TU WIEN Informatics

Konzeption und Implementierung einer Cloud-unterstützten Peer-to-Peer-Backuplösung

P2P-Backuplösung mit Cloud-Unterstützung

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Gregor Lucny, BSc
Matrikelnummer 00725836

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Thomas Grechenig

Wien, 11. Mai 2022

Unterschrift Verfasser

Unterschrift Betreuung

TU WIEN Informatics

Conception and implementation of a cloud-supported peer-to-peer-backup-solution

P2P-Backupsolution with cloud-support

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Gregor Lucny, BSc
Registration Number 00725836

to the Faculty of Informatics

at the TU Wien

Advisor: Thomas Grechenig

Vienna, 11th May, 2022

Signature Author

Signature Advisor

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Konzeption und Implementierung einer Cloud-unterstützten Peer-to-Peer-Backuplösung

P2P-Backuplösung mit Cloud-Unterstützung

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Gregor Lucny, BSc
Matrikelnummer 00725836

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Thomas Grechenig

Wien, 11. Mai 2022

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Gregor Lucny, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 11. Mai 2022

Gregor Lucny

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich während der Verfassung meiner Diplomarbeit unterstützt, motiviert und angetrieben haben.

Mein erster Dank gebührt meinem Diplomarbeitsbetreuer Martin Maier, der mich während der Verfassung meiner Diplomarbeit betreut hat. Er hat mir die Chance gegeben, bei meinem dritten Anlauf meine Diplomarbeit abzuschließen und mich über ein Jahr lang unterstützt. Seine Genauigkeit und sein konstruktives Feedback haben mir sehr dabei geholfen diese Diplomarbeit zu verfassen.

Des Weiteren möchte ich mich bei meiner Lebensgefährtin Michaela Tiefenbacher bedanken, die mir den Ansporn gegeben hat, die Diplomarbeit und mein Master-Studium doch noch abzuschließen. Sie hat des öfteren meine Diplomarbeit auf Grammatik und Rechtschreibung Korrektur gelesen. Vor allem in der letzten Phase hatte sie viel Verständnis für meine langen Abende zur Verfassung der Arbeit.

Ich möchte mich ebenfalls bei meinem engen Freund Magnus Köhler bedanken, der für technische Diskussionen zur Verfügung stand und mit mir Ideen und Lösungen meiner Diplomarbeit besprochen hat.

Zuletzt möchte ich mich noch bei meinen Eltern Gabriele und Peter Lucny bedanken, die bis zuletzt an den Abschluss der Arbeit geglaubt haben.

Acknowledgements

I would like to thank everyone who supported and motivated me during the writing of my diploma thesis.

First I want to express my thanks to my supervisor Martin Maier who supervised me while I wrote my diploma thesis. He gave me the chance to complete my thesis on my third attempt and accompanied me for over a year. His accuracy and constructive feedback really helped me to write this diploma thesis.

Many thanks also go to my partner Michaela Tiefenbacher. She encouraged me to finalize my diploma thesis and my master-studies. She proof read my diploma thesis multiple times. Especially during the last few weeks she showed patience for the long evenings I spent on writing my thesis.

Additionally I want to thank my dear friend Magnus Köhler who was always available for technical discussions. I was able to brainstorm and discuss ideas and solutions of my diploma thesis with him.

At last I want to thank my parents Gabriele and Peter Lucny, who neved stopped believing in me to finish my thesis.

Kurzfassung

In der heutigen Zeit befindet sich eine Vielzahl an Informationen in digitaler Form auf Computern. Wir verlassen uns auf die Vertraulichkeit, Integrität und Verfügbarkeit dieser Informationen, obwohl diese durch Softwarefehler, Hardwaredefekte und Katastrophen oder durch Bedienungsfehler und Angriffe manipuliert, zerstört oder von Unbefugten gelesen werden könnten. Backupsysteme können vor Datenverlust schützen. Anforderungen wie die Durchführung von Off-Site-Backups oder eine Änderung der Sicherungsinfrastruktur werden nicht von allen Backupsystemen unterstützt. Eine eingehende Analyse der Sicherheitsziele der Anwender ist oft nicht erkennbar. Proprietäre Lösungen können nicht selbst überprüft oder weiterentwickelt werden. Dadurch ergibt sich eine Abhängigkeit zum Hersteller bei neuen technologischen Entwicklungen. Peer-to-Peer-Systeme haben aufgrund von fluktuierenden Teilnehmern das Problem der Verwaltung und Verfügbarkeit anderer Teilnehmer. Im privaten Umfeld führt eine asymmetrische Bandbreite auch zu langsamen Sicherungen zu mehreren Peers. Lösungen mit zentraler Infrastruktur wie Cloud-Anbieter haben einen Single-Point-of-Failure und führen oft zu einem Vendor-Lock-In bei dem Anbieter.

In dieser Arbeit wird ein Backupsystem vorgeschlagen, welches sich vorrangig an Privatpersonen richtet. Es nutzt sowohl Peer-to-Peer- als auch Cloud-Komponenten und basiert auf echten sozialen Beziehungen der Teilnehmer. Es werden die Anforderungen an das Backupsystem definiert, die im Peer-to-Peer- und Cloud-Umfeld erfüllt werden müssen. Es wird eine Sicherheitsanalyse auf Basis der Lösungsidee und den Anforderungen durchgeführt und dabei der Schutzbedarf ermittelt. Daraus werden mit einer Bedrohungs- und Risikoanalyse Gefährdungen abgeleitet und darauf aufbauend Sicherheitsmaßnahmen definiert. Die Anforderungen und die Ergebnisse der Sicherheitsanalyse sind die Basis für die Konzeption und Proof-of-Concept-Implementierung der Backuplösung in Java. Diese wird anschließend mit einem definierten Testdatenset erprobt und die Erfüllung der Anforderungen und Maßnahmen strukturiert evaluiert.

Die Diplomarbeit zeigt, dass die vorgeschlagene Lösung im privaten Umfeld zur Datensicherung eingesetzt werden kann. Es können jedoch nicht alle erarbeiteten Sicherheitsmaßnahmen wie beispielsweise die Verschlüsselung der eingebetteten Datenbank nach aktuellen Sicherheitsempfehlungen umgesetzt werden.

Keywords: *Backupsystem, Datensicherung, Datenwiederherstellung, Peer-to-Peer, Cloud, Sicherheitsanalyse*

Abstract

A lot of information is found in digital form on computers. We rely on the confidentiality, integrity and availability of this information, although it could be manipulated, destroyed or accessed by unauthorized parties because of software or hardware defects, catastrophies, user errors or attacks. Backup solutions can protect against data loss. Requirements such as the execution of offsite-backups or changes in the backup-infrastructure are not always supported by available solutions. A thorough analysis regarding the user's security goals is often missing. Proprietary solutions cannot be audited or further developed by the user. This leads to a dependency on the software-provider. Peer-to-peer systems have to deal with fluctuating peers and thus problems regarding the management and availability of peers. Because of asymmetric internet connections in private settings backups are slow when backing up to multiple peers. Central solutions from cloud-providers have a single-point-of-failure and result in vendor-lock-in.

This thesis proposes a backup system which addresses mainly individuals, not businesses or corporations. It uses peer-to-peer and cloud components and is based on real social connections between peers. This thesis defines the requirements for the peer-to-peer based and cloud supported backup system. A security analysis including an evaluation of the security goals is done based on the proposed solution and its defined requirements. A threat and risk analysis is conducted to define security measures. The requirements and measures are the basis for the design and implementation of the backup solution in Java. The backup solution is tested with a defined set of data and the fulfillment of all defined requirements and measures is evaluated.

The thesis shows that the proposed solution can be used in a private setting to backup data. However, not all identified measures like the encryption of the embedded database can be implemented according to current security recommendations.

Keywords: *backup system, data backup, data recovery, peer-to-peer, cloud, security analysis*

Inhaltsverzeichnis

KAPITEL 1

Einleitung

1.1 Problemstellung

Immer mehr wichtige Informationen befinden sich in digitaler Form auf Computern. Die Bandbreite geht von persönlichen Emails, Urlaubsfotos und privaten Videos bis hin zu digitalisierten Verträgen, Geschäftsberichten und anderen unternehmensrelevanten Unterlagen. In der Welt der IT ist jeder Teilnehmer, egal ob privater Benutzer oder multinationales Unternehmen, ständigen Bedrohungen ausgesetzt. Angriffe auf Privatpersonen betreffen oftmals die Verschlüsselung von persönlichen Daten und die Erpressung eines Geldbetrages, um wieder Zugriff auf diese Daten zu erhalten. Jede Person ist daher angehalten, ihre relevanten Daten vor Verlust zu schützen. Auch Unternehmen sind interessante Ziele für Angriffe. Dies wird auch in den Nachrichten immer wieder berichtet[1,2,3,4]. Im Laufe dieser Angriffe werden meist Daten zerstört, überschrieben oder auf unzureichend geschützte Datenbestände und Backups zugegriffen. Diese Daten enthalten oft sensible Informationen und ihr Verlust oder der Zugriff durch Unbefugte kann zu finanziellem Schaden oder Imageverlust führen oder die Rechte der Betroffenen verletzen. Daher sollten alle für eine Person oder ein Unternehmen relevanten Daten vertraulich gespeichert, vor Fremdzugriffen geschützt und vor dem Verlust bewahrt werden. Aber auch andere Gründe wie ein Hardwareausfall, ein Irrtum des Dateneigentümers, ein Software- oder Konfigurationsfehler oder eine Katastrophe können zu Datenverlust führen.

[1] https://orf.at/stories/3210927/ - besucht am 24.04.2022
[2] https://orf.at/stories/3219724/ - besucht am 24.04.2022
[3] https://www.faz.net/aktuell/wirtschaft/unternehmen/hessen-hacker-erbeuten-daten-und-wollten-versicherung-erpressen-17448520.html - besucht am 24.04.2022
[4] https://www.golem.de/news/vielfliegerprogramm-hacker-stehlen-millionen-air-india-meilen-1606-121811.html - besucht am 24.04.2022

Es werden daher Mechanismen benötigt um Daten vor Datenverlust zu schützen. Diese Mechanismen müssen eine Reihe von Schutzzielen erfüllen, um geeignete Backuplösungen darzustellen. Vertraulichkeit wird benötigt, damit nur autorisierte Parteien Zugriff auf diese Daten haben. Die Integrität der Daten sowie eine gute Verfügbarkeit sind notwendig, um Datensicherungen oder -wiederherstellungen zeitnah durchführen zu können.

Es existieren bereits einige Tools, mit denen die Datensicherung durchgeführt werden kann. Die Lösungen treten in unterschiedlicher Art auf: Backups können auf lokalen oder externen Festplatten angelegt werden oder werden auf Netzwerkspeichern im lokalen Netzwerk abgelegt. Es gibt Verfahren, mit denen die Datensicherung mithilfe von Peer-to-Peer-Technologien auf dezentralen Systemen durchgeführt wird. In den letzten Jahren haben sich auch Cloud-Anbieter mit Lösungen zur Datensicherung etabliert. Die Möglichkeiten reichen dabei von der Verwendung der kostenlosen Online-Speicher wie beispielsweise Dropbox und Google Drive zu kostenpflichtigen Enterprise-Backuplösungen.

Diese Tools können zur Datensicherung verwendet werden. Allerdings unterstützen diese die geforderten Schutzziele wie Vertraulichkeit, Integrität oder Verfügbarkeit oft nur teilweise. Einige Lösungen sind impraktikabel, nur in veralteter Version oder nur als Proof-of-Concept verfügbar. Peer-to-Peer basierte Lösungen haben aufgrund ihrer dezentralen Architektur einige Schwachpunkte (Zuo u. a. [88]). Bei Systemen mit hoher Teilnehmerfluktuation entsteht erhöhter Kommunikations- und Speicherbedarf zur Sicherstellung der Verfügbarkeit. Des Weiteren kann es zu ungleicher Lastverteilung und damit zur Belastung oder dem Ausfall einzelner Peers kommen. Kermarrec u. a. [37] nennen als Probleme im Peer-to-Peer-Umfeld die geringe Upload-Verfügbarkeit und die geringe Upload-Bandbreite der einzelnen Teilnehmer. Im Jahr 2020 war bei Festnetzanschlüssen die weltweite durchschnittliche Upload-Geschwindigkeit 44,1 Mbps gegenüber einer Download-Geschwindigkeit von 84,33 Mbps, bei mobilen Anschlüssen betrug die Upload-Geschwindigkeit 10,99 Mbps gegenüber 34,82 Mbps beim Download [75]. Aufgrund dieser Diskrepanz ist die Datenmenge, die ein Peer-to-Peer-Teilnehmer in einer bestimmten Zeit bei einem anderen Peer sichern kann, beschränkt. Nach Silva u. a. [71] ergeben sich bei Verwendung einer Cloud-Umgebung zusätzliche Herausforderungen in Bezug auf die Privatsphäre: beispielsweise sind die geltenden Datenschutzbestimmungen je nach Standort des Rechenzentrums unterschiedlich. Da der Dateneigentümer keine physische Kontrolle über die Daten hat, muss er dem Cloud-Anbieter bis zu einem gewissen Maß vertrauen. Zaman u. a. [86] sprechen auch explizit von dem Risiko, dass staatliche Behörden Cloud-Anbieter anweisen können ihnen Zugriff auf die Daten ihrer Nutzer zu gewähren. Bei Cloud-Anbietern müssen auch die Kosten durch die langfristige Datenspeicherung und Entwicklungsmöglichkeiten durch Vendor-Lock-In berücksichtigt werden.

In Anbetracht dieser Möglichkeiten ist es für den Endbenutzer schwer zu bewerten, welche der Lösungen die eigenen Anforderungen erfüllt. Aufgrund der Komplexität der Materie ist nicht klar, wie sich beispielsweise die Nutzung eines Cloud-Anbieters zur Datensicherung auf den Wunsch nach Vertraulichkeit der eigenen Daten auswirkt.

1.2 Motivation

Es existieren eine Reihe von Tools, mit denen die Datensicherung durchgeführt werden kann. Allerdings unterstützen diese die geforderten Schutzziele oft nur teilweise oder sind unpraktikabel. Folgende Themen sind in verfügbaren Lösungen problematisch:

- Die Datensicherung ist nur lokal oder auf entfernten Systemen möglich. Ein transparentes Verfahren, unabhängig von der Speicherung im lokalen Netzwerk oder per Internet, vereinfacht den Backupprozess.
- Die gesicherten Daten werden nicht verschlüsselt gespeichert. Dies erlaubt den Zugriff auf die Backups durch Unbefugte. Andernfalls müssen die Backups in einem zweiten Schritt verschlüsselt werden.
- Zur Steuerung der Datensicherung wird ein zentraler Server verwendet. Dieser ist als Single-Point-of-Failure angreifbar und eine Schwachstelle.
- Proprietäre Sicherungslösungen führen zu einmaligen oder laufenden Kosten.
- Die Verwendung von proprietären Protokollen und Diensten bei der Datensicherung bzw. Verschlüsselung lässt keinen Einblick auf die korrekte Umsetzung von Sicherheitsmaßnahmen im Umgang mit den zu sichernden Daten zu.
- Die Konfigurierbarkeit kryptographischer Methoden ist nur unzureichend gegeben.
- Es sollen nur geänderte Datenblöcke (im Gegensatz zu vollständigen Dateien) gesichert werden.

Durch diese Arbeit werden mehrere vorhandene Techniken zur Datensicherung, Kryptographie, Kommunikation in verteilten Systemen usw. kombiniert, um die oben beschriebenen Probleme zu lösen. Als Resultat wird die Durchführung von Sicherungen sicherer und einfacher.

1.3 Zielsetzung

Es wird eine Peer-to-Peer-Backuplösung mit Cloud-Unterstützung vorgeschlagen, die vorhandene soziale Beziehungen nutzt. Es werden Anforderungen und Schutzziele für die Sicherung von Daten in dieser Backuplösung definiert. Aufgrund dessen werden in einer Sicherheitsanalyse Gefährdungen ermittelt und notwendige Sicherheitsmaßnahmen formuliert. Darauf aufbauend wird eine Anwendung zur Datensicherung konzipiert, die den in dieser Arbeit ermittelten Anforderungen und Schutzzielen entspricht. Mit diesem Konzept wird eine Proof-of-Concept-Implementierung der Anwendung durchgeführt. Diese Anwendung soll die durchgeführten Backups auf andere Systeme im Peer-to-Peer-Netzwerk verteilen und sich dabei Cloud-Komponenten bedienen. Durch die weite Verteilung soll

die Verfügbarkeit der Daten gesichert werden. Kryptographische Methoden sollen Vertraulichkeit und Integrität der Daten gewährleisten. Durch Einsatz der Anwendung soll der Aufwand zur Datensicherung reduziert und die Herstellung von Backups vereinfacht werden. Die Backuplösung soll dabei helfen, dass die Datensicherung unter Berücksichtigung von Vertraulichkeit, Integrität und Verfügbarkeit der Daten geschieht.

1.4 Aufbau der Arbeit

Kapitel 2 beschäftigt sich mit relevanten Grundlagen wie Kryptographie und Datensicherung nach dem aktuellen Stand der Technik. In *Kapitel 3* werden die Lösungsidee und die Einsatzumgebung sowie die Anforderungen an das Backupsystem formuliert. *Kapitel 4* beinhaltet die Sicherheitsanalyse des geplanten Backupsystems. Ausgehend von einer Schutzbedarfsermittlung werden eine Bedrohungs- und Risikoanalyse erstellt und davon passende Sicherheitsmaßnahmen abgeleitet. In *Kapitel 5* wird die Architektur und Basiskonzepte des Backupsystems erklärt. Es werden Details zur Proof-of-Concept-Implementierung wie die eingesetzten kryptografischen Verfahren beschrieben und die Hauptanwendungsfälle erklärt. *Kapitel 6* erprobt die Backuplösung anhand eines Testdatensets und dokumentiert die Ergebnisse. Die Umsetzung der definierten Anforderungen und abgeleiteten Maßnahmen wird strukturiert evaluiert. In *Kapitel 7* werden die Ergebnisse der Arbeit nochmals zusammengefasst und ein Ausblick auf die weitere Entwicklung gegeben.

KAPITEL 2

Grundlagen

In diesem Kapitel werden Grundlagen zu den Themenbereichen der Diplomarbeit vorgestellt. Zuerst wird ein Überblick über IT-Sicherheit gegeben (*Kapitel 2.1*). Darin werden Begriffe wie Schutzziele und Informations- und Unternehmenswerte erläutert und das Vorgehen bei der Durchführung einer Sicherheitsanalyse beschrieben. Des Weiteren wird auf das Themengebiet der Kryptographie und genutzte kryptographische Verfahren und deren Nutzen zur Wahrung von Schutzzielen der IT-Sicherheit eingegangen. Grundlagen zu verteilten Systemen werden im *Kapitel 2.2* beschrieben. Der letzte Bereich dieses Kapitels betrifft die Datensicherung in *Kapitel 2.3*.

2.1 IT-Sicherheit

Sicherheit spielt im heutigen Informationszeitalter eine wichtige Rolle. Informationen, egal ob privater oder geschäftlicher Natur, liegen oft in digitaler Form vor. Diese Informationen dienen unterschiedlichen Zwecken, unter anderem als Sicherung von privaten Fotos oder Geschäftsdokumenten, zur Übermittlung an Dritte zur Vertragsabwicklung bzw. zur Analyse und Aufbereitung von Statistiken. Im Rahmen dieser Schritte ist es theoretisch an vielen Stellen möglich, auf diese Informationen zuzugreifen und sie für Zwecke, die ursprünglich nicht vorgesehen waren, zu verwenden. Die unvorhergesehene Verwendung von Informationen durch Dritte ist aber meist nicht gewünscht. In vielen Fällen muss ein Zugriff durch Dritte auch aufgrund gesetzlicher Vorgaben wie der Datenschutz-Grundverordnung (DSGVO) [18] verhindert werden. Die IT-Sicherheit beschäftigt sich mit den verschiedenen Arten der Sicherheit von Informationen und ihren Schutzzielen.

2.1.1 Sicherheit

Es gibt eine Reihe von Standards und Organisationen, die sich mit dem Thema Sicherheit im IT-Kontext beschäftigen. Das Bundesamt für Sicherheit in der Informationstechnik

(BSI) hat als Standardreihe das Grundschutzhandbuch [9] veröffentlicht. Im BSI-Standard 200-1 [6] wird eine Unterscheidung zwischen der Informationssicherheit und der IT-Sicherheit getroffen und folgendermaßen definiert:

> Informationssicherheit hat als Ziel den Schutz von Informationen jeglicher Art und Herkunft. Dabei können Informationen sowohl auf Papier, in Rechnersystemen oder auch in den Köpfen der Nutzer gespeichert sein. IT-Sicherheit beschäftigt sich an erster Stelle mit dem Schutz elektronisch gespeicherter Informationen und deren Verarbeitung.

Die Sicherheit von Informationen kann dabei durch beabsichtigte Handlungen wie Angriffe und unbeabsichtigte wie durch Katastrophen, Programmfehler oder menschliche Fehler gefährdet sein.

Die Norm ISO/IEC 27000 [21] der International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) gewährt einen Überblick über Informationssicherheit und definiert grundlegende Begriffe. Die Informationssicherheit wird über Schutzziele definiert (*Kapitel 2.1.2*). Die Informationssicherheit ist gewährleistet, wenn beispielsweise die Vertraulichkeit, Integrität und Verfügbarkeit von Informationen gewährleistet ist. Weitere Schutzziele können jedoch in die Definition miteinbezogen werden. Die Norm ISO/IEC 27001 [20] beschäftigt sich auch mit den Anforderungen, die an ein Informationssicherheitsmanagementsystem (ISMS) gestellt werden. Ein ISMS umfasst die Verfahren, Richtlinien, Tätigkeiten und Ressourcen die verwendet werden, um die Informationssicherheit zu gewährleisten. Die Norm ISO/IEC 27002 [22] enthält eine Liste von Maßnahmen, die bei der Erreichung der Informationssicherheit helfen.

Eckert [26] definiert die IT-Sicherheit folgendermaßen:

> IT-Sicherheit hat die Aufgabe, Unternehmen und deren Werte (Know-How, IT-Sicherheit, Kundendaten, Personaldaten) zu schützen und wirtschaftliche Schäden, die durch Vertraulichkeitsverletzungen, Manipulationen oder auch Störungen der Verfügbarkeit von Diensten des Unternehmens entstehen können, zu verhindern.

Eckert [26] unterscheidet den Begriff Sicherheit im Kontext von IT-Sicherheit noch weiter: Die *Funktionssicherheit* eines Systems ist die Eigenschaft, dass die Ist-Funktionalität des Systems mit der geplanten Funktionalität übereinstimmt. Das Programm funktioniert unter normalen (geplanten) Umständen wie gewünscht. Die *Informationssicherheit* ist die Eigenschaft, dass in einem funktionssicheren System nur autorisierte Benutzer auf Informationen zugreifen oder diese ändern können. Der *Datenschutz* bezeichnet die Fähigkeit einer Person, über die Weitergabe der eigenen personenbezogenen Daten und deren Verwendung zu verfügen. Dies wird unter anderem in der DSGVO [18] geregelt.

2.1.2 Schutzziele

Unter Schutzzielen [26] bzw. Informationssicherheitszielen [7] versteht man Anforderungen an ein System, die erfüllt werden müssen, um dieses und die enthaltenen Daten vor Gefahren und Bedrohungen zu schützen. Von Spitz u. a. [73] bzw. Kriha u. a. [41] werden Schutzziele auch als Sicherheitsdienste bezeichnet. Durch die Definition von Schutzzielen wird die zu erreichende Informationssicherheit und damit der angestrebte Schutz von Informationen definiert. Schutzziele können für Teilbereiche eines Systems mit unterschiedlichem Sicherheitsniveau festgelegt werden. Vom BSI [7] werden die drei Sicherheitsniveaus *Sehr hoch*, *Hoch* und *Normal* für die Priorisierung der Schutzziele verwendet. Diese Festlegung des Sicherheitsniveaus ist abhängig von der Schwere der Auswirkungen auf eine Organisation bei Eintritt einer Gefahr. Dabei kann es sich um finanzielle Einbußen, aber auch um den Imageverlust eines Unternehmens oder den Verlust unwiederbringlicher Informationen handeln. Schutzziele können sich auch aus gesetzlichen Vorgaben ergeben, beispielsweise um Anforderungen an den Datenschutz [18] oder Aufbewahrungspflichten zu erfüllen. Ein Schutzziel definiert noch keine Maßnahmen zur Erreichung der Anforderungen. Diese werden im Rahmen eines ISMS abhängig von den Anforderungen und dem gewünschten Sicherheitsniveau festgelegt um die angestrebte Informationssicherheit zu erreichen.

Die Liste der möglichen Schutzziele ist nicht eindeutig definiert. In [7, 10, 20] werden die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit verwendet. Eckert [26] beschreibt unter anderem auch noch das Schutzziel Authentizität.

2.1.2.1 Vertraulichkeit

Ein System gewährleistet Vertraulichkeit, wenn der unautorisierte Zugriff auf Informationen verhindert wird. Es ist somit nur den berechtigten Parteien möglich, die vertraulichen Informationen zu lesen. Eine Email beispielsweise gewährleistet Vertraulichkeit, wenn nur der gewünschte Empfänger der Email diese lesen kann.

2.1.2.2 Integrität

Integrität bezeichnet die Eigenschaft eines Systems, wenn Daten nur bei entsprechender Autorisierung verändert werden können oder eine unautorisierte Änderung bemerkt wird. Es ist also notwendig mittels Zugriffskontrollen festzulegen, wer auf Daten zugreifen und diese verändern darf. Nur berechtige Parteien dürfen diese Daten verändern. Da es Systeme gibt, in denen dies nicht absolut sicher verhindert werden kann, muss es möglich sein eine unautorisierte Veränderung festzustellen. Dadurch können nachträglich geeignete Maßnahmen getroffen werden um den eintretenden Schaden zu begrenzen.

2.1.2.3 Verfügbarkeit

Ein System ist verfügbar, wenn dazu berechtigte Benutzer oder Systeme dieses im Rahmen ihrer Berechtigung benutzen können. Die Verfügbarkeit kann unbeabsichtigt

oder durch nicht autorisierte Parteien beeinträchtigt werden. Dies kann passieren, indem beispielsweise die Ressourcenzuteilung auf dem System falsch konfiguriert wurde und dadurch Wartezeiten bei der Nutzung eines Dienstes entstehen. Die Verfügbarkeit kann auch durch absichtliche Maßnahmen wie Denial of Service (DoS)-Angriffe beeinträchtigt werden. Dabei wird das System mit Anfragen überlastet, sodass es berechtigte Anfragen nicht mehr verarbeiten kann.

2.1.2.4 Verbindlichkeit

Verbindlichkeit wird oft auch als Zuordenbarkeit oder Nicht-Abstreitbarkeit bezeichnet. Damit ist gemeint, dass der Urheber einer Aktion nachweisbar nicht abstreiten kann, dass er dafür verantwortlich ist. Die Erstellung oder Änderung von Daten lässt sich ihm eindeutig zuordnen. Diese Eigenschaft ist beispielsweise bei Internetgeschäften notwendig, um auch ohne handschriftliche Unterschrift rechtsverbindliche Verträge abschließen zu können.

2.1.2.5 Authentizität

Die Authentizität ist das Wissen über die Identität eines Kommunikationspartners. Dies ist unabhängig davon, ob der Kommunikationspartner ein Benutzer wie beispielsweise eine angemeldete Person oder ein technisches System ist. Die Authentifikation bezeichnet den Vorgang um die Authentizität eines Kommunikationspartners zu verifizieren. Zur Authentifikation können Eigenschaften wie der Fingerabdruck einer Person oder ein Iris-Scan verwendet werden. Eine weitere Möglichkeit ist die Authentifikation durch den Besitz bestimmter Gegenstände wie einer Chipkarte oder eines Personalausweises. Eine dritte Möglichkeit besteht durch den Nachweis von Wissen, wie beispielsweise ein PIN oder ein Passwort. Ein typisches Beispiel dafür ist die Eingabe des Passwortes bei Logins, etwa beim eigenen Computer oder bei Diensten im Internet.

Bei der Authentizität wird zwischen einseitiger und gegenseitiger Authentizität unterschieden. Bei der einseitigen Authentizität identifiziert sich nur ein Kommunikationspartner gegenüber dem anderen. Bei der gegenseitigen Authentizität wissen beide Partner über die Identität des anderen Bescheid. Von Kersten u. a. [38] wird dies auch *Peer Entity Authentication* genannt.

2.1.2.6 Zusammenhänge zwischen Schutzzielen

Schutzziele können voneinander abhängen bzw. die Voraussetzung zur Erfüllung anderer Schutzziele sein. So ist die Authentizität und Verbindlichkeit von Daten nur möglich, wenn auch deren Integrität gewährleistet ist, da ansonsten die Daten manipuliert werden könnten. Vertraulichkeit setzt manchmal Zugriffsregeln voraus, aufgrund welcher entschieden wird, welche Benutzer vertrauliche Informationen einsehen dürfen. Für diese Zugriffsregeln ist wiederum die Authentizität des Benutzers notwendig.

Abbildung 2.1: Ablauf der Sicherheitsanalyse

2.1.3 Informationswerte und Unternehmenswerte

Zur Erfüllung der Informationssicherheit müssen die Informationswerte eines Unternehmens geschützt werden. Informationen können in elektronischer Form auf Datenträgern wie einer Festplatte, in materieller Form als Ausdrucke oder als Wissen der Mitarbeiter vorliegen. Diese sind für den Erfolg eines Unternehmens entscheidend und müssen daher wie Anlagegüter oder Wirtschaftsgüter geschützt werden [7].

Im BSI-Standard [7] werden nicht nur Informationswerte, sondern auch andere Unternehmenswerte betrachtet. Dabei handelt es sich neben Informationen auch um Geschäftsprozesse, Anwendungen und IT-Systeme, die für den Unternehmenserfolg relevant sind. Für deren Funktionsweise sind eine entsprechende Infrastruktur wie Serverräume und Kommunikationsverbindungen, die Interaktion zwischen unterschiedlichen Systemen sowie personelle Ressourcen notwendig. Daher müssen diese wie Informationswerte vor Bedrohungen geschützt werden. Informationswerte und Unternehmenswerte werden auch unter dem Begriff *Assets* zusammengefasst und werden in weitere Folge in dieser Arbeit als *Assets* bezeichnet.

2.1.4 Sicherheitsanalyse

Mithilfe von Schutzzielen und den Assets werden die Sicherheitsanforderungen an ein System definiert. Es gibt jedoch verschiedene Verfahren, um eine Sicherheitsanalyse für ein IT-System durchzuführen. Eine Möglichkeit zur Durchführung dieser Sicherheitsanalyse ist das Vorgehen nach dem vom BSI publizierten IT-Grundschutz-Standard [6, 7]. Dieser Standard dient als Hilfestellung zum strukturierten Vorgehen zur Erkennung und Umsetzung von notwendigen IT-Sicherheitsmaßnahmen. Diese Methodik richtet sich an die Einführung eines Sicherheitsprozesses in Unternehmen oder Institutionen und betrachtet alle Komponenten, die für ein ganzheitliches Sicherheitskonzept notwendig sind. Die Erfüllung der Schutzziele hängt somit nicht nur von technischen Komponenten und Maßnahmen ab, sondern ist auch von Geschäftsprozessen und der Unternehmensstruktur abhängig.

Bei der Durchführung einer Sicherheitsanalyse werden die Schritte aus *Abbildung 2.1* durchlaufen:

1. Festlegung von Rahmenbedingungen und des Geltungsbereichs: Zur Durchführung einer Sicherheitsanalyse nach BSI [7] werden vorab Rahmenbedingungen, in denen das zu betrachtende System läuft, ermittelt. Dazu gehört die Analyse von Geschäftszielen, relevanten Parteien, der Organisationsstruktur, abhängigen anderen

Systemen, den verarbeiteten Informationen und Geschäftsprozessen sowie welche Sicherheitsanforderungen von diesen gestellt werden. Diese Rahmenbedingungen haben beispielsweise Auswirkung auf den zu ermittelnden Schutzbedarf und damit auf die notwendigen Maßnahmen zur Wahrung der Schutzziele. Soll nur ein Teil der Geschäftsprozesse oder der Anwendungen betrachtet werden, müssen diese im Rahmen dieses Schrittes entsprechend abgegrenzt werden.

Rahmenbedingungen, Abgrenzungen und Annahmen zu der geplanten Backuplösung werden am Beginn von *Kapitel 4.1* aufgelistet.

2. Strukturanalyse: Die Erhebung der Assets geschieht im Rahmen der Strukturanalyse. Diese analysiert die funktionalen Eigenschaften und den Verwendungszweck eines Systems [26], in welchem Umfeld das betrachtete System laufen soll und beinhaltet den aktuellen Ist-Stand von Hardware, Software, Betriebssystemen und Netzwerktopologien sowie anderen Umfeldfaktoren. Unterstützend dazu wird ein Netzwerkplan erstellt um eine Übersicht über alle betroffenen Komponenten zu erhalten. Zusätzlich kommt es zur Erfassung von Räumen und Gebäuden der Organisation. Anschließend werden ähnliche Einheiten gruppiert und nach Relevanz für die Ziele der Organisation eingeteilt. Dies hat den Vorteil der Komplexitätsreduktion bei der Sicherheitsanalyse und erleichtert die Durchführung von festgelegten Maßnahmen, da die Anzahl unterschiedlicher Assets reduziert wird.

   Die Strukturanalyse wird im Rahmen dieser Arbeit in *Kapitel 4.1.1* durchgeführt.

3. Schutzbedarfsfeststellung: Bei der Schutzbedarfsfeststellung werden mögliche Bedrohungen und deren Auswirkungen für das Asset bzw. die gesamte Organisation untersucht. Dabei werden noch keine konkreten Schadensfälle aufgelistet, sondern die Schwere der Auswirkungen bei Schadenseintritt betrachtet. Bei den Auswirkungen kann es sich beispielsweise um einen finanziellen Schaden, Imageverlust oder den Verlust von Informationen handeln. Basierend auf der Strukturanalyse wird ermittelt welcher Schutzbedarf je Schutzziel für alle identifizierten Assets erreicht werden muss. In [7] werden dafür folgende Schutzbedarfskategorien empfohlen:

   - *Normal* entspricht einem begrenzten und überschaubaren Schaden.
   - *Hoch* entspricht einem beträchtlichen Schaden.
   - *Sehr Hoch* entspricht einem katastrophalen, existenzbedrohenden Schaden.

   Diesen Kategorien werden mögliche Schadensszenarien zugeordnet um die anschließende Bewertung je Asset zu erleichtern. Das BSI schlägt dazu folgende Szenarien vor: „Verstoß gegen Gesetze/Vorschriften/Verträge“, „Beeinträchtigung des informationellen Selbstbestimmungsrechts“, „Beeinträchtigung der persönlichen Unversehrtheit“, „Beeinträchtigung der Aufgabenerfüllung“, „negative Innen- oder Außenwirkung“ und „finanzielle Auswirkungen“. Bei der Ermittlung des Schutzbedarfs je Asset wird für jedes Schutzziel der erwartete Schaden anhand der Szenarien geschätzt. Die höchste so ermittelte Schutzbedarfskategorie wird als Schutzbedarf für das Schutzziel des Assets herangezogen.

Der Schutzbedarf der konzeptionierten Backuplösung wird in *Kapitel 4.1.2* ermittelt.

4. Bedrohungs- und Risikoanalyse: Auf Basis der Strukturanalyse und dem erhobenen Schutzbedarf werden im Rahmen der Bedrohungs- und Risikoanalyse Bedrohungen für die identifizierten Assets ermittelt. Der BSI-Standard unterstützt bei der Erarbeitung der Gefährdungen durch das IT-Grundschutz-Kompendium [10]. Dieses besteht aus Bausteinen mit Gefährdungen und Maßnahmen um diesen entgegen zu wirken. Dabei werden jedem Asset die Bausteine aus dem IT-Grundschutz-Kompendium zugeordnet, welche das Asset bestmöglich abbilden. Ein Baustein enthält eine kurze Beschreibung der betrachteten Komponente, des Systems oder der Vorgehensweise. Zu jedem Baustein sind elementare Gefährdungen und für den Baustein spezifische Gefährdungen angeführt. Elementare Gefährdungen beschreiben sehr allgemeine Gefahren wie Naturkatastrophen, Gefahren durch Feuer oder Wasser aber auch Gefahren durch Schadprogramme oder Sabotage. Bei den elementaren Gefährdungen ist zusätzlich angegeben, welche Schutzziele davon betroffen sind [8]. Die spezifischen Gefährdungen ergeben sich aus den elementaren Gefährdungen und stellen übliche, genauer definierte Bedrohungen für den betrachteten Baustein dar. Die spezifischen Gefährdungen sind nicht vollständig, detaillieren aber die möglichen Sicherheitsprobleme, welche ohne Gegenmaßnahmen für diesen Baustein entstehen können.

   Da der BSI-Grundschutzkatalog nur übliche Gefährdungen auflistet und keine vollständige Liste darstellt, kann noch eine erweiterte Betrachtung der Bedrohungen und Risiken notwendig sein. Dies trifft vor allem bei Assets mit sehr hohem Schutzbedarf zu oder wenn diese nur ungenügend durch die Grundschutz-Bausteine abgebildet werden. Eine erweiterte Betrachtung ist auch notwendig, wenn die Assets nicht in vom Standard vorgesehenen Einsatzszenarien verwendet werden. In diesem Fall werden zu den vorhandenen Gefährdungen zusätzliche neue Gefährdungen ermittelt. Mit diesem Vorgehen kann damit je Asset eine Liste der möglichen Gefährdungen erstellt werden, die auf das betrachtete System oder auf das Systemumfeld einwirken.

   Zu jedem Baustein werden noch empfohlenen Maßnahmen angeführt, um vor dessen Gefährdungen zu schützen. Die Grundschutzkataloge dienen somit als Hilfestellung bei der Bedrohungsanalyse, da nur die für ein System relevanten Bausteine ausgewählt und deren Maßnahmen umgesetzt werden müssen. Sie unterstützen auch dabei, eine gesamtheitliche Betrachtung aller Aspekte eines Systems durchzuführen.

   In *Kapitel 4.2* werden die identifizierten Gefahren für die geplante Backuplösung im Rahmen einer Bedrohungs- und Risikoanalyse ermittelt.

5. Auswahl von Maßnahmen: Anhand der Bausteine und der ermittelten Bedrohungen lassen sich Maßnahmen ableiten um diesen Bedrohungen zu begegnen. Der Grundschutzkatalog enthält dafür pro Baustein eine Reihe von Sicherheitsanforderungen, die für den Schutz des betrachteten Bausteins relevant sind. Diese Anforderungen stellen Maßnahmen zum Schutz des Bausteins und der definierten Schutzziele vor

den identifizierten Bedrohungen dar. Abhängig von den Rahmenbedingungen oder dem Einsatzbereich können daraus Maßnahmen gewählt werden, um den Baustein abzusichern [7].

Die ermittelten Maßnahmen werden in weiterer Folge umgesetzt, um den identifizierten Bedrohungen zu begegnen. Wenn die Sicherheitsanalyse für ein bereits bestehendes System durchgeführt wird, dienen diese Maßnahmen als Prüfplan. Dieser wird verwendet, um die ermittelten mit den bereits bestehenden Maßnahmen des Systems zu vergleichen und fehlende festzustellen.

Kapitel 4.3 dieser Arbeit beinhaltet die getroffenen Maßnahmen zur Behandlung der Bedrohungen aus *Kapitel 4.2*.

2.1.5 Kryptographie

Im Duden [5] ist die Kryptographie folgendermaßen definiert:

> Teilgebiet der Informatik, das sich mit der Entwicklung und Bewertung von Verfahren der Verschlüsselung geheimer Daten befasst

Die Kryptographie ist die Wissenschaft, die sich mit dem Verschlüsseln von Informationen befasst. Dies betrifft nicht nur die Anwendung von Verschlüsselungsverfahren, sondern auch die Entwicklung dieser Verfahren. Zur Verschlüsselung wird grundsätzlich ein Schlüssel verwendet, der nur autorisierten Personen bekannt ist. Der Kryptographie steht die Kryptoanalyse gegenüber. Sie beschäftigt sich mit dem Entschlüsseln von verschlüsselten Informationen durch Angriffe auf den Verschlüsselungsalgorithmus und dem Versuch, den geheimen Schlüssel zu erfahren. Diese beiden Wissenschaften bilden zusammen die Kryptologie [26, 59, 73]. Kryptographische Verfahren werden unter anderem dazu verwendet, die in *Kapitel 2.1.2* definierten Schutzziele zu gewährleisten [27]. Von Schmeh [67] wird die Kryptographie als die Lehre der Verschlüsselung von Daten beschrieben. Die Kryptographie wird hier als Teilgebiet der Netzwerksicherheit gesehen, da vernetzte Computer größeren Gefährdungen ausgesetzt sind als unvernetzte.

Bei der Verschlüsselung von Daten geht es im Wesentlichen darum, die Informationen so zu verfälschen, dass sie nicht ohne Wissen des Schlüssels rekonstruiert werden können [26]. Die Informationen in einer verschlüsselten Nachricht werden so gegenüber Dritten geheim gehalten. Nur Personen mit dem geeigneten Schlüssel können den Inhalt der Nachricht lesen. Die Umkehrung der Verschlüsselung wird als Entschlüsselung bezeichnet.

Ein wesentliches Gesetz in der Kryptographie ist Kerckhoffs Prinzip [59]: Es besagt, dass die Sicherheit eines Verschlüsselungsverfahrens nur von der Geheimhaltung des Schlüssels und nicht von der Geheimhaltung des Ver- und Entschlüsselungsalgorithmus abhängen darf. Würde die Sicherheit eines Verfahrens nur auf seiner Geheimhaltung beruhen, würde bei Bekanntwerden des Verschlüsselungsalgorithmus das Verfahren auf einen Schlag unsicher werden. Wird der Verschlüsselungsalgorithmus hingegen öffentlich gemacht, kann

er beispielsweise von Experten auf Fehler und Schwächen untersucht und so seine Sicherheit geprüft werden. Je länger ein Algorithmus bekannt und Attacken dagegen erfolglos sind, desto stärker ist das Vertrauen in die Sicherheit des Algorithmus. Diese Methodik ist ein wichtiger Bestandteil bei der Entwicklung aktueller Verschlüsselungsverfahren für starke Kryptographie [27]. Es ist daher wichtig, ein Verfahren zu wählen, dass bekannt ist und als sicher gilt [26].

Ein *sicherer* Verschlüsselungsalgorithmus wird von Ertel [27] folgendermaßen definiert:

- Der finanzielle Aufwand zur Entschlüsselung von verschlüsselten Daten übersteigt deren Wert.
- Die Zeit, die für die Entschlüsselung benötigt wird, ist größer als die Zeit, für die die Daten geheim gehalten werden müssen.
- Es ist nicht genügend Ciphertext, der mit einem bestimmten Schlüssel erzeugt wurde, verfügbar um den Schlüssel zu ermitteln.

Eckert [26] bezeichnet diese Algorithmen auch als *praktisch sicher*. Wenn verschlüsselte Daten uneingeschränkt zur Verfügung stehen und trotzdem nicht auf die entschlüsselten Daten geschlossen werden kann, gilt nach Ertel [27] ein Verschlüsselungsalgorithmus als *uneingeschränkt sicher*. Von Eckert [26] wird dies auch *absolute Sicherheit* genannt.

Ein weiteres wichtiges Kriterium für Verschlüsselungsverfahren ist, dass der Schlüsselraum der verfügbaren Schlüssel möglichst groß ist [26]. Je größer der Schlüsselraum, desto länger würde ein Angreifer benötigen, um durch Ausprobieren den richtigen Schlüssel für eine verschlüsselte Nachricht zu erraten. Der Schlüsselraum muss daher groß genug sein, um den Aufwand eines Angreifers beim Durchsuchen des Schlüsselraums nicht mehr vertretbar zu machen.

Das BSI gibt Richtlinien für kryptographische Verfahren und die zu verwendenden Schlüssellängen aus [11]. Die Bundesnetzagentur [13] verweist auf den Senior Officials Group Information Systems Security (SOG-IS)-Kryptokatalog [69], der definiert, welche asymmetrischen Verfahren und Hashverfahren bei der Signaturerzeugung verwendet werden sollten. Dieser enthält eine Vereinbarung über evaluierte, sichere und gegenseitig anerkannte Verschlüsselungsverfahren, Hash-Funktionen und Schlüssellängen zwischen Regierungsorganisationen innerhalb der EU. Auf BlueKrypt [28] finden sich die Zusammenfassungen von Empfehlungen zu kryptographischen Verfahren und Schlüssellängen von Organisationen wie dem BSI, dem National Institute of Standards and Technology (NIST), der Agence nationale de la sécurité des systèmes d'information (ANSSI) und der Information Assurance Directorate (IAD)-National Security Agency (NSA).

