. Der Schlüssel wird dann zusätzlich mit RSA-2048 verschlüsselt und an die Datei angehängt. So lässt sich die Datei mit anderen Cloudfogger Nutzern teilen. Dies ist eine gute Lösung.

Tabelle 35: Bewertung sichere Dateiablage OwnCloud/Cloudfogger

Ober-/Unterziel	**Gewichtung**	**Anmerkungen**	**Note**
Sichere Dateiablage	**40**		
Einsicht der Daten nur durch Teammitglieder möglich	20	Verschlüsselung der Daten vom Client durch Cloudfogger	3
Änderungen sind Mitgliedern zuzuordnen	*10*	*Kein Log der Nutzernamen, nur Speicherung der Änderungszeit*	*4*
Schutz vor Hardwaredefekten	*10*	*Sicherung der Daten durch Serverstruktur*	*2*
Rücksprung zu früherem Datenstand möglich	*10*	*Mehrere alte Dateiversionen pro Datei werden vorgehalten*	*2*
Unbefugte Änderungen werden erkannt und nicht festgeschrieben	15	Nutzung von TLS/SSL	2
Beidseitige Authentifikation bei Datenübertragung	10	Nutzung von TLS/SSL und Authentifikation über Nutzerdaten	2
Verschlüsselung clientseitig	25	Cloudfogger bietet nach Einrichtung eine gute, automatische und clientseitige Verschlüsselung	2
Zwischennote			**2,7**

[60] [16]http://www.cloudfogger.com/de/home/security.aspx

Tabelle 35: Bewertung sichere Dateiablage OwnCloud/Cloudfogger zeigt das OwnCloud in Zusammenarbeit mit Cloudfogger eine akzeptable Lösung für den Bereich sichere Dateiablage darstellt. Dieser Lösungsansatz genügt allen Muss-Kriterien. Eine Verbesserung des Dateiänderungslogs würde die Note weiter verbessern.
Im nächsten Abschnitt wird das Oberziel **ad hoc-Teambildung** bewertet:

1. Die Rechteverwaltung soll allein vom Teamgründer oder einer autorisierten Person ausgehen:
   In OwnCloud ist es möglich jedem User Rechte als Gruppenadministrator zu vergeben. Dieser kann dann eigene User für sein Team erstellen. Wenn schon jedes Teammitglied einen Account besitzt, kann der Teamgründer einen Ordner für das Team freigeben. Hier können gezielt Rechte für jede Freigabe erstellt werden. Dies ist sehr flexibel und eine gute Lösung.
2. Die Erstellung eines Teams soll allein vom Teamgründer abhängen:
   Haben alle Teammitglieder einen OwnCloud Account, ist dies möglich. OwnCloud unterstützt die Anbindung an ein bestehendes LDAP[61]. So kann der Administrator für alle HAW-Hamburg Mitarbeiter einen OwnCloud Zugang bereitstellen. Ist dies nicht der Fall, muss entweder der OwnCloud Administrator entsprechende Accounts anlegen oder der Teamleader muss vom Administrator eine Gruppe und Gruppenadministratorrechte zugewiesen bekommen. Dann ist es dem Teamleader möglich eigene Accounts für seine Teammitglieder anzulegen. Insgesamt ist dies eine befriedigende Lösung.
3. Es soll eine sichere Nutzeridentifikation und Authentifikation erfolgen:
   OwnCloud unterstützt die Einbindung in ein LDAP. Daher können Passwörter und Nutzernamen nach HAW-Hamburg Richtlinien erzwungen werden. Dies hat den Vorteil, dass der Account direkt von HAW-Hamburg verifiziert wird. Dies entspricht den gewünschten Richtlinien für Authentifikation und Identifikation. Werden die Accounts direkt in OwnCloud angelegt, können beliebige Passwörter[62] vergeben werden. Dies entspricht nicht den Anforderungen. Cloudfogger bietet keine Möglichkeit ein bestehendes LDAP zu verwenden. Cloudfogger nutzt E-Mail Verifikation. Die Sicherheit der Passwörter wird in keiner Weise erzwungen. Cloudfogger empfiehlt zwar die Verwendung sicherer Passwörter auf der Webseite, erzwingt die Nutzung aber nicht im Client. Es kann ein Passwort mit einem Zeichen genutzt werden. Dies führt zu

[61] Lightweight Directory Access Protocol
[28] http://msdn.microsoft.com/en-us/library/windows/desktop/aa367008%28v=vs.85%29.aspx
[62] Siehe Tabelle 6: Sichere Passwörter

Abzügen in der Bewertung. Durch den gravierenden Sicherheitsmangel in Cloudfogger führt dies zu einer insgesamt schlechten Bewertung.