Bei der Verschlüsselung werden zwei Verfahren unterschieden: die symmetrische und die asymmetrische Verschlüsselung. Bei dem symmetrischen Verfahren müssen sowohl Sender als auch Empfänger den geheimen Schlüssel kennen. Eine Herausforderung hierbei ist die sichere Übertragung des gemeinsamen Schlüssels. Die asymmetrische Verschlüsselung

behebt dieses Problem durch ein Schlüsselpaar, den privaten und öffentlichen Schlüssel. Der öffentliche Schlüssel kann dabei jedem Kommunikationspartner bekannt sein, der zugehörige private Schlüssel ist nur jeweils einem Kommunikationspartner bekannt.

2.1.5.1 Symmetrische Verschlüsselung

Die symmetrische Verschlüsselung bezeichnet alle Verschlüsselungsverfahren, bei denen zum Verschlüsseln von Daten derselbe Schlüssel wie zur Entschlüsselung verwendet wird. Dies bedeutet, dass auch beide Kommunikationspartner den geheimen Schlüssel kennen müssen. Die Sicherheit der symmetrischen Verschlüsselung hängt insofern nicht nur von der Stärke des Verschlüsselungsverfahrens und der Länge des Schlüssels ab, sondern auch von der geheimen Übertragung des Schlüssels an den Kommunikationspartner und der sicheren Speicherung des Schlüssels bei beiden Partnern.

Es gibt zwei Verfahren, um Daten symmetrisch zu verschlüsseln: die Strom- und die Blockverschlüsselung. Bei der Stromverschlüsselung wird eine Bitfolge der Daten mit einer Folge von Schlüsselbits verschlüsselt. Die Schlüsselbits werden dabei durch den geheimen Schlüssel und einen Initialwert, auf den sich beide Kommunikationspartner einigen, vom Verschlüsselungsalgorithmus erzeugt. Ein Pseudo-Zufallszahlengenerator wird mit dem Initialwert initialisiert und erzeugt deterministisch eine Schlüsselbit-Folge, welche die Eigenschaften einer echten zufälligen Folge aufweist. Als Verschlüsselungsfunktion wird häufig das Datenbit mit dem Schlüsselbit addiert und Modulo 2 gerechnet. Die Stromverschlüsselung wird meistens verwendet, wenn die zu verschlüsselnden Daten nicht vollständig vorliegen oder das Warten auf eine bestimmte Datenmenge nicht praktikabel ist. Sie wird auch bei unsicheren Transportkanälen benutzt, da Übertragungsfehler auf diese Art nur einzelne Bits betreffen. Ein anschauliches Beispiel dafür ist die Übertragung von Telekommunikations- oder Videodaten. Beispielsweise wird dieses Verfahren im A5-Algorithmus der GSM-Verschlüsselung verwendet [26].

Bei der Blockverschlüsselung wird der Klartext in Blöcke fixer Größe geteilt. Typische Größen sind binäre Potenzen wie 128 Bit oder 256 Bit. Der letzte Block wird mit einem Füllmuster, dem Padding, zur angegebenen Blockgröße aufgefüllt. Jeder Block wird separat von den anderen verschlüsselt. Dadurch sind die verschlüsselten Daten auch in Blöcke unterteilt. Die verschlüsselten Daten ergeben sich aus der Aneinanderreihung der verschlüsselten Blöcke. Bei der Entschlüsselung wird ebenfalls wieder jeder Block einzeln entschlüsselt. Blockverschlüsselung wird hauptsächlich für Anwendungsfälle eingesetzt, in denen die zu verschlüsselnden Daten vollständig vorliegen [26].

Ein Vorteil der symmetrischen Verschlüsselungsverfahren ist die effiziente Datenverschlüsselung im Gegensatz zu asymmetrischen Verfahren (*Kapitel 2.1.5.2*). Der große Nachteil der symmetrischen Verschlüsselung ist, dass sowohl derjenige, der die Daten verschlüsselt, als auch der Empfänger der Daten den symmetrischen Schlüssel kennen muss. Ein großes Problem dabei ist die sichere Übertragung dieses Schlüssels [59]. Gelangt der Schlüssel in Besitz eines Dritten, kann dieser ebenfalls die verschlüsselten Informationen entschlüsseln.

Im Extremfall wird der Schlüssel persönlich übergeben, um das Risiko des Abhörens der Schlüsselübertragung zu verhindern.

Ein weiteres Problem stellt die Verteilung der Schlüssel dar. Jeweils zwischen zwei Teilnehmern muss ein individueller Schlüssel verwendet werden. Bei `n` Teilnehmern in einem Netzwerk müssen daher für jedes Paar (insgesamt `n * (n - 1) / 2`) Schlüssel, verteilt werden [59]. Zur Schlüsselvereinbarung zwischen zwei Teilnehmern kann das Diffie-Hellman (DH)-Verfahren verwendet werden. Dabei berechnen beide Teilnehmer lokal einen Sitzungsschlüssel, der zur Verschlüsselung der Kommunikation verwendet werden kann. Der Schlüssel selbst wird nie übertragen. Das DH-Verfahren ist jedoch anfällig für Manipulation, da sich die Kommunikationspartner vor der Schlüsselvereinbarung nicht gegenseitig authentifizieren [26].

AES Der Advanced Encryption Standard (AES) [65] ist ein symmetrisches Verschlüsselungsverfahren und wurde als Nachfolger des Data Encryption Standard (DES) entwickelt, da dieser als nicht mehr sicher gilt. Zur Ermittlung eines neuen Verschlüsselungsverfahrens wurde vom NIST ein Wettbewerb veranstaltet [27]. 2001 wurde das Verfahren Rijndael, der Gewinner des Wettbewerbs, unter dem Namen AES standardisiert. Die Verschlüsselung eines Klartextblocks erfolgt im AES in mehreren Runden. Die Anzahl der Runden ist von der Schlüssellänge abhängig. Der Vorteil des AES ist seine einfache Struktur und eine gute Performance. Der AES verwendet Blockverschlüsselung mit einer Blocklänge von 128 Bit und einer Schlüssellänge von 128, 192 oder 256 Bit. Zusätzlich kann bei der Verwendung von AES ein Betriebsmodus und je nach Betriebsmodus ein Padding-Verfahren angegeben werden. Das Padding-Verfahren steuert, nach welchem Füllmuster der letzte zu verschlüsselnde Block aufgefüllt wird [26].

Die Sicherheit des AES ist in der Vergangenheit eingehend analysiert worden. Praxisrelevante Angriffe gibt es kaum. Diese gehören zu den Seitenkanal-Angriffen oder Timing-Angriffen. Für diese Angriffe müssen bestimmte Bedingungen zutreffen wie beispielsweise, dass der Angriff auf dem gleichen Computer erfolgt, auf dem die Verschlüsselung durchgeführt wird. Es existieren auch Angriffe um die Anzahl der Verschlüsselungsrunden zu reduzieren. Nach Eckert [26] haben diese Angriffe für die Praxis jedoch noch keine Relevanz, daher gilt der AES weiterhin als sehr sicher. Der Einsatz von AES zur symmetrischen Verschlüsselung wird auch von mehreren Institutionen wie beispielsweise den BSI empfohlen [11, 34, 35, 69].

Betriebsmodi von Blockverschlüsselungsverfahren Ein Blockverschlüsselungsverfahren kann mit unterschiedlichen Betriebsmodi verwendet werden. Der Betriebsmodus hat Auswirkungen darauf, wie die Blöcke verschlüsselt werden, auf die notwendigen Eingabeparameter und beispielsweise auch auf die Notwendigkeit eines Paddings. Dementsprechend eignet sich jeder Betriebsmodus für unterschiedliche Anwendungsgebiete. Nachfolgend werden die in dieser Arbeit verwendeten Betriebsmodi bei der Verwendung von Blockverschlüsselungsverfahren beschrieben [26]:

Bei Cipher Block Chaining (CBC) wird jeder Datenblock mit dem vorhergehenden bereits verschlüsselten Block per XOR verknüpft. Beim ersten Block wird ein Initialisierungsvektor (IV) zur XOR-Verknüpfung verwendet. Mit dieser Methode wird bei gleichen Datenblöcken die Erzeugung gleicher Cipherblöcke verhindert. Ein einfaches Wiedererkennen eines verschlüsselten Blocks und der Rückschluss auf die Echtdaten ist daher nicht möglich. Der Nachteil dieses Betriebsmodus besteht darin, dass sich ein Übertragungsfehler in einem verschlüsselten Block sich auch auf die Entschlüsselung des Nachfolgeblocks auswirkt. Des Weiteren können bei einer längeren Datenfolge nicht einzelne Blöcke ver- oder entschlüsselt werden, da alle vorhergehenden Blöcke für den Vorgang benötigt werden. Für den letzten Block wird ein Padding benötigt. Eine Variante dieses Betriebsmodus ist Cipher Block Chaining - Ciphertext Stealing (CBC-CS) [25]. Dabei wird der letzte zu verschlüsselnde Block, anstatt ihn mit einem Padding-Verfahren aufzufüllen, mit den verschlüsselten Daten des vorletzten Blockes befüllt.

Bei dem Betriebsmodus Counter Modus (CTR) kann eine Blockverschlüsselung als Stromverschlüsselung verwendet werden. Bei diesem Modus wird je Datenblock ein eigener Schlüssel erzeugt. Dazu wird eine Zufallszahl, *Nounce* genannt, verwendet. Diese ist kein Geheimnis des Verschlüsselungsverfahrens. Die Zufallszahl wird mit einem Zähler verknüpft und mit dem Verschlüsselungsverfahren und dem geheimen Schlüssel verschlüsselt um einen Schlüssel fixer Länge zu erzeugen. Dieser wird per XOR mit dem eigentlichen Datenblock verknüpft um diesen zu verschlüsseln. Zur Verschlüsselung von Folgeblöcken wird der Zähler verändert. Dies kann beispielsweise durch Hochzählen des Zählers passieren. Aus der Zufallszahl und dem veränderten Zähler wird danach der Schlüssel für den nächsten Datenblock generiert und dieser mit dem eigentlichen Datenblock verknüpft. Ein wichtiges Kriterium dieses Betriebsmodus ist, dass sich der Zähler über einen langen Zeitraum nicht wiederholt. Vorteil dieses Verfahrens ist, dass auf jeden Block einer Datenfolge zugegriffen werden kann, da der Schlüssel je Block unabhängig von den Datenblöcken ist. Jeder Schlüssel kann mithilfe der Zufallszahl und dem Zähler des Blocks erzeugt werden. Daher können die Schlüssel der einzelnen Datenblöcke bereits vorab generiert werden, um die Ver- und Entschlüsselung von Blöcken parallel durchzuführen.

AE und AEAD Bei Authenticated Encryption (AE) handelt wird ein symmetrisches Verschlüsselungsverfahren in einem Betriebsmodus betrieben, bei dem zusätzlich zur Vertraulichkeit auch die Authentizität und Integrität von Daten gewährleistet und dies innerhalb eines Verfahrens umgesetzt wird. Verfahren, bei denen zusätzlich zu den verschlüsselten Daten auch unverschlüsselte Daten (Associated Data (AD)) authentisiert und in die Integritätsprüfung einbezogen werden können, werden Authenticated Encryption with Associated Data (AEAD) genannt. Dies ist beispielsweise für Header-Informationen relevant, die einsehbar aber nicht veränderbar sein sollen. Bei diesem Betriebsmodus werden aus den Daten, den AD, dem Schlüssel und einem IV die verschlüsselten Daten und ein Authentifizierungs-Tag berechnet. Bei der Entschlüsselung wird zusätzlich das Authentifizierungs-Tag einbezogen und es kommt zu einem Fehler, wenn die Daten nicht authentisiert werden können [26].

Galois/Counter-Mode (GCM) ist ein AEAD-Betriebsmodus und basiert auf dem CTR. Das Verfahren wird daher ebenfalls dazu verwendet, um mittels Blockverschlüsselung eine Stromverschlüsselung durchzuführen. Der IV wird als erster Zähler verwendet um mithilfe des geheimen Schlüssels und des Verschlüsselungsverfahrens den ersten Schlüssel für den ersten Datenblock zu erzeugen. Alle weiteren Schlüssel für die folgenden Datenblöcke werden durch Verändern des Zählers generiert. Im Gegensatz zum CTR bietet er die Funktionen von AEAD, wodurch neben der Verschlüsselung auch die Integrität und Authentizität der Daten erreicht wird.

2.1.5.2 Asymmetrische Verschlüsselung

Als asymmetrische Verschlüsselungsverfahren werden alle Verfahren bezeichnet, die mit zwei Schlüsseln arbeiten. Asymmetrische Verschlüsselungsverfahren bauen auf Einwegfunktionen auf. Eine Einwegfunktion ist eine Funktion $f(x) = y$, deren Wert y effizient berechnet werden kann. Umgekehrt existiert aber kein Verfahren, um aus y per Umkehrfunktion x zu berechnen. Die aktuellen asymmetrischen Verschlüsselungsalgorithmen bauen auf zwei mathematischen Problemen auf. Das erste Problem ist die Komplexität, eine Zahl in ihre Primfaktoren zu zerlegen, d. h. sie zu faktorisieren. Die Berechnung des Produkts zweier Primzahlen ist sehr schnell durchführbar, die Zerlegung dieses Produkts in die ursprünglichen Primzahlen sehr schwierig. Das zweite Problem ist die Berechnung von diskreten Logarithmen. Dies beruht darauf, dass sich das Ergebnis einer Exponentiation $f(x) = a^x mod n = y$ sehr effizient berechnen lässt. Umgekehrt ist es aber sehr schwer aus dem Ergebnis y per Logarithmus x zu ermitteln.

Alle in der asymmetrischen Verschlüsselung verwendeten Einwegfunktionen haben eine Falltüre, um die Umkehrung der Funktion für Berechtigte mit zusätzlichen Informationen effizient zu ermöglichen. Auf Grundlage dieser mathematischen Beziehung verwenden asymmetrische Verschlüsselungsverfahren zwei Schlüssel. Um das Problem der Schlüsselübergabe der symmetrischen Verfahren zu beheben, wird einer der Schlüssel veröffentlicht. Dieser wird als Public Key bezeichnet. Dies ermöglicht es einem Kommunikationspartner, seinen öffentlichen Schlüssel beispielsweise auch ungesichert über Email an andere Personen zu schicken. Der zweite, geheime Schlüssel wird als Private Key bezeichnet und dient als Falltüre der Einwegfunktion. Bei der Datenübertragung verschlüsselt der Absender die Daten mit dem bekannten Public Key des Empfängers. Nur der Inhaber des zugehörigen Private Keys, der Empfänger der Daten, kann diese Nachricht mit seinem Private Key entschlüsseln [26, 38, 67].

Zusätzlich können diese Schlüssel auch verwendet werden, um ihre Herkunft und Integrität zu gewährleisten. Dazu werden die Daten mit dem privaten Schlüssel des Erstellers der Nachricht verschlüsselt. Da dies mit dem privaten Schlüssel des Erstellers passiert, kann dies nur von ihm durchgeführt werden. Die verschlüsselten Daten werden an einen Empfänger übertragen. Der Empfänger der Nachricht entschlüsselt diese mit dem öffentlichen Schlüssel des Erstellers und erhält die ursprünglichen Daten. Da dieses Verfahren die Verschlüsselung der gesamten Nachricht erfordert, ist es nicht gebräuchlich.

Stattdessen wird per Hashwert eine Signatur erzeugt und diese zur Überprüfung verwendet [26].

Der Vorteil dieser Verschlüsselungsverfahren ist die einfachere Verteilung der Schlüssel im Gegensatz zu einem symmetrischen Verfahren. Für `n` Teilnehmer sind nur `n` Schlüssel notwendig, da pro Teilnehmer nur ein Schlüsselpaar benötigt wird. Asymmetrische Verfahren können auch für digitale Signaturen oder zur Authentifizierung der Kommunikationspartner genutzt werden [38].

Der Nachteil der asymmetrischen Verschlüsselung ist der hohe Rechenaufwand, da die Verschlüsselungsalgorithmen auf komplizierten mathematischen Problemen basieren. Für die Verschlüsselung von großen Datenmengen sind diese Verfahren daher eher ungeeignet.

RSA Das Verschlüsselungsverfahren Rivest, Shamir, Adleman (RSA) [64] wurde von Ronald Rivest, Adi Shamir und Leonard Adleman entwickelt. Es basiert auf dem Problem der Faktorisierung großer Zahlen. RSA wurde nie standardisiert, ist mittlerweile aber ein de-facto Standard für asymmetrischen Verschlüsselung und wird als eingesetztes kryptografisches Verfahren empfohlen [11, 34–36].

Beim Einsatz von RSA kann bei der Schlüsselerzeugung die Schlüssellänge festgelegt werden. Zusätzlich sollte für die Verschlüsselung ein Padding-Verfahren angegeben werden. Ohne Padding-Verfahren bleiben während der Verschlüsselung Muster aus den Originaldaten in den verschlüsselten Daten erhalten. Dadurch sind die verschlüsselten Daten beispielsweise mittels statistischen Analysen angreifbar. Der Einsatz eines Padding-Verfahrens verhindert dies durch das Einführen von zufälligen Padding-Mustern vor der Verschlüsselung [26].

2.1.5.3 Hashverfahren

Mit einer Hashfunktion werden Daten beliebiger Länge auf einen Wert fixer Länge abgebildet. Hashverfahren sind ebenfalls Einwegfunktionen. Dies bedeutet, dass der Hashwert einer Nachricht sehr effizient, umgekehrt die Nachricht aber nicht aus dem Hashwert berechnet werden kann. Da beliebige Daten auf diesen Adressbereich abgebildet werden, können verschiedene Daten den gleichen Hashwert generieren. Dieses Verhalten wird Kollision genannt. Da Kollisionen die Identifizierbarkeit von Daten zu einem Hashwert beeinträchtigen, müssen Hashverfahren kollisionsresistent sein. Schwach kollisionsresistente Hashverfahren haben die Eigenschaft, dass es praktisch nicht möglich ist, zu einer vorgegebenen Nachricht und deren Hashwert eine zweite Nachricht zu ermitteln, welche den selben Hashwert erzeugt. Stark kollisionsresistente Hashverfahren haben die Eigenschaft, dass es praktisch nicht möglich ist, zwei beliebige Nachrichten zu finden, welche denselben Hashwert erzeugen. Die kollisionsresistenten Eigenschaften eines Hashverfahrens haben zur Folge, dass die Hashwerte ähnlicher Daten sehr unterschiedlich sind und wie zufällig erscheinen [26]. Dieses Verhalten wird auch als Zufallsorakel bezeichnet [67].

Die Unterscheidung zwischen schwach und stark kollisionsresistenten Hashverfahren spielt bei Angriffen eine Rolle. Schwache Hashfunktionen sind gegen Angriffe resistent, bei denen

ein Angreifer zu einer gegebenen Nachricht und deren Hashwert eine weitere Nachricht erzeugt, die den gleichen Hashwert produziert. Die ursprüngliche Nachricht wird dabei immer weiter leicht verändert, bis eine Kollision gefunden wird. Dieser Angriff wird als *Substitutionsattacke* bezeichnet. Starke Hashfunktionen sind gegen Angriffe resistent, bei denen ein Angreifer das Nachrichtenpaar selbst wählen kann. Ein Angreifer erstellt dabei so lange Nachrichtenpaare, bis er zwei Nachrichten, eine unverdächtige und eine schädigende mit dem gleichen Hashwert, gefunden hat. Die Nachrichten können vorab berechnet und gespeichert werden, um ein entsprechendes Paar schnell zu finden. Dieser Angriff wird als *Geburtstagsangriff* bezeichnet [67].

Schmeh [67] nennt einige Anwendungsmöglichkeiten für Hashverfahren. Hashing wird dazu verwendet, um Daten adressierbar zu machen und so schnelle Suchen oder Zugriffe zu ermöglichen. Beispielsweise werden Daten gehasht. Der berechnete Hashwert wird mit der gewünschten zuzuordnenden Information in einer geordneten Liste abgelegt. Bei der Suche nach dieser zugeordneten Information werden die Daten wieder gehasht und der ermittelte Wert mit den Hashwerten in der geordneten Liste verglichen. Bei Übereinstimmung werden die zugeordneten Informationen gefunden. Hashverfahren werden auch verwendet, um die Integrität von Daten zu gewährleisten. Beispielsweise wird vor der Datenübertragung der Hashwert der Daten gebildet und anschließend mit den Daten übertragen. Nach der Übertragung berechnet der Empfänger der Daten ebenfalls den Hashwert und vergleicht diesen mit dem empfangenen. Bei einer fehlerfreien Übertragung müssen die Hashwerte übereinstimmen. Hashverfahren werden auch bei Signaturen verwendet. Diese dienen dazu, den Ersteller und die Integrität von Nachrichten zu überprüfen. Ein weiteres Anwendungsgebiet ist die gehashte Speicherung von Passwörtern. Anstatt Passwörter im Klartext zu speichern, werden diese gehasht abgelegt. Bei der Passwortkontrolle wird das eingegebene Passwort wieder gehasht und mit dem gespeicherten Hashwert verglichen. Dadurch ist der Vergleich des Passworts möglich, ohne dass dem System das Klartextpasswort bekannt sein muss.

SHA Der Secure Hash Algorithm (SHA) [26] erzeugt 160 Bit Hashwerte mit einer Blockgröße von 512 Bit. Nachdem 2004 und 2005 Angriffe auf den SHA-1 entdeckt wurden gilt dieser nicht mehr als sicher. Die Nachfolger von SHA-1 sind SHA-224, SHA-256, SHA-384 sowie SHA-512 und werden unter dem Begriff SHA-2 zusammengefasst. SHA-224 und SHA-256 erzeugen Hashwerte mit 224 bzw. 256 Bit und arbeiten mit einer Blockgröße von 512 Bit. SHA-384 und SHA-512 erzeugen Hashwerte mit 384 bzw. 512 Bit und verwenden eine Blockgröße von 1024 Bit. 2012 wurde vom NIST das Hashverfahren Keccak als SHA-Nachfolger SHA-3 [4] standardisiert. Dieser erzeugt Hashwerte mit 224, 256, 384 und 512 Bit.

2.1.5.4 MAC

Hashverfahren dienen unter anderem dazu die Integrität von Daten zu überprüfen. Mit ihnen ist es nicht möglich, die Authentizität des Datenerstellers zu kontrollieren. Zur Prüfung der Authentizität muss ein Message Authentication Code (MAC) verwendet

werden. Ein MAC ist eine erweiterte Hashfunktion, die zusätzlich noch einen geheimen symmetrischen Schlüssel verwendet. Dieser Schlüssel ist nur den beiden Kommunikationsteilnehmer bekannt, welche die authentifizierten Daten austauschen möchten. Im Zusammenhang mit MACs wird daher auch von Hashfunktionen mit Schlüssel gesprochen [26, 67].

Beim Einsatz eines MAC generiert der Datenersteller mithilfe der Daten und des geheimen Schlüssels den MAC. Die Daten und der errechnete MAC werden an den Datenempfänger gesendet. Dieser überprüft den MAC, indem er ebenfalls aus den empfangenen Daten und dem ihm bekannten geheimen Schlüssel einen neuen MAC errechnet und ihn mit dem empfangenen vergleicht. Da MAC-Verfahren auf Hashfunktionen aufbauen verwenden diese meistens bereits existierende Hashverfahren (*Kapitel 2.1.5.3*). Dadurch profitieren die MAC-Verfahren von deren Effizienz, Kollisionsresistenz und anderen Eigenschaften [26].

Eine Erweiterung des MAC ist der Keyed-Hash Message Authentication Code (HMAC). Bei einem HMAC wird der geheime Schlüssel verwendet, um die Anfangswerte der verwendeten Hashfunktion zu beeinflussen. Dabei wird zuerst der Schlüssel auf die Größe der Verarbeitungsblöcke der Hashfunktion gebracht. Ist der Schlüssel zu groß, wird er mit der Hashfunktion verkleinert. Danach werden, falls nötig, fehlende Bytes bis zur Blockgröße mit Nullen aufgefüllt. Der so bearbeitete Schlüssel wird mit dem speziellen String *ipad* per XOR verknüpft, vor die Nutzdaten gestellt und gehasht. Das Ergebnis wird an die XOR-Verknüpfung des Schlüssels mit dem speziellen String *opad* angehängt und nochmal gehasht, um den HMAC zu ergeben. Durch diese Anpassungen haben sich bisher keine Angriffe auf das HMAC-Verfahren ergeben [2, 3, 40].

Einen MAC unter Verwendung eines symmetrischen Verschlüsselungsverfahrens zu generieren heißt CBC-MAC [26] oder Cipher-based Message Authentication Code (CMAC) [72]. Dabei wird ein symmetrisches Verschlüsselungsverfahren wie AES im CBC-Modus betrieben und zur Verschlüsselung der Daten verwendet. Der letzte Block des generierten Ciphertextes ist der Hashwert des Verfahrens [26].

Die Verwendung eines MACs erlaubt die Überprüfung der Authentizität des Datenerstellers. Dies gilt aber nur, solange der geheime Schlüssel nur beiden Kommunikationspartnern bekannt ist. Da sie beide den Schlüssel kennen müssen, können beide einen MAC für die gleiche Nachricht erstellen. Es ist daher nicht möglich, einen MAC genau einem Teilnehmer zuzuordnen. Der MAC kann daher nicht als Beweis für die Verbindlichkeit einer Nachricht herangezogen werden [26].

2.1.5.5 Geschützte Netzwerkübertragungen

Um Daten, die über ein ungesichertes Netzwerk übertragen werden, zu schützen, kann Transport Layer Security (TLS) verwendet werden. TLS ist ein standardisiertes Protokoll und wird verwendet, um die Authentifikation von Kommunikationspartnern sicherzustellen, eine verschlüsselte Datenübertragung zwischen diesen Kommunikationspartner zu ermöglichen und um die Integrität der übertragenen Daten sicherzustellen. Die Authenti-

fikation wird durch den Einsatz von öffentlichen und privaten Schlüsseln erreicht (*siehe Kapitel 2.1.5.2*). Daten werden mithilfe eines durch TLS ausgemachten Sitzungsschlüssels verschlüsselt. Zur Sicherstellung der Integrität werden MACs verwendet.

TLS steht in mehreren Versionen zur Verfügung, wobei die aktuellste Version 1.3 ist. Die verwendeten Verfahren stehen bei der Verwendung von TLS nicht fest und unterscheiden sich zwischen den TLS-Versionen. Die Verfahren werden wie beispielsweise die genaue Wahl des Verschlüsselungsverfahrens erst beim Verbindungsaufbau von beiden Kommunikationspartner vereinbart und dann für die Dauer der TLS-Session verwendet [26].

TLS baut auf dem Transmission Control Protocol (TCP) auf und ist im Open Systems Interconnection (OSI)-Modell in der Sitzungsschicht (Ebene 5) angesiedelt. Darüberliegende Protokolle können ihrerseits diese Sitzungsschicht zur verschlüsselten Datenübertragung nutzen. Beispielsweise wird HTTP über TLS abgesichert, genannt HTTPS, um über das Internet über abgesicherte Verbindungen mit anderen Servern zu kommunizieren [26].

2.2 Verteilte Systeme

In der Literatur existieren mehrere Definitionen zu verteilten Systemen. Die Definition nach Tanenbaum u. a. [80] ist die folgende:

> Ein verteiltes System ist eine Ansammlung unabhängiger Computer, die den Benutzern wie ein einzelnes kohärentes System erscheinen.

Schill u. a. [66] beschreiben ein verteiltes System als eine Zusammenstellung von Komponenten, die erst in ihrer Gesamtheit ein funktionierendes System bilden. Das System kann seine Funktion nur durch die Zusammenarbeit aller Komponenten erbringen. Ein typisches Merkmal eines verteilten Systems ist unter anderem der gemeinsame Zugriff auf Ressourcen wie auf gemeinsame Datenbanken. Verteilte Systeme können auch dazu benutzt werden, um mittels Parallelisierung von Aufgaben die Rechenleistung eines Systems zu erhöhen. Eine ähnliche mögliche Zielsetzung ist die Lastverteilung. Dabei werden die Aufgaben an mehrere gleiche Server aufgeteilt, um Lastspitzen und Engpässen entgegen zu wirken. Verteilte Systeme werden auch verwendet, um Anforderungen wie Fehlertoleranz, Verfügbarkeit und Ausfallsicherheit zu erfüllen. Dabei werden Anwendungen mit replizierenden Eigenschaften gebaut, sodass bei Ausfall eines Servers ein weiterer die Aufgaben übernehmen kann. Dies wird vor allem in kritischen Bereichen, in denen Anwendungen ausfallsicher funktionieren müssen, verwendet. Verteilte Systeme können auch genutzt werden, um ein System skalierbar zu halten. Ein System ist skalierbar, wenn dessen Kapazitäten durch Hinzufügen neuer Ressourcen wie neuer Hardware erweitert werden kann, ohne dass das System geändert werden muss [66]. Neben der Skalierung von Ressourcen nennen Luntovskyy u. a. [45] auch noch die geografische Verteilung sowie den Verwaltungsaufwand als Dimensionen für die Skalierung eines verteilten Systems. Geografische Skalierung bedeutet, dass einzelne Ressourcen geografisch verteilt sein können ohne

die Leistung des verteilten Systems gravierend zu beeinflussen. Ein verteiltes System, das in seiner Verwaltung skalierbar ist, kann sich über viele unabhängige Organisationen erstrecken, ohne dass dessen Verwaltungsaufwand überproportional steigt.

In den folgenden Kapiteln werden drei Architekturen verteilter Systeme vorgestellt. Es werden die Client-Server-Architektur, Cloud-Architektur und die Peer-to-Peer-Architektur mit ihren Vor- und Nachteilen erklärt.

2.2.1 Client-Server-Architektur

Die Client-Server-Architektur ist die klassische Architektur eines verteilten Systems. Der Client sendet über das Netzwerk eine Anfrage an den Server. Dabei kann es sich beispielsweise um einen Methodenaufruf oder den Aufruf einer Webseite handeln. Der Server ist für die Verarbeitung der Anfrage zuständig. Nach der serverseitigen Verarbeitung schickt der Server dem Client eine Antwort zu seiner Anfrage, wie das Ergebnis einer Berechnung, einen Statuscode oder eine Webseite [66].

Diese zweistufige Architektur kann zu einer mehrstufigen Architektur ausgebaut werden. Ein Server kann sich anderer Server zur Beantwortung einer Anfrage bedienen. In den meisten Fällen stehen dahinter weitere Systeme wie andere Server, Datenbanken oder ähnliches. Beispielsweise wird bei einer dreistufigen Architektur üblicherweise die Persistenzschicht auf einen eigenen Datenbankserver ausgelagert [66].

Ein Nachteil der Client-Server-Architektur ist die Schlüsselrolle des Servers. Alle Clients müssen mit ihm kommunizieren. Fällt der Server aus, können die Clients ihre Aufgaben ebenfalls nicht mehr wahrnehmen [66].

Weitere Komplexität entsteht, wenn das verteilte System skaliert werden muss, um beispielsweise Lastspitzen zu bewältigen oder die Ausfallsicherheit zu erhöhen. Für die Skalierung müssen Server aufgerüstet oder neue Server in Form von Clustern aufgebaut werden. Dies kostet Zeit, Geld und erfordert das entsprechende Wissen. Außerdem entstehen bei der Nutzung von Clustern Aufwände zur Synchronisation und Konfliktbehandlung zwischen den Servern eines Clusters [66].

2.2.2 Cloud-Architektur

Eine Weiterentwicklung der Client-Server-Architektur stellt die Cloud-Architektur dar. Ein Client kommuniziert wieder mit einem Server, allerdings steht dieser nicht direkt als physischer Server, sondern nur virtuell zur Verfügung. Alle Ressourcen des Cloud-Anbieters werden mit Hilfe von Virtualisierungstechniken gebündelt und als virtuelle Ressourcen bereitgestellt. Einem virtuellen Server wird je nach Bedarf Speicherplatz, Rechenleistung, Netzwerkkapazität und ähnliches zugeteilt und wieder entfernt, sobald diese nicht mehr benötigt werden. Für den Betreiber des Servers ist es dabei irrelevant, woher aus dem Netzwerk des Cloud-Anbieters die Ressourcen zum Betrieb seines Servers kommen. Dabei kann es sich um einen einzelnen Serverstandort oder mehrere verteilte

Rechenzentren handeln [66]. Cloud-Architekturen lassen sich somit gut in Hinblick auf Ressourcen und ihre geografische Verteilung skalieren [45].

Cloud-Architekturen bieten nach Luntovskyy u. a. [45] somit folgende Vorteile:

- Ressourcen sind dynamisch verfügbar und an den Bedarf anpassbar.
- Technische Infrastruktur, die innerhalb einer Organisation nicht verfügbar ist, muss nicht aufgebaut werden, sondern wird von einem Cloud-Anbieter genutzt. Beispiele dafür sind Speicherplatz zur Datensicherung oder CPU-Zeit für rechenintensive Aufgaben.
- Durch die Bündelung von Ressourcen an einer Stelle können unterschiedliche Organisationen die selbe Hardware nutzen. Dies führt zu geringeren Kosten für jede Organisation.

Durch die Nutzung von Cloud-Anbietern entstehen auch Nachteile. Da der Cloud-Anbieter sowie dessen Infrastruktur in Aufbau und Standort für einen Nutzer intransparent ist, lassen sich Datenschutz- und Sicherheitsfragen schwerer beantworten. Es entsteht ein gewisser Kontrollverlust über die verarbeiteten Daten, die Service-Qualität und es kommt zu einer Abhängigkeit gegenüber dem Cloud-Anbieter. Je nach genutztem Service führt die Nutzung eines Cloud-Anbieters auch zu einer gewissen Bindung an genau diesen Anbieter, da eigene Services nicht ohne Aufwand zu einem anderen Cloud-Anbieter migriert werden können [45].

2.2.3 Peer-to-Peer-Architektur

Im Gegensatz zur Client-Server-Architektur und Cloud-Architektur kommunizieren die Clients (die *Peers*) in der Peer-to-Peer-Architektur direkt miteinander. Sie nehmen dabei sowohl die Rolle des Clients als auch die des Servers ein und stellen Anfragen an andere Peers oder beantworten diese. Die Peers sind daher gleichberechtigte Systeme mit gleichen oder ähnlichen Funktionen. Ein zentraler Server wird nicht benötigt. Ein Peer führt damit sowohl Client- als auch Serveraufgaben durch [66].

Peer-to-Peer-Netzwerke können auf unterschiedliche Arten realisiert werden. Eine grundlegende Unterscheidung dabei ist die Strukturierung des Netzwerkes. Bei unstrukturierten Peer-to-Peer-Netzen gibt es keinen Zusammenhang und keine spezifische Zuordnung zwischen Daten und Peers. Es kann nicht von vornherein festgestellt werden, auf welchem Knoten bestimmte Daten gefunden werden können. Suchanfragen an bestimmte Inhalte müssen daher an eine unbestimmte Anzahl an Peers verteilt werden, um zu einem möglichen Ergebnis zu kommen. Im Gegensatz dazu kann bei strukturierten Peer-to-Peer-Netzwerken von den Daten auf ihren Standort innerhalb des Netzwerks geschlossen werden [46, 77, 80].

Unstrukturierte Peer-to-Peer-Netzwerke lassen sich noch weiter unterteilen: in zentralisierte Peer-to-Peer-Netzwerke, reine Peer-to-Peer-Netzwerke und hybride Peer-to-Peer-Netzwerke. Bei zentralisierten Peer-to-Peer-Netzwerken existiert ein zentraler Server als Koordinator. Dieser stellt grundlegende Dienste für Kommunikationsaufbau, das Finden anderer Peers oder das Suchen von Inhalten zur Verfügung. Die übrige Kommunikation findet direkt zwischen den Peers statt. Reine Peer-to-Peer-Netzwerke besitzen keinen zentralen Server und jegliche Kommunikation läuft direkt zwischen den Peers. Bei hybriden Peer-to-Peer-Netzwerken agieren einige Peers als Koordinatoren und übernehmen die Aufgaben des zentralen Koordinierungsservers. Diese können das Netzwerk jedoch verlassen. Verlässt ein Peer mit Koordinator-Funktion das Netzwerk wird ein anderer Peer in die Rolle eines Koordinators erhoben [45, 66].

Bei strukturierten Peer-to-Peer-Netzwerken kann von den Daten auf einen Peer geschlossen werden. Im Normalfall wird dieser Zusammenhang zwischen Daten und Peer per Hash-Funktionen realisiert. Diese Strukturierung erfordert dafür die kontinuierliche Wartung der Lokation der Inhalte. Vorteil dieser Struktur ist das schnelle Finden von Daten und die deterministische Zeitspanne, die das Routing zum Zielpeer in Anspruch nimmt. Der Nachteil dieser Struktur ist eine zwingend fixe Netzwerkstruktur oder ein größerer Wartungsaufwand der beim Kommen und Gehen von Peers entsteht. Wenn sich die Peers innerhalb des Netzwerkes ändern hat dies Einfluss darauf, welchem Peer mittels Hashfunktion welche Daten zugeordnet sind. Daher müssen kontinuierlich Daten zwischen Peers verschoben werden [77].

Nach Schill u. a. [66] bieten sich Peer-to-Peer-Systeme vor allem parallelisierbare und kollaborative verteilte Systeme an. Rechenintensive Operationen können in Teilprobleme zerlegt und durch mehrere Peers verarbeitet werden. Die kollaborative Nutzung zielt auf die ungenutzten Ressourcen innerhalb eines Peer-to-Peer-Netzwerks ab. Freier Speicherplatz oder ungenutzte Netzwerkkapazität können verwendet werden, um Daten verteilt zu speichern oder anderen Peers zur Verfügung zu stellen.

Der Nachteil besteht in der dynamischen Struktur von Peer-to-Peer-Netzwerken. Es muss mit neuen Peers, die dem Netzwerk beitreten, und Peers, die das Netzwerk verlassen (genannt *Churn*), umgegangen werden. Dies erfordert Wartungsaufwand und belastet das Netzwerk. Die dynamische und dezentrale Struktur macht es schwieriger, sichere Aussagen über Metriken wie Antwortzeiten oder Verfügbarkeiten zu treffen, da dies von den Peers, deren Ressourcen und Netzwerkverbindungen abhängt. Je nach Struktur des Netzwerks können beispielsweise Suchabfragen länger dauern als bei einer herkömmlichen Client-Server-Architektur, da es keinen zentralen Anlaufpunkt für jeden Peer gibt.

2.3 Datensicherung

Barth [1] beschreibt ein Backup als eine Kopie von Daten zu einem bestimmten Zeitpunkt. Ein Backup wird durchgeführt, um diese im Fall von Datenverlust wiederherstellen zu können. Datenverlust kann durch Zerstörung der Daten aufgrund von Naturkatastrophen, wegen Unfällen oder Hardwaredefekten wie einer defekten Festplatte auftreten. Es kann

aber auch dazu kommen, wenn Daten irrtümlich oder absichtlich gelöscht oder verändert werden, wie beispielsweise bei einem Fehler in einem Programm oder durch gezielte Angriffe auf das IT-System. In jedem Fall wird ein Backup benötigt, um den alten Stand der Daten wiederherzustellen.

Ein Backup ist im Normalfall keine hochaktuelle Sicherheitskopie der zu sichernden Daten, spiegelt also nicht den Stand des Online-Systems wieder. Backups werden in periodischen Abständen, je nach Anwendungsfall beispielsweise wöchentlich, täglich oder stündlich durchgeführt. Diese Backups werden eine Zeit lang aufbewahrt, um den Datenbestand bei Notwendigkeit auf einen der verschiedenen Zeitpunkte zurückzusetzen. In den meisten Fällen wird das aktuellste Backup zur Wiederherstellung herangezogen. Wenn aber beispielsweise die Wiederherstellung einer Datei notwendig ist, die bereits vor einigen Monaten gelöscht wurde, ist dies mit einem Backup, dass zu diesem Zeitpunkt angefertigt wurde, ebenfalls möglich [62]. Ältere Backups werden meistens nach einiger Zeit gelöscht, durch neue Backups überschrieben oder in der Menge reduziert, sodass nur noch ein monatliches Backup älterer Datenbestände vorhanden ist. Ein Backup dient daher vorrangig der Wahrung der Verfügbarkeit der Daten. Zweitens dienen Backups einer Versionierung der Daten, abhängig von deren Durchführung und Lebensdauer [1].

Mit Backups werden verschiedenste Daten gesichert. Die Art der Daten hängt vom Einsatzszenario ab. Üblicherweise werden Verzeichnisstrukturen mit Dateien wie beispielsweise Bildern, Filmen oder Dokumenten gesichert. Eine Sicherung kann aber auch Konfigurations- oder Anwendungsdaten umfassen. Es kann aber auch ein Image eines Computersystems gesichert werden, um ein komplettes System wiederherstellbar zu machen.

2.3.1 Backuparten

Backups können auf verschiedene Weise durchgeführt werden. In [1, 51, 61, 78] wird zwischen der vollen Sicherung (*Full Backup*), der differenziellen Sicherung (*Differential Backup*) und der inkrementellen Sicherung (*Incremental Backup*) unterschieden.

2.3.1.1 Volle Sicherung

Bei der vollständigen Sicherung werden alle Daten gesichert, d. h. das komplette Laufwerk, alle Ordner und Dateien, die Partition oder ähnliches. Es werden alle Daten gesichert, unabhängig davon, ob sich diese geändert haben. Wird regelmäßig eine Vollsicherung durchgeführt, benötigen die Sicherungen ein Vielfaches der zu sichernden Daten. Dies führt zu einer Belastung der System- und Netzwerkressourcen und einer langen Sicherungsdauer.

2.3.1.2 Differenzielle Sicherung

Bei der differenziellen Sicherung werden nur jene Daten gesichert, die sich seit der letzten vollen Sicherung geändert haben oder neu dazugekommen sind. Sie setzt also auf der letzten vollen Sicherung auf und benötigt dadurch weniger Speicherplatz und Zeit als eine

neue volle Sicherung. Die Belastung der System- und Netzwerkressourcen ist dadurch ebenfalls geringer als bei einem vollen Backup. Zur Wiederherstellung der Daten zu einem differenziellen Backup werden jedoch zwei Sicherungsstände, das aktuelle differenzielle und das volle Backup, benötigt. Preston [62] nennt diese Art auch *Cumulative Incremental Backup*.

2.3.1.3 Inkrementelle Sicherung

Bei der inkrementellen Sicherung werden nur jene Daten gesichert, die sich seit einer beliebig auswählbaren vorhergehenden Sicherung geändert haben. Dieses Verfahren benötigt daher noch weniger Speicherplatz und Zeit als eine differentielle Sicherung. Die Belastung auf System- und Netzwerkressourcen ist bei dieser Backupart die geringste. Zur Wiederherstellung der Daten muss jedoch auf alle vorhergehenden Backups bis zur letzten Vollsicherung zugegriffen werden. Dies führt zu einem längeren Zeitaufwand bei der Datenwiederherstellung. Außerdem kann ein Sicherungszeitpunkt verloren gehen, wenn einer der vorhergehenden notwendigen Sicherungsstände nicht verfügbar ist.

Zur Unterscheidung der Backups werden Backuplevel eingeführt. Eine Sicherung mit Level 0 entspricht dabei einem vollen Backup. Die inkrementelle Sicherung mit Level 1 sichert alle Daten, die sich seit dem letzten vollen Backup geändert haben. Ein Level X Backup sichert alle Änderungen seit dem letzten Backup mit niedrigeren Level. Beispielsweise werden bei einem Level 10 Backup alle Änderungen seit dem letzten niedrigeren Backup (Level 9) gesichert. Existiert kein Level 9 Backup, sind alle Änderungen seit Level 8 betroffen. Je länger die Sicherung eines Levels zurückliegt, desto mehr Daten müssen im nächsten Level gespeichert werden. Barth [1] empfiehlt, Backups nur bis Level 2 anzulegen, um im Wiederherstellungsfall nur eine geringe Anzahl an Sicherungen zu benötigen.