Tabelle 36: Bewertung ad hoc-Teambildung OwnCloud/Cloudfogger

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Ad hoc-Teambildung	**25**		
Rechteverwaltung vom Teamgründer oder autorisierter Person	*33*	*Gute und flexible Rechtevergabe*	*2*
Erstellung eines Teams allein vom Teamgründer abhängig	*33*	*Bei bestehenden OwnCloud Accounts problemlos, ansonsten Abhängigkeit vom OwnCloud Administrator*	*3*
Sichere Nutzeridentifikation und Authentifikation	33	Einbindung in LDAP und AD möglich, Cloudfogger erlaubt die Verwendung sehr unsicherer Passwörter (1 Zeichen Passwort erlaubt)	4
Zwischennote			**3**

Tabelle 36: Bewertung ad hoc-Teambildung OwnCloud/Cloudfogger zeigt, dass OwnCloud/Cloudfogger im Bereich ad hoc-Teambildung eine befriedigende Lösung mit einigen Schwächen darstellt. Das Muss-Kriterium der *sicheren Nutzeridentifikation und Nutzerauthentifikation* kann nicht erfüllt werden, da nicht auf hinreichend sichere Passwörter geachtet wird.
Im folgenden Abschnitt wird das Oberziel **heterogene Netzwerke** bewertet:

1. Die Kommunikation soll mittels anerkannter Netzwerkprotokolle erfolgen:
   Für die Kommunikation zwischen Client und Server wird TLS/SSL verwendet. Dies entspricht den momentanen Empfehlungen[63] für sichere Übertragung und ist eine gute Lösung.

63

[19]https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102_pdf.pdf?__blob=publicationFile

Tabelle 37: Bewertung heterogene Netzwerke OwnCloud/Cloudfogger

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Heterogene Netzwerke	**25**		
Kommunikation mittels anerkannter Netzwerkprotokolle	100	Verwendung von TLS/SSL zur Datenübertragung	2
Zwischennote			**2**

Tabelle 37: Bewertung heterogene Netzwerke OwnCloud/Cloudfogger zeigt, dass OwnCloud gut für die Verwendung in heterogenen Netzwerken geeignet ist. Dies folgt aus der Verwendung von Standards für die Datenübertragung.
Die Bewertung der **Qualität** von OwnCloud zusammen mit Cloudfogger wird im folgenden Abschnitt vorgenommen:

1. Die Bewertung der Kosten setzt sich wie folgt aus den Kosten für OwnCloud und Cloudfogger zusammen:
   OwnCloud ist ein Open Source Produkt und von daher kostenfrei zu betreiben. Cloudfogger ist ebenfalls kostenfrei erhältlich. Der Betreiber von Cloudfogger behält sich vor den kommerziellen Gebrauch später mit Kosten zu versehen. Dies gibt Abzüge in der Bewertung. Insgesamt ist dies eine befriedigende Lösung.
2. Die Funktionalität von OwnCloud und Cloudfogger ist gut.
   Es konnten keine Unterschiede zwischen beworbenen und tatsächlichen Funktionen festgestellt werden. Cloudfogger verschlüsselt die Dateien zuverlässig bevor diese von OwnCloud synchronisiert werden.
3. Die Zuverlässigkeit der Lösung wird als gut angesehen.
   Es gab keine Probleme beim Anwenden des Anwendungsfalls Qualitätstest[64].
4. Die Benutzbarkeit der Produkte wird wie folgt bewertet:
   OwnCloud und Cloudfogger haben jeder für sich ein einheitliches intuitives Design. Abzüge gibt es dafür, dass ein höherer Konfigurationsaufwand besteht. Der OwnCloud Server muss einmalig für das Unternehmen eingerichtet werden. Es müssen Nutzer für OwnCloud und Nutzer für Cloudfogger angelegt werden. Insgesamt geschieht die Einrichtung schnell und intuitiv. Die Einrichtung beider Accounts dauert insgesamt circa 10 Minuten. Sind die Programme erst eingerichtet, erfolgt die Verschlüsselung und Synchronisation automatisch. Die Bereitstellung einer Datei erfolgt automatisch nach dem Speichern in dem entsprechenden Ordner. Dies geschieht innerhalb von einer Minute. Der Anwendungsfall Qualitätstest ist so in 22 Minuten abgeschlossen. Insgesamt ist dies ein befriedigendes Ergebnis.

[64] Siehe Tabelle 17: Anwendungsfall Qualitätstest

5. Die Effizienz der Lösung ist befriedigend. Es muss ein Server für OwnCloud eingerichtet werden. Da diese in der HAW-Hamburg virtuelle Server sind und eine gute Infrastruktur vorhanden ist, ist kein zusätzlicher Hardwareaufwand nötig. Daher gibt es hierfür nur einen geringen Abzug. Insgesamt ist dies eine befriedigende Lösung.
6. Die Wartbarkeit dieser Lösung ist insgesamt befriedigend.
   OwnCloud wird aktiv weiterentwickelt. Neue Versionen erscheinen in regelmäßigen Abständen. Wurde OwnCloud aus einem Repository installiert, werden die Updates automatisch eingespielt. Ansonsten müssen diese von Hand installiert werden. Insgesamt ist dies eine befriedigende Lösung.
7. Die Übertragbarkeit von OwnCloud und Cloudfogger ist gut.
   Cloudfogger bietet automatische Clientupdates an. Der OwnCloud Server bietet fertige Linux Packages zur Installation an. Wird dieser Weg genutzt, können durch die Updatemechanismen des Betriebssystems automatisch Updates eingespielt werden. Bei Nutzung einer anderen Variante ist ein automatisches Update nicht möglich. OwnCloud ist für alle gängigen Betriebssysteme erhältlich. Dies schließt Android und IOS mit ein. Cloudfogger hingegen ist nicht für Linux erhältlich. Insgesamt führt dies zu einer befriedigenden Lösung.

Tabelle 38: Bewertung Qualität OwnCloud/Cloudfogger

Ober-/Unterziel	**Gewichtung**	**Anmerkungen**	**Note**
Qualität	**10**		
Kosten	*40*	*OwnCloud kostenfrei, Cloudfogger eventuell später nicht mehr kostenlos*	*3*
Funktionalität	*10*	*Keine Beanstandung*	*2*
Zuverlässigkeit	*10*	*Anwendungsfall Qualitätstest ohne Probleme ausführbar*	*2*
Benutzbarkeit	*10*	*Abzüge für den erhöhten Konfigurationsaufwand*	*3*
Effizienz	*10*	*Aufsetzten des OwnCloud Servers notwendig*	*3*
Wartbarkeit	*10*	*Automatische Updates möglich*	*3*
Übertragbarkeit	*10*	*Cloudfogger unterstützt kein Linux*	*3*
Zwischennote			**2,8**

Tabelle 38: Bewertung Qualität OwnCloud/Cloudfogger zeigt, dass dieser Lösungsansatz im Bereich Qualität eine befriedigende Lösung darstellt.
Anhand der ausgewerteten Zwischennoten der Oberziele wird im folgenden Abschnitt eine **Gesamtnote** für OwnCloud/Cloudfogger ermittelt.

Tabelle 39: Gesamtbewertung OwnCloud/Cloudfogger

Ober-/Unterziel	Gewichtung	Anmerkungen	Note
Sichere Dateiablage	40		2,7
Ad hoc-Teambildung	25		3
Heterogene Netzwerke	25		2
Qualität	10		2,8
Gesamtnote			**2,6**

Tabelle 39: Gesamtbewertung OwnCloud/Cloudfogger zeigt, dass dieser Lösungsansatz eine befriedigende Lösung darstellt. In Zukunft kann die Verwendung von Cloudfogger als Verschlüsselungstool problematisch werden, da Cloudfogger sich die Einführung einer kostenpflichtigen Variante vorhält. Positiv ist zu sehen, dass die OwnCloud Community die Integration einer clientseitigen Verschlüsselung plant.

4.2.1 Fazit der Bewertungen

In diesem Kapitel werden abschließend die Bewertungen der Lösungen verglichen.