2.3.2 Deduplikation

Deduplikation bezeichnet ein Verfahren zur Erkennung von Duplikaten in Daten und deren Entfernung vor der Datenspeicherung. Ein mehrfach vorkommender Datensatz einer Datenmenge wird dabei nur einmal gespeichert. Alle weiteren Vorkommen des redundanten Datensatzes werden nur referenziert. Durch die Anwendung von Deduplizierung wird daher der insgesamt notwendige Speicherplatz verringert. Die Originaldaten sind jedoch ohne Verluste wiederherstellbar [62].

Diese Eigenschaften führen dazu, dass Deduplikation bei der Erstellung von Backups eingesetzt wird [79]. Gleiche Daten eines Backups werden dadurch platzsparend nur ein Mal gespeichert. Der Deduplizierungsumfang kann noch erweitert werden. Die Deduplizierung kann für alle Sicherungen einer Quelle durchgeführt werden. Wenn davon ausgegangen wird, dass sich pro Backup nur ein geringer Teil der Daten ändert, kann damit ein Großteil der Daten jedes Backups dedupliziert werden. *Abbildung 2.2* zeigt ein Beispiel für die Deduplizierung zwischen mehreren Dateien und Dateiversionen. Gleiche Datensätze

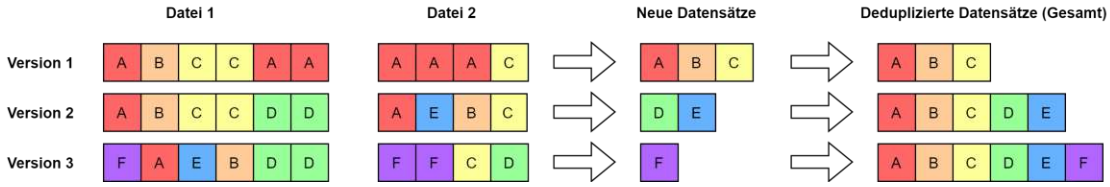

Abbildung 2.2: Deduplizierung zwischen mehreren Dateien und Dateiversionen

werden erkannt und jeder Datensatz nur einmal gesichert. Bei neuen Dateiversionen wird die gesamte Sicherung nur um neue Datensätze erweitert.

Die nächste Erweiterung wäre die Einbindung mehrerer Datenquellen. Wenn mehrere Datenquellen zu einem Zielgerät sichern, kann die Deduplizierung zwischen den Daten der unterschiedlichen Datenquellen durchgeführt werden. Dieses Prinzip lässt sich vor allem in größeren Organisationen auf immer mehr Geräte anwenden. Diese platzsparenden Eigenschaften führten laut Preston [62] dazu, dass in Rechenzentren mittlerweile hauptsächlich Festplatten zur Datensicherung verwendet werden.

2.3.2.1 Deduplizierungsort

Die Durchführung der Deduplizierung kann bei der Datenquelle oder am Zielort der Speicherung geschehen. Daher wird zwischen *Source Deduplication* und *Target Deduplication* unterschieden. Je nach Ort der Deduplizierung hat dies Auswirkungen auf die Performance und mögliche Funktionen des Backupsystems [62].

- *Target Deduplication*: Bei Target Deduplication wird die Deduplizierung vom empfangenden System, dass die Daten sichert, durchgeführt. Die sichernden Systeme senden ihre Daten zum Backupserver, welcher die Deduplizierung durchführt und nur neue Daten speichert. Die Quellsysteme müssen von diesem Verfahren nichts wissen. Es wird noch zwischen *Inline-Deduplication* und *Post-Process-Deduplication* unterschieden. Bei der *Inline-Deduplication* wird die Deduplizierung im Speicher ausgeführt, bevor die Daten vom Backupsystem auf eine Festplatte gespeichert werden. Bei der *Post-Process-Deduplication* werden die Daten zuerst geschrieben und anschließend von einem asynchronen Prozess dedupliziert.
- *Source Deduplication*: Dabei wird die Deduplizierung bei der Datenquelle von der installierten Backup-Software durchgeführt. Diese stellt fest, welche Daten sich geändert haben und ob diese im Backupsystem bereits vorhanden sind. Nur neue Daten werden zum Backupsystem geschickt und gesichert.

Die Vorteile von *Target Deduplication* sind, dass ein bestehendes Backupverfahren sehr einfach dadurch ergänzt oder ersetzt werden kann. Volle oder inkrementelle Backups können von einem Backupserver anstatt auf ein Band oder eine Festplatte einfach auf ein System gespeichert werden, dass *Target Deduplication* unterstützt. Der Einsatz von

Source Deduplication erfordert die Installation von spezieller Backupsoftware auf jedem Quellgerät [62].

Der Nachteil von *Target Deduplication* ist, dass der Backupserver zur Durchführung der Deduplizierung Zugriff auf die zu sichernden Daten benötigt. Je nach Datenformat der Backups oder bei verschlüsselten Daten ist dies nicht immer einfach möglich. *Source Deduplication* kann die Quelldaten direkt lesen, daher ist die Verpackung in einen Backup-Container oder die Verschlüsselung der Daten nach der Deduplizierung durchführbar. Durch die Deduplizierung auf Quellseite müssen auch weniger Daten an den Backupserver gesendet werden.

2.3.2.2 Deduplizierungsarten

Deduplikation kann auf unterschiedliche Art angewendet werden. Die Daten werden dazu je nach Verfahren in Blöcke unterschiedlicher Größe eingeteilt. Mandagere u. a. [47] nennt folgende Algorithmen zur Einteilung der Blöcke und der Deduplikation:

- *Datei (Whole File Hashing)*: Die Deduplikation erfolgt auf Dateiebene. Der Inhalt einer Datei wird gehasht und dieser Wert als Signatur verwendet. Dateien mit gleicher Signatur können dedupliziert werden. Meister u. a. [49] nennen dieses Verfahren auch *Whole-File Chunking*.
- *Fixe Blöcke (Fixed Block Hashing)*: Die Daten werden in Blöcke fixer Größe zerlegt. Pro Block wird ebenfalls wieder ein Hashwert erzeugt. Die Blöcke werden anhand der Hashwerte dedupliziert. Ein Problem dieser Methode ist, dass eine Änderung innerhalb einer Datei alle nachfolgenden Blöcke verschiebt. Gleichbleibende Blöcke innerhalb der Datei werden dadurch nicht mehr erkannt. Meister u. a. [49] nennen dieses Verfahren *Static Chunking*.
- *Variable Blöcke (Variable Block Hashing)*: Die flexibelste Variante zur Durchführung von Deduplizierung ist die Einteilung von Daten in Blöcke variabler Größe. Dabei wird die Datei abhängig vom Dateninhalt in Blöcke zerlegt. Eine Block-Grenze wird dabei als Anker bezeichnet. Beim Ändern einer Datei ändert sich genau der eine, betroffene Block. Andere, verschobene Blöcke werden aufgrund ihres Dateninhalts wieder erkannt und können weiterhin dedupliziert werden.
- *Applikationsspezifische Blöcke (Application-specific Chunking [49])*: Bei diesem Verfahren wird anwendungsspezifisches Wissen zur Blockeinteilung verwendet. Die Blockeinteilung erfolgt beispielsweise abhängig vom Dateityp unterschiedlich. Liu u. a. [44] beschreiben, dass bei der spezifischen Einteilung von Emails, HTML- und MP3-Dateien eine höhere Kompression von 20% bis 50% im Gegensatz zur variablen Blockeinteilung erreicht wird.

Mandagere u. a. [47] beschreiben die Auswirkungen bei der Wahl des Algorithmus zur Deduplikation auf einige Eigenschaften der Anwendung. Bei der Anwendung von variablen

Blöcken wird die höchste Platzeinsparung erzielt. Den höchsten Aufwand zur Datenwiederherstellung wird bei der Verwendung von fixen oder variablen Blöcken benötigt. Die Deduplikation auf Basis einer Datei oder fixer Blöcke benötigt weniger CPU Ressourcen, da nur Hashwerte, aber keine Anker innerhalb der Daten berechnet werden müssen. Dies passiert jedoch auf Kosten des benötigten Speicherplatzes. Diese Verfahren belasten daher die Netzwerkkapazitäten stärker als die Verwendung variabler Blöcke.

Zur Anwendung von Deduplikation ist die Führung von Metadaten zu den eingeteilten Blöcken und ihren Vorkommen in den Originaldaten notwendig. Je genauer dedupliziert wird, desto mehr Speicherplatz kann bei der Deduplikation gespart werden. Die notwendigen Metadaten zur Verwaltung werden jedoch mehr. Die Einteilung einer Datei in kleinere Blöcke führt zu deren Fragmentierung. Dies hat Auswirkungen auf die Performance des Systems. In der Studie von Meyer u. a. [50] wird gezeigt, dass die Anwendung von Deduplikation auf Dateiebene Einsparungen bei der Anwendung von Deduplikation auf Blockebene erreicht.

2.3.2.3 Source Deduplication als Backupart

Der Einsatz von Source Deduplication erlaubt eine weitere Möglichkeit zur Durchführung von inkrementellen Sicherungen. Bei der ersten Sicherung wird eine volle Sicherung durchgeführt. Es werden alle Daten in Blöcke eingeteilt, an den Backupserver geschickt und gesichert.

Alle folgenden Sicherungen sind nur mehr inkrementell. Sobald eine Datei im Quellsystem geändert wurde und eine Sicherung angestoßen wird, wird diese Datei in Blöcke eingeteilt. Alle Blöcke werden im Quellsystem dedupliziert. Bereits im Backupsystem gespeicherte Blöcke werden verworfen. Es werden nur neue Blöcke an den Backupserver übermittelt und dort gespeichert. Bereits gesicherte Daten werden somit nie nochmal zum Backupserver übermittelt. Am Backupserver existiert jeder Block nur genau ein Mal.

Der Vorteil von Source Deduplication ist, dass sich regelmäßige Backups wie inkrementelle Backups verhalten und somit immer nur eine Minimalmenge an Daten übertragen und gespeichert werden muss. Bei der Datenwiederherstellung verhält sich der gesicherte Datenbestand jedoch wie bei einer vollen Sicherung, da alle Blöcke aller Sicherungsstände vorhanden und referenziert werden können. Dadurch kann der aktuelle Datenstand einfach aus den Blöcken dieses Datenstandes wiederhergestellt werden.

2.3.3 Backuplagerung

Zusätzlich zur Durchführung von Backups ist auch der Lagerort von Backups für eine funktionierende Backupstrategie wichtig. Täglich durchgeführte Backups sind nur dann nützlich, wenn das gewählte Speichermedium zuverlässig ist. Im Einsatzfall müssen die gesicherten Daten verfügbar und verwendbar sein. Im Fall einer Katastrophe dürfen die Datenbackups nicht zerstört werden. Die durchgeführten Backups müssen daher schnell und zuverlässig zur Datenwiederherstellung verwendet werden können.

2.3.3.1 Online- und Offline-Speicherung

Bei der Lagerung von Backups wird zwischen Online- und Offline-Speicherung unterschieden. Dies hat Auswirkungen auf die Verfügbarkeit der gesicherten Daten. Bei einer Online-Speicherung sind die Sicherungsdaten jederzeit verfügbar. Die Sicherung geschieht häufig über ein Netzwerk wie das Internet. Nachteil dieser Speicherart ist, dass die Sicherung von außen angreifbar ist. Durch die aktive Netzwerkverbindung können Angreifer oder infizierte benachbarte Systeme das Sicherungssystem attackieren.

Bei der Offline-Speicherung sind die gesicherten Daten nicht über das Netzwerk erreichbar. Preston [62] nennt dies auch *Air Gap*. Dabei kann es sich beispielsweise um externe Festplatten oder Universal Serial Bus (USB)-Sticks handeln, die nach der Durchführung der Sicherung vom System getrennt werden. Bänder oder optische Medien eignen sich ebenfalls zur Offline-Speicherung von Backups. Nachteil dieser Art ist, dass die Sicherungen potenziell weniger aktuell sind als bei der Online-Speicherung. Die Durchführung einer Sicherung ist im Regelfall aufwändiger als bei der Online-Speicherung, da die Sicherungsmedien erst verfügbar gemacht und nach der Sicherung der *Air Gap* wieder hergestellt werden muss. Daraus folgt, dass auch im Wiederherstellungsfall die Sicherungsdaten nicht sofort zur Verfügung stehen.

2.3.3.2 Backupmedien

Ein Backup kann auf verschiedenen Speichermedien abgelegt werden. Es darf aber nicht auf dem gleichen physischen Medium wie die Originaldaten, z.B. einer zweiten Partition der gleichen eingebauten Festplatte, abgelegt werden. Dies würde beim Ausfall der Festplatte dazu führen, dass auch das Backup verloren geht. Deswegen müssen nach Preston [62] Backups auf mindestens zwei anderen physischen Medien als die Originaldaten abgelegt werden. Kandidaten zur Ablage von Backups sind folgende [62]:

- *Bänder*: Bänder sind die älteste Form der Speichermedien zur Datensicherung. Sie sind in drei Bereichen zur Speicherung von Backups besser geeignet als Festplatten: sie sind billiger, haben eine höhere Schreibzuverlässigkeit und haben bessere Eigenschaften bei der Langzeitspeicherung von Daten. Bänder eignen sich dafür nicht gut für die Speicherung von inkrementellen Backups. Aufgrund der technischen Funktionsweise von Magnetbändern werden diese signifikant langsamer, sobald kleine Datenmengen geschrieben werden. Diese verhindern, dass das Band mit voller Geschwindigkeit bewegt werden kann.

- *Optische Medien*: Bei optischen Medien handelt es sich um DVDs oder Blu-rays, die per Laser beschrieben werden. Optische Medien werden kaum zur regelmäßigen Datensicherung verwendet, da das Beschreiben im Gegensatz zu Bändern oder Festplatten länger dauert und ihr Uncorrected Bit Error Rate (UBER) ebenfalls schlechter ist. Sie werden daher eher als zusätzliches Sicherungsmedium für die Langzeitspeicherung von wichtigen Informationen verwendet.

- *Festplatten*: Bei der Datensicherung auf Einzelfestplatten werden eingebaute Festplatten, Festplatten mit Wechselrahmen oder externe Festplatten mit z.B. USB oder Firewire verwendet. Die Datensicherung auf Einzelfestplatten ist jedoch problematisch, da diese entweder im Computer mit den Quelldaten eingebaut sind oder im Fall von Wechselfestplatten oder externen Festplatten meistens in der Nähe gelagert werden. Dies führt im Katastrophenfall dazu, dass sowohl die Originaldaten als auch die Festplatte mit den Backupdaten verloren gehen (*siehe Kapitel 2.3.3.3*).

2.3.3.3 On- und Off-Site-Speicherung

Generell wird zwischen On- und Off-Site Speicherung der gesicherten Daten unterschieden. Bei der On-Site Speicherung werden die Daten-Backups in räumlicher Nähe zur Datenquelle gelagert. Dies bedeutet beispielsweise die Lagerung auf internen oder externen Festplatten, optischen Medien oder Magnetbändern im gleichen Raum oder Gebäude. On-Site Backups sind schnell durchführbar, im Anlassfall zeitnah verfügbar und meist billiger als Off-Site Backups. Im Fall einer Katastrophe (z.B. im Fall eines Feuers oder einer Überflutung) können sie aber vernichtet werden [61].

Off-Site Backups werden an einem vom Ursprungsort entfernten Ort oder einem anderen geografischen Gebiet gespeichert. Die zu sichernden Daten werden an einem entfernten Server gelagert, damit diese im Katastrophenfall sicher vor der Zerstörung sind. Nach Preston [62] muss daher mindestens ein Backupmedium Off-Site gelagert werden. Bei der Verwendung eines Cloud-Anbieters muss der Standort der Cloud-Server berücksichtigt werden, damit sich diese in einem anderen Gebiet befinden. Besonders wichtig ist dies, wenn sich auch die Original-Daten in der Cloud befinden.

Ein Off-Site-Standort kann dabei von mehreren Datenquellen zur Datensicherung verwendet werden. Der Nachteil von Off-Site Backups ist, dass die Datensicherung auf diesem Weg aufwändiger ist. Es muss die entsprechende Infrastruktur zur effizienten, schnellen und sicheren Durchführung hergestellt werden [61].

2.3.4 Recovery Time Objective (RTO) und Recovery Point Objective (RPO)

Die gewählte Backupart und die Lagerung des Backups haben Auswirkungen auf die gesicherten Systeme. Für Design und Wartung eines Systems werden zwei Metriken verwendet: RTO und RPO. Das RTO definiert die Zeitspanne, die eine Wiederherstellung des Systems nach einem Ausfall dauern darf. Das RTO beginnt mit dem Start des Ausfalls und endet erst, nachdem das ausgefallene System wieder einsatzbereit ist. Dies bedeutet, dass eine Datenwiederherstellung innerhalb der Zeitspanne des RTOs stattfindet, diese damit aber noch nicht endet. Beispielsweise könnte es notwendig sein, dass nach der Datenwiederherstellung andere Systeme neu gestartet werden. Die Zeitspanne des RTOs sollte für eine Organisation verkraftbar sein und kann sich für unterschiedliche Bereiche einer Organisation unterscheiden [62].

Das RPO wird ebenfalls als Zeitspanne definiert und entspricht dem akzeptablen Datenverlust nach dem Ausfall eines Systems. Dies bedeutet beispielsweise, dass nach einem Ausfall alle Daten der letzte Stunde nicht wiederhergestellt werden können. Je kleiner das RPO, desto häufiger muss eine Datensicherung durchgeführt werden.

2.3.5 Cloud- und Peer-to-Peer-Lösungen für Off-Site-Datensicherung

In *Kapitel 2.3.3.3* wurde erläutert, dass für eine funktionierende Datensicherung mindestens ein Off-Site Lagerort vorhanden sein muss. Eine Lösungsvariante ist die Nutzung eigener Firmenstandorte an einem anderen geografischen Ort und der dort vorhandenen Infrastruktur, um Daten sicher abzulegen. In vielen Fällen, beispielsweise für Privatpersonen oder kleine Unternehmen, ist dies jedoch aufgrund der Rahmenbedingungen schlichtweg nicht möglich, daher muss in diesem Fall auf fremde Off-Site-Speicherorte zurückgegriffen werden. Zur Verwendung kommen dabei meist Cloud- oder Peer-to-Peer-basierte Lösungen, die folgend beschrieben werden.

2.3.5.1 Cloud-Lösungen

Die erste Kategorie betrifft Cloud-Lösungen von Herstellern, welche die gesicherten Daten auf den eigenen Hersteller-Servern speichern. Ein Vertreter ist CrashPlan[1] von der Firma Code42 Software. CrashPlan dient dazu Daten von einem Computer auf anderen, ausgewählten Systemen zu sichern. CrashPlan richtet sich an kleine Unternehmen. Für die Datensicherung wird monatlich pro Gerät bezahlt. Die Software läuft auf Windows, Linux und Mac.

Die Sicherung kann für beliebige, auswählbare Ordner erfolgen. Die Datensicherung erfolgt bei CrashPlan kontinuierlich oder zeitgesteuert zu einem festgelegten Zeitpunkt. Wird eine neue oder geänderte Datei erkannt, wird diese in Blöcke eingeteilt. Nur Blöcke, die noch nicht gesichert wurden, werden weiterverarbeitet. Diese Blöcke werden komprimiert, verschlüsselt und an die CrashPlan-Server zur Sicherung gesendet. Alternativ kann die Datensicherung auch auf externen Festplatten oder Network Attached Storage (NAS) abgelegt werden [17].

Zur Benutzung der Software ist ein Account mit Passwort und Email-Adresse erforderlich. Benutzer-Schlüssel werden in einem eigenen Bereich auf den CrashPlan-Servern abgelegt. Die Datenverschlüsselung mittels AES-256 durchgeführt. Die Client-Server-Kommunikation wird mittels TLS verschlüsselt [16].

Für Administratoren steht eine Management-Console zum Hinzufügen neuer Benutzer oder neuer Geräte zur Verfügung. Damit kann beispielsweise das Passwort eines Benutzers zurückgesetzt oder dessen Daten wiederhergestellt werden.

CrashPlan verwendet eigenen Server zur Verwaltung der Benutzer und zur Sicherung der Backupdaten. Im Falle eines Serverausfalls sind diese Funktionen nicht mehr gegeben.

[1] `https://www.crashplan.com/` - besucht am 25.04.2022

CrashPlan kann auch zum Sichern von Daten auf externe Laufwerke oder andere Computer im Local Area Network (LAN) verwendet werden, da dafür kein zentraler Server benötigt wird. Dies entspricht aber nicht dem ursprünglichen Ziel, die Backupdaten auf entfernte Systeme auszulagern. Die Verfügbarkeit der Daten wird somit beeinträchtigt.

Da die Einrichtung von CrashPlan über Administratoren erfolgt gibt es eine zentrale Stelle, die auf die Backupdaten eines Benutzers zugreifen kann. Dies ist eine Gefährdung für Vertraulichkeit, Verfügbarkeit und Integrität der Backupdaten.

CrashPlan ist nach dem EU-U.S. Privacy Shield zertifiziert. Es ist trotzdem zu bedenken, dass das Unternehmen in den Vereinigten Staaten ansässig ist. Werden Daten auf den Servern von CrashPlan gespeichert, unterliegen sie damit potenziell dem Zugriff der amerikanischen Behörden [15].

CrashPlan ist ein proprietäres Produkt. Der Sourcecode ist nicht frei verfügbar. Er kann nicht eingesehen und einer Analyse unterzogen werden. Es bleibt somit ein Restrisiko für den Benutzer bei der Benutzung der Software, da deren Funktionsweise nicht überprüft werden kann.

2.3.5.2 Lösungen mit Nutzung anderer Cloud-Anbieter

Diese Kategorie betrifft Lösungen, welche sich anderer Cloud-Anbieter zur Datensicherung bedienen. Ein Beispiel dafür ist Duplicati[2]. Bei Duplicati werden Backupdaten auf unterschiedlichen Speicherlösungen abgelegt. Dazu gehören Cloud-Speicher wie beispielsweise Microsoft OneDrive oder Google Drive. Alternativ können auch Server verwendet werden, die Protokolle wie FTP, SSH oder WebDAV unterstützen. Zusätzlich ist auch die Ablage auf Netzwerklaufwerken oder externe Festplatten möglich. Der Sourcecode von Duplicati ist Open Source. Duplicati läuft auf den Betriebssystemen Windows, Linux und MacOS.

Die verwendeten Speicherbackends müssen mit den notwendigen Zugangsdaten konfiguriert werden. Es können beliebige Ordner im Dateisystem zur Datensicherung gewählt werden. Duplicati führt in regelmäßigen, konfigurierbaren Zeitintervallen die Datensicherung durch. Dies kann je Datenquelle und -ziel konfiguriert werden.

Von Duplicati werden Dateien in 100KB große Blöcke eingeteilt. Basierend auf diesen Blöcken wird Deduplizierung angewandt. Neue Blöcke werden anhand ihres SHA-256 Hashwertes erkannt und weiterverarbeitet. Bereits bekannte Blöcke und ihre Hashwerte werden in einer lokalen SQLite-Datenbank abgelegt. Bereits gesicherte Blöcke werden nicht nochmal gesichert. Vor dem Upload werden mehrere Blöcke zusammengefasst und mittels Zip/Deflate oder 7z/LZMA2 komprimiert und anschließend verschlüsselt [24].

Die Verschlüsselung erfolgt symmetrisch mittels AES-256. Alternativ kann lokal Gnu Privacy Guard (GPG) installiert und zur Verschlüsselung verwendet werden [23].

[2] `https://www.duplicati.com/` - besucht am 25.04.2022

Die Verifizierung der Verfügbarkeit der gesicherten Blöcke erfolgt durch regelmäßige Downloads der Daten. Dazu werden zufällige Backupdaten heruntergeladen, der Inhalt wiederhergestellt und dessen Integrität geprüft [23].

Der Fokus auf die Datensicherung bei Cloud-Anbietern ist notwendig, um die Off-Site-Speicherung der Backupdaten zu ermöglichen. Eine reine lokale Speicherung der Daten auf lokalen Servern oder externen Festplatten würde im Katastrophenfall zu Datenverlust führen. Da die Backupdaten inkrementell anwachsen wird der Umfang der Sicherungsdaten sukzessive größer. Die kostenlosen Angebote der Cloud-Anbieter sind zum Sichern dieser Daten nicht ausreichend, daher müssen die kostenpflichtigen Modelle verwendet werden.

Duplicati speichert die zu sichernden Daten immer nur auf einem Zielsystem. Die gleichzeitige Sicherung von Daten auf mehreren Cloud-Providern ist nicht möglich. Es können die gleichen Daten als zusätzliche Sicherungen auf anderen Zielsystemen eingerichtet werden, dies erfordert jedoch zusätzlichen Konfigurationsaufwand. Des Weiteren ist damit die übergreifende Wiederherstellung von Daten aus mehreren Datenquellen nicht möglich. Dies bedeutet, dass im Zweifelsfall nur eine Datensicherung vorhanden ist.

Wenn mehrere Sicherungsorte für zu sichernde Daten eingerichtet werden, dann muss der Backup-Prozess mehrfach durchlaufen werden. Dies bedeutet pro Datei die mehrfache Einteilung in Blöcke, Hashsummenberechnung, der Upload zum Cloud-Provider etc. Für Haushaltsanschlüsse mit asymmetrischer Internet-Verbindung verzögert der mehrfache Upload außerdem den Abschluss der Datensicherung.

2.3.5.3 Peer-to-Peer-basierte Lösungen

Während der Verfassung dieser Diplomarbeit wurde nach Peer-to-Peer basierten kommerziellen oder Open-Source-Lösungen gesucht. Einige Lösungen wurden nur als Prototypen für wissenschaftliche Arbeiten erstellt [42, 63, 85]. BuddyBackup ist ein Vertreter der Peer-to-Peer-basierten Lösungen und ermöglicht die Sicherung von Daten bei anderen Privatpersonen wie Familienmitgliedern, Freunden und Kollegen. BuddyBackup wurde während der Verfassung dieser Arbeit nicht mehr aktiv gewartet und hat mittlerweile die Einstellung des Betriebs angekündigt [3]. Zum 18.04.2022 wurde keine aktiv gewartete Lösung gefunden, die nach dem Peer-to-Peer-Prinzip funktioniert und Sicherungsdaten bei anderen Teilnehmern ablegt.

[3] https://blog.buddybackup.com/2022/02/21/buddybackup-is-closing-down/ - besucht am 24.04.2022

KAPITEL 3

Lösungsansatz

Dieses Kapitel beinhaltet den Lösungsansatz des geplanten Peer-to-Peer-Backupsystems mit Cloud-Unterstützung. In *Kapitel 3.1* werden die Anforderungen an das Backupsystem ermittelt. Es folgt in *Kapitel 3.2* die Einsatzumgebung, in der das Backupsystem eingesetzt wird. *Kapitel 3.3* beschreibt die geplante Lösung.

Das Backupsystem soll von einer Gruppe einander bekannten Personen genutzt werden. Dabei kann es sich beispielsweise um die Mitglieder einer Familie oder eines Freundeskreises handeln. Jede dieser Personen möchte ihre Daten sichern. Die Backuplösung baut auf einer Menge an Systemen auf, die von den einzelnen Teilnehmern betrieben werden und welche die Software des Backupsystems installiert haben. Diese Systeme kooperieren und stellen sich gegenseitig freien Speicherplatz zur Sicherung der Daten zur Verfügung.

Das Backupsystem ist kein reines Peer-to-Peer-System oder Cloud-System, sondern wird als hybrides System konzeptioniert. Cloud-Komponenten werden verwendet, um Herausforderungen des Peer-to-Peer-Umfelds zu mitigieren.

Die Einschränkung der Zielgruppe auf einander bekannte Personen bietet einige Vorteile und hat Auswirkungen auf die Architektur der Lösung. In Netzwerken, die auf echten sozialen Netzwerken basieren, ist die Kooperationsbereitschaft der Teilnehmer höher. Es ist kein Anreiz notwendig, damit Teilnehmer anderen Personen Speicherplatz zur Verfügung stellen. Situationen, in denen ein Teilnehmer nur den Speicherplatz anderer Teilnehmer nutzt und selbst keinen zur Verfügung stellt, können so vermieden oder außerhalb des Backupsystems durch persönlichen Kontakt gelöst werden. Bei bekannten Personen ist die Wahrscheinlichkeit ähnlicher Lebensgewohnheiten höher und damit entsteht eine ähnliche Verfügbarkeit ihrer Systeme. Dies erhöht die Wahrscheinlichkeit, dass mehrere Teilnehmer gleichzeitig zur Datensicherung zur Verfügung stehen und reduziert die Anzahl der notwendigen Sicherungskopien, um die Verfügbarkeit der Daten zu gewährleisten [88].

Die soziale Beziehung ermöglicht auch informelle Kommunikation zwischen den Teilnehmern. Sollten beispielsweise im Wiederherstellungsfall keine Teilnehmer verfügbar sein kann eine Person andere Teilnehmer kontaktieren und darum bitten, ihre Systeme für den Wiederherstellungsprozess zu aktivieren. Oder Teilnehmer können sich untereinander über Systemausfälle informieren oder nachfragen, sollte ein Teilnehmer seit längerer Zeit nicht mehr verfügbar sein. Dies reduziert ebenfalls die Anzahl der notwendigen Sicherungskopien und verringert die Komplexität der Lösung, um tote Systeme zu erkennen.

Ein weiterer Grund für die Einschränkung der Zielgruppe ist, dass Peer-to-Peer-Netzwerke, die auf sozialen Beziehungen aufbauen, stabiler sind. Das bedeutet, dass weniger Fluktuation bei den teilnehmenden Systemen auftritt. Dies führt zu weniger Wartungsaufwand für das Backupsystem, da die Verfügbarkeit von Systemen weniger streng geprüft werden muss und weniger Sicherungskopien von fehlenden Systemen auf neue verschoben werden müssen [48].

3.1 Anforderungen an das Peer-to-Peer-Backupsystem mit Cloud-Unterstützung

Ein sicheres Backupsystem muss eine Reihe von Anforderungen erfüllen. Das vorgestellte Backupsystem dieser Diplomarbeit soll Daten auf den Systemen anderer Teilnehmer sichern. Die Kommunikation soll dabei möglichst direkt zwischen den Peers ablaufen. Die Integration von Cloud-Komponenten dient der Unterstützung des Backup- und Wiederherstellungsvorgangs. Folgend werden die Anforderungen, die an die Lösung gestellt werden, aufgelistet.

AF-1 **Datensicherung**: Die primäre Aufgabe des Backupsystems ist die regelmäßige Durchführung von Datensicherungen. Die Angabe von einem oder mehreren Verzeichnissen zur Datensicherung muss möglich sein. Die Datensicherung wird in regelmäßigen, konfigurierbaren Abständen durchgeführt. Die Daten werden in entsprechend gesicherter Form an anderer Stelle vom Backupsystem abgelegt.

AF-2 **Erkennung und Sicherung von Änderungen**: Bei der initialen Datensicherung wird eine vollständige Sicherung aller Nutzdaten durchgeführt. Bei weiteren Sicherungsläufen werden inkrementelle Sicherungen angewendet. Es werden Änderungen der Nutzdaten im Bezug zum letzten gesicherten Datenbestand erkannt und nur diese Änderungen im Backupsystem abgelegt. Unveränderte Daten werden nicht nochmal gesichert.

AF-3 **Off-Site Speicherung der Backups an einem anderen geografischen Standort**: Die Speicherung der Backups erfolgt Off-Site. Dies bedeutet, dass die gesicherten Daten physisch an einem anderen Ort wie einem anderen Gebäude gesichert werden und in den meisten Fällen nur über das Internet erreichbar sind (siehe *Kapitel 2.3.3*).

Off-Site Backups sind notwendig, um beispielsweise im Katastrophenfall Sicherungen an einem Ort abseits der Katastrophe zur Verfügung zu haben. Dies gewährleistet die Verfügbarkeit der Daten falls alle On-Site Backups zerstört werden.

AF-4 **Vertrauliche Speicherung der Backups**: Die Backups müssen vertraulich gespeichert werden. Nur der Dateneigentümer hat die Möglichkeit die gesicherten Daten zu lesen um auf einen alten Datenstand zuzugreifen oder von einem beliebigen System eine Wiederherstellung durchzuführen. Die Verwendung durch andere Teilnehmer des Backupsystems oder durch Fremde, die Zugriff auf die Sicherungsdaten erhalten, ist nicht möglich. Die vertrauliche Speicherung von Backups ist zur Erfüllung von AF-3 notwendig. Andernfalls würde die Gefahr bestehen, dass unbefugte Zugriff auf die Datensicherungen erhalten, diese lesen und auf unerwünschte Art verwenden.

AF-5 **Datenwiederherstellung**: Die Wiederherstellung der Originaldaten auf Basis der Sicherungen muss möglich sein. Die Wiederherstellung kann für jeden Zeitpunkt, an dem eine Sicherung durchgeführt wurde, erfolgen (d.h. es muss nicht immer das aktuellste Backup wiederhergestellt werden). Es wird immer der gesamte Datenbestand wiederhergestellt.

AF-6 **Erkennung von unvollständigen oder manipulierten Sicherungen**: Das Backupsystem muss die Verfügbarkeit und Integrität der gesicherten Daten gewährleisten, damit eine Datenwiederherstellung laut AF-5 möglich ist. Das Fehlen von Teilen des Backups oder eine Manipulation der gesicherten Daten muss erkannt werden.

AF-7 **Kompensation von fehlenden oder fehlerhaften Backupdaten**: Die Backuplösung muss Mechanismen vorsehen, um fehlende oder fehlerhafte Backupdaten bestmöglich zu kompensieren. Durch die Korrektur fehlerhafter Daten oder deren erneute Sicherung wird deren Verfügbarkeit gewährleistet. Dies kann erforderlich sein, wenn ein Teilnehmer das Backupsystem verlässt oder Sicherungsdaten am System des sichernden Teilnehmers manipuliert werden. Die Sicherungsdaten müssen derart gespeichert werden, um den Ausfall von bis zu zwei Teilnehmern kompensieren zu können.

AF-8 **Vertrauliche Kommunikation zwischen Teilnehmern**: Zum Schutz der Kommunikation muss der Datenaustausch zwischen Teilnehmern des Backupsystems vertraulich geschehen. Es darf einem Angreifer nicht möglich sein, den Datenverkehr während der Durchführung einer Datensicherung oder Datenwiederherstellung zu lesen. Dies erlaubt die Nutzung von Netzwerken mit unbekannten Teilnehmern wie dem Internet zur Datenübertragung.

AF-9 **Nutzung des freien Speicherplatzes anderer Teilnehmer zur Datensicherung**: Die Sicherungsdaten werden langfristig nur auf den Systemen anderer

Teilnehmer des Backupsystems gespeichert. Die Cloud-Komponenten speichern Sicherungsdaten nur temporär.

Nach Meyer u. a. [50] verwenden die Hälfte der Computernutzer weniger als 40% ihres verfügbaren Speicherplatzes. Die Backuppartner stellen sich somit gegenseitig ungenutzte Ressourcen zur Datensicherung zur Verfügung. Toka u. a. [83] untersuchten die Speicherkosten von Daten in Peer-to-Peer- und Cloud-Lösungen und kommen zu dem Schluss, dass die Speicherung im Peer-to-Peer-Umfeld günstiger ist. Daher werden als finaler Ablageort nur Systeme anderer Teilnehmer verwendet.

AF-10 **Bekannte Teilnehmer innerhalb des Backupsystems**: Die Teilnehmer des Backupsystems müssen einander bekannt sein. Dies erlaubt die Wahl von bestimmten Teilnehmern, mit denen zur Datensicherung kooperiert wird und denen vertraut wird. Die Kommunikation mit fremden Personen finden nicht statt.

Bei bekannten Teilnehmern eines Netzwerks besteht außerhalb des Netzwerks eine reale Beziehung zwischen den Teilnehmern, beispielsweise sozialer oder unternehmerischer Art. Die Untersuchungen von Sharma u. a. [70] und Gracia-Tinedo u. a. [32] zeigen, dass in Netzwerken bekannter Teilnehmer eine höhere Datenverfügbarkeit erreicht wird als bei unbekannten Teilnehmern. Dies ergibt sich durch eine höhere Kooperationsbereitschaft und ähnliche Lebensgewohnheiten. Nach Gracia-Tinedo u. a. [32] führt dies auch dazu, dass Algorithmen zum Abschätzen der notwendigen Redundanzen zu hohe Redundanzwerte ermitteln. In Friend-to-Friend Netzwerken kann daher eine geringere Redundanz bei der Datensicherung akzeptiert werden. Zuo u. a. [88] zeigen, dass in Friend-to-Friend Netzwerken die Ressourcenausnutzung des angebotenen Speicherplatzes fairer funktioniert als zwischen unbekannten Teilnehmern. Die Einführung zusätzlicher Mechanismen zur Kontrolle der Ressourcennutzung ist in diesem Fall daher nicht unbedingt erforderlich. Die Nutzung von sozialen Zusammenhängen zwischen Teilnehmern führt daher zu einer geringeren Ressourcenbelastung für das gesamte Backupsystem.

AF-11 **Technische Möglichkeit zur Authentifizierung der Teilnehmer**: Zur Erfüllung von AF-10 muss es eine technische Möglichkeit zur gegenseitigen Authentifizierung der Teilnehmer geben.

AF-12 **Kommunikation der Teilnehmer ohne zentralen Server**: Die Kommunikationspartner des Backupsystems kommunizieren direkt miteinander. Die gewählte Architektur folgt dabei dem eines Peer-to-Peer-Netzwerks (*Kapitel 2.2.3*). Das Finden von anderen Backuppartnern, die Sicherung, Verifizierung und Wiederherstellung von Daten muss ohne zentralen Server möglich sein. Ein zentraler Server würde dem Backupsystem einen Single-Point-of-Failure hinzufügen und würde das System beispielsweise für DoS-Angriffe angreifbar machen.

AF-13 **Minimierung der Sicherungszeit**: Es müssen Mechanismen zur Minimierung der Sicherungszeit eingesetzt werden.

Nach Dell'Amico u. a. [19] und Li u. a. [43] entstehen die meisten Datenverluste, weil ein System vor Vervollständigung einer Sicherung ausfällt. In Zusammenhang mit AF-3 stellt die verfügbare Upload-Bandbreite einen limitierenden Faktor für die Geschwindigkeit der Datensicherung dar [75].

AF-14 **Sichere Standardwerte bei kryptographischen Verfahren**: Die Anwendung muss, basierend auf dem aktuellen Stand der Technik, als sicher erachtete kryptographische Verfahren als Standardwerte verwenden.

AF-15 **Zugang zu Quellcode der Anwendung**: Ein Benutzer der Backuplösung muss die Möglichkeit haben, die beschriebene Funktionsweise der Anwendung und die eingesetzten Techniken zu verifizieren. Dazu muss der Quellcode der Anwendung zugänglich sein.

Khanjani u. a. [39] zeigen, dass bezüglich der Themen Sicherheit, Fehler und Testen eine Open-Source-Anwendung einer Closed-Source-Anwendung vorzuziehen ist. Schryen u. a. [68] messen die Sicherheit eines Softwareproduktes anhand der Fehler und Patchzyklen. Am Beispiel von OpenOffice und MS Office wurde festgestellt, dass OpenOffice nach den vorgeschlagenen Metriken sicherer ist. Penha-Lopes [60] unterstreicht die Bedeutung von Open-Source für die Transparenz und Vertrauenswürdigkeit von Software. Dies ermöglicht im Fall der Backuplösung beispielsweise die Kontrolle der verwendeten Verschlüsselungsverfahren oder die Verifikation, dass keine privaten Schlüssel an andere Teilnehmer übertragen werden. Der Zugang zum Quellcode ermöglicht auch die Durchführung von Reviews und Fehlerbehebungen durch Dritte.

3.2 Einsatzumgebung

In *Kapitel 3* wurde zu Beginn festgelegt, dass die Zielgruppe der Backuplösung Teilnehmer sind, welche einander kennen. Daher wird davon ausgegangen, dass die Lösung hauptsächlich in privaten Haushalten oder kleinen Unternehmen eingesetzt wird. Es wird ein Netzwerkplan mit den Komponenten von privaten Haushalten bzw. kleinen Unternehmen erstellt, um diese Einsatzumgebung genauer zu definieren. Dieser Netzwerkplan enthält die für die Backuplösung relevante IT-Systeme, deren Verbindungen untereinander und Verbindungen nach außen. Folgend werden einige Merkmale bzw. Möglichkeiten dieses Umfelds aufgelistet:

- Kleine Netzwerkstrukturen unterschiedlicher Ausprägung
- Clients in Form von Stand-PCs oder Laptops sowie Server mit unterschiedlicher Architektur
- Unterschiedliche Betriebssysteme (Windows, Linux, . . .)
- Die Teilnehmer kommunizieren über ein IP-Netzwerk, in dem auch Datenverkehr anderer Anwendungen und Systeme stattfindet

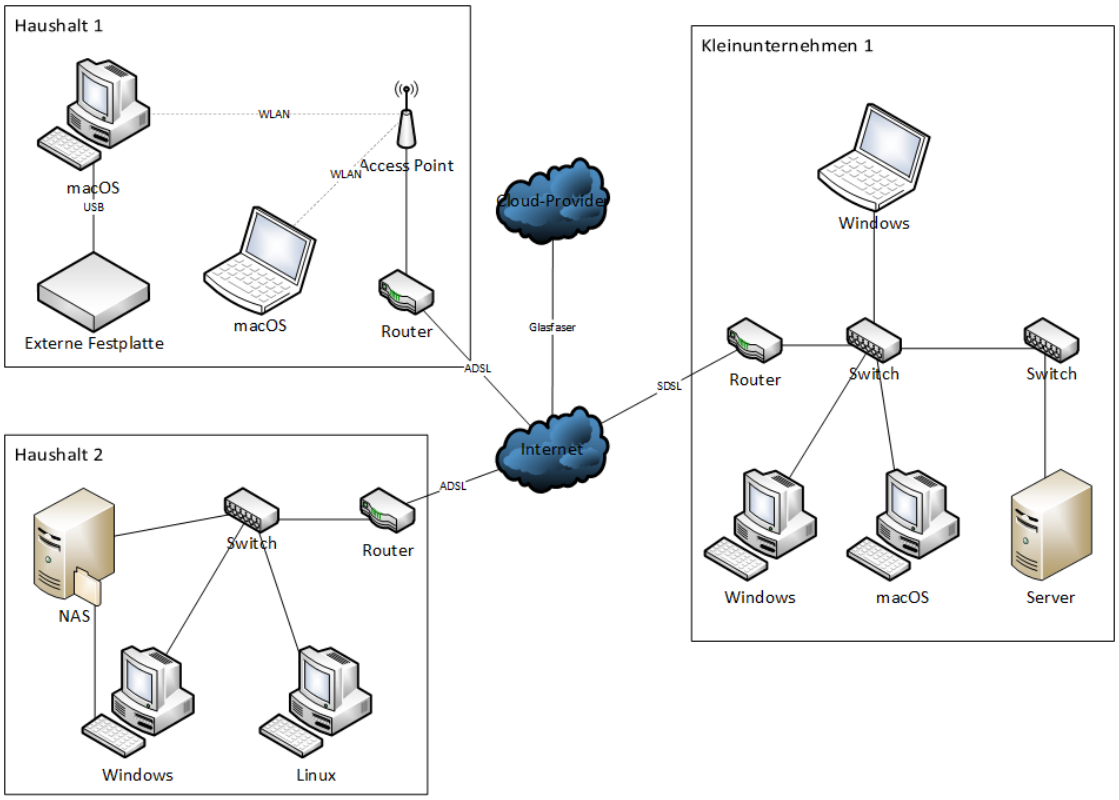

Abbildung 3.1: Netztopologie der Einsatzumgebung

- Symmetrische und asymmetrische Internetverbindungen

Ausgehend von diesen Überlegungen und den Anforderungen aus *Kapitel 3.1* stellt der Netzwerkplan in *Abbildung 3.1* die Netzwerktopologie des Einsatzumfeldes der Backuplösung dar. In der Backuplösung existieren mehrere Haushalte oder Kleinunternehmen mit unterschiedlicher Netzwerktopologie. Die Kommunikation untereinander wird über eine Breitband-Internet-Anbindung ermöglicht. Eine Studie für den genauen Bandbreitenbedarf wird im Rahmen dieser Diplomarbeit nicht durchgeführt. Die notwendige Bandbreite hängt von der zu sichernden Datenmenge ab und muss adäquat dimensioniert sein. Der Internet-Anschluss sollte über unbegrenztes Datenvolumen verfügen bzw. sollte es sich bei dem verfügbaren Datenvolumen um ein Vielfaches der zu sichernden Datenmenge handeln. Unterstützend wird ein Teil der Kommunikation über einen Cloud-Provider abgewickelt. Dieser besteht aus einer für das Backupsystem nicht transparenten und sich verändernden Infrastruktur.