Tabelle 40: Vergleich Lösungsansätze

Lösungsansatz	Note
USB-Stick	3,5
E-Mail	3,4
Wuala	3,3
OwnCloud/Cloudfogger	2,6

Tabelle 40: Vergleich Lösungsansätze zeigt die Noten der einzelnen Lösungsansätze. Generell schließen die Lösungen mit zentraler Datenhaltung, Wuala und OwnCloud/Cloudfogger, besser ab, als die Lösungen mit direkter Datenübergabe. Dies zeigt, dass die zentrale Dateiverwaltung besser funktioniert, als die Daten direkt von Person zu Person zu übergeben. Verantwortlich hierfür ist, dass die Synchronisation der Daten bei direkter Datenübergabe wesentlich komplexer ist. Die größten Schwierigkeiten der Cloud Dienste liegen in der sicheren Dateiablage und der Nutzerauthentifikation.
Auffällig an den ausgewerteten Noten ist, dass kein Lösungsansatz gut abgeschnitten hat. Keine der Lösungen konnte allen Muss-Kriterien gerecht werden. Der beste Ansatz für eine Lösung ist, in diesem Szenario, OwnCloud zusammen mit Cloudfogger. OwnCloud und Cloudfogger haben beide das Problem, dass keine sicheren Passwörter erzwungen werden. Diesem Fehler kann durch gute Absprachen innerhalb des Teams entgegengewirkt werden. Dies ist keine gute Lösung, könnte aber übergangsweise ausreichen. Wenn nur sichere Passwörter verwendet werden, erfüllt OwnCloud/Cloudfogger alle Muss-Kriterien des Szenarios. Ebenfalls besteht die Möglichkeit die entsprechenden Fehler in OwnCloud selbst zu beheben oder diese in der Community zu diskutieren und als Feature Request bei Own-

Cloud anzugeben. So ist es möglich diesen Lösungsansatz zu verbessern und den bestehenden Ansprüchen anzupassen. Cloudfogger bietet diese Möglichkeit nicht, da dieses Produkt nicht Open Source ist. Hier muss der Hersteller aktiv werden.

5 Fazit

In dieser Arbeit wurde untersucht, wie eine sichere Dateiablage bei ad hoc-Teams in heterogenen Netzwerken sinnvoll verwirklicht werden kann. Anhand des Szenarios wurden vier Vertreter verschiedener Lösungskategorien gewählt und dann anhand einer Nutzwertanalyse bewertet.
Für die Nutzwertanalyse wurden die Oberziele aus dem Szenario herausgearbeitet. Die entstandenen drei Oberziele *sichere Dateiablage, ad hoc-Teambildung* und *heterogene Netzwerke* wurden jeweils in Unterziele aufgebrochen. Zusätzlich zu den herausgearbeiteten Oberzielen wurde die Qualität der Lösung als Oberziel hinzugefügt. Die Oberziele wurden nach Wichtigkeit priorisiert. Nachdem die Unterziele herausgearbeitet wurden, konnten diese gewichtet und in Muss- und Kann-Kriterien eingeteilt werden. So war es möglich, dass die Unterziele je nach Wichtigkeit mehr Einfluss auf die Endbewertung haben konnten.
Nachdem das Grundgerüst des Zielsystems aufgestellt wurde, konnte eine technische Analyse der Unterziele erfolgen. Anhand der Analyse konnte festgelegt werden unter welchen Umständen ein Unterziel als erfüllt gelten kann. Es wurden Erwartungen an die technische Ausführung der Unterziele gestellt.
Nach der Analyse der Teilsysteme war das Zielsystem komplett aufgestellt. Dieses wurde als einheitlicher Bewertungsmaßstab für die vier ausgewählten Lösungsansätze angewendet. Die Lösungsansätze waren USB-Stick, E-Mail, Wuala und OwnCloud/Cloudfogger.
Laut Nutzwertanalyse konnte OwnCloud/Cloudfogger als beste Lösung abschließen. Auffällig war, dass ein Dienst wie Wuala, der aktiv mit der Sicherheit des Produktes wirbt, Grundsätze der IT-Sicherheit missachtet. Für die Datenübertragung wird ein geheimes Protokoll verwendet, was ein klarer Verstoß gegen das Kerckhoffs'sche Prinzip[65] ist. Keine Lösung konnte alle Muss-Kriterien des Zielsystems erfüllen.
Dies zeigt, dass das komplexe Thema IT-Sicherheit immer noch nicht im Fokus der Softwareentwicklung steht oder auch noch nicht komplett verstanden wurde. Gerade im Bereich des verteilten Zugriffs von mehreren Nutzern auf verschlüsselte Dateien sind bessere Lösungen erforderlich.