Das Backupsystem eines Teilnehmers muss aus mindestens vier Teilnehmern bestehen, d.h. die Datensicherung erfolgt bei mindestens drei anderen Teilnehmern. Dies ist notwendig, um die Verfügbarkeit der Sicherungen zu gewährleisten und wird für AF-7 Kompensation von fehlenden oder fehlerhaften Backupdaten benötigt.

3.3 Beschreibung der Lösung

Dieses Kapitel gibt einen Überblick über die geplante Backuplösung. Das gesamte Backupsystem wird von mehreren Teilnehmern betrieben. Jeder Teilnehmer hat Daten, welche nicht verloren gehen dürfen. Diese Teilnehmer stellen sich gegenseitig freien Speicherplatz ihrer Systeme zur Datensicherung zur Verfügung.

Ein Teilnehmer kennt die anderen Teilnehmer, denen er vertraut und mit denen er zusammenarbeiten möchte. Diese Teilnehmer werden lokal als Sicherungspartner konfiguriert. Umgekehrt muss auch jeder dieser Sicherungspartner den Teilnehmer konfigurieren. Erst wenn dies durch beide Teilnehmer durchgeführt wurde ist die Kommunikation möglich. Nur auf Systemen dieser Teilnehmer werden eigene Daten gesichert und es werden auch nur Sicherungsdaten dieser Teilnehmer akzeptiert.

Am Quellsystem werden die zu sichernden Ordner konfiguriert. Diese werden in regelmäßigen Abständen auf Änderungen überprüft. Zur Minimierung des Kommunikations- und Speicheraufwandes wird bei Durchführung einer Sicherung Source Deduplication eingesetzt. Dabei werden geänderten Daten am Quellsystem erkannt, in Blöcke eingeteilt und nur neue Blöcke im Backupsystem verteilt. Bestehende, bereits früher vorhandene Blöcke werden übersprungen, da sie bereits im Backupsystem gesichert wurden. Dazu wird lokal ein Index geführt. Dieser enthält Informationen darüber, welche Blöcke bereits im Backupsystem gesichert wurden und zu welchen Dateien diese Blöcke gehören.

Zur Wahrung der Verfügbarkeit der gesicherten Daten werden diese zu mehreren Teilnehmern repliziert. Das bedeutet, dass jeder Block mehrfach, bei jeweils anderen Teilnehmern, gespeichert wird. Der Speicherort der Blöcke wird ebenfalls im Index mitprotokolliert. Dabei wird zu jedem Block vermerkt, bei welchen Teilnehmern er gesichert wurde.

Die Zielgruppe des Backupsystems sind Privatpersonen und kleine Unternehmen. Es wird nicht davon ausgegangen, dass alle Systeme dieser Teilnehmer jederzeit verfügbar sind. Des Weiteren verfügt jeder Teilnehmer im Normalfall über einen Internetanschluss mit asynchroner Download-/Upload-Geschwindigkeit. Trotzdem soll eine Datensicherung möglichst schnell vonstattengehen. Daher werden zur Unterstützung des Backupverfahrens Cloud-Komponenten eingesetzt. Zu sichernde Blöcke werden nicht direkt an einen anderen Teilnehmer gesendet, sondern zuerst bei einem Cloud-Onlinespeicher zwischengespeichert. Andere Teilnehmer, die diese Blöcke sichern sollen, erhalten vom Quellsystem eine Nachricht mit dem Speicherort in der Cloud. Dadurch können die Daten vom Cloud-System mit der vollen Download-Bandbreite heruntergeladen werden, auch wenn das Quellsystem nicht verfügbar ist. Nach dem Download informiert das Zielsystem das Quellsystem über die durchgeführte Sicherung des Blocks. Nachdem ein Block durch genügend Systeme gesichert wurde, wird er vom Quellsystem wieder aus dem Cloud-Onlinespeicher entfernt.

Abgesehen vom Cloud-Onlinespeicher zum Austausch der Sicherungsblöcke läuft die Kommunikation zwischen den einzelnen Systemen direkt wie in einem Peer-to-Peer-Netzwerk ab. Sie soll ohne zentralen Server zur Vermittlung zwischen den einzelnen Teilnehmern

funktionieren. Diese Kommunikation dient der Übertragung von Steuerungsnachrichten. Dazu gehören beispielsweise eine Nachricht vom Quell- zum Zielsystem mit dem Cloud-Speicherstandort eines zu sichernden Blockes. Umgekehrt bestätigt das Zielsystem auf diesem Kommunikationskanal die erfolgte Sicherung. Im Rahmen dieser Diplomarbeit wird davon ausgegangen, dass eine direkte Netzwerkkonnektivität zwischen den Systemen der Teilnehmer besteht und diese direkt per IP-Protokoll miteinander kommunizieren können. Dies kann mit Mechanismen wie Virtual Private Networks (VPNs), Network Address Translation (NAT), Port-Forwarding oder ähnlichem erreicht werden, ist jedoch nicht im Umfang dieser Arbeit enthalten.

Da die Daten eines Teilnehmers auf Fremdsystemen gesichert werden, die nicht seiner Kontrolle unterliegen, müssen sie vertraulich gespeichert werden. Mit aktuellen kryptographischen Verfahren wird sichergestellt, dass nur der Dateneigentümer die ursprünglichen Daten aus den Sicherungsdaten wiederherstellen kann.

KAPITEL 4

Sicherheitsanalyse

Dieses Kapitel umfasst die Sicherheitsanalyse des Backupsystems und behandelt die zu mitigierenden Sicherheitsproblematiken. In *Kapitel 4.1* wird eine Schutzbedarfsermittlung durchgeführt, um relevante Schutzziele für das Backupsystem und die zu speichernden Daten zu identifizieren. Anschließend werden mit einer Bedrohungsanalyse mögliche Bedrohungen oder Angriffe auf das System aufgezeigt (*Kapitel 4.2*). Abschließend werden in *Kapitel 4.3* Sicherheitsmaßnahmen zur Begegnung der Bedrohungen erläutert.

4.1 Schutzbedarfsermittlung

Im folgenden Kapitel werden die Schritte zur Feststellung des Schutzbedarfs der Backuplösung durchgeführt. Dazu wird nach dem in *Kapitel 2.1.4* beschriebenen Vorgehen des BSI vorgegangen.

Die Definition der Rahmenbedingungen und des Geltungsbereichs der Sicherheitsanalyse ist nur eingeschränkt möglich. Da die Backuplösung nicht für eine bestimmte Organisation mit konkreter Struktur, bekannten Geschäftsprozessen und genauen Geschäftszielen entwickelt bzw. eingesetzt wird, werden Annahmen für Rahmenbedingungen und den Geltungsbereich getroffen:

- Zielgruppe sind Privatpersonen und kleine Unternehmen. Die Backuplösung wird im privaten oder kleinunternehmerischen Bereich mit asymmetrischen Internetverbindungen eingesetzt.
- Die zu sichernden Daten werden nicht nach Inhalt und Sicherheitsanforderungen unterschieden. Daher wird für die zu sichernden Daten von einem sehr hohen Sicherheitsniveau ausgegangen.

- Die Sicherheitsanalyse betrachtet das selbst entwickelte Backupsystem, die zu sichernden Daten sowie die zur Sicherung notwendigen Komponenten. Dazu gehören beispielsweise Netzwerkkomponenten und Speichersysteme, die im Rahmen des Backupsystems verwendet werden.
- Das Ziel des Backupsystems ist die Sicherung und Wiederherstellung der schützenswerten Daten. Weitere Geschäftsprozesse oder der Nutzen der Daten im unternehmerischen Umfeld werden im Rahmen der Sicherheitsanalyse nicht betrachtet.

4.1.1 Strukturanalyse und Identifikation der Assets

In Schritt *Kapitel 2.1.4 - 2.* der Sicherheitsanalyse werden bei der Strukturanalyse die funktionalen Eigenschaften des geplanten Systems erfasst sowie der Ist-Stand der Unternehmensprozesse, Anwendungen und Informationen, die im Rahmen des Geltungsbereichs vom zukünftigen Sicherheitskonzept betroffen sind [7]. Die Anforderungen an die Backuplösung wurden bereits in *Kapitel 3.1* ermittelt. Die zu entwickelnde Backuplösung ist nicht auf ein bestimmtes Unternehmen zugeschnitten und generisch verwendbar, daher werden bei der Strukturanalyse spezifische Unternehmensprozesse, Unternehmensanwendungen und Informationen ausgeklammert. Ebenso werden die örtlichen Gegebenheiten wie Gebäude oder Räume, in denen die IT-Systeme laufen, nicht betrachtet. Bei der Datensicherung werden bezüglich der zu sichernden Informationen keine Einschränkungen vorgenommen. Es soll möglich sein, alle Informationen, die in Form von Dateien in Verzeichnissen vorliegen, zu sichern. Von der Strukturanalyse werden daher nur die Backupanwendung, ihre zu sichernden Daten und relevante Infrastrukturkomponenten untersucht.

Der nächste Schritt zur Ermittlung der Infrastrukturkomponenten ist die Erstellung eines Netzwerkplans zur Ermittlung der betroffenen IT-Systeme, deren Verbindungen untereinander und Verbindungen nach außen. Dieser wurde bereits in *Kapitel 3.2* in *Abbildung 3.1* erarbeitet. Aus dem Netzwerkplan lassen sich Gruppierungen ähnlicher IT-Systeme und Netzwerkkomponenten ableiten. Dies erleichtert in späterer Folge die Erhebung der Bedrohungen sowie der erforderlichen Maßnahmen, um je Gruppe ein bestimmtes Sicherheitsniveau zu erreichen. Vom BSI werden diese Gruppierungen auch Bausteine genannt [10].

Die folgende Auflistung definiert die zur Beschreibung der Backuplösung notwendigen Assets sowie die aus dem Netzwerkplan abgeleiteten Assets, welche für die Sicherheitsanalyse des Backupsystems relevant sind:

AS-1 **Backuplösung**: Die Software der eingesetzten Backuplösung mit den Anforderungen aus *Kapitel 3.1*. Das Asset umfasst auch die Informationen, die für die Funktionsfähigkeit der Lösung notwendig sind. Dazu gehören beispielsweise der Index zur Verwaltung der Sicherungsdaten, die Konfiguration der bekannten Teilnehmer sowie deren Authentifizierungs-Informationen, wie z.B. Zertifikate anderer Teilnehmer.

AS-2 **Nutzdaten**: Alle Informationen, die auf einem Quellsystem in Form von Dateien vorhanden sind und vom Besitzer als sicherungswürdig erachtet werden, sind zu schützen. Es wird davon ausgegangen, dass diese Daten unverschlüsselt und ungesichert vorliegen.

AS-3 **Sicherungsdaten**: Bezeichnet alle Daten, die auf Systemen anderer Teilnehmer zur langfristigen Datensicherung abgelegt werden. Die Sicherungsdaten dienen zur Wiederherstellung der Nutzdaten.

AS-4 **Netzwerk-Infrastruktur**: Bezeichnet alle Geräte zum Zugang des LANs und Internets. Dazu gehören Router, Switch und WLAN-Access-Points, aber auch die Internet-Verbindung an sich über einen kabelgebundenen Anbieter oder das Mobilfunknetz.

AS-5 **Endgeräte**: Enthält Computer oder Server zur privaten oder unternehmerischen Nutzung. Auf diesen Geräten wird die Backuplösung betrieben.

AS-6 **Lokale Speicher**: Alle Geräte und Medien zur Speicherung von Daten, die lokal bei den Endgeräten gelagert sind. Dazu gehören beispielsweise externe Festplatten, Wechseldatenträger, NAS-Server oder USB-Sticks.

AS-7 **Cloud-Speicher**: Bezeichnet alle Systeme zur Sicherung von Daten in der Cloud.

Die ermittelten Assets AS-1 bis AS-7 bilden die relevanten Assets der Backuplösung und sind der Ausgangspunkt für die weitere Sicherheitsanalyse.

4.1.2 Schutzbedarfsfeststellung

Nach der Strukturanalyse erfolgt die Feststellung des Schutzbedarfs. Dazu werden für alle identifizierten Assets die Schadensauswirkungen bei Schadenseintritt definiert. Dabei handelt es sich um eine qualitative Aussage über die Schadenshöhe. Bei der Durchführung der Schutzbedarfsfeststellung nach BSI beziehen sich die Schadensauswirkungen auf den Erfolg eines Unternehmens. Die Schadensauswirkungen werden im Rahmen dieser Diplomarbeit auf die Funktionsfähigkeit der Backuplösung bezogen und konkrete Szenarien wie der Verstoß gegen Gesetze oder Verträge nicht explizit berücksichtigt. Es werden die Schutzziele aus *Kapitel 2.1.2* sowie die empfohlenen Kategorien *normal*, *hoch* und *sehr hoch* aus *Kapitel 2.1.4 - 3.* verwendet. Zusätzlich wird die Kategorie *nicht anwendbar* für Schutzziele verwendet, die für das betroffene Asset nicht von Relevanz sind. Das Schutzziel Verbindlichkeit wird nicht berücksichtigt, da die Nicht-Abstreitbarkeit von gesicherten Daten kein Ziel der Backuplösung ist.

4.1.2.1 Schutzbedarf von AS-1 Backuplösung

- Vertraulichkeit *hoch*: Die Anwendung enthält die Konfiguration mit bekannten Teilnehmern, die eigenen und fremden Authentifizierungsinformationen und Metadaten

zu den Sicherungsdaten. Ein Angreifer kann bei Zugriff Informationen über den Inhalt des Quellsystems oder die Kommunikationspartner erhalten oder sich als das Quellsystem ausgeben.

- Integrität *sehr hoch*: Für die korrekte Funktionsweise des Backupsystems muss eine Manipulation an der Anwendung erkannt werden. Eine kompromittierte Anwendung könnte dazu führen, dass keine eigenen oder fremden Daten gesichert werden ohne dass dies von dem Benutzer bemerkt wird.

- Verfügbarkeit *normal*: Die Anwendung wird lokal installiert. Sollte die lokale Installation ausfallen kann diese neu installiert werden. Der Ausfall eines Backupsystems beeinträchtigt andere Teilnehmer nur geringfügig, da diese über andere Teilnehmer zur Datensicherung und -wiederherstellung verfügen.

- Authentizität *hoch*: Die Anwendung muss sich bzw. den Teilnehmer, den sie repräsentiert, eindeutig nachweisen können. Ohne die erfolgreiche Authentifizierung findet keine Kommunikation mit anderen Teilnehmern statt.

4.1.2.2 Schutzbedarf von AS-2 Nutzdaten

- Vertraulichkeit *sehr hoch*: Die Dateien enthalten Informationen, die nur vom Eigentümer eingesehen und verwendet werden dürfen. Im geplanten Einsatzumfeld werden wahrscheinlich auch personenbezogene Daten oder vertrauliche Daten gespeichert. Da der genaue Inhalt der Dateien unbekannt ist wird deren Schutzbedarf sicherheitshalber auf sehr hoch kategorisiert.

- Integrität *sehr hoch*: Die Nutzdaten können nur durch ein Replikat wiederhergestellt werden. Wenn keine integre Kopie der Sicherungsdaten vorhanden ist können die Daten nicht wiederhergestellt werden.

- Verfügbarkeit *hoch*: Die Verfügbarkeit der Nutzdaten ist grundsätzlich nur während der Datensicherung notwendig. Im Wiederherstellungsfall müssen sie jedoch in akzeptabler Zeit rekonstruiert werden. Die Verfügbarkeit anderer Teilnehmer zur Wiederherstellung der Nutzdaten kann auch außerhalb des Backupsystems angefordert werden, um den Wiederherstellungsvorgang zu beschleunigen. Es ist vom Backupsystem dafür Sorge zu tragen, dass genügend Sicherheitskopien der Sicherungsdaten bei anderen Teilnehmern existieren.

- Authentizität *nicht anwendbar*: Andere Teilnehmer erhalten vom Quellsystem die Anforderung zur Datensicherung. Da der Teilnehmer bereits authentifiziert ist und die Sicherungsdaten nur gelagert werden ist keine Authentifizierung erforderlich.

4.1.2.3 Schutzbedarf von AS-3 Sicherungsdaten

- Vertraulichkeit *normal*: Für die Sicherungsdaten ist kein erhöhter Schutzbedarf erforderlich, da deren Vertraulichkeit vor Verlassen des Quellsystems von der Backuplösung durch Verschlüsselung gewährleistet wird.
- Integrität *sehr hoch*: Die Integrität der Sicherungsdaten muss gewährleistet werden, da nur durch eine integre Kopie die ursprünglichen Nutzdaten wiederhergestellt werden können.
- Verfügbarkeit *hoch*: Die Verfügbarkeit der Sicherungsdaten ist notwendig, um die regelmäßige Prüfung der Integrität der Sicherungsdaten zu ermöglichen. Die Fehlererkennung muss regelmäßig erfolgen, ist aber nicht zeitkritisch. Des Weiteren werden die Sicherungsdaten für den Wiederherstellungsfall benötigt. Bei Totalausfall eines Teilnehmers können Sicherungsdaten von anderen Teilnehmern angefordert werden. Da ein Totalausfall außerhalb des Systems kommuniziert werden kann ist es dem Inhaber des Quellsystems im Anlassfall möglich, die Replizierung der Sicherungsdaten auf andere Systeme anzustoßen.
- Authentizität *nicht anwendbar*: Die Sicherungsdaten wurden von einem authentifizierten Teilnehmer übermittelt, daher ist keine Authentifizierung der Sicherungsdaten erforderlich.

4.1.2.4 Schutzbedarf von AS-4 Netzwerk-Infrastruktur

- Vertraulichkeit *normal*: Die Netzwerkkomponenten selbst enthalten keine Informationen. Die Vertraulichkeit der übertragenen Daten muss durch die Backuplösung gewährleistet werden.
- Integrität *normal*: An Netzwerkkomponenten werden keine besonderen Integritätsanforderungen gestellt, da die Integrität der Sicherungsdaten durch die Backuplösung sichergestellt wird. Manipulierte Netzwerkkomponenten könnten übertragene Sicherungsdaten verändern. Dies wird jedoch durch die Backuplösung im Rahmen der Verifizierung der Sicherungsdaten erkannt.
- Verfügbarkeit *normal*: Ohne funktionierende Netzwerk-Infrastruktur kann die Backuplösung für einen Teilnehmer nicht verwendet werden. Der Ausfall eines Teilnehmers führt jedoch nur zu geringen Einschränkungen der anderen Teilnehmer. Diese können in der Zwischenzeit auf andere Teilnehmer zur Datensicherung und -wiederherstellung ausweichen. Ein Quellsystem kann für die Dauer des Ausfalls der Netzwerkinfrastruktur keine Datensicherung durchführen. Dies führt aber zu einer Lücke in der Backupfolge für die Dauer des Ausfalls. Ein Totalausfall kann durch einfachen Austausch der Netzwerkkomponente behoben werden. Sobald das Netzwerk wieder verfügbar ist kann die Datensicherung fortgesetzt und wieder auf alle bisherigen Backups zugegriffen werden.

- Authentizität *nicht anwendbar*: Die Authentizität der Netzwerkkomponenten ist nicht notwendig, da diese nur für die Datenübertragung verwendet werden. Es bestehen keine Anforderungen, welche die Authentizität der Netzwerkkomponenten erfordern.

4.1.2.5 Schutzbedarf von AS-5 Endgeräte

- Vertraulichkeit *normal*: Auf Endgeräten werden möglicherweise Sicherungsdaten anderer Teilnehmer gespeichert sowie die Backuplösung installiert. Die Vertraulichkeit der gesicherten Daten wird durch die Backuplösung selbst gewährleistet.
- Integrität *normal*: Auf Endgeräten werden möglicherweise Sicherungsdaten anderer Teilnehmer gespeichert sowie die Backuplösung installiert. Deren Integrität wird durch Mechanismen der Backuplösung sichergestellt. An die Endgeräte selbst werden keine gesonderten Integritätsanforderungen gestellt.
- Verfügbarkeit *normal*: Ohne Endgerät kann die Backuplösung nicht verwendet werden. Der Ausfall eines Teilnehmers führt jedoch nur zu geringen Einschränkungen der anderen Teilnehmer, da diese für die Datensicherung und -wiederherstellung auf andere Teilnehmer ausweichen können. Bei Totalausfall eines Systems kann dies außerhalb der Backuplösung an andere Teilnehmer kommuniziert werden, damit diese die Sicherung ihrer Daten bei anderen Teilnehmern durchführen können.
- Authentizität *nicht anwendbar*: Die Authentizität der Endgeräte ist nicht notwendig, da Authentizität eines Teilnehmers über die Backuplösung abgewickelt wird. Es bestehen keine Anforderungen, welche die Authentizität der Endgeräte erfordern.

4.1.2.6 Schutzbedarf von AS-6 Lokale Speicher

- Vertraulichkeit *normal*: Auf lokalen Speichern werden Sicherungsdaten anderer Teilnehmer abgelegt. Die Vertraulichkeit dieser Daten wird durch die Backuplösung gewährleistet.
- Integrität *normal*: Es gilt das gleiche wie beim Schutzziel Vertraulichkeit: Die Integrität der Sicherungsdaten wird durch Anforderungen der Backuplösung sichergestellt.
- Verfügbarkeit *normal*: Der Ausfall eines lokalen Speichers betrifft nur andere Teilnehmer der Backuplösung. Aufgrund der Replikation der gesicherten Daten führt ein Ausfall nur zu geringen Auswirkungen auf andere Teilnehmer. Ein Totalausfall der lokalen Speicherlösung kann unter den Teilnehmern direkt kommuniziert werden, um die Sicherung ihrer Daten auf andere Systeme anzustoßen.
- Authentizität *nicht anwendbar*: Es besteht keine Anforderung zur Authentifizierung eines lokalen Speichers gegenüber der Backuplösung, daher ist dieses Schutzziel nicht notwendig.

4.1.2.7 Schutzbedarf von AS-7 Cloud-Speicher

- Vertraulichkeit *normal*: Der Zugang zum Cloud-Speicher kann nicht kontrolliert werden, daher muss der unrechtmäßige Zugriff auf Daten verhindert werden. Die Vertraulichkeit der Daten wird durch die Backuplösung gewährleistet.
- Integrität *normal*: Die Integrität der Sicherungsdaten am Cloud-Speicher muss gewährleistet oder überprüfbar sein. Wenn fehlerhafte Daten aus dem Cloud-Speicher als Sicherung abgelegt werden führt dies dazu, dass diese Replikate nicht zur Datenwiederherstellung genutzt werden können. Manipulierte Daten müssen am Zielsystem erkennbar sein, damit keine fehlerhaften Daten als Sicherung abgelegt werden und das Quellsystem nicht fälschlicherweise eine Erfolgsmeldung über die Datensicherung erhält. Die Integrität der Sicherungsdaten wird durch die Backuplösung sichergestellt.
- Verfügbarkeit *hoch*: Der Speicher des Cloud-Providers ist Bestandteil der Backupstrategie des Backupsystems. Für die Zeit eines Ausfalls kann kein einziger Teilnehmer die Datensicherung bei anderen Teilnehmern durchführen. Dies führt zu einer Lücke in der Backupfolge jedes Teilnehmers. Sobald der Cloud-Speicher wieder verfügbar ist, kann die Datensicherung fortgesetzt werden. Die Datenwiederherstellung ist von einem Ausfall des Cloud-Speichers nicht betroffen.
- Authentizität *normal*: Die Authentifizierung beim Cloud-Provider ist notwendig, damit nur der Benutzer des Quellsystems dieses zur Ablage seiner Sicherungsdaten nutzen kann. Andernfalls könnten andere Benutzer den Backupprozess stören, indem sie Sicherungsdaten am Cloud-Speicher manipulieren oder löschen.

4.2 Bedrohungs- und Risikoanalyse

In *Kapitel 4.1* wurde das Umfeld und die relevanten Schutzziele der geplanten Anwendung ermittelt. Auf dieser Basis wird eine Bedrohungsanalyse mithilfe des IT-Grundschutz-Kompendiums für die Backuplösung durchgeführt. Als Einschätzung für das Backupsystem wird festgelegt, dass nur Assets mit Schutzzielen mit Sicherheitsniveau *hoch* oder *sehr hoch* für die Bedrohungsanalyse zu berücksichtigen sind. Zur Ermittlung der Bedrohungen wird zuerst jedem Asset eine Reihe von Bausteinen zugeordnet, welche das Asset bestmöglich beschreiben. Bausteine, die nicht klar einem Asset der Backuplösung zugeordnet werden können oder im Rahmen einer Organisation zu sehen sind, werden nicht berücksichtigt. Dies ist beispielsweise bei den Bausteinen unter *ORP* (Organisation und Personal) oder *INF* (Infrastruktur) der Fall, da das Einsatzumfeld der Backuplösung nicht klar definiert ist.

Im Rahmen der Bedrohungsanalyse werden nur Bedrohungen analysiert, die sich aus dem aktuellen Umfeld des Prototypen und dem aktuellen Stand der Technik ergeben. Bedrohungen, die während dem längerfristigen Einsatz des Prototypen entstehen können, sind nicht betrachtet. Ein Beispiel dafür wäre die Bedrohung durch technische Entwicklungen,

welche dazu führen, dass die eingesetzten kryptografischen Verfahren die Schutzziele der Backuplösung nicht mehr erfüllen.

Zur Ermittlung der Bedrohungen werden anhand dieser Bausteine die elementaren Gefährdungen bestimmt. Dabei werden je Baustein nur jene elementaren Gefährdungen berücksichtigt, die ein Schutzziel mit Sicherheitsniveau *hoch* oder *sehr hoch* des Assets bedrohen. Aus den relevanten elementaren Gefährdungen werden unter Zuhilfenahme der spezifischen Gefährdungen aus dem IT-Grundschutzkompendium die Bedrohungen für jedes Asset der Backuplösung erarbeitet [10]. Es werden nur Bedrohungen betrachtet, die Auswirkungen auf die Backuplösung oder ein notwendiges System zum Betrieb der Backuplösung haben. Dies betrifft beispielsweise von der Anwendung genutzte Hardwarekomponenten (wie Festplatten) oder den Netzwerkpfad zwischen den kommunizierenden Teilnehmern. Details zu den Gefährdungen können im IT-Grundschutzkompendium [10] bzw. BSI-Standard [8] nachgeschlagen werden.

4.2.1 Bedrohungen zu AS-1 Backuplösung

Das Asset AS-1 Backuplösung wird mit den Bausteinen APP.6 Allgemeine Software und CON.1 Kryptokonzept abgebildet. Die Bedrohungen leiten sich aus den elementaren und spezifischen Gefährdungen laut BSI ab [10]:

BD-1 **Konzeptionsfehler in der Backuplösung**: Bei der Konzeption der Backuplösung können Konzeptionsfehler entstehen, die Auswirkung auf die Funktionsfähigkeit oder Schutzziele der Backuplösung haben und dazu führen, dass die Backuplösung nicht wie vorgesehen funktioniert.

BD-2 **Modifikation der Software durch Bezug aus unzuverlässiger Quelle oder Schadprogramme**: Eine Manipulation der Backuplösung kann auftreten, wenn diese aus einer unzuverlässigen Quelle bezogen wird. Die Backuplösung wird dadurch nicht wie vorgesehen funktionieren. In diesem Fall könnte die Software beispielsweise ohne aktivierten Backup-Mechanismus laufen oder Sicherungsdaten von anderen Teilnehmern immer verwerfen. Schadprogramme könnten ähnliche Auswirkungen haben oder sich über die Backuplösung im Backupsystem verteilen.

BD-3 **Offenlegung schützenswerter Informationen**: Ein Angreifer könnte Zugang zu geschützten Informationen wie der lokalen Konfiguration der Kommunikationspartner oder den Metainformationen der gesicherten Daten erlangen. Des Weiteren kann ein Angreifer versuchen die direkte Kommunikation der Teilnehmer abzuhören.

BD-4 **Manipulation oder Integritätsverlust von Informationen**: Informationen der Backuplösung könnten vorsätzlich geändert werden, um den Betrieb der Backuplösung zu stören. Ein Angreifer könnte beispielsweise durch Entfernung aller Sicherungspartner die Durchführung von Backups verhindern. Die Integrität der

Informationen kann auch durch Hardwarefehler, Schadprogramme oder ähnliches beeinträchtigt werden.

BD-5 **Software-Schwachstellen oder -Fehler in Kryptomodulen**: Schwachstellen und Fehler in kryptografischen Modulen, welche von der Backuplösung verwendet werden, beeinträchtigen die Sicherheit der eingesetzten kryptografischer Verfahren. Dies kann beispielsweise dazu führen, dass eigentlich vertrauliche Sicherungsdaten durch Dritte gelesen werden können oder das sich Daten nicht mehr vom Eigentümer entschlüsseln lassen.

BD-6 **Unsichere kryptographische Algorithmen oder Produkte**: Unsichere kryptografische Verfahren können von einem Angreifer in realistischer Zeit gebrochen werden um auf geschützte Informationen zuzugreifen oder diese zu manipulieren.

BD-7 **Kompromittierung des kryptografischen Schlüssels des Teilnehmers**: Ein Angreifer könnte in Besitz der kryptografischen Schlüssel des Teilnehmers zu kommen. Dies kann beispielsweise geschehen, wenn kryptografische Schlüssel ungeschützt gespeichert werden oder ein Zugriff durch den Arbeitsspeicher möglich ist. Dadurch wäre eine Entschlüsselung vertraulich gespeicherter Daten möglich.

BD-8 **Gefälschte Zertifikate anderer Teilnehmer**: Ein Angreifer könnte dem Teilnehmer ein gefälschtes Zertifikat eines anderen Teilnehmers unterschieben. Dies würde dazu führen, dass die Kommunikation mit diesem Teilnehmer nicht mehr möglich ist oder stattdessen mit dem Angreifer kommuniziert wird.

4.2.2 Bedrohungen zu AS-2 Nutzdaten

Für das Asset AS-2 Nutzdaten wird der Baustein CON.3 Datensicherungskonzept verwendet. Es werden jene Bedrohungen betrachtet, die sich auf die Sicherung der schützenswerten Daten beziehen.

BD-9 **Fehlende Wiederherstellbarkeit der Nutzdaten**: Die Datensicherung ist nur von Nutzen, wenn aus den Sicherungsdaten die ursprünglichen Daten wiederhergestellt werden können. Die Wiederherstellung kann beispielsweise fehlschlagen, wenn die Sicherungsdaten unvollständig oder nicht mehr integer sind. Ohne Test der Wiederherstellung kann nicht sichergestellt werden, dass diese aus weiteren Gründen wie Konfigurationsfehlern, Netzwerk-Timeouts oder ähnlichem nicht funktioniert.

BD-10 **Zugriff auf Daten durch unautorisierte Personen**: Andere Teilnehmer oder Angreifer können auf die Sicherungsdaten zugreifen, wenn diese nicht vertraulich gespeichert sind.

BD-11 **Manipulation der Daten**: Zur Sicherung markierte Nutzdaten könnten durch Angreifer, Schadprogramme, Hardwarefehler oder ähnliches manipuliert werden

oder vom Eigentümer aus Versehen verändert werden, bevor die Datensicherung abgeschlossen wurde.

BD-12 **Datenverlust**: Die Nutzdaten könnten gelöscht werden, nachdem für diese die Sicherung gestartet, aber bevor die Datensicherung abgeschlossen wurde.

4.2.3 Bedrohungen zu AS-3 Sicherungsdaten

Für das Asset AS-3 Sicherungsdaten werden die Bausteine CON.2 Datenschutz und CON.3 Datensicherungskonzept verwendet. Es werden jene Bedrohungen betrachtet, die sich auf die Sicherungsdaten am Sicherungsort betreffen.

BD-13 **Ungeeignete Aufbewahrung der Datenträger von Datensicherungen**: Je nach Aufbewahrungsort der Sicherungsdaten kann ein Angreifer auf diese zugreifen und sie manipulieren. Des Weiteren wäre eine unbeabsichtigte Veränderung der Sicherungsdaten möglich, wenn die speichernden Datenträger unter ungünstigen klimatischen Bedingungen gelagert werden.

BD-14 **Manipulation oder Integritätsverlust der Sicherungsdaten**: Die Sicherungsdaten der Backuplösung könnten vorsätzlich durch Teilnehmern oder Angreifer manipuliert werden. Dadurch können die Originaldaten aus den Sicherungsdaten nicht mehr wiederhergestellt werden. Die Integrität der Sicherungsdaten kann durch Schadprogramme, Hardwarefehler oder ähnliches beeinträchtigt werden.

BD-15 **Verlust von Sicherungsdaten**: Sicherungsdaten der Backuplösung könnten gelöscht werden, um die Wiederherstellung der Originaldaten zu verhindern.

4.2.4 Bedrohungen zu AS-4 Netzwerk-Infrastruktur , AS-5 Endgeräte und AS-6 Lokale Speicher

Alle Schutzziele der Assets AS-4 Netzwerk-Infrastruktur , AS-5 Endgeräte und AS-6 Lokale Speicher wurden in der Schutzbedarfsfeststellung in *Kapitel 4.1.2* mit dem Schutzbedarf *normal* oder *nicht anwendbar* bewertet. Da in der Bedrohungsanalyse nur die Sicherheitsniveaus *hoch* und *sehr hoch* berücksichtigt werden kommt es zu keiner Erhebung der spezifischen Bedrohungen für diese Assets.

4.2.5 Bedrohungen zu AS-7 Cloud-Speicher

Für das Asset AS-7 Cloud-Speicher mit den Bausteinen SYS.1.1 Allgemeiner Server, SYS.1.8 Speicherlösungen und OPS.2.2 Cloud-Nutzung sind die Bedrohungen folgend aufgelistet:

BD-16 **Verlust der Sicherungsdaten beim Cloud-Anbieter**: Werden Daten auf Cloud-Speichern abgelegt kann es zum Verlust der abgelegten Daten kommen. Dies

kann durch Konfigurationsfehler, Angriffe, eine Änderung des Speicherkontingents, durch Aktionen von Mitarbeitern des Cloud-Anbieters oder ähnlichem geschehen.

BD-17 **Überlastung von Cloud-Servern**: Durch eine Unterdimensionierung des Cloud-Servers oder durch DoS-Angriffe kann es zu dessen Überlastung kommen. Dadurch kann dieser legitime Anfragen nur mehr verzögert oder nicht mehr beantworten. Im Backupsystem kann es zu Datenverlust kommen, da die Datensicherung nicht abgeschlossen werden kann.

BD-18 **Abhängigkeit von einem Cloud-Diensteanbieter (Kontrollverlust)**: Die Nutzung eines Cloud-Anbieters führt zu einer gewissen Abhängigkeit zu diesem Anbieter. Ausgelagerte Daten können nicht vollständig kontrolliert werden. Das Backupsystem ist darauf angewiesen, dass der Cloud-Anbieter notwendige Sicherheitsmaßnahmen korrekt umsetzt.

BD-19 **Beeinträchtigung der eigenen Dienste durch andere Kunden des Cloud-Diensteanbieters**: Da sich bei Cloud-Anbietern mehrere Kunden eine gemeinsame Infrastruktur teilen, müssen deren Daten und Ressourcen korrekt voneinander getrennt werden. Ist dies nicht der Fall kann das Nutzungsverhalten eines Kunden die Systeme der anderen Kunden beeinflussen. Im schlimmsten Fall kann auf Daten anderer Kunden zugegriffen und diese Daten ausgelesen, manipuliert oder gelöscht werden.

BD-20 **Ausfall des Cloud-Diensteanbieters**: Ein Ausfall des Cloud-Anbieters führt bei der Backuplösung dazu, dass von keinem Teilnehmer eine Datensicherung durchgeführt werden kann.

BD-21 **Verstoß gegen rechtliche Vorgaben**: Viele Cloud-Anbieter operieren im internationalen Umfeld und haben ihre Rechenzentren in anderen Staaten. Damit unterliegen diese anderen nationalen Gesetzen als der Eigentümer der Originaldaten. Dies kann dazu führen, dass je nach Art der gesicherten Daten, bei Verwendung des Cloud-Anbieters gegen rechtliche Rahmenbedingungen wie Datenschutz oder Informationspflichten verstoßen wird.

Zu den ermittelten Bedrohungen lassen sich eine Reihe von Sicherheitsmaßnahmen treffen, um diesen zu begegnen. Diese werden im folgenden *Kapitel 4.3* beschrieben.

4.3 Sicherheitsmaßnahmen

Für jedes Asset wurde in *Kapitel 4.2* eine Liste an Bedrohungen für die Schutzziele mit *hohem* oder *sehr hohem* Sicherheitsniveau ermittelt. Dieses Kapitel beschreibt die Maßnahmen, die gewählt werden, um die Bedrohungen zu verhindern oder zu mitigieren. Dadurch werden die Schutzziele im Backupsystem bestmöglich erfüllt. In *Kapitel 4.3.1* werden alle getroffenen Maßnahmen erläutert. In *Kapitel 4.3.2* folgt eine genaue Zuordnung jeder Maßnahme zu den Bedrohungen, die durch die Maßnahme behandelt werden.

4.3.1 Maßnahmen

Es folgen die Maßnahmen, die im Rahmen der Backuplösung eingesetzt werden. Eine Maßnahme kann mehrere Assets betreffen und mehreren Bedrohungen entgegenwirken. Eine grundsätzliche Maßnahme ist die Umsetzung der Anforderungen aus *Kapitel 3.1* der Backuplösung, da diese Anforderungen bereits Einfluss auf den Schutzbedarf und die Bedrohungsanalyse hatten.

MA-1 **Review des Konzepts der Backuplösung**: Die Anforderungen und das grobe Konzept der Backuplösung wurden in *Kapitel 3* beschrieben. Das Detailkonzept folgt in Folgekapiteln. Diese Diplomarbeit wird während der Verfassung mehrfach durch mit der Thematik vertraute Kollegen aus dem universitären Umfeld geprüft. Im Rahmen dessen wird auch das Konzept der Backuplösung wiederholten Reviews unterzogen. Bei Abschluss der Diplomarbeit wird ein Peer-Review durchgeführt. Dieser Prozess minimiert das Risiko von Konzeptionsfehlern.

MA-2 **Einsatz von Analysetools zur Erkennung von Software-Konzeptionsfehlern und -Schwachstellen**: Während der Entwicklung des Prototypen werden Tools eingesetzt, welche den Quellcode der Anwendung sowie alle verwendeten Software-Bibliotheken analysieren. Fehler und Schwachstellen im Quellcode werden zur Entwicklungszeit erkannt und können bei Relevanz behoben werden. Die verwendeten Software-Bibliotheken werden auf bekannte Schwachstellen untersucht. Dadurch können für unsichere Software-Bibliotheken mit relevanten Schwachstellen Maßnahmen wie beispielsweise ein Upgrade der Bibliotheks-Version durchgeführt werden.

MA-3 **Empfohlener Einsatz von Sicherheitsmaßnahmen gegen Schadprogramme**: Dem Anwender wird empfohlen geeignete Maßnahmen vorzusehen, um Schadprogramme am System zu identifizieren. Dazu gehört der Einsatz von betriebssystemeigenen Sicherheitsmechanismen oder die Verwendung eines Virenscanners. Dadurch kann Schadsoftware erkannt und entfernt sowie eine Neuinstallation des Systems oder der Backuplösung vorgenommen werden. Die Entscheidung über den Einsatz obliegt jedoch dem Anwender, die Backuplösung wird dies nicht überprüfen.

MA-4 **Verfügbarkeit des Sourcecodes in einem öffentlich verfügbaren Repository**: Die Backuplösung wird unter einer OpenSource-Lizenz entwickelt und steht in Form eines öffentlich verfügbaren Sourcecode-Repositories zur Verfügung. Dies ermöglicht die Neuinstallation bei Problemen oder Fehlern. Da es sich bei der Backuplösung nur um einen Prototypen handelt wird keine Wartung der Backuplösung gewährleistet. Die Verfügbarkeit des Sourcecodes erlaubt es anderen Personen, im Anlassfall Anpassungen oder Fehlerbehebungen für die Backuplösung durchzuführen. Der Sourcecode wird jedoch entsprechend abgesichert, sodass eine Änderung nur nach Review durch den Ersteller dieser Diplomarbeit in die Codebasis aufgenommen wird.

MA-5 **Verifikation der Applikationsdateien per Hash-Wert**: Der Anwender muss erkennen können, ob die Applikationsdateien der Backuplösung modifiziert wurden. Um dies zu ermöglichen wird mit dem Sourcecode der Backuplösung auch ein Hash-Wert zur Verfügung gestellt, mit dem der Anwender die Integrität der Applikationsdateien mit den Bordmitteln des Betriebssystems prüfen kann.

MA-6 **Verwendung von als derzeit sicher geltenden kryptografischen Verfahren**: Für alle Einsatzszenarien von Kryptografie wird auf die aktuellen Empfehlungen von Einrichtungen wie dem BSI oder der NSA zurückgegriffen. Diese Empfehlungen wurden von der Fachwelt auf Funktionsweise und Sicherheitslücken untersucht und sind für die Benutzung bis zu einem gewissen Zieljahr verwendbar.

MA-7 **Verschlüsselung der Konfigurations- und Metadaten**: Um die Vertraulichkeit der lokalen Konfigurations- und Metadaten, die zum Betrieb der Backuplösung notwendig sind, zu gewährleisten, müssen diese lokal mit einem in MA-6 gewählten Verfahren verschlüsselt gespeichert werden.

MA-8 **Integritätsprüfung der Konfigurations- und Metadaten**: Eine ungeplante Manipulation der lokalen Konfigurations- und Metadaten wird erkannt, indem diese mithilfe kryptografischer Verfahren um Metadaten wie Hashwerte ergänzt werden, um eine Integritätsprüfung zu ermöglichen. Wird eine Änderung der Daten erkannt, kann der Benutzer eine Neuinstallation und Neukonfiguration der Backuplösung vornehmen, um mit der korrekt funktionierenden Datensicherung fortzufahren.

MA-9 **Verschlüsselung der Kommunikation mit anderen Teilnehmern**: Die direkte Kommunikation mit anderen Teilnehmern wird über öffentliche Netzwerke übertragen, daher erfolgt die Datenübertragung verschlüsselt.

MA-10 **Gesicherte Ablage von kryptografischen Schlüsseln**: Kryptografische Schlüssel werden immer vertraulich abgelegt, um den Zugriff durch unautorisierte Personen zu verhindern. Die Nutzung von privaten Schlüsseln ist nur möglich, wenn diese verschlüsselt und durch ein Geheimnis in Form eines Passwortes gesichert sind. Werden von der Anwendung Schlüssel generiert und gespeichert werden diese ebenfalls verschlüsselt und durch ein vom privaten Schlüssel abgeleitetes Geheimnis gesichert.

MA-11 **Geeignete Erzeugung von Schlüsseln in der Backuplösung**: Kryptografische Schlüssel werden von geeigneten Schlüsselgeneratoren der Backuplösung nach aktuellen kryptografischen Verfahren und Standards erzeugt. Dazu wird den Empfehlungen von Einrichtungen wie dem BSI gefolgt. Diese Verfahren zur Schlüsselgenerierung sind in der Backuplösung hinterlegt, um Fehlkonfigurationen zu verhindern.

MA-12 **Entzug der Kommunikationsrechte von anderen Teilnehmern**: Der Anwender kann individuell entscheiden, mit welchen anderen Teilnehmern er zusammenarbeitet. Dies ist bereits im Konzept der Backuplösung vorgesehen. Einem

Benutzer muss es möglich sein einem anderen Teilnehmer die Erlaubnis zur Kommunikation wieder zu entziehen. Dies ermöglicht die Unterbindung weiterer Kommunikation mit Teilnehmern denen nicht mehr vertraut wird. Für den Fall der Kompromittierung des privaten Schlüssels muss der Betroffene alle Kommunikationspartner informieren. Dies ermöglicht das Unterbrechen der Verbindung bei allen Partnern.

MA-13 **Bestätigung des öffentlichen Schlüssels anderer Teilnehmer**: Der Anwender muss der Backuplösung beim Hinzufügen eines neuen Kommunikationspartners dessen öffentlichen Schlüssel bekanntgeben. Die Backuplösung zeigt während diesem Vorgang ein signifikantes Merkmal wie beispielsweise den Fingerprint dieses Schlüssels an. Dieses Merkmal muss der Anwender auf einem Weg außerhalb der Backuplösung von diesem Teilnehmer erhalten. Der Anwender kann damit das angezeigte Merkmal mit dem erwarteten vergleichen und die Korrektheit des Schlüssels bestätigen.

MA-14 **Erzeugung von Sicherungsdaten unabhängig von Verteilung bei anderen Teilnehmern**: Während der Sicherung werden aus den Nutzdaten möglichst schnell Sicherungsdaten erzeugt und lokal abgelegt. Es wird nach lokaler Verarbeitung einer zu sichernden Datei nicht darauf gewartet, dass deren Sicherungsdaten in der Cloud abgelegt oder an andere Teilnehmer verteilt werden. Es wird sofort mit der Verarbeitung der nächsten Datei fortgefahren, um möglichst rasch für alle Nutzdaten Sicherungsdaten zu erzeugen.

MA-15 **Mehrere Replikate der Sicherungsdaten**: Die Speicherung von Replikaten ist im Konzept der Backuplösung bereits vorgesehen. Von allen Sicherungsdaten werden mehrere Replikate auf mindestens drei unterschiedliche Teilnehmer verteilt. Dies gewährleistet die Verfügbarkeit der Daten, sollte ein Teilnehmer ausfallen, die Sicherungsdaten manipuliert werden, das Speichersystem Fehler erzeugen oder ähnliches.

MA-16 **Regelmäßige Verifikation der Sicherungsdaten bei anderen Teilnehmern**: Die Sicherungsdaten bei anderen Teilnehmern müssen in regelmäßigen Intervallen auf deren Verfügbarkeit und Integrität überprüft werden. Dazu werden kryptografische Verfahren eingesetzt, damit ein anderer Teilnehmer den Nachweis über den Besitz der Sicherungsdaten erbringen kann. Das Verifikationsintervall wird von der Backupanwendung vorkonfiguriert und ist änderbar. Jeder Anwender kann das Intervall zur Verifikation seiner Sicherungsdaten anpassen.

MA-17 **Verschlüsselung der Sicherungsdaten mit derzeit als sicher geltenden symmetrischem Verfahren**: Die Sicherungsdaten werden vor Verlassen der Datenquelle mit einem in MA-6 gewählten Verfahren verschlüsselt, um deren Vertraulichkeit zu gewährleisten. Dadurch ist die Speicherung bei anderen Teilnehmern oder in der Cloud möglich, ohne das unautorisierte Personen Zugriff

auf den Dateninhalt erhalten. Der Schlüssel ist nur dem Dateneigentümer bekannt. Aus Effizienzgründen und zur Schonung der Systemressourcen werden die Sicherungsdaten symmetrisch verschlüsselt.

MA-18 **Integritätsprüfung der Sicherungsdaten**: Da die Sicherungsdaten im Wiederherstellungsfall von nicht kontrollierbaren Systemen kommen, muss im Wiederherstellungsfall deren Korrektheit bzw. eine Manipulation erkannt werden können. Dazu werden alle Sicherungsdaten mithilfe von kryptografischen Verfahren aus MA-6 um Metadaten wie Hashwerte ergänzt, welche eine Integritätsprüfung ermöglichen. Dadurch können im Wiederherstellungsfall fehlerhafte Sicherungsdaten erkannt und alternativ von einem anderen Teilnehmer angefordert werden.

MA-19 **Verzögerte Löschung der lokalen Sicherungsdaten**: Die Datenquelle darf die lokal vorbereiten Sicherungsdaten nicht sofort nach dem Upload zum Cloud-Anbieter löschen. Dies darf erst geschehen, sobald von einer ausreichenden Anzahl an Teilnehmern eine Bestätigung über die erfolgte Datensicherung eingelangt ist. Bei Daten- oder Integritätsverlust der Sicherungsdaten beim Cloud-Anbieter können diese dadurch erneut hochgeladen werden.

MA-20 **Integritätsprüfung der Cloud-Sicherungsdaten**: Sichernde Teilnehmer erhalten die Sicherungsdaten aus dem Cloud-System. Da dieses nicht kontrolliert werden kann, muss die Korrektheit der Daten bzw. eine Manipulation erkannt werden. Dazu werden alle Sicherungsdaten mithilfe von kryptografischen Verfahren aus MA-6 um Metadaten ergänzt, um eine Integritätsprüfung durch sichernde Teilnehmer zu ermöglichen. Dadurch können fehlerhafte Sicherungsdaten erkannt und die Datenquelle darüber informiert werden.

MA-21 **Direkte Teilnehmer-Kommunikation bei fehlenden/nicht integren Daten im Cloud-Speicher**: Wenn ein Teilnehmer eine Anforderung zur Sicherung von Daten nicht abschließen kann weil diese am Cloud-Speicher fehlen oder nicht integer sind, dann muss dieser Teilnehmer die Datenquelle darüber informieren. Dadurch können die zu sichernden Daten erneut hochgeladen werden, um die Vervollständigung der Datensicherung zu ermöglichen.

MA-22 **Minimale Anforderungen an den Cloud-Speicher**: Zur Wahrung der Portabilität des Cloud-Speichers werden nur minimale Anforderungen an diesen gestellt. Dieser muss über technische Schnittstellen die Speicherung und Entfernung einer Datei unterstützen sowie eine Schnittstelle zur Verfügung stellen, um zu dieser Datei eine öffentliche URL zu erzeugen. Dies ermöglicht den möglichst einfachen Wechsel des Cloud-Anbieters.

MA-23 **Verwendung mehrerer Cloud-Speicher**: Zur Wahrung der Verfügbarkeit des Cloud-Speichers müssen mehrere Cloud-Speicher angebunden werden. Dies führt bei Überlastung oder Ausfall eines Cloud-Anbieters zur weiteren reibungslosen Funktionsweise der Backuplösung. In Kombination mit MA-22 können dadurch in Zukunft zusätzliche Cloud-Anbieter einfacher angebunden werden.

MA-24 **Sorgfältige Auswahl des Cloud-Anbieters**: Bei der Wahl des Cloud-Anbieters muss mit Sorgfalt vorgegangen werden. Es dürfen nur Cloud-Anbieter gewählt werden, die regelmäßige Sicherheitsaudits durchführen lassen und über eine ausreichende Informationssicherheit verfügen. Dieser Nachweis lässt sich in Form eine Zertifizierung, beispielsweise nach IT-Grundschutz oder ISO/IEC 27001 erbringen.

MA-25 **Leitfaden für Tausch des privaten Schlüssels**: Bei Kompromittierung des privaten Schlüssels muss dessen Tausch möglich sein. Dessen Änderungen betrifft auch andere kryptologische Artefakte der Backuplösung wie beispielsweise die verschlüsselten Backupdaten, die entsprechend migriert werden müssen. Es muss ein Leitfaden zur notwendigen Vorgehensweise beim Tausch des privaten Schlüssels erstellt werden.

4.3.2 Zuordnung der Maßnahmen zu Bedrohungen

Die Maßnahmen aus *Kapitel 4.3.1* sollen vor bestimmten Bedrohungen schützen oder deren Auswirkungen mitigieren. Diese Zuordnung ist in *Tabelle 4.1* abgebildet.

	MA-1	MA-2	MA-3	MA-4	MA-5	MA-6	MA-7	MA-8	MA-9	MA-10	MA-11	MA-12	MA-13	MA-14	MA-15	MA-16	MA-17	MA-18	MA-19	MA-20	MA-21	MA-22	MA-23	MA-24	MA-25
BD-1	X	X																							
BD-2			X	X	X																				
BD-3							X		X																
BD-4								X																	
BD-5		X				X																			
BD-6						X					X														
BD-7										X		X													X
BD-8						X		X				X	X												
BD-9																X									
BD-10																	X								
BD-11			X											X	X										
BD-12			X											X	X										
BD-13																		X							
BD-14			X															X		X					
BD-15			X												X										
BD-16																			X		X				
BD-17																						X	X		
BD-18																						X	X	X	
BD-19	X																X		X	X	X			X	
BD-20																							X	X	
BD-21																								X	

Tabelle 4.1: Zuordnung von Maßnahmen zu Bedrohungen

KAPITEL 5

Konzeption und Implementierung des Peer-to-Peer-Backupsystems mit Cloud-Unterstützung

Dieses Kapitel stellt die Konzepte der Backuplösung genauer dar. Grundlage für die Konzeption sind die Anforderungen aus *Kapitel 3.1* sowie die aus der Sicherheitsanalyse abgeleiteten Maßnahmen aus *Kapitel 4.3.1*. In *Kapitel 5.1* wird das generelle Konzept der Datensicherung, Begriffe und das Umfeld des Backupsystems definiert. *Kapitel 5.2* beinhaltet Entscheidungen und Begründungen über die verwendeten Cloud-Provider, eingesetzte Technologien sowie die eingesetzte kryptografische Verfahren und referenziert auf die dadurch erfüllten Maßnahmen. Danach folgt in *Kapitel 5.3.1* das detaillierte Konzept der Datensicherung, gefolgt vom Konzept der Datenverifikation (*Kapitel 5.3.2*) und dem Konzept zur Wiederherstellung der Originaldaten (*Kapitel 5.3.3*).

5.1 Basiskonzept und Systemarchitektur

Dieses Kapitel gibt einen groben Überblick über die Funktionsweise und Architektur des Backupsystems. Folgend werden einige Begriffe für die Konzeption und Implementierung des vorgeschlagenen Systems definiert. In den folgenden Kapiteln werden mithilfe dieser Begriffe die Architektur und Komponenten des Backupsystems erläutert.

- Backupsystem: Das Backupsystem ist der Verbund der Personen, die untereinander Daten sichern. Jede Person bildet mit seinen gewählten Kommunikationspartnern ein eigenes Backupsystem.

- Backupanwendung: Die Backupanwendung bezeichnet die Softwareanwendung zur Durchführung von Backups mit allen dazugehörigen Komponenten (siehe AS-1 Backuplösung)
- Teilnehmer: Ein Teilnehmer des Backupsystems ist eine Person, die innerhalb des Backupsystems mit anderen Teilnehmer kooperiert und damit das Backupsystem bildet. Ein Teilnehmer wählt andere Teilnehmer aus, mit denen er innerhalb des Backupsystems zusammenarbeiten möchte. Jeder Teilnehmer betreibt die Backupanwendung auf seinem Hardware-System, um eigene Daten zu sichern und Backup-Daten anderer Teilnehmer zu empfangen.
- Knoten/System: Ein Knoten bzw. System ist ein konkretes Hardwaresystem wie der Computer oder Server eines Teilnehmers. Auf einem Knoten wird die Backupanwendung installiert um Teil des Backupsystems zu werden. Ein Teilnehmer betreibt immer genau einen Knoten. Aufgrund dieser eindeutigen Zuordnung zwischen Teilnehmer und Knoten werden die Begriffe Teilnehmer, Knoten und System in weiterer Folge synonym verwendet.
- Die Definition von AS-2 Nutzdaten , AS-3 Sicherungsdaten , AS-4 Netzwerk-Infrastruktur , AS-5 Endgeräte , AS-6 Lokale Speicher und AS-7 Cloud-Speicher findet sich in *Kapitel 4.1.1.*
- Datenblock: Ein Datenblock bezeichnet beliebige Daten, die vom Backupsystem verwaltet und gesichert werden. Beispielsweise werden Nutzdaten zur Verwaltung in kleinere Datenblöcke aufgeteilt. Diese haben standardmäßig eine Größe von 500KB. 500KB wurden gewählt, damit der Up- und Download von Datenblöcken in kleinen Zeiteinheiten durchführbar ist und Unterbrechungen im Datenverkehr nur kleine Datenmengen betreffen. Die Größe der Datenblöcke kann aber in der Backupanwendung konfiguriert werden.
- Sicherungsblock: Ein Sicherungsblock entspricht einem verschlüsselten Datenblock. Der Inhalt eines Sicherungsblocks ist vertraulich und mit Metadaten zur Integritätssicherung versehen. Ein Sicherungsblock kann zu anderen Teilnehmern oder dem Cloud-Speicher verteilt werden.
- Replikat: Ein Replikat bezeichnet einen Sicherungsblock bei einem anderen Teilnehmer. Das Backupsystem sorgt dafür, dass pro Sicherungsblock mehrere Replikate bei anderen Teilnehmern gespeichert werden.
- Metadaten: Jegliche Daten, die von der Backupanwendung intern zur Verwaltung der Sicherungen verwendet werden, werden als Metadaten bezeichnet. Dabei handelt es sich beispielsweise um Hash-Werte von Datenblöcken oder um Verifizierungsdaten zur Überprüfung der Replikate.

Eine Verdeutlichung der Beziehung zwischen den einzelnen Teilnehmern der Backuplösung wird in *Abbildung 5.1* gezeigt. Jeder Teilnehmer bestimmt, mit welchen anderen

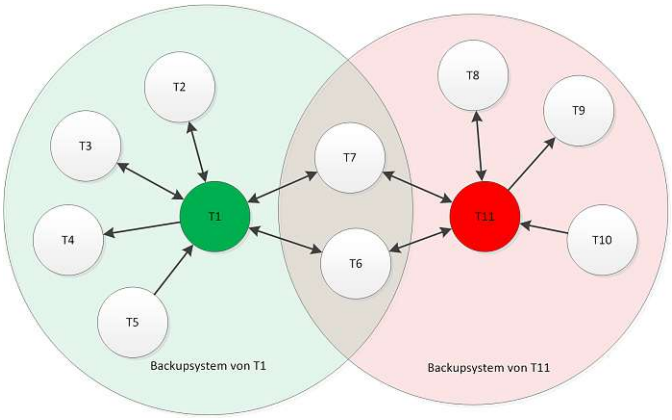

Abbildung 5.1: Grundlegender Aufbau des Backupsystems

Teilnehmern in der Backuplösung zusammengearbeitet wird. Zwei Teilnehmer können nur miteinander kommunizieren, wenn beide dieser Zusammenarbeit zustimmen. Beispielsweise entscheidet sich Teilnehmer T1 mit T2 bis T7 zusammenzuarbeiten und umgekehrt. Das selbe gilt für Teilnehmer T11 mit T6 bis T10. Da sich Teilnehmer T1 und T11 aber nicht gegenseitig als Kommunikationspartner akzeptiert haben, nehmen sie nicht am Backupsystem des jeweils anderen teil und können generell nicht miteinander kommunizieren. Nur die Teilnehmer T7 und T8 können sowohl mit T1 und T11 kommunizieren. Jeder Teilnehmer bildet somit für sich ein eigenes Backupsystem mit seinen Kommunikationspartnern.

Je Teilnehmer wird noch unterschieden, ob von diesem Teilnehmer Sicherungsdaten akzeptiert werden und ob dieser Teilnehmer Sicherungsziel sein darf. Um Daten bei einem anderen Teilnehmer zu sichern muss dieser lokal als Sicherungsziel konfiguriert sein. Gleichzeitig muss dieser Teilnehmer auch Backups von der Datenquelle erlauben. Am Beispiel von *Abbildung 5.1* bedeutet dies, dass der Teilnehmer T1 Daten bei T4 sichern kann, aber keine Sicherungsdaten von T4 akzeptiert. Umgekehrt erlaubt der Teilnehmer T1 die Ablage von Sicherungsdaten von T5, sichert aber keine Daten bei diesem Teilnehmer.

5.1.1 Systemarchitektur

Das Backupsystem wird als verteilte Anwendung konzipiert und verwendet sowohl Konzepte der Peer-to-Peer-Architektur (*siehe Kapitel 2.2.3*) also auch der Cloud-Architektur (*siehe Kapitel 2.2.2*). Die Architektur des Backupsystems wird in *Abbildung 5.2* dargestellt.

Die Backupanwendung wird bei jedem Teilnehmer installiert und beinhaltet zusätzlich eine eingebettete, persistente Datenbank, mit der über JDBC kommuniziert wird. In dieser Datenbank werden Konfigurationsdaten wie die gesicherten Verzeichnisse, Metainformationen über die zu sichernden Daten, die konfigurierten Teilnehmer oder laufende Wiederherstellungen gehalten. Die Backupanwendung verwendet I/O-Zugriffe, um auf den

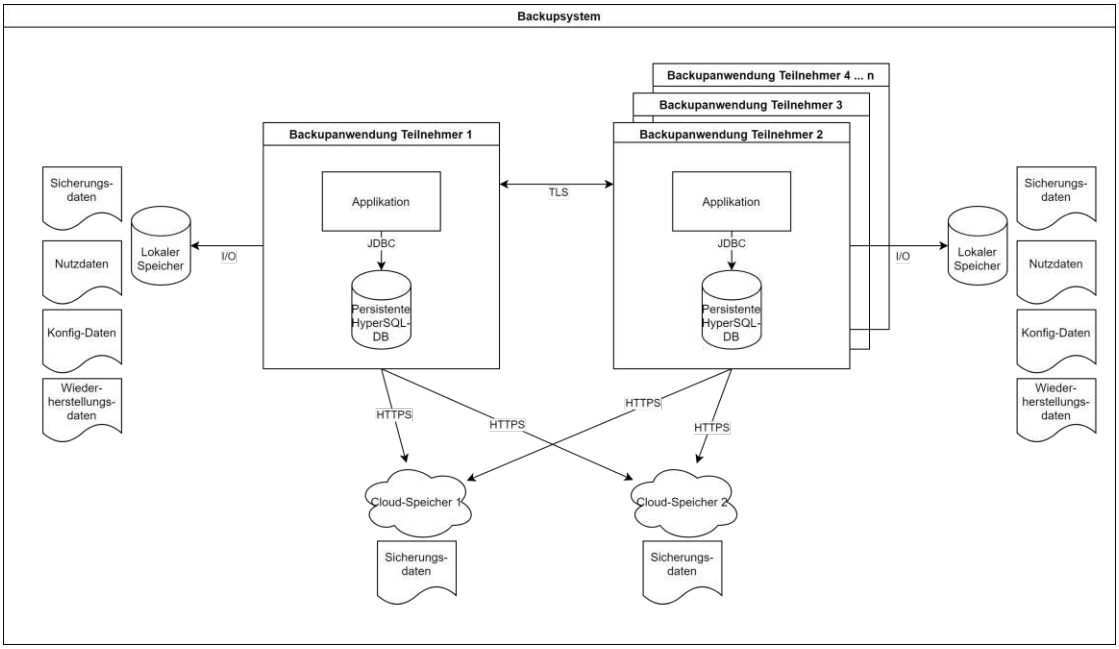

Abbildung 5.2: Architektur des Backupsystems

lokalen Speicher zuzugreifen. Auf diesem befinden sich die zu sichernden Nutzdaten sowie die daraus generierten Sicherungsdaten, welche an andere Teilnehmer zur Datensicherung verteilt werden. Die Sicherungsdaten anderer Teilnehmer werden ebenfalls im lokalen Speicher abgelegt. Des Weiteren sind noch grundlegende Daten wie das Schlüsselpaar des Teilnehmers oder die Initialkonfiguration der Backupanwendung für den ersten Start auf dem lokalen Speicher hinterlegt. Die Kommunikation zwischen Teilnehmern erfolgt direkt per TLS.

In der Backupanwendung jedes Teilnehmers werden die zu nutzenden Cloud-Speicher konfiguriert. Mit diesen kommuniziert die Anwendungssoftware per HTTPS zur Ablage von Sicherungsdaten. Andere Teilnehmer verwenden ebenfalls HTTPS um diese Sicherungsdaten abzuholen und im eigenen lokalen Speicher zu sichern.

5.1.2 Peer-to-Peer- und Cloud-Architektur

Das Backupsystem verwendet sowohl Peer-to-Peer- als auch Cloud-Konzepte zur Erfüllung seiner Anforderungen und Sicherheitsziele. Folgend werden Details zu den entsprechenden Konzepten erläutert und Referenzen zu den damit erfüllten Anforderungen und Maßnahmen angegeben.

5.1.2.1 Peer-to-Peer-Architektur

Teilnehmer kommunizieren untereinander immer direkt wie in einem Peer-to-Peer-Netzwerk (*siehe Kapitel 2.2.3*) ohne Verwendung des Cloud-Providers oder eines zentralen

Servers. Jeder Teilnehmer wird durch ein Public-Key/Private-Key-Schlüsselpaar identifiziert. Ein Teilnehmer kann von der Backupanwendung entweder ein neues Schlüsselpaar generieren lassen oder ein existierendes Schlüsselpaar zur Verfügung stellen. Bei der Generierung wird zusätzlich ein Zertifikat mit dem öffentlichen Schlüssel erzeugt. Mit der Generierung des Schlüsselpaares wird der Maßnahme MA-11 Geeignete Erzeugung von Schlüsseln in der Backuplösung zur sicheren Erzeugung von Schlüsseln entsprochen. Die Nutzung eines bestehenden Schlüsselpaares ist beispielsweise bei der Reinstallation der Backupanwendung notwendig, damit diese auf die Sicherungsdaten des Benutzers zugreifen kann.

Zur Authentifizierung eines Teilnehmers wird dessen Zertifikat mit dem öffentlichen Schlüssel verwendet. Dieses muss an alle anderen Teilnehmer, mit welchen kommuniziert werden soll, verteilt werden. Zusätzlich muss auf einem anderen Transportkanal ein identifizierendes Merkmal des Zertifikats übermittelt werden, damit der Empfänger dessen Korrektheit prüfen kann (siehe MA-13 Bestätigung des öffentlichen Schlüssels anderer Teilnehmer). Umgekehrt müssen auch die Zertifikate der anderen Teilnehmer in der eigenen Backupanwendung konfiguriert werden um die Kommunikationsverbindung aufzubauen. Mithilfe von kryptographischen Verfahren kann damit die gegenseitige Authentifizierung der Teilnehmer durchgeführt werden (siehe AF-11 Technische Möglichkeit zur Authentifizierung der Teilnehmer). Wie in MA-12 Entzug der Kommunikationsrechte von anderen Teilnehmern definiert, wird damit die Kommunikation nur mit bekannten Teilnehmern garantiert und die nachträgliche Unterbindung der Kommunikation zu ausgewählten Teilnehmern ermöglicht. Die Schlüsselpaare bzw. Zertifikate der Teilnehmer werden auch dazu verwendet, um die direkte Kommunikation der Teilnehmer zu verschlüsseln (siehe MA-9 Verschlüsselung der Kommunikation mit anderen Teilnehmern).

Zur direkten Kommunikation benötigt jeder Teilnehmer eine Adresse, um im Backupsystem erreichbar zu sein. Unter dieser Adresse kann der Knoten des Teilnehmers Verbindungsanfragen anderer Knoten entgegennehmen. Ein Knoten kann mehrere Adressen besitzen. Für die Erreichbarkeit über ein öffentliches Netzwerk wie das Internet ist eine öffentliche Netzwerkadresse in Form einer IP-Adresse oder eines DNS-Eintrags notwendig. Befinden sich mehrere Knoten im gleichen LAN genügt eine private IP-Adresse des gemeinsamen lokalen Netzwerks. Die Kommunikation zwischen Knoten erfolgt über TLS. Es kann auch Knoten mit dynamische Adresse geben oder Knoten deren Adresse sich je nach Aufenthaltsort ändert. Dies wäre beispielsweise bei einem Laptop der Fall der an verschiedenen Orten benutzt wird. In diesem Fall ist die direkte Kommunikation mit diesem Knoten nur möglich, wenn dieser die Verbindung zu einem Knoten mit statischer Adresse aufbaut. Wie in *Kapitel 3.3* beschrieben wird im Rahmen dieser Diplomarbeit davon ausgegangen, dass zwischen einzelnen Knoten eine direkte Netzwerkverbindung besteht.

5.1.2.2 Cloud-Architektur

Sicherungsdaten werden nicht direkt an andere Teilnehmer gesendet sondern werden im Cloud-Speicher eines Cloud-Anbieters abgelegt. Jeder Teilnehmer benötigt dazu einen eigenen Bereich bei einem Cloud-Anbieter, andere Teilnehmer haben auf diesen Bereich keinen Zugriff.

Der Cloud-Speicher ist über Authentifizierungsmaßnahmen abgesichert. Diese hängen vom verwendeten Cloud-Anbieter ab. Dabei kann es sich beispielsweise um eine Authentifizierung basierend auf Username und Passwort oder per generiertem Access-Token handeln. Die Registrierung beim Cloud-Anbieter wird nicht von der Backupanwendung unterstützt und muss manuell durch den Anwender durchgeführt werden. Die jeweiligen Authentifizierungsdaten werden danach in der Backupanwendung hinterlegt. Daher kann nur die Backupanwendung des Anwenders direkt Daten im seinem Cloud-Speicher ablegen, lesen oder löschen.

Nach dem Upload der Sicherungsdaten wird über das API des Cloud-Anbieters eine öffentlich erreichbar Uniform Resource Locator (URL) zu den abgelegten Daten erzeugt. Das Quellsystem übermittelt daraufhin jenen Teilnehmern, welche die Speicherung der Sicherungsdaten durchführen sollen, eine direkte Nachricht mit der Download-URL der Sicherungsdaten. Jeder Teilnehmer ladet die Sicherungsdaten über die URL herunter, speichert sie auf seinem System und bestätigt dem Quellsystem die Sicherung der Daten. Erst nach erfolgter Sicherung durch genügend Teilnehmer verwendet das Quellsystem das API des Cloud-Anbieters, um die Sicherungsdaten aus dem Cloud-Speicher zu löschen. Dies entspricht der Maßnahme MA-19 Verzögerte Löschung der lokalen Sicherungsdaten .

Zur Unterstützung dieses Ablaufs muss ein Cloud-Anbieter folgende Schnittstellen für den Cloud-Speicher zur Verfügung stellen, welche die minimalen Anforderungen an den Cloud-Speicher darstellen (MA-22 Minimale Anforderungen an den Cloud-Speicher):

- Upload einer Datei in den Cloud-Speicher
- Freigabe einer gespeicherten Datei per öffentlich erreichbarer URL. Diese muss zufällige Pfad-Bestandteile enthalten um zu verhindern, dass Angreifer die öffentlich erreichbare URL erraten können.
- Auflistung aller Dateien im Cloud-Speicher
- Löschen einer Datei aus dem Cloud-Speicher

Von der Backuplösung werden zwei Cloud-Anbieter angebunden und zufällig beim Upload der Sicherungsdaten verwendet. Dies wirkt Störungen entgegen und setzt die Maßnahme MA-23 Verwendung mehrerer Cloud-Speicher um. Eine Störung kann der zeitweilige Ausfall eines Cloud-Anbieters oder auch eine langfristige Störung wie die Einstellung

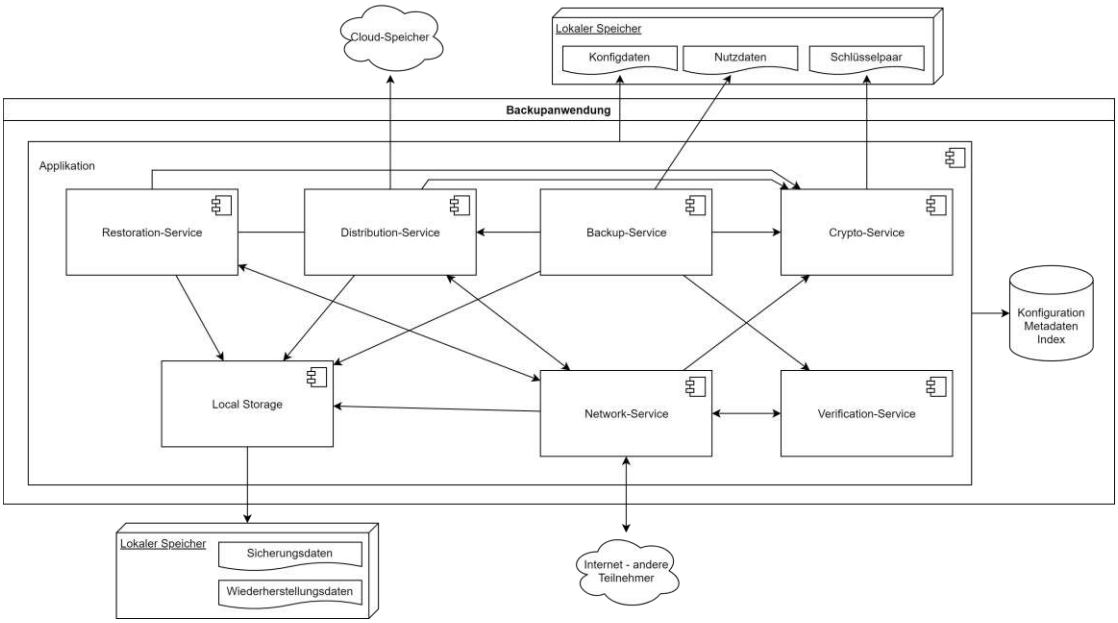

Abbildung 5.3: Komponenten der Lösung

des Betriebs des Cloud-Anbieters sein[1]. Ebenso können Änderungen an den Nutzungsbedingungen des Cloud-Anbieters, der Entwicklung zu einem kostenpflichtigen Dienst oder der Größe des nutzbaren Cloud-Speichers zu einer schlechteren Verfügbarkeit bis zur Unbenutzbarkeit des Cloud-Anbieters führen. Durch die Verwendung mehrere Cloud-Anbieter werden im Störungsfall Aufgaben vom zweiten Cloud-Anbieter übernommen, wodurch es zu keiner Unterbrechung bei der Datensicherung kommt. Die Verteilung auf mehrere Cloud-Anbieter führt zusätzlich dazu, dass der notwendige Speicherplatz für die temporären Sicherungsdaten reduziert wird, wodurch kostenlose Angebote genutzt werden können.

Vorteil dieser Cloud-Strategie ist, dass die initiale Datensicherung sofort erfolgen kann, unabhängig davon, ob bzw. wie viele andere Teilnehmer gerade verfügbar sind. Des Weiteren müssen dadurch Sicherungsdaten nur einmal vom Quellsystem hochgeladen werden, anstatt die Sicherungsdaten je sicherndem Teilnehmer separat zu übertragen. Dies führt zu einer Verkürzung der Upload-Zeit je Sicherungsblock, wodurch sich die gesamte Backupzeit verkürzt. Die Sicherungszeit wird somit wie in AF-13 Minimierung der Sicherungszeit definiert reduziert um das Risiko von Datenverlust zu minimieren.

5.1.3 Komponenten der Lösung

Die Backupanwendung ist in mehrere Komponenten eingeteilt, von denen jede ihren eigenen Aufgabenbereich hat. Die Komponenten sind in *Abbildung 5.3* dargestellt und

[1] `https://www.derstandard.at/story/2000134151597/rueckzug-von-amazon-microsoft-und-co-russland-geht-der-datenspeicher` - besucht am 24.04.2022

werden folgend erläutert:

- Konfiguration/Metadaten/Index: In der internen Datenbank werden Konfigurationsdaten, ein Index über alle Nutzdaten und Metadaten über den Zustand der Nutz- und Sicherungsdaten gespeichert. Pro Teilnehmer werden seine Adresse und sein Zertifikat gespeichert. Andere Konfigurationsdaten wie gesicherte Verzeichnisse oder angebundene Cloud-Provider werden beim ersten Start der Applikation aus dem lokalen Speicher geladen und unveränderbar in der Datenbank abgelegt. Für die Nutz- und Sicherungsdaten wird ein Index aller gesicherten Dateien mit den gesicherten Versionen sowie den Blöcken, aus denen jede Dateiversion besteht, geführt. Je Block sind die sichernden Teilnehmer und Metadaten zur Verifikation gespeichert. Im Fall einer Datenwiederherstellung werden Metadaten über den Fortschritt abgelegt. Details dazu finden sich in *Kapitel 5.1.4*.

- Crypto-Service: Das Crypto-Service ist die zentrale Stelle für die Verwaltung des Schlüsselpaares des Benutzers und der Zertifikate anderer Teilnehmer und dient auch der Generierung von kryptografischen Schlüsseln.

- Network-Service: Das Network-Service dient der Kommunikation mit anderen Teilnehmern. Informationen zur Authentifizierung aller Teilnehmer werden über das Crypto-Service bereitgestellt. Das Network-Service wird verwendet, um Nachrichten an andere Teilnehmer zu verschicken. Es ist auch für den Empfang der Nachrichten von anderen Teilnehmern und deren Weiterleitung an die korrekten Komponenten der Backupanwendung verantwortlich.

- Local-Storage: Der Local-Storage wird verwendet, um im lokalen Speicher Sicherungsdaten abzulegen. Dabei handelt es sich um Sicherungsdaten, die aus den eigenen Nutzdaten erzeugt und zu anderen Teilnehmern repliziert werden. Während einer laufenden Datenwiederherstellung werden die angeforderten Nutzdaten durch den Local-Storage im lokalen Speicher gesammelt.

  Der Local-Storage wird auch von Empfängern von Sicherungsdaten verwendet, um die Daten anderer Teilnehmer zu sichern. Während einer Datenwiederherstellung werden angeforderte Blöcke aus dem Local-Storage geholt und an das Quellsystem gesendet.

- Backup-Service: Das Backup-Service wird zur Konfiguration der zu sichernden Verzeichnisse und zur Durchführung der Sicherungen verwendet. Es ist verantwortlich für die Anlage und Administration von Metadaten für alle zu sichernden Dateien. Dateien werden während einer Sicherung aus dem lokalen Speicher gelesen und verarbeitet. Dateien, die gesichert werden müssen, werden in Blöcke eingeteilt, per Crypto-Service verschlüsselt und im Local-Storage abgelegt. Ihre Metadaten werden über das Backup-Service aktualisiert. Während der Durchführung des Backups wird zusätzlich kontrolliert, ob für einen Block genügend Metadaten zur Verifizierung verfügbar sind und im Anlassfall die Erzeugung dieser Metadaten über das

Verification-Service veranlasst. Das Backup-Service veranlasst auch die Verteilung der Sicherungsdaten über das Distribution-Service.

- Distribution-Service: Das Distribution-Service kümmert sich um den Upload der Sicherungsblöcke aus dem Local-Storage zu den konfigurierten Cloud-Speichern und verwendet das Network-Service, um andere Teilnehmer über neue zu sichernde Blöcke im Cloud-Speicher zu informieren. Erkennt das Distribution-Service das für einen Block zu wenige Replikate im Backupsystem vorhanden sind ist es auch dafür verantwortlich, diese von sichernden Teilnehmern anzufordern und wieder im Backupsystem zu verteilen.

  Als empfangender Teilnehmer wird das Distribution-Service vom Network-Service über einen zu sichernden Block informiert, ladet diesen vom Cloud-Speicher herunter und legt diesen im Local-Storage ab.

- Verification-Service: Das Verification-Service führt in regelmäßigen Abständen mithilfe von Metadaten eine Überprüfung der Replikate bei anderen Teilnehmern durch und speichert das Ergebnis der Überprüfung je Block und Teilnehmer. Sind für Sicherungsdaten zu wenige Verifikationsdaten verfügbar, werden die entsprechenden Sicherungsdaten von anderen Teilnehmern über das Network-Service angefordert, die Metadaten zur Verifizierung generiert und in den lokalen Metadaten abgelegt.

- Restoration-Service: Das Restoration-Service ist für die Wiederherstellung von Nutzdaten zuständig. Es startet und verwaltet laufende Wiederherstellungen. Über das Network-Service werden Sicherungsdaten von anderen Teilnehmern angefordert. Dieses leitet eingehende Sicherungsdaten an das Restoration-Service weiter, welche überprüft, per Crypto-Service entschlüsselt und im Local-Storage gespeichert werden. Wenn genügend Sicherungsdaten verfügbar sind werden diese zur Wiederherstellung der Nutzdaten verwendet.

5.1.4 Informations-Modell

Die Backupanwendung benötigt eine Reihe von Informationen zur korrekten Funktionsweise. Diese werden in der lokalen Datenbank gespeichert oder an andere Teilnehmer verteilt und werden folgend beschrieben.

5.1.4.1 Konfiguration von Teilnehmern

User: Die Daten anderer Teilnehmer, mit denen die lokale Backupanwendung das Backupsystem bildet, werden durch *User* abgebildet (*siehe Abbildung 5.4*). Jeder *User* hat eine eindeutige *id*, mit der er innerhalb der Backupanwendung referenziert wird. *certificate* beinhaltet das Zertifikat mit dem öffentlichen Schlüssel des Teilnehmers. Ohne Zertifikat erfolgt keine Authentifizierung des anderen Teilnehmers wodurch eine Kommunikation verhindert wird. Die Merkmale *allowBackupFromUser* bzw. *allowBackupToUser* geben an, ob von diesem Teilnehmer Sicherungsdaten akzeptiert werden bzw. ob der Teilnehmer

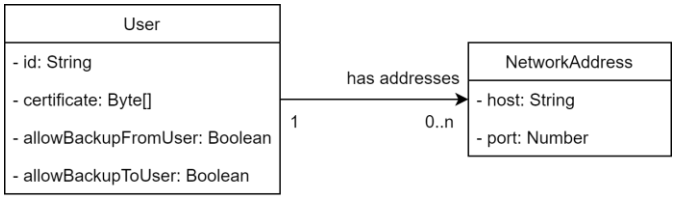

Abbildung 5.4: Datenmodell der Teilnehmer

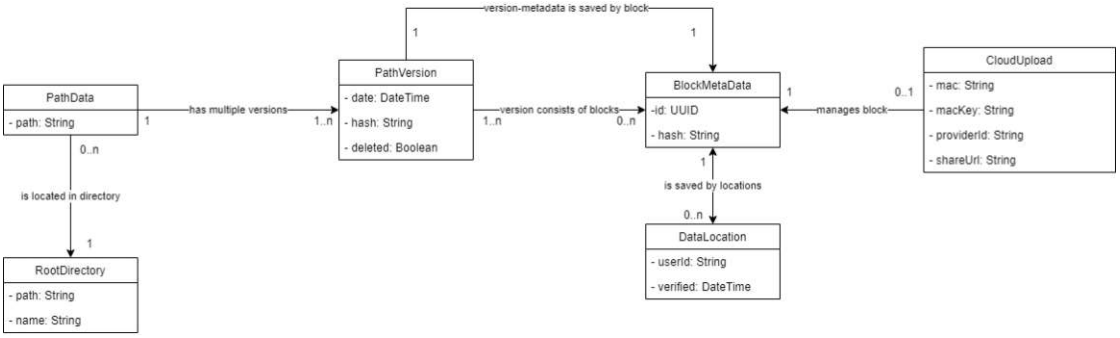

Abbildung 5.5: Datenmodell der Sicherungen

Ziel für eigene Sicherungsdaten ist. Jeder Teilnehmer verfügt des Weiteren über eine Liste von Netzwerkadressen über die er erreichbar ist.

5.1.4.2 Datenmodell zur Verwaltung von Sicherungen

In *Abbildung 5.5* wird das Datenmodell zur Verwaltung der Sicherungen gezeigt. Diese Daten werden im lokalen Index gespeichert.

RootDirectory: Der Anwender konfiguriert Verzeichnisse die von der Backupanwendung gesichert werden. Für jedes Verzeichnis wird ein Name angegeben um die Wiederherstellung des Verzeichnisses anhand des Namens zu erleichtern.

PathData: Jede Datei innerhalb des Verzeichnisses eines *RootDirectory* wird durch einen *PathData*-Eintrag repräsentiert. Unter *path* wird der Pfad relativ zum Stammverzeichnis inkl. dem Dateinamen der Datei gespeichert.

PathVersion: Für jede Datei (abgebildet durch einen *PathData*-Eintrag) können mehrere Dateiversionen (*PathVersion*) extistieren. *date* beinhaltet den Sicherungszeitpunkt der Dateiversion, unter *hash* wird der Hashwert der gesamten Datei zur Änderungserkennung gespeichert. Eine Dateiversion wird für neue Dateien angelegt oder wenn sich der Hash der Datei im Vergleich zur aktuellsten gespeicherten Dateiversion geändert hat. Der Marker *deleted* wird gesetzt sobald bei einer Sicherung erkannt wird das die Datei gelöscht wurde. Jede *PathVersion* besteht aus einer Reihe von Datenblöcken.

Diese Datenblöcke werden als *BlockMetaData*-Einträge abgebildet. Werden die Datenblöcke in der vorkommenden Reihenfolge der *PathVersion* aneinandergereiht kann die

Originaldatei wiederhergestellt werden.

BlockMetaData: Ein Datenblock wird durch einen *BlockMetaData*-Eintrag verwaltet. Dieser hat eine eindeutige *id* in Form einer UUID, mit welcher im Index auf den Datenblock referenziert werden kann. Die Verwendung einer künstlichen ID hat den Vorteil, dass sie keine Rückschlüsse auf den Inhalt des Blocks zulässt. Dadurch kann diese ID auch zur Referenzierung des Blocks bei anderen Teilnehmern verwendet werden. In dieser Arbeit wird diese ID auch als *Block-ID* bezeichnet. Bei der Speicherung eines Daten- oder Sicherungsblocks wird die Block-ID als Dateiname verwendet. Damit kann von einem Daten- oder Sicherungsblock immer auf den entsprechenden *BlockMetaData*-Eintrag geschlossen werden. Zusätzlich wird im *BlockMetaData*-Eintrag der Hashwert des Datenblocks zur Wiedererkennung gespeichert. Dadurch können gleiche Blöcke in unterschiedlichen Dateien bzw. -versionen erkannt, müssen nur einmal im Index abgelegt und im Backupsystem gesichert werden.

Zusätzlich zu den Datenblöcken müssen auch die Informationen einer *PathVersion* gesichert werden um eine Wiederherstellung der Nutzdaten bei Verlust des lokalen Index zu ermöglichen. Diese Metadaten werden in *Kapitel 5.1.4.4* erläutert. Zur Vereinheitlichung des Sicherungs-, Verifikations- und Wiederherstellungsvorganges werden diese in einem eigenen neuen Datenblock verpackt. Dieser wird ebenso als *BlockMetaData* verwaltet, verschlüsselt und als Sicherungsblöcke an andere Teilnehmer verteilt.

DataLocation: Pro *BlockMetaData* werden die sichernden Teilnehmer, die ein Replikat des Sicherungsblocks vorhalten, in einer *DataLocation* gespeichert. *userId* enthält die ID des *Users*, unter *verified* wird der Zeitstempel der letzten erfolgreichen Verifizierung dieser Sicherung abgelegt. Das Backupsystem sorgt nach MA-15 Mehrere Replikate der Sicherungsdaten dafür, dass je Datenblock mindestens drei verifizierte Replikate verfügbar sind. Ein Replikat wird alle 14 Tage überprüft und gilt als verifiziert, wenn dessen letzte Überprüfung maximal 21 Tage zurückliegt. Bei unverifizierten *DataLocations* versucht die Backupanwendung die Verifikation weiter und löscht den *DataLocation*-Eintrag erst nach 30 Tagen. Parallel wird der entsprechende Sicherungsblock von einem anderen Teilnehmer angefordert und neu verteilt, um mindestens drei verifizierte Replikate herzustellen (*siehe Kapitel 5.3.2.2*). Das Verifikationsintervall, die Dauer, bis ein Replikat als nicht verifiziert gilt und die Dauer bis zur Löschung sind konfigurierbar.

CloudUpload: Zur Verteilung der Sicherungsblöcke werden diese zu einem Cloud-Speicher hochgeladen. Die relevanten Metadaten für jeden Upload werden als *CloudUpload* gespeichert. *mac* und *macKey* enthalten den MAC des Sicherungsblocks, damit dessen Integrität vom empfangenden Teilnehmer geprüft werden kann. Das Merkmal *providerId* bezeichnet den Cloud-Anbieter, bei dem der Sicherungsblock hochgeladen wurde. Unter *shareUrl* befindet sich die öffentlich erreichbare URL zum Download des Sicherungsblocks.

5.1.4.3 Verifikationsdaten

Abbildung 5.6 zeigt die Daten die zur Verifikation der Replikate notwendig sind. Verifikationsdaten werden in der lokalen Datenbank gespeichert.