[65] Siehe Kapitel 2.1 Kerckhoffs-Prinzip

6 Ausblick

Aktuelle Datenschutzskandale, die unter anderem durch die National Security Agency verursacht wurden, zeigen, dass eine sichere Speicherung der Daten notwendig ist. Durch diese Skandale wird sichere Datenspeicherung für die breite Masse interessant. Es ist nötig die Lücke zwischen dem komfortablen Speichern in der Cloud und der Datensicherheit zu schließen. Wie diese Arbeit zeigt, gibt es bereits gute Ansätze, aber alle getesteten Lösungen zeigen noch Schwächen.

Neben Wuala gibt es noch andere Produkte, die die Sicherheit der Nutzerdaten versprechen. Es zeichnen sich zwei verschiedene Ansätze ab. Der erste Ansatz ist eine clientseitige Verschlüsselung. Mega[66]wäre ein weiterer Anbieter, der diesen Ansatz verfolgt. Es wäre zu evaluieren inwieweit dieser Anbieter die Schwächen von Wuala vermeidet. Der andere Ansatz ist, die Daten auf dem eigenen Server direkt am Arbeitsplatz zu speichern und trotzdem die Vorteile der Cloud zu genießen. Mit OwnCloud wäre dies möglich, aber die Konfiguration ist nicht einfach. Im Gegensatz hierzu wäre Protonet[67] eine Alternative. Protonet bietet einen kompletten, vorkonfigurierten Server inklusive der Hardware an. Diese kleinen Server könnten dann direkt vom Teamleader angeschlossen und betrieben werden. Auch in diesem Fall müsste geprüft werden, inwiefern die angelegten Sicherheitsstandards von diesem Produkt eingehalten werden können.

Momentane Lösungen bauen auf asynchronen Verschlüsselungen und der Verteilung der öffentlichen Schlüssel auf. Dieses Verfahren kann zur Lösung dieser Probleme verwendet werden, aber die schlechten Noten der Bewertung zeigen, dass dies kompliziert und nicht optimal ist.

Entweder muss die Struktur zum Austausch öffentlicher Schlüssel verbessert und vereinheitlicht werden oder andere Ansätze müssen vorangetrieben werden. Die momentane Struktur sieht vor, dass die Vertraulichkeit eines Zertifikates von der Vertraulichkeit der ausstellenden Zertifizierungsstelle abgeleitet wird. Allerdings ist es nicht leicht herauszufinden, ob der Zertifizierungsstelle vertraut werden kann. Es gilt zu überlegen, ob ein staatliches Zertifizierungssystem interessant wäre. Eventuell würde ein staatliches Prüfsiegel für Zertifizierungsstellen ausreichen. Weiterhin wäre eine komfortable Bereitstellung des öffentlichen Schlüssels ein wünschenswertes Ziel.

Neben diesen Problemen wurde beobachtet, dass sehr gute Ansätze durch simples Missachten von IT-Sicherheitsstandards Lücken in der Sicherheit aufweisen. Entweder ist dies

[66] [21] https://mega.co.nz/

[67] [20] http://www.protonet.info/de/

Unwissen oder durch Strukturprobleme in der Software ist das Einhalten einfacher Richtlinien nur sehr schwer möglich. Dies zeigt, dass IT-Sicherheit nicht später nachimplementiert, sondern von Anfang an berücksichtigt werden sollte.
Die IT ist auf dem richtigen Weg, aber noch nicht im umfassend sicheren Datenverkehr angekommen.

Literaturverzeichnis

[1] F. Petitcolas, "la cryptographie militaire," [Online]. Available: http://www.petitcolas.net/fabien/kerckhoffs/#english. [Accessed 30 July 2013].

[2] C. Eckert, IT-Sicherheit, Oldenbourg: Oldenbourg Wissenschaftsverlag GmbH, 2013.

[3] D. D. J. Berwanger, D. E. Winter and J.-H. Krumme, "Gabler Wirtschaftslexikon, Stichwort: juristische Person," Springer Gabler Verlag, , [Online]. Available: http://wirtschaftslexikon.gabler.de/Archiv/1028/juristische-person-v13.html. [Accessed 2 August 2013].