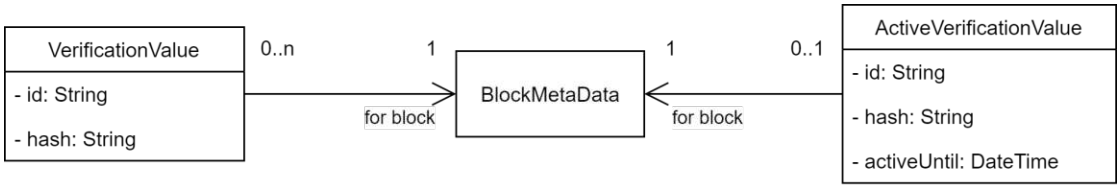

Abbildung 5.6: Datenmodell zur Verifikation von Sicherungsdaten

VerificationValue: *VerificationValues* werden zur Verifizierung von Replikaten verwendet. *VerificationValues* referenzieren den *BlockMetaData*-Eintrag des ursprünglichen Datenblocks und beinhalten eine *id*. Diese wird dem Sicherungsblock vorangestellt und aus der Kombination der Hashwert *hash* berechnet. Verifikationsdaten werden während der Datensicherung erzeugt. Wie in *Kapitel 5.1.4.2* beschrieben werden Replikate alle 14 Tage überprüft und dabei die Verifizierungsdaten gewechselt. Pro Block werden maximal 12 *VerificationValues* gespeichert. Damit kann jeder Sicherungsblock für sechs Monate verifiziert werden. Die Generierung während einer Sicherung erfolgt nur, wenn sechs oder weniger *VerificationValues* für den *BlockMetaData*-Eintrag vorhanden sind. Dies hat den Grund, dass bei einer Sicherung nicht jeder Datenblock, dem nur eine *VerificationValue* fehlt, vollständig verschlüsselt und verarbeitet werden soll. Wenn nach ca. drei Monaten nur mehr sechs oder weniger *VerificationValues* verfügbar sind und der Datenblock bei einer Sicherung verarbeitet wird werden neue Werte generiert. Es kann der Fall eintreten, dass für einen Datenblock keine *VerificationValues* mehr existieren. In diesem Fall wird der Sicherungsblock von einem anderen Teilnehmer angefordert um neue *VerificationValues* zu generieren.

ActiveVerificationValue: Je *BlockMetaData* gibt es maximal einen aktiven Verifikationsdaten-Eintrag. Dessen Verifikationsdaten werden bis auf Widerruf bei Verifizierungsanfragen verwendet. In einer *ActiveVerificationValue* ist in *activeUntil* gespeichert, wie lange diese Verifikationsdaten verwendet werden. Nach Ablauf des Zeitpunkts wird der Eintrag gelöscht und durch einen neuen *VerificationValue* mit neuer begrenzter Gültigkeit ersetzt.

5.1.4.4 Datenmodell der gesicherten Metadaten

Neben den eigentlichen Sicherungsdaten müssen auch Metadaten aus dem lokalen Index bei anderen Teilnehmern gesichert werden. Diese sind notwendig, damit bei Verlust der lokalen Datenbank die Metadaten zur Verwaltung der Sicherungen wiederhergestellt werden können. Die gesicherten Metadaten sind in *Abbildung 5.7* dargestellt.

PathDataVersion: Ein *PathDataVersion*-Block beinhaltet Teile der Informationen aus *PathData* und *PathVersion* und repräsentiert genau eine Version einer gesicherten Datei zu einem bestimmten Zeitpunkt. Das Merkmal *rootDirectoryId* beinhaltet die ID des *RootDirectory*, *path* den Dateipfad innerhalb des Ordners. In *date*, *hash* und *deleted* befinden sich die Werte der gesicherten *PathVersion*. Unter *blockIds* sind die IDs aller Datenblöcke, aus denen die Datei besteht, in der richtigen Reihenfolge enthalten.

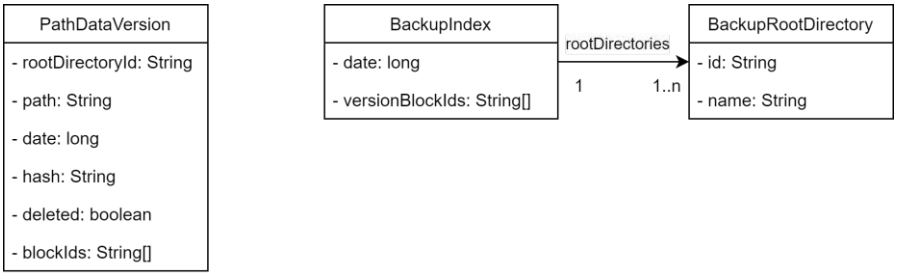

Abbildung 5.7: Datenmodell der gesicherten Metadaten

Wenn alle Datenblöcke mit der entsprechenden ID in der vorkommenden Reihenfolge aneinandergereiht werden kann die Originaldatei wiederhergestellt werden.

Ein *PathDataVersion*-Block wird wie ein normaler Datenblock behandelt. Er erhält eine ID, wird verschlüsselt und als Sicherungsblock verteilt. Andere Teilnehmer kennen den Inhalt dieser Blöcke nicht.

Aus einer *PathDataVersion* kann nur genau eine Datei wiederhergestellt werden. Die *PathDataVersion*-Blöcke werden über alle Teilnehmer verteilt und haben bei den Teilnehmern keine Ordnung. Daher können bei Verlust des lokalen Index nicht nicht ohne weiteres gezielt alle *PathDataVersions* einer bestimmten Sicherung ermittelt werden.

BackupIndex: Bei jedem Sicherungslauf wird ein *BackupIndex*-Block erzeugt. Ein *BackupIndex* besteht aus dem Zeitpunkt der Sicherung (*date*), den Block-IDs aller *PathDataVersion*-Blöcke dieser Sicherung sowie der Liste von konfigurierten *RootDirectories*. Ein *BackupIndex*-Block erhält eine Block-ID in Form von `IDX_TIMESTAMP`, wobei `TIMESTAMP` dem Zeitpunkt der Sicherung entspricht. Dies ermöglicht die Unterscheidung von anderen Sicherungsblöcken und die Ermittlung des aktuellsten *BackupIndex* bei einem sichernden Teilnehmer. Anhand des *BackupIndex* können für einen bestimmten Backupzeitpunkt die relevanten *PathDataVersion*-Blöcke identifiziert, diese angefordert und so der lokale Index zur Wiederherstellung einer spezifischen Sicherung hergestellt werden. Ein *BackupIndex*-Block wird ebenfalls wie ein normaler Datenblock behandelt, verschlüsselt und als Sicherungsblock verteilt.

5.1.4.5 Datenmodell zur Verwaltung von Wiederherstellungen

Im Fall der Wiederherstellung von Nutzdaten müssen von anderen Teilnehmern Sicherungsdaten angefordert werden. Die entschlüsselten Datenblöcke werden danach pro Datei aneinandergereiht um die Originaldatei wiederherzustellen. Die Anforderung aller Sicherungsdaten hängt von der Verfügbarkeit anderer Teilnehmer ab und ist ein länger laufender Vorgang, daher muss der Status der Wiederherstellung in der Backupanwendung mitgeführt werden. Das notwendige Datenmodell ist in *Abbildung 5.8* abgebildet.

RestoreBlockData: Zur Wiederherstellung eines Datenblocks wird ein *RestoreBlockData*-Eintrag angelegt. Dieser repräsentiert einen Datenblock, der noch von anderen

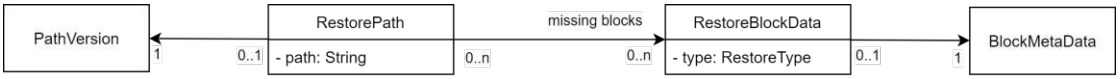

Abbildung 5.8: Datenmodell zur Verwaltung von Wiederherstellungen

Teilnehmern angefordert werden muss. Zusätzlich wird im Merkmal *type* die Priorität der Anforderung angegeben: Sicherungsblöcke zur Wiederherstellung von Nutzdaten werden mit hoher Priorität angefordert. Im Fall des Verlusts des lokalen Index müssen auch Metadaten zu historischen Dateiversionen wiederhergestellt werden. Diese Sicherungsblöcke werden mit normaler Priorität angefordert und nachrangig behandelt. Wenn ein Datenblock einlangt wird der entsprechende *RestoreBlockData*-Eintrag gelöscht.

RestorePath: Die Wiederherstellung von Dateien wird über *RestorePath*-Einträge verwaltet. Ein *RestorePath* zeigt auf die wiederherzustellende *PathVersion* und beinhaltet in *path* den absoluten Pfad, an dem die Datei wiederhergestellt werden soll. Zusätzlich werden pro *RestorePath* alle fehlenden Blöcke bzw. deren *RestoreBlockData*-Einträge, die zur Wiederherstellung der Datei notwendig sind, referenziert. Zu Beginn einer Wiederherstellung werden von einem *RestorePath* alle Blöcke, aus denen die Datei besteht, referenziert. Mit fortschreitender Anforderung von Sicherungsblöcken werden diese Einträge gelöscht. Wenn von einem *RestorePath* keine *RestoreBlockData*-Einträge referenziert werden sind alle Datenblöcke zur Wiederherstellung der Datei lokal verfügbar.

5.1.5 Systemanforderungen und Umfeld

Die Backupanwendung soll auf möglichst vielen Systemen und auf unterschiedlichen Rechnerarchitekturen einsetzbar sein. Der Einsatz ist auf üblichen Heimcomputern, Laptops und Servern geplant, die nach [74, 76] zu einem Großteil mit Windows, MacOS oder Linux betrieben werden. Daher werden als Ziel-Betriebssysteme, unter denen die Backupanwendung laufen soll, Windows, MacOS und Linux definiert. Der Einsatz auf mobilen Endgeräten ist kein Ziel dieses Prototypen.

Der Betrieb der Backupanwendung setzt das Vorhandensein eines Massenspeichers zur Datenspeicherung voraus. Auf diesem werden die konfigurierten Teilnehmer und Metadaten abgelegt. Des Weiteren werden verschlüsselte Sicherungsblöcke vor dem Versand an die Cloud oder während der Datenwiederherstellung auf dem Massenspeicher zwischengespeichert. Aufgrund dieser Zwischenspeicherung muss am Massenspeicher freier Speicherplatz in der Größe der zu sichernden Daten verfügbar sein.

Wie in *Kapitel 3.2* beschrieben muss das Backupsystem eines Teilnehmers aus mindestens drei zusätzlichen Teilnehmern bestehen, damit pro Datenblock drei Replikate hergestellt werden können. Empfohlen wird die Konfiguration von mindestens vier Teilnehmern, um beispielsweise den Ausfall eines Teilnehmers zu kompensieren.

5.2 Technische Entscheidungen für die Proof-of-Concept-Implementierung

Im folgenden Kapitel werden technologische Entscheidungen für die Implementierung der Backuplösung besprochen.

5.2.1 Beschreibung des verwendeten Cloud-Anbieters

Für die Funktionsweise der Backuplösung ist wie in *Kapitel 5.1.2.2* beschrieben die Nutzung eines Cloud-Anbieters mit Cloud-Speicher mit bestimmten Minimalfunktionen notwendig. Es werden die Cloud-Speicher Google Drive[2] sowie eine selbst gehostete Nextcloud[3]-Instanz genutzt. Jedem Cloud-Speicher wird intern eine Provider-ID zugeordnet, um diese innerhalb der Backupanwendung zu unterscheiden. Diese wird beispielsweise für den Upload von Sicherungsdaten zum Cloud-Speicher benötigt.

5.2.1.1 Google Drive

Bei Google Drive handelt es sich um einen Cloud-Speicher von Google, in dem beliebige Dateien abgelegt werden können. Das Google Drive-API [29] bietet die in *Kapitel 5.1.2.2* geforderten Methoden für Dateioperationen und Dateifreigaben an. Google Drive verwendet die global verteilte Infrastruktur von Google Cloud. Privatpersonen stehen 15 GB Speicherplatz kostenlos für Daten zur Nutzung innerhalb von Google Drive, Gmail, Fotos etc. zur Verfügung [30]. Da die bei Google Cloud eingesetzte Infrastruktur, die Schutzmaßnahmen für Datenschutz und Privatsphäre, die eingesetzten kryptographischen Verfahren und andere Sicherheitsmaßnahmen nicht überprüft werden können, muss diesbezüglich den Angaben von Google vertraut werden. Dies wird dadurch gestützt, dass Google Cloud unabhängige Verifizierungen ihrer Maßnahmen durchführen lässt und beispielsweise nach ISO/IEC 27001 zertifiziert ist [31]. Damit wird der Maßnahme MA-24 Sorgfältige Auswahl des Cloud-Anbieters entsprochen.

Dem Cloud-Speicher Google Drive wird die Provider-ID *at.lucny.p2pbackup.cloud.googledrive.service.GoogleDriveStorageServiceImpl* zugeordnet.

5.2.1.2 Nextcloud Files

Bei Nextcloud Files handelt es sich um einen Cloud-Speicher, der zur Ablage, Synchronisierung und Freigabe von Dateien verwendet wird. Der Quellcode[4] von Nextcloud ist frei verfügbar und kann eingesehen werden. Der Quellcode wurde bereits von externen Stellen wie Veracode oder der NCC Group auf Schwachstellen überprüft [54]. Nextcloud kann als gehostete Enterprise-Version bei einer Reihe von Infrastruktur-Anbietern betrieben werden. Eine Nextcloud-Instanz kann auch selbst, beispielsweise auf einem privaten Server oder im eigenen Unternehmen, betrieben werden, sofern dieser Server

[2] `https://www.google.at/drive/about.html` - besucht am 24.04.2022
[3] `https://nextcloud.com/` - besucht am 24.04.2022
[4] `https://github.com/nextcloud` - besucht am 24.04.2022

den Systemanforderungen genügt. Dadurch lässt sich eine Nextcloud-Instanz auf einer beliebigen Infrastruktur betreiben, die nach Maßnahme MA-24 Sorgfältige Auswahl des Cloud-Anbieters entsprechend geprüft werden kann.

Es wird eine Nextcloud-Instanz in Version 20.0.14 auf einem privaten Server des deutschen Anbieters all-inkl.com[5] verwendet. Es stehen für Nextcloud Files 50 GB Speicherplatz zur Verfügung. Das REST-API [52, 53] von Nextcloud bietet die in *Kapitel 5.1.2.2* geforderten Schnittstellen an. Dem Cloud-Speicher Nextcloud Files wird die Provider-ID *at.lucny.p2pbackup.cloud.nextcloud.service.NextcloudStorageServiceImpl* zugeordnet.

5.2.2 Technologien

Für die Implementierung der Backuplösung sind ausgehend von den Anforderungen und dem Umfeld einige Entscheidungen bezüglich der zu nutzenden Technologien erforderlich. Folgend werden die genutzten Technologien vorgestellt und deren Auswahl begründet.

5.2.2.1 Java

Die Implementierung des Prototyps erfolgt in der Programmiersprache Java[6]. Die Laufzeitumgebung von Java ist auf den in *Kapitel 5.1.5* definierten Zielplattformen der Backupanwendung verfügbar. Für Java besteht eine große Community, die eine Reihe von Problemen, beispielsweise die Implementierung von kryptographischen Verfahren, bereits gelöst hat. Es kann auf eine Vielzahl an verfügbaren Programmbibliotheken zurückgegriffen werden. Java befindet sich unter den Top 3 der weltweit populärsten Programmiersprachen [14, 82]. Damit ist von einer langfristigen Unterstützung neuer Rechnerarchitekturen und einer Weiterentwicklung der Sprache auszugehen. Java wird in Version 17.0.1[7] verwendet.

5.2.2.2 HyperSQL-DB

Zur Speicherung der lokalen Metadaten wird eine Embedded HyperSQL-Datenbank[8] verwendet. Durch den Einsatz einer Datenbank können Metadaten der Anwendung strukturiert abgelegt und per SQL durchsucht werden. Die HyperSQL-Datenbank ist vollständig in Java implementiert, wird als zusätzliche Bibliothek mit der Backupanwendung paketiert und beim Start der Anwendung automatisch gestartet. Der Betrieb einer separat installierten Datenbank ist daher nicht erforderlich. Die Daten der Datenbank liegen persistent im Dateisystem vor. Es wird eine HyperSQL-Datenbank verwendet, da diese die verschlüsselte Ablage der persistenten Datenbank-Daten unterstützt [84] (siehe MA-7 Verschlüsselung der Konfigurations- und Metadaten).

[5] `https://all-inkl.com/` - besucht am 24.04.2022
[6] `https://java.com/de/` - besucht am 25.04.2022
[7] `https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html` - besucht am 24.04.2022
[8] `http://hsqldb.org/` - besucht am 24.04.2022

5.2.3 Kryptographische Verfahren

Folgend werden je Einsatzgebiet die verwendeten kryptografischen Verfahren inkl. der notwendigen Parameter wie Schlüssellängen oder Betriebsmodi erläutert. Die Wahl der Verfahren und Parameter basiert auf aktuellen Empfehlungen, um die Schutzziele der Backuplösung nach aktuellem Stand der Technik bestmöglich zu erfüllen (siehe MA-6 Verwendung von als derzeit sicher geltenden kryptografischen Verfahren) und deren Verfügbarkeit in den eingesetzten Technologien.

Die technische Richtlinie des BSI mit Empfehlungen für Algorithmen und Schlüssellängen gilt nur für den Zeitraum bis Ende 2027. Eine darüber hinausgehende Empfehlung wird nicht abgegeben, da für den Zeitraum danach keine Aussage über technologische Entwicklungen, Fortschritte in Bezug auf die Leistungsfähigkeit von Computern oder neue Angriffe auf kryptografische Verfahren abgegeben werden kann. Für Daten, deren Vertraulichkeit längerfristig gewährleistet werden muss, wird empfohlen von Beginn an möglichst starke Verfahren aus der Empfehlung zu wählen [11].

Als weitere Quelle dient die Empfehlung der SOG-IS über zu verwendende Verfahren und Schlüssellängen [69]. Teilnehmer der SOG-IS sind unter anderem nationale Behörden wie das BSI oder die ANSSI.

5.2.3.1 Sicherstellung der Integrität der Applikationsdateien

Die Backupanwendung steht sowohl als Source-Code als auch in Form von kompilierten Applikationsdateien zur Verfügung. Bei Nutzung der Applikationsdateien muss deren Integrität überprüfbar sein. Daher wird zusätzlich zum Source-Code auch ein Hash-Wert der Applikationsdateien zur Verfügung gestellt. Dieser muss mit Betriebssystemmitteln der definierten Ziel-Betriebssysteme (*siehe Kapitel 5.1.5*) verifizierbar sein.

Auf Empfehlung des BSI [11] werden SHA-512-Hashes verwendet. Die Berechnung eines SHA-512-Hashes ist mit Betriebssystemmitteln sowohl unter Windows mit `Get-FileHash`[9], unter MacOs mittels `shasum` als auch mit Linux mittels `sha512sum`[10] möglich. Dies ermöglicht die Überprüfung der Applikationsdateien vor Verwendung der Backupanwendung und entspricht MA-5 Verifikation der Applikationsdateien per Hash-Wert .

5.2.3.2 Generierung von Zufallszahlen

Für die meisten kryptografischen Verfahren werden Zufallszahlen benötigt. Diese werden beispielsweise bei der Schlüsselerzeugung oder als IV bei symmetrischen Verschlüsselungsverfahren verwendet. Die Generierung der Zufallszahlen soll laut [69] mit echten Zufallszahlen erfolgen. Da die Qualität von echten Zufallszahlen auf Systemen oft nur

[9] `https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-filehash` - besucht am 24.04.2022

[10] `http://manpages.ubuntu.com/manpages/xenial/de/man1/sha512sum.1.html` - besucht am 24.04.2022

schwer zu bewerten ist wird empfohlen diese nur als Seed für einen der deterministischen Zufallszahlen-Generatoren *HMAC-DRBG*, *Hash-DRBG* oder *CTR-DRGB* zu verwenden.

Unter Java hängt der standardmäßig verwendete Zufallszahlengenerator von der Java-Version, dem Betriebssystem und den Sicherheitseinstellungen in der Datei *java.security* der Java-Installation ab [56]. Daher kann nicht garantiert werden welcher Zufallszahlengenerator von der Backupanwendung eingesetzt wird. Aufgrund dessen wird bei der Erstellung von Zufallszahlen der Empfehlung von [69] gefolgt. Unter Java werden die Zufallszahlen-Generators des Betriebssystems (Windows `Windows-PRNG`, Linux `NativePRNG`) eingesetzt um einen Seed zu erzeugen. Dieser Seed wird zur Initialisierung des `DRGB`-Zufallszahlen-Generators verwendet. Dieser nutzt intern einen *Hash-DRBG* mit SHA-256 [55]. Die Erzeugung von als derzeit sicher geltenden Zufallszahlen ist notwendig, um die Maßnahmen MA-6 Verwendung von als derzeit sicher geltenden kryptografischen Verfahren und MA-11 Geeignete Erzeugung von Schlüsseln in der Backuplösung zu erfüllen.

5.2.3.3 Authentifikation eines Teilnehmers

Ein Teilnehmer muss sich gegenüber den anderen Teilnehmern des Backupsystems authentisieren. Dafür muss jeder Teilnehmer eine Möglichkeit zur Identifikation besitzen. Ein Teilnehmer wird über ein Zertifikat und den zugehörenden Private-Key identifiziert (*siehe Kapitel 2.1.5.2*). Damit andere Teilnehmer die Authentifizierung durchführen können, wird das Zertifikat an alle Teilnehmer, mit denen kommuniziert werden soll, verteilt. Nach Empfehlung des BSI [11] und des IAD der NSA [34–36] werden für den zugrunde liegenden Public- und Private-Key RSA-Schlüsselpaare verwendet. Es wird eine Schlüssellänge von 3072 Bit für einen langfristigen Einsatz empfohlen. Das Schlüsselpaar sowie das Zertifikat mit dem öffentlichen Schlüssel werden beim ersten Start der Backupanwendung selbstständig erzeugt, um die geeigneten kryptografischen Parameter zu gewährleisten (siehe MA-11 Geeignete Erzeugung von Schlüsseln in der Backuplösung).

5.2.3.4 Gesicherte Ablage des Schlüsselpaars des Anwenders

Das Schlüsselpaar eines Teilnehmers (*siehe Kapitel 5.2.3.3*) wird von der Backupanwendung gesichert abgelegt, um es vor Ausspähung oder Manipulation zu schützen. Es ist mit einem Passwort abgesichert, dass nur dem Anwender bekannt ist.

Zur Speicherung des generierten Schlüsselpaares wird ein PKCS #12 Keystore verwendet. Nach Empfehlung von [69] wird als Schlüsselableitungsfunktion PBKDF2 mit HMAC-SHA-512 als Hashfunktion mit einem 128 Bit Salt eingesetzt, um aus dem Passwort des Anwenders ein Geheimnis zur Verschlüsselung des Keystores zu erzeugen. Die Verschlüsselung erfolgt mit AES-256[11] im CBC-Betriebsmodus mit PKCS5-Padding. Dieses Verfahren wird in Java unter dem Namen `PBEWithHmacSHA512AndAES_256` unterstützt [57].

5.2.3.5 Überprüfung des Zertifikats eines anderen Teilnehmers

Zur Kommunikation mit einem anderen Teilnehmer müssen dessen Verbindungsdaten wie die Internet Protocol (IP)-Adresse und dessen Zertifikat der eigenen Backupanwendung bekannt sein. Diese Daten werden vom Teilnehmer an den Anwender übermittelt. Im Rahmen des Imports dieser Daten werden dem Anwender der Hash des Zertifikats in Form eines SHA3-512-Hashes als Hexwert angezeigt. Der Hash des Zertifikats wird dem Anwender über einen anderen zusätzlichen Kommunikationskanal übermittelt. Die Übereinstimmung beider Hashes muss bestätigt werden um den Teilnehmer der Backupanwendung hinzuzufügen. Mit diesem Verfahren wird MA-13 Bestätigung des öffentlichen Schlüssels anderer Teilnehmer entsprochen.

5.2.3.6 Einsatz von TLS

Um die Vertraulichkeit der Kommunikation zwischen zwei Teilnehmern und deren Authentizität zu gewährleisten wird TLS eingesetzt (*siehe Kapitel 2.1.5.5*). TLS bedient sich dazu der Schlüsselpaare der Teilnehmer aus *Kapitel 5.2.3.3*. Die Authentifizierung erfolgt auf Netzwerkschicht über das TLS-Protokoll. Beide Teilnehmer müssen sich gegenseitig authentifizieren, um eine Kommunikation zu ermöglichen. Die darauffolgende Datenkommunikation wird per TLS verschlüsselt (siehe MA-9 Verschlüsselung der Kommunikation mit anderen Teilnehmern).

Nach Empfehlung des BSI [12] wird TLS in Version 1.3 verwendet. Zur Übertragung wird nur die Verwendung der empfohlenen Cipher-Suiten *TLS_AES_128_GCM_SHA256*, *TLS_AES_256_GCM_SHA384* und *TLS_AES_128_CCM_SHA256* erlaubt. Die Schlüssellänge eines RSA-Schlüsselpaars mit einer Schlüssellänge von 3072 Bit entspricht bereits der empfohlenen Mindestlänge beim Einsatz von TLS.

5.2.3.7 Schlüsselableitung aus Master-Schlüssel

Zur Verschlüsselung von Daten wird ein Schlüssel benötigt. Dieser soll aus Effizienzgründen für die Verwendung von symmetrischer Verschlüsselung geeignet sein. Aus dem Private-Key des Teilnehmers wird per Schlüsselableitungsfunktion ein Master-Schlüssel mit 256 Bit erzeugt. Nach Empfehlung von [69] wird die Schlüsselableitungsfunktion PBKDF2 mit HMAC-SHA-512 als Hashfunktion eingesetzt. In Java ist diese unter `PBKDF2WithHmacSHA512` verfügbar. Die Länge des Salts beträgt 128 Bit.

Der Master-Schlüssel wird beim Anwendungsstart berechnet und wird ausschließlich dafür eingesetzt, um daraus weitere Schlüssel mithilfe der gleichen Ableitungsfunktion abzuleiten. Dieses Verfahren hat den Vorteil, dass das Passwort des Teilnehmers nur zum Zugriff auf dessen Private-Key verwendet wird. Der Private-Key wird in Folge wieder nur kurzfristig verwendet, da aus diesem der Master-Schlüssel erzeugt wird und alle weiteren Ableitungen von diesem ausgehen.

Zur Erzeugung weiterer Schlüssel wird je Verwendungszweck die Ableitungsfunktion mit einem fixen Salt aufgerufen. Die erzeugte Schlüssellänge hängt vom Verwendungszweck

ab. Pro Verwendungszweck bleibt der genutzte Schlüssel gleich, d.h. es wird beispielsweise zur Verschlüsselung eines Datenblocks immer der gleiche Schlüssel verwendet. Das beschriebene Verfahren gewährleistet, dass alle kryptografischen Schlüssel in der Backupanwendung nach aktuellen Standards erzeugt werden (siehe MA-11 Geeignete Erzeugung von Schlüsseln in der Backuplösung).

5.2.3.8 Verschlüsselung der lokalen Metadaten

Die Konfigurationsdaten über andere Teilnehmer und ihre Zertifikate sowie der lokale Index werden in Form einer Datenbank abgelegt. Diese wird zur Gewährleistung der Vertraulichkeit verschlüsselt gespeichert (siehe MA-7 Verschlüsselung der Konfigurations- und Metadaten). Es wird das symmetrische Verschlüsselungsverfahren AES im Modus CBC mit dem Padding-Verfahren PKCS5 verwendet (*siehe Kapitel 2.1.5.1*). Die Schlüssellänge beträgt 256 Bit. Der Schlüssel wird aus dem Master-Schlüssel mit dem Salt *databaseKey* abgeleitet. Für den IV wird die Länge von 128 Bit gewählt. Dieses Verschlüsselungsverfahren und der Modus wurden gewählt, da sie den Kriterien des BSI entsprechen und von der HyperSQL-DB unterstützt werden (*siehe Kapitel 5.2.2.2*).

Die Empfehlung des BSI sieht vor, dass sich bei der Nutzung von Blockverschlüsselung im CBC-Modus der IV während der gesamten Nutzungszeit nicht wiederholen darf. Die Verschlüsselung der HyperSQL-DB erlaubt jedoch nur die Festlegung eines IV bei der Erstanlage der Datenbank. Bei weiteren Nutzungen und Persistierung der Datenbank-Daten wird immer der gleiche IV verwendet womit der Empfehlung nicht entsprochen wird. MA-7 Verschlüsselung der Konfigurations- und Metadaten kann daher nicht vollständig umgesetzt werden.

5.2.3.9 Integritätssicherung der lokalen Metadaten

Die HyperSQL-DB unterstützt zur Verschlüsselung keinen kombinierten Betriebsmodus wie GCM, mit dem neben der Vertraulichkeit auch die Integrität und Authentizität der Daten sichergestellt werden könnte. Daher wird die verschlüsselte Datenbank zusätzlich mit einem MAC (*siehe Kapitel 2.1.5.4*) versehen um die Integrität der Datenbank zu gewährleisten (siehe MA-8 Integritätsprüfung der Konfigurations- und Metadaten). Der Schlüssel zur Generierung des MAC wird ebenfalls aus dem Master-Schlüssel abgeleitet. Da dieser nur dem Betreiber der aktuellen Backupanwendung bekannt ist, kann der MAC nur von diesem erzeugt werden. Beim Beenden der Backupanwendung wird der MAC mithilfe des Schlüssels und der persistenten Daten der lokalen Datenbank erzeugt. Beim Start der Backupanwendung wird der MAC erneut generiert und mit dem hinterlegten Wert verglichen. Dadurch kann die Integrität der lokalen Datenbank überprüft und sichergestellt werden, dass Änderungen nur durch den Benutzer der Backupanwendung erfolgen. Nach Empfehlung des BSI [11] wird ein HMAC eingesetzt. Es wird ein HMAC mit SHA-512 verwendet. Die Länge des für HMAC notwendigen Schlüssels beträgt 128 Bit. Dieser wird aus dem Master-Schlüssel mit dem Salt *databaseMAC* abgeleitet.

5.2.3.10 Verschlüsselung und Integritätssicherung der Sicherungsdaten

Die Sicherungsdaten werden auf den Fremdsystemen anderer Teilnehmer abgelegt. Zur Wahrung ihrer Vertraulichkeit und Integrität müssen die Sicherungsdaten daher verschlüsselt (siehe MA-17 Verschlüsselung der Sicherungsdaten mit derzeit als sicher geltenden symmetrischem Verfahren) und mit einer Möglichkeit zur Integritätsprüfung (siehe MA-18 Integritätsprüfung der Sicherungsdaten) bei der Datenwiederherstellung versehen werden. Die Verschlüsselung erfolgt aus Effizienzgründen symmetrisch. Der Schlüssel zur Verschlüsselung wird aus dem Master-Schlüssel mit dem Salt *blockEncryption* abgeleitet. Jeder Sicherungsblock wird mit dem gleichen abgeleiteten Schlüssel verschlüsselt.

Für diese kombinierte Anforderung aus Verschlüsselung und Integritätssicherung wird nach Empfehlung des BSI [11] AES im AEAD-Betriebsmodus GCM eingesetzt (*siehe Kapitel 2.1.5.1*). Die Schlüssellänge beträgt 256 Bit. Dieser Betriebsmodus ermöglicht während der Verschlüsselung gleichzeitig die Berechnung eines Tags, mit dem die Authentizität der Daten geprüft werden kann. Die Block-ID eines Sicherungsblocks wird als AD in den Verschlüsselungsvorgang miteinbezogen. Damit ist die Integritätssicherung der Block-ID eines Sicherungsblocks möglich. Für den IV wird nach Empfehlung eine Länge von 96 Bit gewählt. Dieser wird für jede Verschlüsselung neu zufällig generiert und den verschlüsselten Daten vorangestellt. Das BSI empfiehlt eine Tag-Länge von >= 96 Bit, im SOG-IS-Kryptokatalog [69] werden mindestens 128 Bit genannt. Daher wird für den Authentisierungs-Tag die Länge von 128 Bit gewählt.

5.2.3.11 Integritätsprüfung der Sicherungsdaten durch sichernden Teilnehmer

Die Sicherungsdaten werden von einem Teilnehmer vom Cloud-Speicher heruntergeladen und lokal abgelegt. Dem sichernden Teilnehmer ist der Inhalt des Sicherungsblocks nicht bekannt, da dieser von der Datenquelle verschlüsselt wurde. Es muss jedoch die Integrität des Sicherungsblocks überprüft werden um auszuschließen, dass dieser im Cloud-Speicher oder am Transportweg manipuliert wurde (siehe MA-20 Integritätsprüfung der Cloud-Sicherungsdaten). Für die Sicherungsdaten wird daher von der Backupanwendung der Datenquelle ein MAC (*siehe Kapitel 2.1.5.4*) erzeugt. Anhand dessen kann der sichernde Teilnehmer die Integrität des Sicherungsblocks prüfen und sicherstellen, dass der Sicherungsblock von der Datenquelle stammt. Der Schlüssel zur Generierung des MAC wird für jeden Sicherungsblock zufällig generiert.

Wie in *Kapitel 5.2.3.10* wird nach Empfehlung des BSI [11] ein HMAC mit SHA-512 verwendet. Die Länge des erzeugten Schlüssels beträgt 128 Bit.

5.2.3.12 Verifikation der Sicherungsdaten bei anderen Teilnehmern

Sicherungsdaten bei anderen Teilnehmern müssen in periodischen Abständen geprüft werden, um ihre Verfügbarkeit und Integrität zu gewährleisten. Die Sicherungsdaten sind verschlüsselt, daher muss die Überprüfung auf Basis der verschlüsselten Daten erfolgen. Die Verifikation erfolgt anhand eines Challenge-Response-Verfahrens. Die Datenquelle

schickt mit der Prüfungsanfrage für Sicherungsdaten eine Challenge an einen sichernden Teilnehmer. Dieser verknüpft die Challenge mit den lokal gespeicherten Sicherungsdaten, berechnet davon einen Hashwert und schickt diesen an die Datenquelle zurück. Diese kann den Hashwert mit einem vorausberechneten Wert vergleichen und so überprüfen, ob der Teilnehmer die Sicherungsdaten noch vorhält (siehe MA-16 Regelmäßige Verifikation der Sicherungsdaten bei anderen Teilnehmern).

Eine Challenge ist eine Zufallszahl mit 128 Bit. Damit ist die Menge der möglichen Werte groß genug um nach [69] eine Vorausberechnung und Speicherung aller möglichen Challenge-Hashwert-Kombinationen für einen Sicherungsblock zu verhindern. Als Hashverfahren kommt nach BSI-Empfehlung [11] SHA3-512 zum Einsatz.

5.2.4 Zusätzliche Maßnahmen

Folgend werden die Maßnahmen beschrieben, die während der Entwicklung zur Wahrung der Schutzziele eingesetzt werden.

5.2.4.1 Prüfung der verwendeten Bibliotheken mittels OWASP Dependency-Check

Zur Implementierung der Backuplösung werden verfügbare Bibliotheken aus anderen Quellen wie beispielsweise das Spring-Framework[11] oder Netty[12] verwendet. Diese können Fehler oder sicherheitsrelevante Schwachstellen enthalten. Zur Absicherung gegenüber diesen Sicherheitsbedrohungen werden während der Implementierung alle genutzten Bibliotheken mit dem Open Web Application Security Project (OWASP) Dependency-Check untersucht [58]. Dieser Check ermittelt, ob es für eine der Bibliotheken einen Eintrag, Common Platform Enumeration (CPE) genannt, in der National Vulnerability Database (NVD) des NIST gibt. Bei einer falschen Zuordnung von Bibliothek zu CPE kann dieser Eintrag lokal ausgenommen werden. Wenn für eine Bibliothek ein CPE existiert werden für diese alle bekannten Schwachstellen, Common Vulnerabilities and Exposures (CVE) genannt, aufgelistet. Die Prüfung der Bibliotheken wird per Maven-Plugin[13] in den Maven-Build der Anwendung integriert und manuell ausgeführt. Pro ermitteltem CVE wird eine der folgenden Maßnahmen gesetzt:

- Die Bibliothek wird auf eine Version ohne Schwachstelle aktualisiert.
- Ist die Schwachstelle für die Backuplösung nicht relevant, weil beispielsweise die betroffene Bibliotheksfunktion nicht genutzt wird, wird die Schwachstelle für zukünftige Checks aus den relevanten Schwachstellen ausgenommen und dies entsprechend in der Backupanwendung dokumentiert.

[11] `https://spring.io/` - besucht am 24.04.2022
[12] `https://netty.io/` - besucht am 24.04.2022
[13] `https://github.com/jeremylong/DependencyCheck/tree/main/maven` - besucht am 24.04.2022

- Es wird eine alternative Bibliothek ohne Schwachstelle eingesetzt.
- Ist keine der obigen Maßnahmen möglich wird versucht, die Sicherheitslücke durch Anpassungen im Prototypen zu mitigieren.

Mit dieser Vorgehensweise wird der Maßnahme MA-2 Einsatz von Analysetools zur Erkennung von Software-Konzeptionsfehlern und -Schwachstellen entsprochen.

5.2.4.2 Untersuchung des Quellcodes mittels SonarQube

Bei der Entwicklung von Software entstehen Fehler. Diese können die Backuplösung im einfachsten Fall in bestimmten Situationen an der korrekten Funktionsweise hindern. Im schlimmsten Fall können sie zu sicherheitsrelevanten Schwachstellen führen. Selbst wenn im Quellcode kein Fehler enthalten ist kann ein schlechter Codestil die Analysierbarkeit oder Änderbarkeit des Quellcodes bei notwendigen Anpassungen in der Zukunft beeinträchtigen.

Zur Minimierung dieser Probleme wird das statische Codeanalyse-Tool SonarQube[14] verwendet. Dieses untersucht den Quellcode auf potenzielle Fehler im Programmfluss, erkennt Sicherheitslücken und weist auf schlechten Codestil hin. Die Sonarqube-Analyse wird per Maven-Plugin in den Maven-Build der Anwendung integriert, manuell ausgeführt und die SonarQube-Informationen an einen lokalen SonarQube-Server geschickt. Sonarqube wird als Docker-Container mit der Standardkonfiguration von Sonarqube zur Klassifizierung von Fehlern verwendet. Im Rahmen der Implementierung werden alle Fehler mit der Klassifizierung *Blocker*, *Critical* und *Major* behoben oder mit einer Begründung als False-Positives markiert. Mit dieser Vorgehensweise wird der Maßnahme MA-2 Einsatz von Analysetools zur Erkennung von Software-Konzeptionsfehlern und -Schwachstellen entsprochen.

5.3 Funktionale Anwendungsfälle

Im folgenden Kapitel werden Design und Implementierung des Backupsystems anhand der funktionalen Anwendungsfälle erläutert. Der Fokus liegt auf den Anwendungsfällen der Datensicherung (*Kapitel 5.3.1*), der Verifikation der Sicherungsdaten zur Sicherstellung der Verfügbarkeit und Integrität (*Kapitel 5.3.2*) und der Datenwiederherstellung in *Kapitel 5.3.3*.

5.3.1 Datensicherung bei Peer-to-Peer-Teilnehmern

Die Sicherung der Nutzdaten besteht aus mehreren Schritten. *Kapitel 5.3.1.1* beschreibt die Verarbeitung eines lokalen Verzeichnisses, dessen Daten gesichert werden sowie die Aktualisierung des lokalen Index und notwendiger Metadaten. In *Kapitel 5.3.1.2* wird der Vorgang zum Upload der Sicherungsdaten zu den konfigurierten Cloud-Speichern

[14] `https://www.sonarqube.org/` - besucht am 24.04.2022

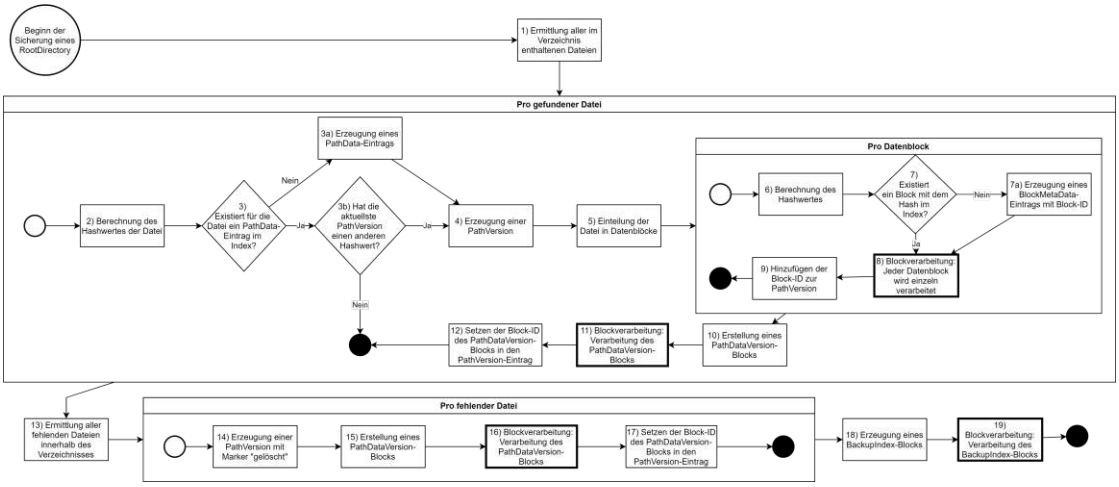

Abbildung 5.9: Ablauf einer Sicherung

beschrieben. *Kapitel 5.3.1.3* erläutert die Verteilung der Sicherungsdaten an andere Teilnehmer.

5.3.1.1 Verarbeitung der Nutzdaten im Backup-Service

Um die Sicherung von Dateien zu ermöglichen werden die zu sichernden Verzeichnisse in der Backupanwendung konfiguriert. Jedes dieser Verzeichnisse wird über das Backup-Service im lokalen Index als *RootDirectory* (*siehe Kapitel 5.1.4.2*) angelegt. Der Vorgang während der Datensicherung ist für jedes konfigurierte Verzeichnis gleich, daher wird in weiterer Folge die Datensicherung nur für ein *RootDirectory* beschrieben.

Eine Sicherung wird vom Anwender manuell über das Backup-Service veranlasst oder geschieht automatisch beim Hinzufügen eines neuen *RootDirectory*. Die dabei durchlaufenen Verarbeitungsschritte werden in *Abbildung 5.9* gezeigt. Zur Vereinfachung wird die Verarbeitung von Datenblöcken innerhalb des Ablaufdiagramms als *Blockverarbeitung* bezeichnet und separat in *Abbildung 5.10* dargestellt. Der dargestellte Ablauf beschreibt nur die Schritte zur Aktualisierung des lokalen Index während einer Sicherung ohne die anschließende Datenverteilung.

Die Datenverteilung an den Cloud-Speicher und andere Teilnehmer erfolgt währenddessen nebenläufig im Distribution-Service auf Basis der Date im lokalen Index und wird in *Kapitel 5.3.1.2* und *Kapitel 5.3.1.3* beschrieben. Die parallele Durchführung der Sicherung der Nutzdaten und Verteilung der Sicherungsdaten entspricht der Umsetzung von MA-14 Erzeugung von Sicherungsdaten unabhängig von Verteilung bei anderen Teilnehmern . Dies führt dazu, dass trotz einer langsamen Internet-Verbindung oder bei Nicht-Verfügbarkeit von anderen Teilnehmern Sicherungsdaten erzeugt werden.

Die gesicherten Verzeichnisse sind im lokalen Index als *RootDirectories* verfügbar. Eine Datensicherung durchläuft im Backup-Service die Schritte aus *Abbildung 5.9*:

1. Innerhalb eines konfigurierten *RootDirectory* werden alle enthaltenen Dateien ermittelt.
2. Pro Datei wird ein SHA-256-Hash erzeugt. Dieser wird zur Erkennung von Änderungen der Datei benötigt.
3. Im lokalen Index wird ermittelt, ob für den Pfad der Datei ein *PathData*-Eintrag vorhanden ist.
   a) Wenn nein: Es wird ein neuer *PathData*-Eintrag mit dem Dateipfad angelegt.
   b) Wenn ja: Für den *PathData*-Eintrag wird die aktuellste gesicherte Dateiversion (*PathVersion*) anhand des Änderungsdatums ermittelt. Entspricht der gespeicherte Hash der Datei-Version dem berechneten der aktuellen Datei bedeutet dies, dass sich die Datei nicht verändert hat. Eine neue Sicherung der Datei ist nicht notwendig und die weitere Verarbeitung dieser Datei wird übersprungen.
4. Es wird eine neue *PathVersion* mit dem aktuellen Zeitstempel und dem berechneten Hashwert angelegt.
5. Es erfolgt die Einteilung der Datei in Datenblöcke zu je 500KB (*siehe Kapitel 5.1*).
6. Für jeden Datenblock wird ein SHA-256-Hash erzeugt. Dieser wird als Fingerprint des Datenblocks zur Wiedererkennung in unterschiedlichen Dateien oder Dateiversionen verwendet.
7. Anhand dieses Hashes wird ermittelt, ob bereits ein *BlockMetaData*-Eintrag mit gleichem Hash-Wert im lokalen Index verfügbar ist.
   a) Wenn nein: Es wird ein neuer *BlockMetaData*-Eintrag erzeugt. Für diesen wird eine eindeutige ID in Form einer UUID erzeugt und zusätzlich der berechnete Hash-Wert zugewiesen. In weiteren Schritten wird die ID auch Block-ID genannt.
8. Jeder Datenblock wird verarbeitet. Dabei werden Metadaten für den Block angelegt und die Verteilung über das Distribution-Service angestoßen. Der genaue Ablauf wird in *Abbildung 5.10* gezeigt und nachfolgend beschrieben.
9. Für jeden verarbeitete Datenblock wird dessen Block-ID in der Reihenfolge des Auftretens innerhalb der Datei der *PathVersion* hinzugefügt.
10. Zur Sicherung der *PathVersion* wird eine *PathDataVersion* erzeugt (*siehe Kapitel 5.1.4.4*) und in einen Datenblock verpackt.
11. Dieser neue Datenblock wird wie vorherige Datenblöcke nach Schritt 8 verarbeitet.
12. Die Block-ID des neuen Datenblocks wird beim *PathVersion*-Eintrag der Datei-Version hinterlegt.

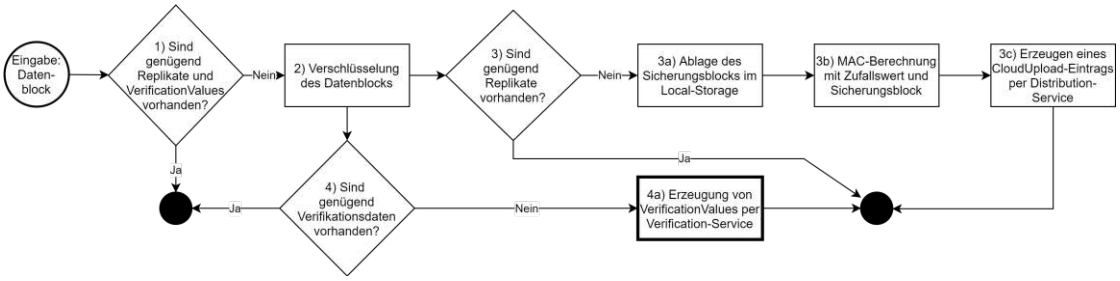

Abbildung 5.10: Verarbeitung eines Datenblocks

13. Nach Verarbeitung aller Dateien, die im Sicherungsverzeichnis vorhanden sind, erfolgt die Ermittlung von fehlenden Dateien. Dies sind somit alle Dateien, die seit der letzten durchgeführten Sicherung gelöscht wurden und betrifft alle *PathData*-Einträge, deren aktuellste Version nicht den Marker gelöscht hat und für die keine Datei im Dateisystem existiert.

14. Für jede fehlende Datei wird eine neue *PathVersion* mit dem aktuellen Zeitstempel und dem Marker gelöscht angelegt.

15. Für die als gelöscht markierte *PathVersion* wird eine *PathDataVersion* erzeugt (*siehe Kapitel 5.1.4.4*) und in einen Datenblock verpackt.

16. Der *PathDataVersion*-Block wird wie in Schritt 8 verarbeitet.

17. Die Block-ID des neuen Datenblocks wird beim *PathVersion*-Eintrag der Datei-Version hinterlegt.

18. Nach Verarbeitung aller Dateien wird für die gesamte Sicherung ein *BackupIndex*-Block erzeugt (*siehe Kapitel 5.1.4.4*). Dieser wird zur Wiederherstellung der Metadaten benötigt. Bei Totalausfall des Systems kann anhand des *BackupIndex* der letzte Stand der Nutzdaten wiederhergestellt werden. Der *BackupIndex* beinhaltet die konfigurierten Verzeichnisse sowie die Block-IDs aller *PathDataVersions* der Sicherung. Details zum *BackupIndex* und der Datenwiederherstellung werden in *Kapitel 5.3.3* erläutert.

19. Der *BackupIndex*-Block wie vorherige Datenblöcke nach Schritt 8 verarbeitet.

Nach Verarbeitung des *BackupIndex* ist die Datensicherung abgeschlossen.

Ein Datenblock kann Nutzdaten oder Metadaten aus dem lokalen Index enthalten und eine beliebige Größe haben. Die Verarbeitung eines Datenblocks (*siehe Abbildung 5.10*) geschieht im Backup-Service und erfolgt für alle Datenblöcke gleich:

1. Zu Beginn wird mithilfe des lokalen Index überprüft, ob für einen Datenblock genügend Replikate im Backupsystem vorhanden sind. Dies ist der Fall, wenn

mindestens drei verifizierte *DataLocations* zu anderen Teilnehmern existieren (*siehe Kapitel 5.1.4.2*). Zusätzlich wird überprüft, ob für den Datenblock genügend Verifikationsdaten vorhanden sind. Dies trifft zu, wenn für dessen Block-ID mehr als sechs *VerificationValues* in den lokalen Metadaten gespeichert sind (*siehe Kapitel 5.1.4.3*). Treffen beide Bedingungen zu wird die Verarbeitung des Datenblocks übersprungen.

2. Trifft eine der Bedingungen nicht zu wird der Datenblock mittels AES im AEAD-Betriebsmodus GCM und einem abgeleiteten Schlüssel (*siehe Kapitel 5.2.3.10*) symmetrisch verschlüsselt. Ergebnis ist ein verschlüsselter Datenblock der Sicherungsblock genannt wird.

3. Wenn zu wenige Replikate des Sicherungsblocks vorhanden sind wird dessen Verteilung angestoßen.

   a) Der Sicherungsblock wird unter seiner Block-ID im Local-Storage abgelegt.

   b) Es wird ein Zufallswert gebildet und damit der HMAC mit SHA-256 des Sicherungsblock berechnet (*siehe Kapitel 5.2.3.11*). Der MAC wird von anderen Teilnehmern zur Integritätsprüfung des Sicherungsblocks nach dem Download vom Cloud-Speicher benötigt.

   c) Anschließend wird über das Distribution-Service ein *CloudUpload*-Eintrag erzeugt um die Verteilung des Sicherungsblocks anzustoßen. Der *CloudUpload*-Eintrag referenziert den Sicherungsblock per Block-ID und beinhaltet dessen MAC und den zur Berechnung notwendigen Zufallswert. Der nächste Schritt in der Datenverteilung ist der Upload zum Cloud-Speicher (*siehe Kapitel 5.3.1.2*).

4. Wenn zu wenige Verifikationsdaten vorhanden sind werden zusätzliche Verifikationsdaten erzeugt.

   a) Anhand des Sicherungsblocks werden über das Verification-Service *VerificationValues* erzeugt. Details dazu werden in *Kapitel 5.3.2.1* erläutert.

5.3.1.2 Ablage der Sicherungsdaten im Cloud-Speicher

Die Verteilung der Sicherungsdaten erfolgt innerhalb des Distribution-Service über die angebundenen Cloud-Speicher. Sicherungsdaten werden einmal in der Cloud abgelegt. Alle sichernden Teilnehmer werden danach direkt aufgefordert die Sicherungsblöcke aus dem Cloud-Speicher herunterzuladen und lokal abzulegen. Dies hat den Vorteil, dass Sicherungsdaten bereits Off-Site (in der Cloud) abgelegt werden können auch wenn keine anderen Teilnehmer verfügbar sind. Teilnehmer können später zum Download der Sicherungsdaten aufgefordert werden. Diese können den Download vor allem im Fall einer größeren Datenmenge durchführen, ohne dass der Eigentümer der Daten verfügbar sein muss.

Die Ablage von Sicherungsdaten im Cloud-Speicher erfolgt, sobald diese im Local-Storage verfügbar sind und ein entsprechender *CloudUpoad*-Eintrag erstellt wurde. Dies bedeutet,

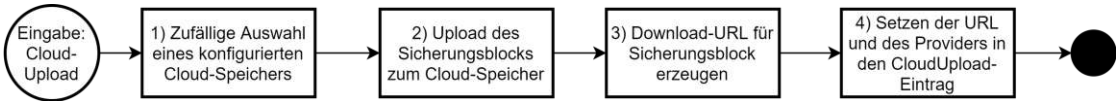

Abbildung 5.11: Ablauf des Cloud-Uploads eines Sicherungsblocks

dass bereits während der laufenden Sicherung eines Verzeichnisses vom Distribution-Service mit der Verteilung von Sicherungsdaten in die Cloud begonnen wird. Zur Ablage von Sicherungsdaten im Cloud-Speicher werden alle *CloudUpload*-Einträge ohne *providerId* und *shareUrl* ermittelt. Dabei handelt es sich um alle Sicherungsblöcke, die noch nicht zu einem Cloud-Speicher hochgeladen wurden. Die Verarbeitung jedes *CloudUpload*-Eintrags wird in *Abbildung 5.11* gezeigt und geschieht nach folgendem Ablauf:

1. Es wird zufällig einer der konfigurierten Cloud-Speicher aus *Kapitel 5.2.1* ausgewählt.
2. Der Sicherungsblock wird über das Cloud-Speicher-API im Cloud-Speicher abgelegt.
3. Für den hochgeladenen Sicherungsblock wird per API-Aufruf eine Download-URL erzeugt. Mit dieser öffentlichen URL kann ein anderer Teilnehmer den Sicherungsblock vom Cloud-Speicher downloaden.
4. Der gerade verarbeitete *CloudUpload*-Eintrag wird um die Provider-ID des verwendeten Cloud-Speichers und die erzeugten URL erweitert. Er steht damit für weitere Cloud-Uploads nicht mehr zur Verfügung.

Nach der Ablage der Sicherungsdaten im Cloud-Speicher werden diese noch nicht aus dem eigenen Local-Storage gelöscht. Für den Fall von Übertragungsfehlern oder einer Manipulation am Cloud-Speicher werden diese laut MA-19 Verzögerte Löschung der lokalen Sicherungsdaten vorerst aufbewahrt und erst nach Herstellung von ausreichen Replikaten gelöscht.

5.3.1.3 Verteilung der Sicherungsdaten an andere Teilnehmer

Nach Upload der Sicherungsdaten zu den Cloud-Speichern wird vom Distribution-Service an andere Teilnehmer die Aufforderung zur Datensicherung geschickt. Sicherungsdaten können nur an Teilnehmer übermittelt werden, für die eine gegenseitige Authentifizierung möglich ist (*siehe Kapitel 5.2.3.3*) und die zum Empfang von Sicherungsdaten konfiguriert wurden. Zur Verteilung werden vom Distribution-Service alle *CloudUpload*-Einträge mit gesetzter *providerId* und *shareUrl* ermittelt. Dabei handelt es sich um alle Sicherungsblöcke, die bereits in einem Cloud-Speicher abgelegt wurden und zum Download für andere Teilnehmer bereit stehen. Die Verarbeitung eines *CloudUpload*-Eintrags bzw. die Verteilung an andere Teilnehmer wird in *Abbildung 5.12* dargestellt:

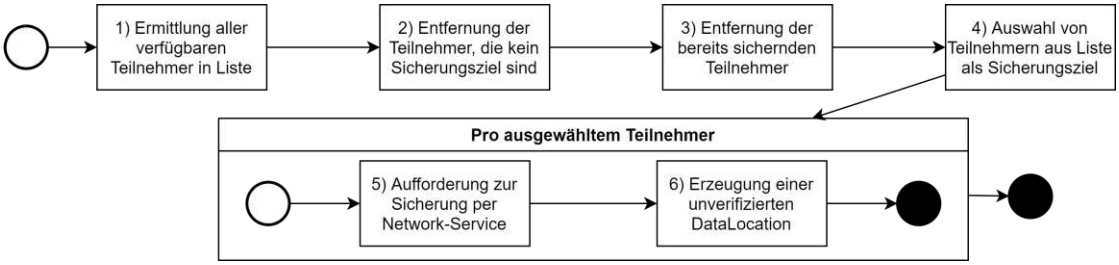

Abbildung 5.12: Verteilung eines Sicherungsblocks an andere Teilnehmer

1. Über das Network-Service werden alle Teilnehmer ermittelt, zu denen aktuell eine Verbindung aufgebaut werden kann. Diese werden in einer temporären Teilnehmerliste gespeichert.

2. Es werden alle Teilnehmer aus der Liste entfernt, die nicht für den Empfang von Sicherungsdaten konfiguriert wurden.

3. Ein *CloudUpload*-Eintrag referenziert per Block-ID einen bestimmten Datenblock. Es werden alle verifizierten *DataLocations*, die dem *BlockMetaData*-Eintrag mit der Block-ID zugeordnet sind, geladen. Die hinterlegten Teilnehmer sichern bereits ein Replikat und werden aus der Teilnehmerliste entfernt. Alle übrigen Teilnehmer der Teilnehmerliste sind Kandidaten um ein Replikat des Sicherungsblocks zu speichern.

4. Aus der Teilnehmerliste werden zufällig `n` Teilnehmer gewählt. `n` entspricht der auf drei fehlenden Anzahl an verifizierten Replikaten, d.h. `n = 3 - s` wobei `s` der Anzahl an aktuell verifizierten Replikate bei anderen Teilnehmern entspricht. Damit versucht die Backupanwendung pro Datenblock die Verfügbarkeit von mindestens drei Replikaten sicherzustellen.

5. Zu jedem der gewählten Teilnehmer wird per Network-Service eine Nachricht geschickt. Diese beinhaltet die Block-ID, die URL zum Download des Sicherungsblocks sowie den MAC-Wert und -Schlüssel zur Integritätsprüfung des Sicherungsblocks. Die Kommunikation erfolgt vertraulich via TLS (*siehe Kapitel 5.2.3.6*).

6. Nach Versand der Nachricht wird ein *DataLocation*-Eintrag erstellt und dem *BlockMetaData*-Eintrag des Blocks zugeordnet. Die *DataLocation* beinhaltet die User-ID des Teilnehmers. Das *verified*-Datum wird 21 Tage in die Vergangenheit gesetzt, damit die Sicherung der Daten bei dem Teilnehmer vorerst als unverifiziert gilt (*siehe Kapitel 5.1.4.2*).

Nach der Aufforderung zur Datensicherung werden die lokalen Sicherungsdaten im LocalStorage vorerst nicht gelöscht (MA-19 Verzögerte Löschung der lokalen Sicherungsdaten). Dies ist notwendig, um eine Manipulation am Cloud-Speicher oder Übertragungsfehler vom Cloud-Speicher zu den sichernden Teilnehmern kompensieren zu können. Die Löschung der lokalen Sicherungsdaten geschieht erst zu einem späteren Zeitpunkt.

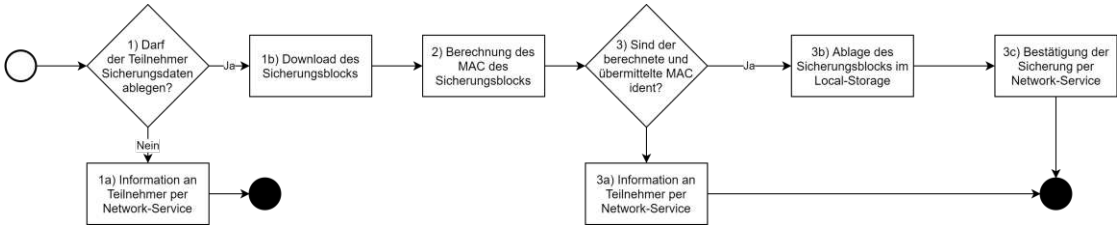

Abbildung 5.13: Speicherung eines Sicherungsblocks am Zielsystem

Ein Teilnehmer, der eine Nachricht zur Sicherung eines Blocks über das Network-Service erhält führt die Schritte aus *Abbildung 5.13* aus:

1. Es wird geprüft, ob der übermittelnde Teilnehmer zur Sicherung von Daten konfiguriert wurde. Nur wenn dies zutrifft dürfen Sicherungsdaten des Teilnehmers lokal abgelegt werden.
   a) Wenn nein: Der übermittelnde Teilnehmer wird über das Network-Service informiert, dass der Sicherungsblock nicht gesichert werden darf. Die Verarbeitung des Sicherungsblocks wird danach abgebrochen.
   b) Wenn ja: Der Sicherungsblock wird über die übermittelte URL heruntergeladen.
2. Es wird mithilfe des Schlüssels aus der Nachricht der HMAC mit SHA-256 des Sicherungsblocks berechnet.
3. Es wird überprüft ob der berechnete MAC-Wert und der in der Nachricht übermittelte MAC-Wert übereinstimmen. Dies entspricht der Umsetzung der Maßnahme MA-20 Integritätsprüfung der Cloud-Sicherungsdaten .
   a) Wenn nein: Die Datenquelle wird nach MA-21 Direkte Teilnehmer-Kommunikation bei fehlenden/nicht integren Daten im Cloud-Speicher per Network-Service direkt über die fehlgeschlagene Prüfung informiert und erhält eine Fehler-Nachricht mit der Block-ID. Der übermittelnde Teilnehmer kann danach gegebenenfalls den Sicherungsblock vom Cloud-Speicher löschen, einen anderen Cloud-Provider wählen und die Sicherungsdaten erneut zum Cloud-Speicher hochladen.
   b) Wenn ja: Der Sicherungsblock wird im Local-Storage gespeichert.
   c) Nach der Speicherung des Sicherungsblocks wird dem übermittelnden Teilnehmer per Network-Service eine Bestätigungsnachricht mit der Block-ID gesendet.

Der Eigentümer der Nutzdaten erhält am Ende des Sicherungsvorganges eine Nachricht vom sichernden Teilnehmer. Diese wird vom Network-Service an das Distribution-Service weitergeleitet. Im Fall einer Bestätigungsnachricht führt dies dazu, dass für den sichernden

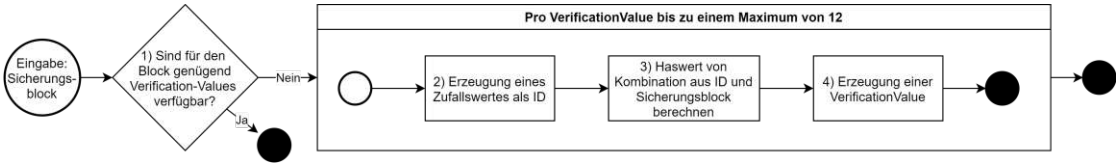

Abbildung 5.14: Generierung von Verifikationsdaten

Teilnehmer das *verified*-Datum der *DataLocation* des gesicherten Blocks auf den aktuellen Zeitpunkt gesetzt wird. Wurden damit drei verifizierte Replikate des Sicherungsblocks hergestellt wird der Sicherungsblock vom Cloud-Speicher sowie dem Local-Storage entfernt und der *CloudUpload*-Eintrag gelöscht. Die Sicherung und Verteilung an andere Teilnehmer ist für den Sicherungsblock abgeschlossen.

Im Fall einer Fehler-Nachricht wird die *DataLocation* des Teilnehmers gelöscht, da dieser den Sicherungsblock nicht speichern konnte. Für den Fall einer fehlgeschlagenen MAC-Prüfung wird zusätzlich der Sicherungsblock vom Cloud-Speicher entfernt und die Merkmale *providerId* und *shareUrl* des *CloudUpload*-Eintrags gelöscht. Der *CloudUpload*-Eintrag wird daraufhin wie in *Kapitel 5.3.1.2* wieder neu zu einem Cloud-Provider hochgeladen.

5.3.2 Datenverifikation

Die Sicherung der Nutzdaten erfolgt, indem lokal Sicherungsdaten erzeugt und diese in Form von Replikaten an andere Teilnehmer verteilt werden. Nach dieser Verteilung werden die lokalen Sicherungsdaten gelöscht. Um zu garantieren, dass die ursprünglichen Nutzdaten wiederherstellbar bleiben, muss die Backupanwendung regelmäßig die Verfügbarkeit und Integrität der Replikate überprüfen. Die Generierung der dafür notwendigen Verifikationsdaten wird in *Kapitel 5.3.2.1* beschrieben. *Kapitel 5.3.2.2* beschreibt den regelmäßigen Ablauf der Verifikation.

5.3.2.1 Erzeugung von Verifikationsdaten

Die Verwaltung der Verifikationsdaten geschieht im Verification-Service. Die Überprüfung von Sicherungsdaten bei anderen Teilnehmern erfolgt über ein Challenge-Response-Verfahren. Dazu sind vorausberechnete Verifikationsdaten notwendig die im lokalen Index abgelegt werden. Die Erzeugung wird während der Sicherung der Dateien angestoßen (*siehe Kapitel 5.3.1.1*). Dabei wird ein Sicherungsblock dem Verification-Service zur Erzeugung von Verifikationsdaten übergeben und dem Ablauf aus *Abbildung 5.14* gefolgt:

1. Für den Sicherungsblock wird überprüft, ob genügend *VerificationValues* (*siehe Kapitel 5.1.4.3*) in den lokalen Metadaten vorrätig sind. Dies ist der Fall, wenn mindestens sechs *VerificationValues* für den Block existieren. In diesem Fall wird die Erzeugung von Verifikationsdaten übersprungen.

2. Andernfalls werden die lokalen *VerificationValues* auf 12 Einträge aufgefüllt. Dazu wird wie in *Kapitel 5.2.3.12* beschrieben je Eintrag eine ID in Form eines Zufallswertes mit 128 Bit erzeugt.

3. Diese wird dem Sicherungsblock vorangestellt und daraus der SHA3-512-Hash berechnet.

4. Die Kombination aus zufälliger ID und Hash-Wert wird als *VerificationValue* für den Block gespeichert.

5.3.2.2 Verifikation der Sicherungsdaten bei anderen Teilnehmern

Die Verifikation der Sicherungsdaten geschieht in periodischen Abständen. Das Datum der letzten Verifizierung eines Replikats wird pro Block und Teilnehmer im Merkmal *verified* der *DataLocation* gespeichert. Sobald dieser Zeitstempel älter als 14 Tage ist wird eine neue Überprüfung dieses Sicherungsblocks beim betroffenen Teilnehmer durchgeführt. Für die Verifizierung wird je Block eine *VerificationValue* als *ActiveVerificationValue* gesetzt. Diese beinhaltet die ID, den Hash-Wert und zusätzlich ein Gültigkeitsdatum bis zu dem diese *ActiveVerificationValue* für Verifizierungen verwendet wird. Eine *ActiveVerificationValue* ist immer für 14 Tage gültig, danach wird sie gelöscht und durch eine der verfügbaren *VerificationValues* ersetzt. Sollten für einen Block keine *VerificationValues* verfügbar sein wird der Sicherungsblock von einem anderen Teilnehmer angefordert um die Neugenerierung von Verifizierungsdaten zu ermöglichen.

Das Verification-Service ermittelt zur Verifikation der Replikate alle *DataLocations* deren Verifizierungsdatum überschritten wurde. Pro Replikat werden die Schritte in *Abbildung 5.15* ausgeführt:

1. Je Replikat wird über das Network-Service überprüft ob der sichernde Teilnehmer erreichbar ist. Ist der Teilnehmer nicht online wird die Überprüfung des Replikats abgebrochen.

2. Es wird ermittelt, ob eine *ActiveVerificationValue* für die Block-ID des Replikats existiert.

   a) Wenn ja: Es wird überprüft, ob die *ActiveVerificationValue* noch gültig ist. Dies ist der Fall, wenn *activeUntil* in der Zukunft liegt.

      i. Wenn nein: Die *ActiveVerificationValue* ist nicht mehr gültig und wird gelöscht.

   b) Wenn nein: Es wird überprüft, ob für die Block-ID des Replikats noch *VerificationValues* verfügbar sind.

      i. Wenn nein: Es wird per Network-Service eine Anfrage an den Teilnehmer zur Wiederherstellung des Sicherungsblocks gestellt. Bei Empfang des Sicherungsblocks werden neue *VerificationValues* generiert. Die Verifikation des Replikats wird abgebrochen.

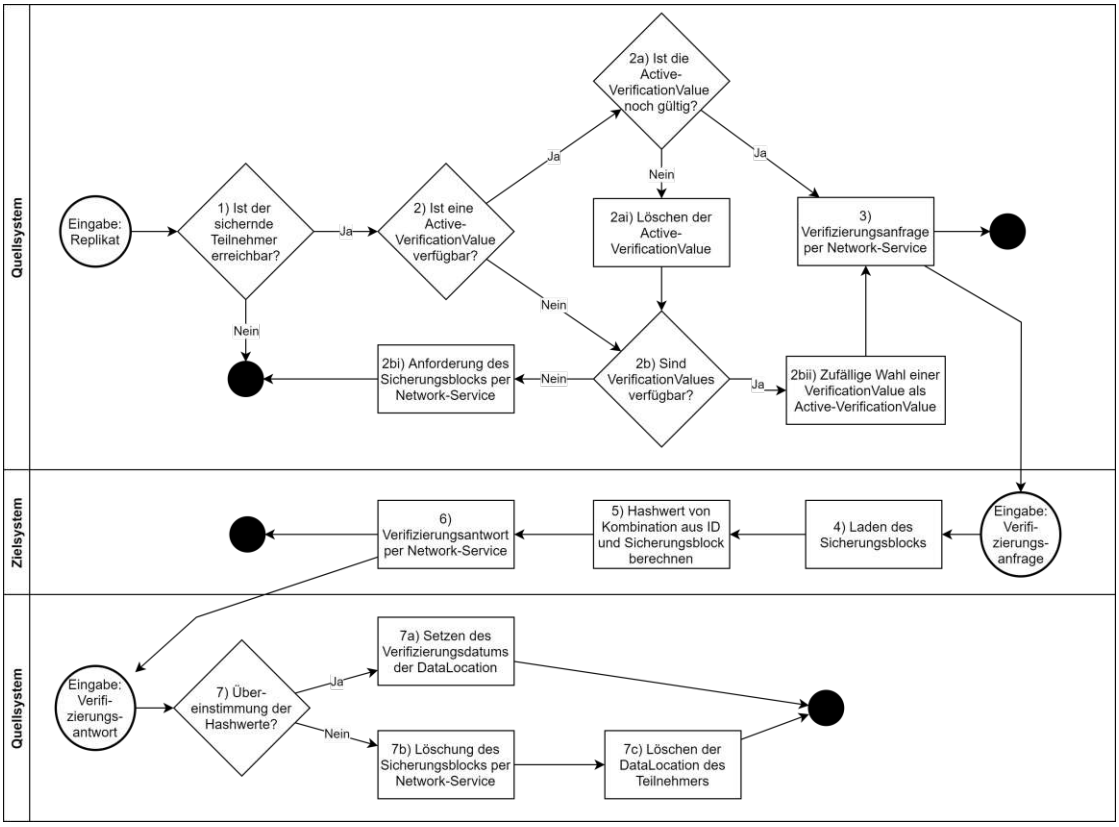

Abbildung 5.15: Ablauf der Verifikation eines Sicherungsblocks beim Quell- und Zielsystem

ii. Wenn ja: es wird eine zufällige *VerificationValue* gewählt und als neue *ActiveVerificationValue* gesetzt. Das Gültigkeitsdatum wird 14 Tage in die Zukunft gesetzt.

3. Die Datenquelle verschickt über das Network-Service eine Nachricht mit der Block-ID des Sicherungsblocks und der ID der *ActiveVerificationValue*. Die Kommunikation zwischen Datenquelle und dem sichernden Teilnehmer erfolgt vertraulich via TLS (*siehe Kapitel 5.2.3.6*).

4. Das Verification-Service des sichernden Teilnehmers erhält über das Network-Service die Aufforderung zur Prüfung. Der Sicherungsblock wird aus dem Local-Storage geladen.

5. Die ID der *ActiveVerificationValue* wird dem Sicherungsblock vorangestellt. Davon wird der SHA3-512-Hash berechnet (*siehe Kapitel 5.2.3.12*).

6. Der Teilnehmer sendet per Network-Service eine Antwort-Nachricht an die Datenquelle. Die Nachricht enthält die Block-ID, die ID der *ActiveVerificationValue* und den berechneten Hash-Wert.

7. Das Verification-Service der Datenquelle vergleicht, ob der übermittelte Hash-Wert mit dem Hash-Wert der *ActiveVerificationValue* übereinstimmt.
   a) Wenn ja: Das Verifizierungsdatum des Replikats wird in der *DataLocation* des Teilnehmers auf den aktuellen Zeitstempel gesetzt.
   b) Wenn nein: Dem sichernden Teilnehmer wird eine Aufforderung zum Löschen des gesicherten Blocks übermittelt, da dieser nicht mehr integer ist.
   c) Anschließend wird die *DataLocation* des Teilnehmers für den entsprechenden Block gelöscht, da dieser das Replikat nicht mehr vorhält.

Wenn während dem Verifikationsvorgang ein Teilnehmer erkennt, dass der zu prüfenden Block nicht mehr verfügbar ist, meldet er dies an die Datenquelle zurück. Diese verfährt dann wie bei einer Unterscheidung der Hashwerte. Sollte ein Teilnehmer nicht auf die Aufforderung zur Verifikation reagieren schickt die Backupanwendung in periodischen Abständen weitere Nachrichten zur Überprüfung. Replikate, deren letzte Verifizierung länger als 21 Tage zurück liegt, gelten als nicht mehr verifiziert. Die Backupanwendung versucht weiterhin eine Verifizierung durchzuführen, kümmert sich aber parallel dazu um die Anforderung des Sicherungsblocks um diesen an einen anderen Teilnehmer zu verteilen und wieder drei Replikate herzustellen. Wenn ein unverifiziertes Replikat später wieder verifiziert wird kann der Fall eintreten, dass für einen Sicherungsblock mehr als drei Replikate im Backupsystem vorhanden sind. Die Backupanwendung reagiert jedoch nicht gesondert auf diesen Fall und verwaltet bzw. überprüft alle vorhandenen Replikate.

5.3.3 Datenwiederherstellung

Die Datenwiederherstellung wird verwendet, um aus Replikaten bei anderen Teilnehmern die ursprünglichen Nutzdaten zu einem bestimmten Zeitpunkt wiederherzustellen. Es wird zwischen drei Arten der Datenwiederherstellung unterschieden:

1. Die Nutzdaten sind lokal nicht mehr verfügbar oder verwendbar, der Index ist jedoch noch verfügbar und integer. Dies ist beispielsweise der Fall, wenn Nutzdaten versehentlich gelöscht wurden oder eine Änderung durchgeführt wurde die rückgängig gemacht werden soll.
2. Sowohl die Nutzdaten als auch der Index sind nicht mehr verfügbar oder verwendbar. Dieser Fall kann beispielsweise eintreten, wenn es beim genutzten Massenspeicher zu einem Hardware-Defekt kommt oder der Teilnehmer Opfer eines Verschlüsselungstrojaners wurde.
3. Die Kompromittierung des Schlüsselpaares ist ein Sonderfall. Historische Nutzdaten sind lokal nicht mehr verfügbar, der Index ist jedoch noch verfügbar und integer. Aufgrund der Schlüsselkompromittierung ist ein Schlüsseltausch erforderlich. Damit kann nicht mehr auf Replikate bei anderen Teilnehmern zugegriffen werden.

Alle drei Fälle der Datenwiederherstellung werden nachfolgend beschrieben.

5.3.3.1 Wiederherstellung mit verfügbaren Metadaten

Mit intaktem Index sind der Backupanwendung alle notwendigen Metadaten zur Wiederherstellung der Nutzdaten bekannt. Es sind Informationen über alle gesicherten Dateien, ihre Dateiversionen und die Datenblöcke, aus denen jede Dateiversion besteht, verfügbar. Je Block sind zusätzlich die sichernden Teilnehmer in den *DataLocations* hinterlegt. Aus diesen Informationen können die notwendigen Datenblöcke zur Wiederherstellung der Nutzdaten zu einem beliebigen Zeitpunkt innerhalb der Sicherungsperiode ermittelt werden.

Die Datenwiederherstellung wird vom Benutzer manuell über das Restoration-Service eingeleitet. Dabei wird der Name des konfigurierten *RootDirectory*, der Zeitpunkt, dessen Datenstand wiederhergestellt werden soll, sowie ein Zielverzeichnis zur Speicherung des damaligen Datenstandes angegeben. Zum Start einer Datenwiederherstellung werden nach folgendem Ablauf Metadaten im lokalen Index angelegt:

1. Anhand des Namens des *RootDirectory* werden alle gesicherten Dateien des Verzeichnisses und die relevanten Dateiversionen (*PathVersion*) ermittelt. Dabei handelt es sich pro Datei um jene Dateiversion, deren Zeitstempel (*date*) mit kürzestem Abstand vor dem gesuchten Sicherungszeitpunkt liegt.

2. *PathVersions* mit dem Marker *deleted* werden ausgefiltert, da die gesicherte Datei zum damaligen Sicherungszeitpunkt aus dem Verzeichnis gelöscht wurde.

3. Pro *PathVersion* wird für jeden Block, aus denen eine Version besteht, im lokalen Index ein *RestoreBlockData*-Eintrag mit hoher Priorität gespeichert (*siehe Kapitel 5.1.4.5*). Der *RestoreBlockData*-Eintrag verweist auf den *BlockMetaData*-Eintrag des Blocks. Anhand dessen fordert die Backupanwendung in weiterer Folge die notwendigen Sicherungsblöcke von anderen Teilnehmern an.

4. Zusätzlich wird für jede *PathVersion* ein *RestorePath*-Eintrag zur Wiederherstellung der Datei gespeichert. Dieser Eintrag beinhaltet eine Referenz auf die *PathVersion*, den absoluten Pfad an dem die Datei wiederhergestellt werden soll sowie Referenzen zu den *RestoreBlockData*-Einträgen zur Anforderung der notwendigen Datenblöcke.

Nach der Initialisierung einer Wiederherstellung fordert die Backupanwendung von anderen Teilnehmern die notwendigen Sicherungsblöcke an. Diese Anforderung wird kontinuierlich vom Restoration-Service ausgeführt. Es werden alle *RestoreBlockData*-Einträge geladen. Jeder *RestoreBlockData* repräsentiert einen Sicherungsblock, der von einem anderen Teilnehmer angefordert werden muss. Pro Eintrag werden die Schritte in *Abbildung 5.16* durchlaufen:

1. Es wird überprüft, ob der Sicherungsblock des *RestoreBlockData*-Eintrags bereits im Local-Storage liegt. Ein Sicherungsblock ist beispielsweise im Local-Storage

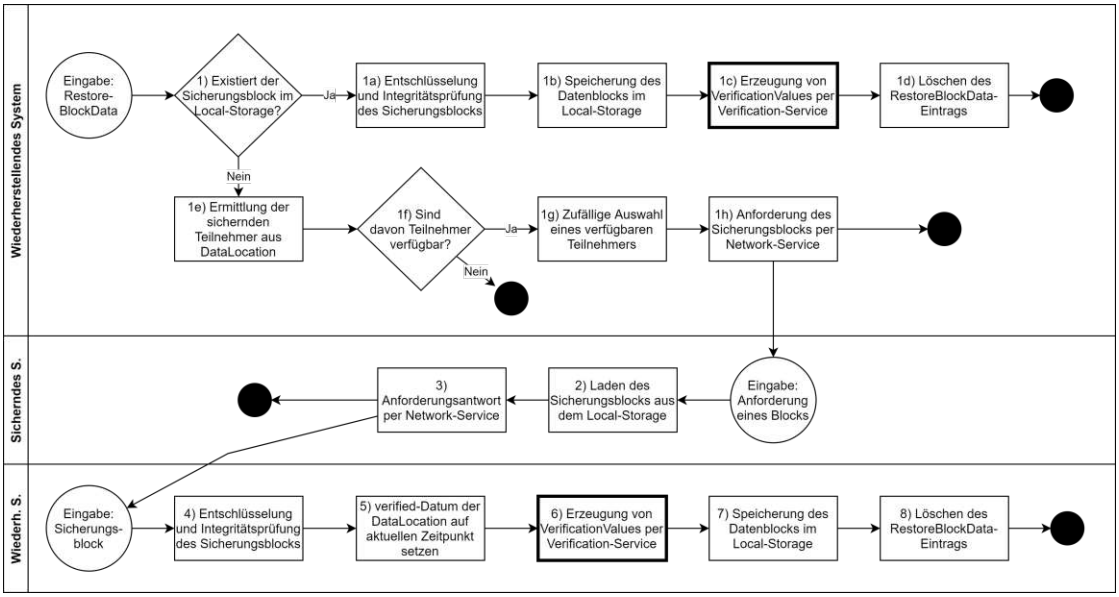

Abbildung 5.16: Anforderung von Sicherungsblöcken und Wiederherstellung der Datenblöcke

verfügbar, wenn er bei der Datensicherung noch nicht an genügend Teilnehmer verteilt wurde (*siehe Kapitel 5.3.1.3*).

a) Wenn ja: Der Sicherungsblock wird symmetrisch mit AES im AEAD-Betriebsmodus GCM entschlüsselt (*siehe Kapitel 5.2.3.10*).

b) Der Datenblock wird im Local-Storage zur Datenwiederherstellung abgelegt.

c) Das Verification-Service wird aufgefordert, falls notwendig, zusätzliche *VerificationValues* zu erzeugen (*siehe Kapitel 5.3.2.1*).

d) Anschließend wird der *RestoreBlockData*-Eintrag aus dem lokalen Index gelöscht. Der *RestoreBlockData*-Eintrag wird aus allen *RestorePath*-Einträgen, die den Block benötigen, entfernt.

e) Wenn nein: Ist ein Sicherungsblock lokal nicht verfügbar erfolgt die Anforderung von anderen Teilnehmern. Anhand der *DataLocations* des Blocks werden die sichernden Teilnehmer ermittelt. Wenn kein sichernder Teilnehmer verfügbar ist wird die Anforderung des Sicherungsblocks übersprungen.

f) Es wird ein zufälliger sichernder Teilnehmer zur Wiederherstellung gewählt.

g) Die Backupanwendung verschickt über das Network-Service eine Nachricht zur Anforderung des Sicherungsblocks mit der gefragten Block-ID an den sichernden Teilnehmer.

2. Der Empfänger der Anforderung ladet den entsprechenden Sicherungsblock aus dem Local-Storage.

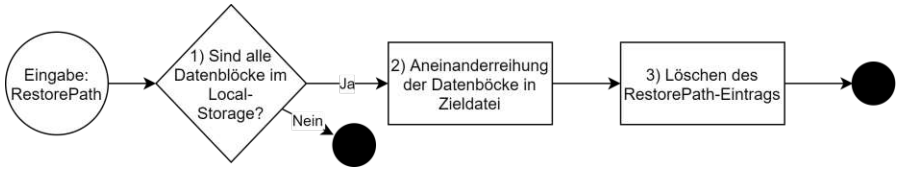

Abbildung 5.17: Wiederherstellung einer Datei

3. Der Sichungsblock wird per Network-Service direkt an den anfordernden Teilnehmer geschickt. Das Network-Service des anfordernden Teilnehmers empfängt den Sicherungsblock und leitet diesen an das Restoration-Service weiter.

4. Der Sicherungsblock wird symmetrisch mit AES im AEAD-Betriebsmodus GCM entschlüsselt (*siehe Kapitel 5.2.3.10*). Dadurch wird die Integrität des Sicherungsblocks geprüft und der Datenblock wird verfügbar.

5. Das Verifizierungsdatum der *DataLocation* des Blocks wird für den Teilnehmer auf den aktuellen Zeitpunkt gesetzt.

6. Per Verification-Service wird die Generierung von Verifikationsdaten angestoßen (*siehe Kapitel 5.3.2.1*). Dieses prüft, ob ausreichend *VerificationValues* verfügbar sind und erzeugt gegebenenfalls zusätzliche Werte.

7. Der Datenblock wird im Local-Storage zur Datenwiederherstellung abgelegt.

8. Der *RestoreBlockData*-Eintrag mit der Block-ID des Datenblocks wird geladen. Der Eintrag wird aus allen *RestorePath*-Einträgen gelöscht. Der *RestoreBlockData*-Eintrag wird aus dem lokalen Index entfernt.

Parallel zur Anforderung von Sicherungsblöcken werden alle Dateien, für welche die notwendigen Datenblöcke verfügbar sind, wiederhergestellt. Der Vorgang wird ebenfalls kontinuierlich vom Restoration-Service ausgeführt. Dazu werden alle *RestorePath*-Einträge geladen. Pro Eintrag wird dem Ablauf in *Abbildung 5.17* gefolgt:

1. Pro *RestorePath*-Einträge wird überprüft, ob für diesen Eintrag bereits alle Blöcke angefordert wurden und im Local-Storage verfügbar sind. Dies ist daran erkennbar, dass der *RestorePath*-Eintrag keine zugeordneten *RestoreBlockData*-Einträge besitzt.

2. Ist dies der Fall wird die Datei am Zielort wiederhergestellt. Dazu werden anhand der referenzierten *PathVersion* die Reihenfolge der Datenblöcke, aus denen die Datei besteht, ermittelt. Die entsprechenden Datenblöcke werden aus dem Local-Storage geladen und am Zielort zu einer Datei zusammengeführt.

3. Nach der Wiederherstellung wird der *RestorePath*-Eintrag gelöscht.

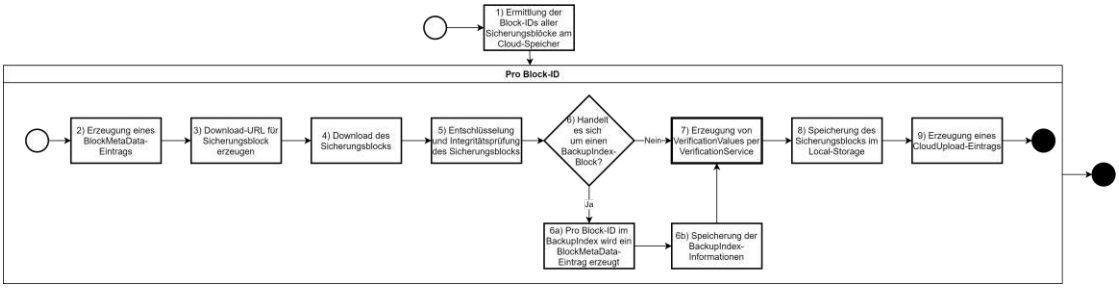

Abbildung 5.18: Wiederherstellung von einem Cloud-Provider nach Totalausfall

Die oben beschriebenen Abläufe werden solange durchlaufen, bis im lokalen Index keine *RestorePath*- und *RestoreBlockData*-Einträge mehr existieren. Zu diesem Zeitpunkt ist die Dateiwiederherstellung abgeschlossen. Danach werden alle wiederhergestellten Dateblöcke aus dem Local-Storage gelöscht.

5.3.3.2 Wiederherstellung nach Totalausfall

Bei einem Totalausfall gehen neben den Nutzdaten auch die lokalen Metadaten und die Konfiguration der Backupanwendung verloren. Um die Verbindung mit anderen Teilnehmern des Backupsystems wieder aufzubauen muss eine Grundkonfiguration der Backupanwendung wiederhergestellt werden. Es muss das eigene Schlüsselpaar eingerichtet werden um die Authentifizierung durch andere Teilnehmer zu ermöglichen und um kryptografische Verfahren nutzen zu können (*siehe Kapitel 5.2.3*). Das Schlüsselpaar wird von der Backupanwendung nicht gesichert und muss in geeigneter Form durch den Endnutzer gesichert werden. Des Weiteren müssen alle Teilnehmer, mit denen bisher kommuniziert wurde, konfiguriert werden. Das Zertifikat eines Teilnehmers sowie dessen Adresse zur Konnektivität sind außerhalb des Backupsystems anzufordern.

Ohne lokale Metadaten können die Nutzdaten nicht wiederhergestellt werden. Daher muss bei einem Totalausfall zuerst der lokale Index rekonstruiert werden, um auf dessen Basis die Wiederherstellung der eigentlichen Nutzdaten nach *Kapitel 5.3.3.1* durchzuführen. Es wird davon ausgegangen, dass nach einem Totalausfall immer die aktuellste Sicherung der Nutzdaten wiederhergestellt werden soll. Daher wird der aktuellste *BackupIndex* zur Rekonstruktion der lokalen Metadaten verwendet. Die Wiederherstellung der lokalen Metadaten wird über das Restoration-Service angefordert.