[4] "Sicherheitstechnische Anforderungen an die Kommunikationsverbindung Telearbeitsrechner - Institution," Bundesamt für Sicherheit in der Informationstechnik, [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m05/m05051.html. [Accessed 6 Oktober 2013].

[5] "Passwörter," Bundesamt für Sicherheit in der Informationstechnik, [Online]. Available: https://www.bsi-fuer-buerger.de/BSIFB/DE/MeinPC/Passwoerter/passwoerter_node.html . [Accessed 2 Juni 2013].

[6] H. Balzert, R. Koschke, H. Balzert, U. Lämmel, P. Liggesmeyer and J. Quante, Lehrbuch Der Softwaretechnik: Basiskonzepte Und Requirements Engineering, 2009.

[7] Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen, "Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung," [Online]. Available: http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/QES/Veroeffentlichungen/Algorithmen/2013Algorithmenkatalog.pdf?__blob=publicationFile&v=1. [Accessed 8 August 2013].

[8] "Verwendung von TLS/SSL," Bundesamt für Sicherheit in der Informationstechnik, [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_

content/m/m05/m05066.html. [Accessed 18 July 2013].

[9] "Gefährliche Kuckuckseier: E-Mails mit falschem Absender," Bundesamt für Sicherheit in der Informationstechnik, [Online]. Available: https://www.bsi-fuer-buerger.de/BSIFB/DE/GefahrenImNetz/GefaelschteAbsenderadressen/gefaelschteabsenderadressen_node.html. [Accessed 5 Oktober 2013].

[10] M. Borgmann, T. Hahn, M. Herfert, T. Kunz, M. Richter, U. Vieberg and S. Vow´e, "On the Security of Cloud Storage Services," Fraunhofer-Institut für Sichere Informationstechnologie, [Online]. Available: https://www.sit.fraunhofer.de/de/angebote/projekte/cloud-studie/. [Accessed 12 September 2013].

[11] L. Meisser, "Fraunhofer study on cloud storage security," Wuala, [Online]. Available: http://wualablog.blogspot.de/2012/05/fraunhofer-study-on-cloud-storage.html. [Accessed 10 Oktober 2013].

[12] D. Grolimund, L. Meisser, S. Schmidt and R. Wattenhofer, "Cryptree: A Folder Tree Structure for Cryptographic File Systems," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, CH-8092 Zurich, [Online]. Available: http://dcg.ethz.ch/publications/srds06.pdf. [Accessed 21 September 2013].

[13] "Cloudfogger Sicherheit," Cloudfogger, [Online]. Available: http://www.cloudfogger.com/de/home/security.aspx. [Accessed 12 Oktober 2013].

[14] Bundesamt für Sicherheit in der Informationstechnik, "Kryptographische Verfahren:Empfehlungen und Schlüssellängen," Bundesamt für Sicherheit in der Informationstechnik, [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102_pdf.pdf?__blob=publicationFile. [Accessed 10 August 2013].

[15] "Protonet," http://www.protonet.info/de/, [Online]. Available: http://www.protonet.info/de/. [Accessed 20 Oktober 2013].

[16] "Mega," https://mega.co.nz/, [Online]. Available: https://mega.co.nz/. [Accessed 20 August 2013].

[17] "Bundesnetzagentur," [Online]. Available: http://www.bundesnetzagentur.de. [Accessed 10 July 2013].

[18] "Dropbox," [Online]. Available: https://www.dropbox.com/. [Accessed 14 July 2013].

[19] "Open Source," [Online]. Available: http://opensource.org/osd-annotated. [Accessed 17 August 2013].

[20] "Wuala," [Online]. Available: http://www.wuala.com/de/ . [Accessed 14 September 2013].

[21] "OwnCloud," [Online]. Available: http://owncloud.org/ . [Accessed 15 September 2013].

[22] "Cloudfogger," [Online]. Available: http://www.cloudfogger.com/de/ . [Accessed 19 September 2013].

[23] "Microsoft Sharepoint," [Online]. Available: http://office.microsoft.com/de-de/sharepoint/. [Accessed 20 Oktober 2013].

[24] "LDAP," Microsoft, [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa367008%28v=vs.85%29.aspx. [Accessed 17 Oktober 2013].

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____ _____