Sicherungsdaten müssen sowohl von den Cloud-Anbietern als auch von den anderen Teilnehmern wiederhergestellt werden. Die Nutzung der Cloud-Anbieter ist notwendig, da vom Backupsystem möglicherweise noch nicht alle Sicherungsdaten zu anderen Teilnehmern verteilt wurden und es Sicherungsblöcke ohne Replikate geben kann. Beispielsweise könnte sich der aktuellste *BackupIndex*-Block nur im Cloud-Speicher befinden. Zur Wiederherstellung der Metadaten wird für jeden konfigurierten Cloud-Anbieter der Ablauf aus *Abbildung 5.18* durchlaufen:

1. Das Restoration-Service ermittelt alle Sicherungsblöcke, die aktuell bei konfigurierten Cloud-Anbietern gespeichert sind.
2. Für jede Block-ID wird im lokalen Index ein *BlockMetaData*-Eintrag angelegt.
3. Für jeden Sicherungsblock wird über den Cloud-Anbieter eine öffentliche URL erzeugt.
4. Jeder Block wird über die öffentliche URL heruntergeladen.
5. Jeder Sicherungsblock wird symmetrisch mit AES im AEAD-Betriebsmodus GCM entschlüsselt, um dessen Integrität zu überprüfen (*siehe Kapitel 5.2.3.10*). Dadurch wird der Datenblock verfügbar.
6. Es wird überprüft, ob es sich bei dem Datenblock um einen *BackupIndex* handelt. Dies ist am Prefix `IDX_TIMESTAMP` der Block-ID erkennbar (*siehe Kapitel 5.1.4.4*).
   a) Wenn ja: Für jede im *BackupIndex* enthaltene Block-ID wird ein *BlockMetaData*-Eintrag angelegt. Diese Block-IDs repräsentieren *PathDataVersion*-Blöcke und enthalten die Metadaten zu allen Dateien dieser Sicherung.
   b) Der *BackupIndex* wird im lokalen Index gespeichert. Diese Informationen sind notwendig, da später vom Benutzer ein bestimmter *BackupIndex* ausgewählt wird, dessen Dateien prior wiederhergestellt werden sollen.
7. Pro Sicherungsblock wird vom Verification-Service die Generierung von Verifikationsdaten angefordert (*siehe Kapitel 5.3.2.1*).
8. Jeder Sicherungsblock wird im Local-Storage abgelegt.
9. Es wird ein *Cloud-Upload*-Eintrag mit der öffentlichen URL erzeugt, um nach der Datenwiederherstellung die Verteilung des Sicherungsblocks wie in *Kapitel 5.3.1.3* beschrieben zu ermöglichen. Dies ist notwendig, da Sicherungsblöcke am Cloud-Anbieter potenziell noch nicht an andere Teilnehmer verteilt wurden.

Nach der Durchführung für alle konfigurierten Cloud-Anbieter wurden alle Metadaten, die möglicherweise nur am Cloud-Anbieter gesichert wurden, wiederhergestellt. Alle Sicherungsblöcke liegen im Local-Storage und können bei der Datenwiederherstellung genutzt werden. Zusätzlich wird die Medadaten-Wiederherstellung von allen konfigurierten Teilnehmern angefordert. Dazu werden pro Teilnehmer die Schritte aus *Abbildung 5.19* durchlaufen:

1. Das Restoration-Service fordert über das Network-Service vom Teilnehmer den aktuellsten *BackupIndex* und die Block-IDs aller gesicherten Blöcke an.
2. Der Teilnehmer ladet den aktuellsten *BackupIndex*-Block aus dem Local-Storage. Dies ist an der Block-ID erkennbar, da diese bei einem *BackupIndex*-Block den Sicherungszeitpunkt enthält.

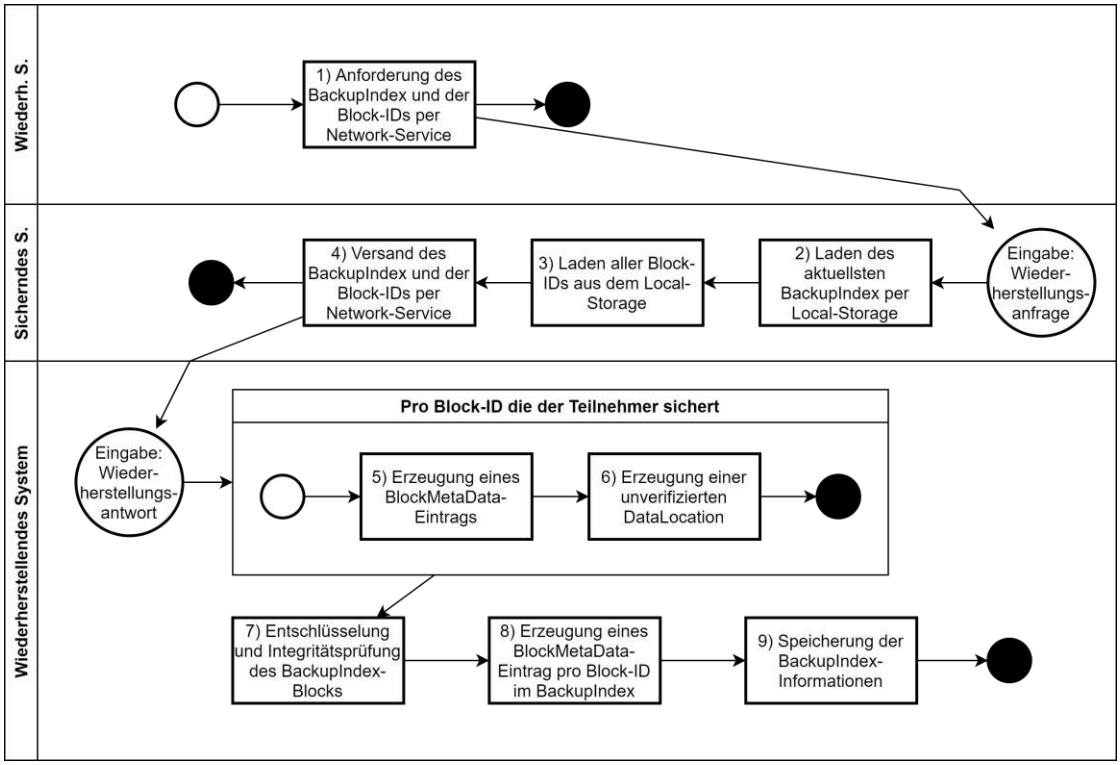

Abbildung 5.19: Wiederherstellung von einem Teilnehmer nach Totalausfall

3. Der Teilnehmer ladet die Liste aller Block-IDs der Sicherungsblöcke, die vom anfordernden Teilnehmer gesichert werden, aus seinem Local-Storage.

4. Der *BackupIndex*-Block und die Block-IDs werden per Network-Service an den anfordernden Teilnehmer geschickt.

5. Das Restoration-Service legt für jede Block-ID, die vom Teilnehmer gesichert wird, im lokalen Index einen *BlockMetaData*-Eintrag an.

6. Pro angelegtem *BlockMetaData*-Eintrag wird eine *DataLocation* für den Teilnehmer angelegt, da dieser Teilnehmer ein Replikat des Blocks sichert. Die *DataLocation* wird noch als unverifiziert markiert, da noch nicht geprüft wurde, ob der Block bei dem Teilnehmer verfügbar und integer ist.

7. Der *BackupIndex*-Block wird symmetrisch mit AES im AEAD-Betriebsmodus GCM entschlüsselt (*siehe Kapitel 5.2.3.10*). Dadurch wird die Integrität des Backup-Index geprüft.

8. Für jede im *BackupIndex* enthaltene Block-ID wird ein *BlockMetaData*-Eintrag angelegt. Diese Block-IDs repräsentieren *PathDataVersion*-Blöcke und enthalten die

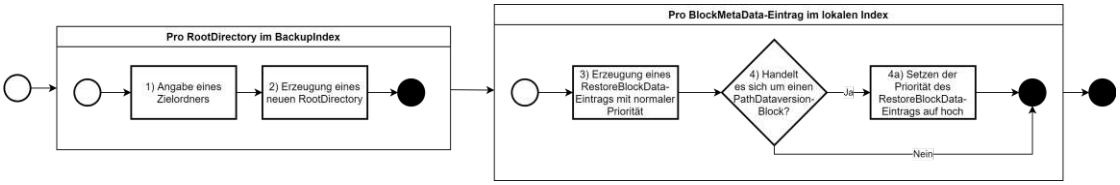

Abbildung 5.20: Start der Wiederherstellung nach Totalausfall

Metadaten zu allen Dateien dieser Sicherung. Es wird keine *DataLocation* angelegt, da nicht bekannt ist, welche Teilnehmer diese *PathDataVersion*-Blöcke sichern.

9. Der *BackupIndex* wird im lokalen Index gespeichert. Diese Informationen sind notwendig, da später vom Benutzer ein bestimmter *BackupIndex* ausgewählt wird, dessen Dateien prior wiederhergestellt werden sollen.

Nach der Wiederherstellung der Metadaten von den Cloud-Anbietern und allen Teilnehmern sind alle im Backupsystem gesicherten Sicherungsblöcke sowie die Sicherungsorte bekannt. Des Weiteren wurden die aktuellsten *BackupIndizes* ermittelt. Von unterschiedlichen Teilnehmer und dem Cloud-Speicher können verschiedene aktuellste *BackupIndex*-Blöcke übermittelt werden, da nicht jeder Teilnehmer zwingend den aktuellsten *BackupIndex* sichert. Für den Fall, dass nicht alle Teilnehmer gleichzeitig verfügbar sind, muss die Wiederherstellung des *BackupIndex* von den Teilnehmern öfter angestoßen werden. Daher können die vorhergehenden Schritte beliebig oft ausgeführt werden. Ist der gewünschte *BackupIndex* verfügbar startet der Anwender die Datenwiederherstellung für diesen*BackupIndex*. Dabei werden initial die in *Abbildung 5.20* gezeigten Metadaten im lokalen Index abgelegt:

1. Für jede *rootDirectoryId* im gewählten *BackupIndex* wird vom Anwender ein neues Zielverzeichnis angegeben. Dieses Zielverzeichnis beinhaltet die wiederhergestellten Nutzdaten und wird zukünftig über die Mechanismen der Backupanwendung gesichert.

2. Die Zielverzeichnisse werden als *RootDirectories* im lokalen Index abgelegt.

3. Für jeden *BlockMetaData*-Eintrag im lokalen Index wird ein *RestoreBlockData*-Eintrag mit normaler Priorität angelegt.

4. Es wird pro *RestoreBlockData*-Eintrag überprüft, ob dessen Block-ID in den *PathDataVersion*-Blöcken des gewählten *BackupIndex* vorkommt.

   a) Wenn ja: Der *RestoreBlockData*-Eintrag erhält eine hohe Priorität. Solange es *RestoreBlockData*-Einträge mit hoher Priorität gibt werden andere Einträge nicht verarbeitet. Dies dient dazu, dass die Daten der letzten Sicherung möglichst rasch wiederhergestellt werden.

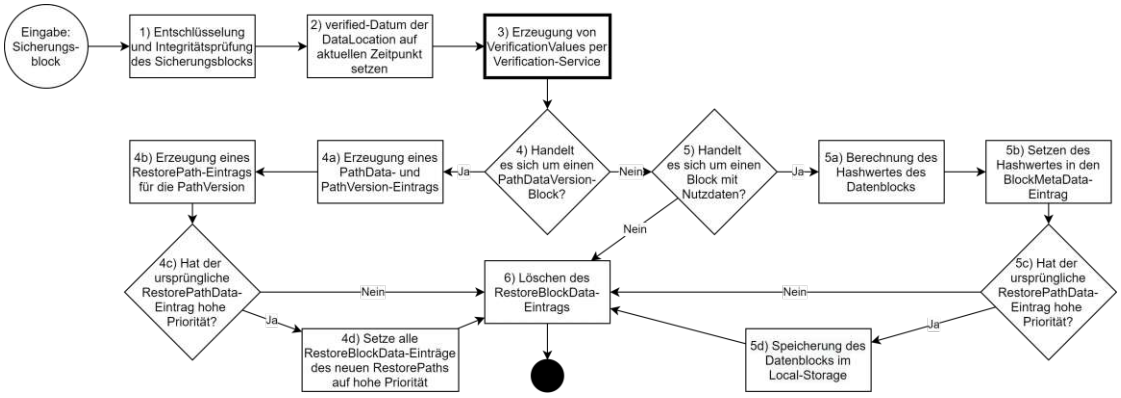

Abbildung 5.21: Verarbeitung von wiederhergestellten Sicherungsblöcken nach Totalausfall

Nach dem Start der Wiederherstellung werden die *RestoreBlockData*-Einträge wie in *Kapitel 5.3.3.1* beschrieben verarbeitet. Sicherungsblöcke werden entweder von anderen Teilnehmern angefordert oder sind bereits im Local-Storage. Dies trifft für alle Sicherungsblöcke zu, die von einem Cloud-Anbieter heruntergeladen wurden. Bei Empfang eines Sicherungsblocks wird alternativ zu *Abbildung 5.16* eine erweiterte Verarbeitung nach *Abbildung 5.21* durchgeführt, um *PathData*- und *PathVersion*-Einträge wiederherzustellen.

1. Der Sicherungsblock wird symmetrisch mit AES im AEAD-Betriebsmodus GCM entschlüsselt (*siehe Kapitel 5.2.3.10*).

2. Das Verifizierungsdatum der *DataLocation* des Blocks wird für den Teilnehmer auf den aktuellen Zeitpunkt gesetzt.

3. Für jeden empfangenen Sicherungsblock werden per Verification-Service Verifikationswerte erzeugt (*siehe Kapitel 5.3.2.1*).

4. Wenn es sich bei dem Datenblock um einen *PathDataVersion*-Block handelt:

   a) Anhand des *PathDataVersion*-Blocks wird ein *PathData*- und *PathVersion*-Eintrag angelegt. Die *PathVersion* wird mit den entsprechenden *BlockMetaData*-Einträgen, welche die Blöcke der Dateiversion repräsentieren, verknüpft.

   b) Es wird ein *RestorePath*-Eintrag für die *PathVersion* erzeugt und mit den *RestoreBlockData*-Einträgen der Blöcke dieser Dateiversion verknüpft. Dadurch wird die Wiederherstellung der Datei wie in *Kapitel 5.3.3.1* beschrieben gestartet.

   c) Wurde der *RestoreBlockData*-Eintrag zur Wiederherstellung des *PathDataVersion*-Blocks mit hoher Priorität versehen?

   d) Wenn dies der Fall ist werden auch alle *RestoreBlockData*-Einträge von Blöcken dieser Dateiversion mit hoher Priorität versehen. Dies ermöglicht eine rasche

Anforderung der Sicherungsblöcke, die für die Wiederherstellung der Daten notwendig sind.

5. Wenn es sich bei dem Datenblock um einen Block mit Nutzdaten handelt:

   a) Es wird der Hash des Datenblocks berechnet.

   b) Im lokalen Index wird der *BlockMetaData*-Eintrag des Datenblocks um den Hashwert aktualisiert. Dadurch kann der Datenblock bei zukünftigen Datensicherungen wiedererkannt werden.

   c) Wurde der *RestoreBlockData*-Eintrag zur Wiederherstellung des Datenblocks mit hoher Priorität versehen?

   d) Ist dies der Fall handelt es sich um einen Datenblock der zur Wiederherstellung der gewünschten Sicherung notwendig ist. Der Datenblock wird im Local-Storage abgelegt. Andernfalls wird der Datenblock nach der Erweiterung des *BlockMetaData*-Eintrags verworfen.

6. Unabhängig von der Art des Blocks wird der *RestoreBlockData*-Eintrag mit der Block-ID entfernt.

Die Datei- und Blockwiederherstellung wird kontinuierlich wiederholt. Notwendige Sicherungsblöcke von wiederherzustellenden Dateien werden als *RestoreBlockData*-Eintrag mit hoher Priorität angelegt und angefordert. Alle anderen Sicherungsblöcke werden nachrangig angefordert und nach der Wiederherstellung der Metadaten verworfen. Die Datenwiederherstellung der Nutzdaten ist abgeschlossen, sobald alle *RestoreBlockData*-Einträge mit hoher Priorität abgearbeitet wurden und keine *RestorePath*-Einträge mehr existieren. Die Wiederherstellung aller Metadaten dauert bis zur Verarbeitung aller *RestoreBlockData*-Einträge.

5.3.3.3 Vorgehen bei Schlüsseltausch

Im Fall der Kompromittierung des Schlüsselpaares ist der sichere Betrieb des Backupsystems gefährdet. Die Sicherungsdaten des kompromittierten Teilnehmers können von einem Angreifer von anderen Teilnehmern angefordert und gelesen oder bei diesen Teilnehmern manipuliert werden. Anderen Teilnehmer können in diesem Fall nicht mehr unterscheiden, ob sie eigene Sicherungsdaten beim kompromittierten Teilnehmer oder bei dessen Angreifer ablegen. Im Fall der Kompromittierung muss daher das Schlüsselpaar getauscht werden.

Am System des Anwenders sind nur die aktuellen Nutzdaten sowie der lokale Index verfügbar. Sicherungsdaten alter Dateiversionen liegen als Replikate bei anderen Teilnehmern. Vor dem Tausch des Schlüsselpaares müssen daher alte, zu sichernde Datenstände wiederhergestellt werden, da diese sonst nicht mehr entschlüsselbar sind und verloren gehen.

Sobald ein Benutzer über die Kompromittierung seines Schlüsselpaares Kenntnis erlangt muss unverzüglich folgendem Ablauf gefolgt werden:

1. Der Benutzer muss mithilfe der Backupanwendung ein neues Schlüsselpaar und Zertifikat erzeugen. Diese dienen ab jetzt zur Identifikation des Benutzers. Das alte Schlüsselpaar muss vorläufig noch zur Entschlüsselung alter Sicherungen aufbewahrt werden.

2. Alle Kommunikationsteilnehmer des Benutzers müssen über den Schlüsseltausch informiert werden. Dies geschieht auf einem Weg außerhalb des Backupsystems. Jedem Teilnehmer wird dabei das neue Zertifikat des Benutzers übermittelt.

3. Jeder Teilnehmer muss unverzüglich das Zertifikat des Benutzers durch das neue austauschen und dabei dessen Korrektheit nach *Kapitel 5.2.3.5* prüfen. Jeder Teilnehmer muss vor dem Schlüsseltausch aktiv bei dem Benutzer die Echtheit des Schlüsseltausches rückfragen. Aufgrund der Bekanntheit der Teilnehmer ist diese Rückfrage über ein Telefonat oder ein persönliches Gespräch möglich. Alternativ kann der betroffene Benutzer dies auch proaktiv per Telefon oder persönlich bestätigen.

4. Nach dem Zertifikatstausch akzeptieren die anderen Teilnehmer nur mehr eine Authentifizierung über das neue Schlüsselpaar des Benutzers. Der Angreifer kann sich gegenüber anderen Teilnehmern nicht mehr mit dem dem alten Schlüsselpaar des Benutzers authentisieren und ein Verbindungsaufbau wird abgelehnt.

5. Der kompromittierte Benutzer startet die Backupanwendung mit beiden Schlüsselpaaren. Das neue Zertifikat und der zugehörige Private-Key werden zur Authentisierung gegenüber anderen Teilnehmern verwendet. Das alte Schlüsselpaar wird für alle anderen kryptografischen Verfahren wie die Entschlüsselung des lokalen Index verwendet.

6. Der Benutzer wählt jene Sicherungsstände aus die nach einem abgeschlossenen Schlüsseltausch nicht verloren gehen dürfen. Diese werden wie in *Kapitel 5.3.3.1* beschrieben wiederhergestellt und müssen vom Benutzer manuell aufbewahrt werden. Sie werden zukünftig nicht mehr von der Backupanwendung verwaltet.

7. Jeder andere Teilnehmer wird über den Abschluss der Datenwiederherstellung informiert. Diese löschen alle gespeicherten Sicherungsdaten des kompromittierten Benutzers.

8. Der kompromittierte Benutzer löscht den lokalen Index und das kompromittierte Schlüsselpaar.

9. Die Backupanwendung wird mit dem neuen Schlüsselpaar und Zertifikat initialisiert. Alle bisherigen Teilnehmer und gesicherten Verzeichnisse werden neu konfiguriert. Die Backupanwendung ist wieder einsatzbereit. Ab diesem Zeitpunkt kann eine Sicherung nach *Kapitel 5.3.1* wieder durchgeführt werden.

KAPITEL 6

Evaluierung der prototypischen Implementierung

In diesem Kapitel wird der Proof-of-Concept des Backupsystems verifiziert. In *Kapitel 6.1* werden die Hauptanwendungsfälle aus *Kapitel 5.3* mit einem Testdaten-Set erprobt. *Kapitel 6.2* schlüsselt die Anforderungen an das Backupsystem und die identifizierten Maßnahmen zur Erreichung der Sicherheitsanforderungen auf. Je Anforderung bzw. Maßnahme wird die Umsetzung innerhalb der Proof-of-Concept-Implementierung erläutert und gegebenenfalls auf vorhergehende Kapitel referenziert.

6.1 Verifikation der Proof-of-Concept-Implementierung

In diesem Kapitel werden die Anforderungen der Backuplösung anhand eines Fallbeispiels getestet. Es wird ein Backupsystem mit sechs Teilnehmern aufgebaut. Mit diesen Teilnehmern werden die Datensicherung, -verifizierung und -wiederherstellung erprobt. Der Testlauf wird dokumentiert und das Ergebnis der Wiederherstellung verifiziert.

6.1.1 Testaufbau und Testdaten

Als Testsystem dient ein Computer mit Intel Core i5-2500K CPU mit 3.30GHz, 8 GB Arbeitsspeicher und Windows 10 Pro (64 Bit), Version 21H1. Für das Backupsystem werden lokal die Teilnehmer T1 bis T6 angelegt. Für jeden Teilnehmer wird eine eigene Instanz der Backupanwendung gestartet. Für den Teilnehmer T1 wird die Anbindung an Nextcloud Files und Google Drive konfiguriert. Die Teilnehmer T2 bis T5 dienen als Sicherungsziele und sichern selbst keine eigenen Daten. Der Teilnehmer T1 verwendet das Testdatenset und sichert dieses bei den Teilnehmern T2 bis T5. Der Teilnehmer T6 wird nicht als Sicherungsziel konfiguriert. Das bedeutet der Teilnehmer T1 darf bei T6 keine Daten sichern.

Als Testdaten dienen eine Ordnerstruktur mit Unterordnern, bestehend aus Bildern, Word-, Excel- und PDF-Dokumenten sowie Videos. Das Testdatenset besteht aus 32 Ordnern und 196 Dateien. Die Gesamtgröße des Datensets beträgt ungefähr 2,5 GB. Die kleinste Datei hat eine Größe von 1.43 KB, die größte 1.77 GB.

6.1.2 Geplanter Testlauf

Der Testlauf besteht aus folgenden Schritten:

1. Teilnehmer 1 wird als einziger Teilnehmer gestartet.
2. Teilnehmer T1 führt eine Sicherung des lokalen Datenverzeichnisses durch. Bereits während der Datensicherung werden die Sicherungsblöcke kontinuierlich zu den Cloud-Anbietern hochgeladen.
3. Die Teilnehmer T2, T3 und T6 werden gestartet und eine Datenverteilung durch T1 durchgeführt. Die Sicherungsdaten werden bei den Teilnehmern T2 und T3 abgelegt. Da der Teilnehmer T6 nicht als Sicherungsziel konfiguriert wurde dürfen bei diesem Teilnehmer keine Daten abgelegt werden. Die Sicherungsblöcke bleiben lokal und am Cloud-Anbieter erhalten, da es nicht möglich ist drei Replikate herzustellen.
4. Die Teilnehmer T4 und T5 werden gestartet und eine Datenverteilung durch T1 durchgeführt. Die Backuplösung stellt mindestens drei Replikate pro Sicherungsblock her und löscht danach die Sicherungsdaten bei Teilnehmer T1 und die Sicherungsdaten am Cloud-Speicher.
5. Ein Sicherungsblock bei Teilnehmer T2 wird manuell verändert. Ein zweiter Sicherungsblock bei T2 wird gelöscht.
6. Die Datenverifikation wird von Teilnehmer T1 angestoßen. Diese erkennt den manipulierten und fehlenden Block bei Teilnehmer T2, fordert die Blöcke von anderen Teilnehmern an und verteilt sie neu im Backupsystem.
7. Die lokalen Metadaten von Teilnehmer T1 werden gelöscht und ein leeres Verzeichnis als neues Datenverzeichnis gewählt um einen Totalausfall der Festplatte zu simulieren.
8. Der Teilnehmer T1 führt eine Wiederherstellung der Nutzdaten für das letzte Sicherungsdatum aus. Nach Angabe des Wiederherstellungsverzeichnisses werden alle Nutzdaten wiederhergestellt.

Der Upload von Sicherungsdaten sowie deren öffentliche Freigabe und die Ermittlung eines Download-Links funktioniert sowohl bei Nextcloud Files als auch bei Google Drive wie erwartet. Bei den ersten Testläufen wurde jedoch festgestellt, dass der Download von Sicherungsdaten von Google Drive über die öffentliche URL wie `https://drive.google.com/`

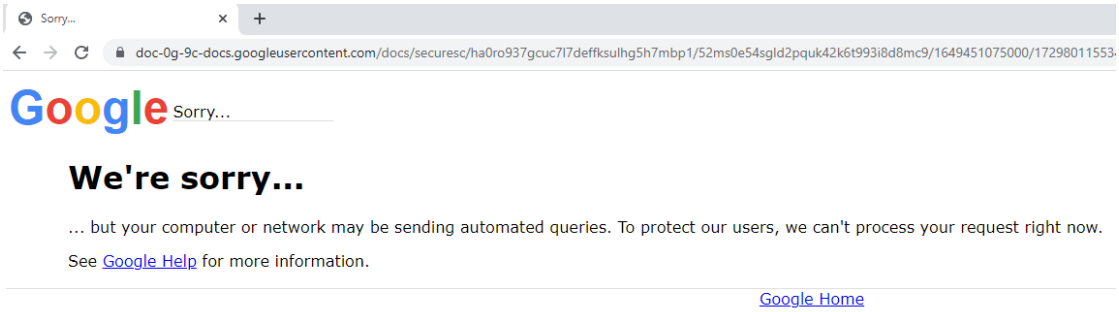

Abbildung 6.1: Google Drive Fehlermeldung

`uc?id=15LBLrTZ0CKjucc9E0qFm76PWB9A7WVHB&export=download` nach einigen Aufrufen mit einem HTTP-Response 403 Forbidden (*siehe Listing 6.1*) beantwortet wird. Der manuelle Aufruf einer entsprechenden Download-URL führt zu einer Fehlerseite (*siehe Abbildung 6.1*). Die Google-Infrastruktur erkennt in den Downloads automatische Abfragen und blockiert zur Sicherheit alle entsprechenden Downloads. Nach einigen Testläufen wurde ermittelt, dass die Geschwindigkeit der Blockierung mit der Anzahl und zeitlichen Nähe der Abfragen in Zusammenhang steht. Nach einem halben Tag sind die entsprechenden URLs wieder erreichbar und erlauben einen Download der Sicherungsdaten. Eine genaue Analyse ist jedoch nicht im Umfang dieser Arbeit enthalten und wäre eine Möglichkeit, die Erkenntnisse dieser Arbeit weiter zu untersuchen.

Listing 6.1: Google Drive HTTP 403

```
java.io.IOException: Server returned HTTP response code: 403 for URL: https://doc-00-9c-
    ↪ docs.googleusercontent.com/docs/securesc/ha0ro937gcuc7l7deffksulhg5h7mbp1/
    ↪ mkagkbt3l7bg54pdtnlfv0g8cd568q95/1649484750000/17298011553413984708/*/10
    ↪ Fjrv2svDVCIaRksx7JCtHCMC6YvLIUC?e=download
```

6.1.3 Testlauf mit Nextcloud Files

Aufgrund der in *Kapitel 6.1.2* beschriebenen Limitierungen mit Google Drive wird der Testlauf nur mit der Nextcloud Files-Anbindung durchgeführt.

6.1.3.1 Datensicherung

Zu Beginn des Testlaufs wird der Teilnehmer T1 gestartet. Die Teilnehmer T2 bis T6 werden noch nicht gestartet, um die Nichtverfügbarkeit dieser Teilnehmer zu diesem Zeitpunkt zu simulieren. Es wird eine Sicherung für das Datenverzeichnis durchgeführt. Durch den Start der Sicherung werden für alle Dateien Blöcke mit 500KB (*siehe Kapitel 5.1*) im Local-Storage in verschlüsselter Form abgelegt. Als Beispiel wird der Block `000E7F42-37D7-4179-9BA2-1C11B82A61C3` in *Abbildung 6.2* gezeigt. Die gezeigte tatsächliche Blockgröße von 501KB ergibt sich, da jeder Sicherungsblock zusätzlich zu den

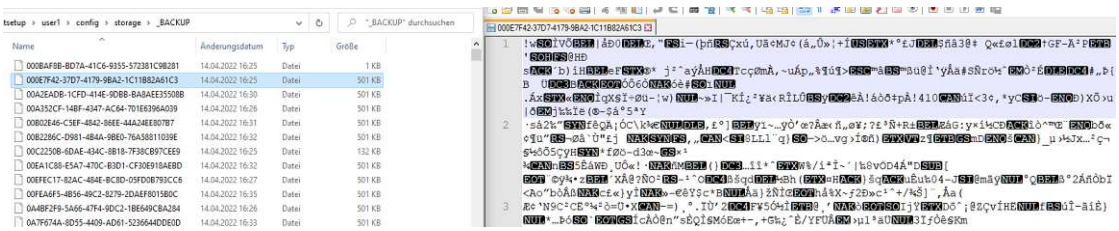

Abbildung 6.2: Verschlüsselte Blöcke im Local-Storage von Teilnehmer T1

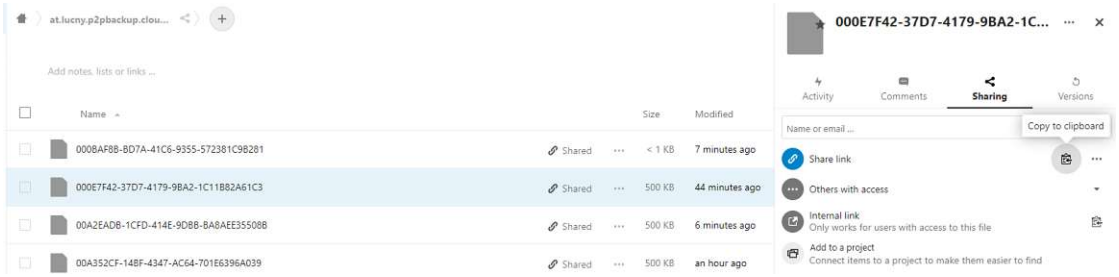

Abbildung 6.3: Verschlüsselte Blöcke im Nextcloud Files Cloud-Speicher

Nutzdaten den zur Verschlüsselung verwendeten IV und den GCM-Authentisierungstag enthält.

Während der Sicherung werden die Sicherungsblöcke bereits zum Cloud-Anbieter hochgeladen und ein Freigabelink erzeugt. Der Block `000E7F42-37D7-4179-9BA2-1C11B82A61C3` ist über eine öffentliche URL[1] erreichbar (*siehe Abbildung 6.3*).

Während der laufenden Sicherung bleiben die Sicherungsblöcke im Local-Storage gespeichert. Der kontinuierliche Datenupload ist in *Listing 6.2* ersichtlich. Die Verifikation und die Verteilung der Sicherungsblöcke wird zum aktuellen Zeitpunkt nicht durchgeführt, da keine anderen Teilnehmer verfügbar sind.

Listing 6.2: Upload der Sicherungsdaten während der Durchführung einer Sicherung

```
16:24:28.006 : backup of path C:\Entwicklung\Testsetup\user1\data
16:24:29.099 : prepare to upload up to 7 blocks
16:24:29.114 : processed 0/7 cloud-upload-entries
16:24:32.957 : processed 7 cloud-upload-entries total
16:24:33.708 : prepare to upload up to 36 blocks
16:24:33.715 : processed 0/36 cloud-upload-entries
16:24:38.813 : processed 10/68 cloud-upload-entries
...
16:25:40.176 : processed 140/565 cloud-upload-entries
16:25:40.221 : processed 100 files
16:25:44.999 : processed 150/605 cloud-upload-entries
```

[1] `https://nextcloud.lucny.at/index.php/s/2NneRcRXbRq8bsw` - besucht am 14.04.2022

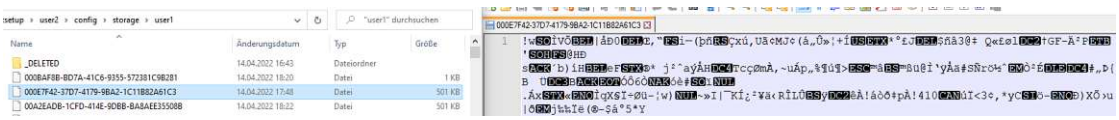

Abbildung 6.4: Sicherungsblöcke im Local-Storage von Teilnehmer T2

Nach erfolgtem Upload zum Cloud-Anbieter werden alle Teilnehmer T2, T3 und T6 gestartet und anschließend mit der Verteilung begonnen. Die Teilnehmer T2 und T3 erhalten alle Sicherungsblöcke. Als Beispiel wird wieder der Block `00AA8EC9-802C-44E7-8768-8DD9CCCD7335` bei Teilnehmer T2 in *Abbildung 6.4* gezeigt.

Mit dem Start der Teilnehmer T4 und T5 und einer erneuten Verteilung werden pro Sicherungsblock mindestens drei Replikate hergestellt sowie alle Sicherungsblöcke im Local-Storage und am Cloud-Anbieter gelöscht. Aufgrund von Verzögerungen wie einer verspäteten Bestätigung einer Sicherung durch einen Teilnehmer kann es vorkommen, dass von einem Sicherungsblock mehr als drei Replikate hergestellt werden (*siehe Listing 6.3*).

Listing 6.3: Statistik der verifizierten Replikate von Teilnehmer T1

```
0 verified replicas: 0 blocks
1 verified replicas: 0 blocks
2 verified replicas: 3598 blocks
3 verified replicas: 1919 blocks
4 verified replicas: 74 blocks
```

6.1.3.2 Datenverifikation

Für die Datenverifikation werden Änderungen an den Sicherungsdaten bei Teilnehmer T2 durchgeführt. Der Block `000E7F42-37D7-4179-9BA2-1C11B82A61C3` wird manuell verändert um die Manipulation des Blocks zu simulieren. Der Block `00A2EADB-1CFD-414E-9DBB-BA8AEE35508B` wird gelöscht. Ein Zurücksetzen der Verifizierungsdaten aller Replikate von Teilnehmer T2 führt dazu, dass alle Sicherungsblöcke des Teilnehmers überprüft werden. In *Listing 6.4* wird die Ausgabe der Backupanwendung gezeigt. Die Verifikation der beiden Sicherungsblöcke schlägt fehlt. Da die Sicherungsblöcke lokal nicht mehr verfügbar sind werden sie von anderen Teilnehmern angefordert und anschließend zur Verteilung zum Cloud-Speicher hochgeladen.

Listing 6.4: Verifikation der Sicherungsblöcke von Teilnehmer T2

```
19:02:29.901 : sent verification request to 6959/6959 blocks. stop and continue with next run
19:02:49.786 : user user2 failed the verification of the block 000E7F42-37D7-4179-9BA2-1
    ↪ C11B82A61C3
19:03:21.161 : unable to verify backup block 00A2EADB-1CFD-414E-9DBB-
    ↪ BA8AEE35508B on user user2: failure was BLOCK_MISSING
19:03:34.795 : prepare to upload up to 2 blocks
19:03:34.802 : processed 0/2 cloud-upload-entries
```

```
19:03:36.825 : processed 2 cloud-upload-entries total
```

6.1.3.3 Datenwiederherstellung

Der Testlauf der Datenwiederherstellung sieht die Simulation eines Festplattenausfalls vor. Dazu wird der lokale Index des Teilnehmers T1 gelöscht. Die Nutzdaten werden in einem leeren Verzeichnis wiederhergestellt. Dadurch kann nach der Wiederherstellung ein Vergleich zwischen den ursprünglichen und den wiederhergestellten Dateien durchgeführt werden.

Zuerst wird der letzte Backup-Index mithilfe des Cloud-Speichers und der Teilnehmer T2 bis T6 ermittelt. In *Listing 6.5* ist erkennbar, dass sowohl der Cloud-Speicher als auch der Teilnehmer T6 keine Sicherungsdaten gespeichert hatten. Von den Teilnehmern T2 bis T5 wurden die IDs aller gesicherten Blöcke sowie ein Backup-Index vom `2022-04-14T14:35:49.318` mit ID `4162605B-3DA3-464D-98E1-15FDD8E17B96` wiederhergestellt.

Listing 6.5: Wiederherstellung des Backup-Index und der IDs gesicherter Blöcke

```
21:39:31.102 : application is now in recovery-mode
21:39:32.037 : recovering 0 blocks from cloud-storage a.l.p.c.n.s.NextcloudStorageServiceImpl
21:39:32.040 : recovered 0 blocks total from cloud-storage a.l.p.c.n.s.
    ↪ NextcloudStorageServiceImpl
21:39:35.533 : application is now in recovery-mode
21:39:35.770 : user user6 saves 0 blocks - restore block-meta-data and add user user6 as
    ↪ data-location
21:39:37.549 : user user2 saves 5591 blocks - restore block-meta-data and add user user2 as
    ↪ data-location
21:39:55.729 : recover backup-index 1649946949318/2022-04-14T14:35:49.318 from user
    ↪ user2
21:39:55.906 : recovered backup-index 1649946949318/2022-04-14T14:35:49.318 from user
    ↪ user2
...
21:40:33.149 : recover backup-index 1649946949318/2022-04-14T14:35:49.318 from user
    ↪ user3
21:40:33.171 : recovered backup-index 1649946949318/2022-04-14T14:35:49.318 from user
    ↪ user3
```

Die Datenwiederherstellung für den Backup-Index mit ID `4162605B-3DA3-464D-98E1-15FDD8E17B96` wird gestartet. Von der Backupanwendung werden kontinuierlich Sicherungsblöcke von anderen Teilnehmern angefordert, entschlüsselt, auf ihre Integrität geprüft und im Local-Storage abgelegt. Sobald für eine wiederherzustellende Datei alle Blöcke verfügbar sind wird diese im Zielordner wiederhergestellt (*siehe Listing 6.6*).

Listing 6.6: Wiederherstellung von Dateien

```
21:49:22.332 : requested 100 blocks for restore
...
```

```
21:49:27.828 : requested 1000 blocks total for restore
21:50:42.063 : start to recover blocks and files
21:50:42.071 : trying to restore 21 files
21:50:42.106 : restored file C:\...\Studienbestaetigung_0725836_066 937_2014W.pdf
...
21:50:42.364 : restored file C:\...\Studienbestaetigung_0725836_066 937_2013W.pdf
21:50:42.379 : restored all currently available files
...
21:58:54.594 : restored file C:\...\IMG_20210123_183027086.jpg
21:58:54.595 : restored all currently available files
21:59:12.428 : all blocks recovered, stop recovery−mode
```

Dieser Prozess wird solange durchlaufen, bis alle Dateien der gewählten Sicherung wiederhergestellt und jeder Block von mindestens einem Teilnehmer angefordert und verifiziert wurde. Nach Wiederherstellung aller Dateien der Sicherung wird der Inhalt des Wiederherstellungsordners mithilfe des ursprünglichen Datenordners geprüft. Dazu wird mittels PowerShell ein SHA-512-Hash jeder Datei erzeugt und mit ihrem Equivalent im anderen Ordner verglichen. Der Vergleich findet keine Unterschiede zwischen beiden Verzeichnissen.

6.2 Verifikation der Systemanforderungen und Sicherheitsmaßnahmen

Folgend sind alle Anforderungen an das Backupsystem sowie die Umsetzung im Rahmen der Proof-of-Concept-Implementierung aufgelistet. Einige Anforderungen werden im Rahmen von identifizierten Maßnahmen umgesetzt.

AF-1 **Datensicherung** : *Kapitel 5.3.1* beschreibt den Ablauf der Datensicherung.

AF-2 **Erkennung und Sicherung von Änderungen** : *Kapitel 5.3.1.1* - Änderungen an Dateien werden anhand eines SHA-256-Hashes erkannt. Nur für geänderte Dateien wird eine Sicherung durchgeführt. Eine Datei wird in Blöcke mit 500KB eingeteilt. Je Block wird anhand eines SHA-256-Hashes erkannt, ob dieser Block bereits im Backupsystem gesichert wurde. Nur neue Blöcke werden von der Backupanwendung gesichert. *Kapitel 5.1.4.2* - Der lokale Index enthält die notwendigen Metadaten zur Änderungserkennung in *PathVersion*- und *BlockMetaData*-Einträgen in einer SQL-Datenbank.

AF-3 **Off-Site Speicherung der Backups an einem anderen geografischen Standort** : *Kapitel 5.1* - Sicherungsdaten werden bei anderen Teilnehmern abgelegt. *Kapitel 5.1.2.1* - Andere Teilnehmer sind per IP- oder DNS-Adresse per Internet erreichbar und können sich daher an einem beliebigen Ort befinden. Dies ermöglicht die Off-Site-Speicherung. *Kapitel 5.3.1.3* beschreibt die Verteilung von Sicherungsdaten an andere Teilnehmer.

AF-4 **Vertrauliche Speicherung der Backups** : Die vertrauliche Sicherung eines Backups wird durch MA-17 Verschlüsselung der Sicherungsdaten mit derzeit als sicher geltenden symmetrischem Verfahren sichergestellt.

AF-5 **Datenwiederherstellung** : *Kapitel 5.3.3.1* beschreibt den Ablauf einer Datenwiederherstellung für einen bestimmten Zeitpunkt. *Kapitel 5.3.3.2* beschreibt den Ablauf einer Datenwiederherstellung für den aktuellsten Zeitpunkt nach einem Totalausfall des Systems des Benutzers.

AF-6 **Erkennung von unvollständigen oder manipulierten Sicherungen** : Fehlerhafte oder manipulierte Sicherungen werden mittels MA-16 Regelmäßige Verifikation der Sicherungsdaten bei anderen Teilnehmern erkannt.

AF-7 **Kompensation von fehlenden oder fehlerhaften Backupdaten** : Fehlende oder fehlerhafte Sicherungen werden durch Einträge im lokalen Index (*siehe Kapitel 5.1.4.2*) erkennbar und werden durch MA-15 Mehrere Replikate der Sicherungsdaten kompensiert, da Sicherungsdaten mehrfach gespeichert werden. Die Backupanwendung fordert von einem Teilnehmer die Sicherungsdaten an und verteilt sie an weitere Teilnehmer bis die erforderliche Verfügbarkeit erreicht wird.

AF-8 **Vertrauliche Kommunikation zwischen Teilnehmern** : Die vertrauliche Kommunikation wird durch MA-9 Verschlüsselung der Kommunikation mit anderen Teilnehmern sichergestellt.

AF-9 **Nutzung des freien Speicherplatzes anderer Teilnehmer zur Datensicherung** : *Kapitel 5.1* - Sicherungsdaten werden bei anderen Teilnehmern abgelegt. *Kapitel 5.1.2.2* - Sicherungsdaten werden bei Cloud-Anbietern nur temporär abgelegt und nach der Verteilung an andere Teilnehmer wieder gelöscht.

AF-10 **Bekannte Teilnehmer innerhalb des Backupsystems** : *Kapitel 5.1.2.1* - Jeder Teilnehmer wird über ein Zertifikat identifiziert. Das Zertifikat jedes Teilnehmers, mit dem kommuniziert werden soll, muss in der Backupanwendung konfiguriert werden. Zur Sicherstellung der Korrektheit des Zertifikats wird MA-13 Bestätigung des öffentlichen Schlüssels anderer Teilnehmer eingesetzt.

AF-11 **Technische Möglichkeit zur Authentifizierung der Teilnehmer** : *Kapitel 5.2.3.3* - Zur Identifikation der Teilnehmer werden Zertifikate und der zugehörige Private-Key verwendet. Bei dem zugrunde liegenden Public- und Private-Key handelt es sich um ein RSA-Schlüsselpaar mit 3072 Bit.

AF-12 **Kommunikation der Teilnehmer ohne zentralen Server** : *Kapitel 5.1.2* beschreibt die direkte Kommunikation der Teilnehmer. Die Cloud-Anbieter werden nur temporär zur Speicherung verwendet.

AF-13 **Minimierung der Sicherungszeit** : *Kapitel 5.1.2.2* beschreibt den sofortigen Upload von Sicherungsdaten zu einem Cloud-Anbieter. Die initiale Datensicherung wird sofort durchgeführt. Ebenso wird die Upload-Bandbreite pro Sicherungsblock

nur einmal belastet. Andere Teilnehmer sichern Sicherungsblöcke vom Cloud-Anbieter, können ihre Download-Bandbreite nutzen und den Download weiterführen, selbst wenn das Quellsystem temporär nicht erreichbar ist.

AF-14 **Sichere Standardwerte bei kryptographischen Verfahren** : Die Nutzung von als sicher erachteten kryptografischen Verfahren wird durch die Maßnahmen MA-6 Verwendung von als derzeit sicher geltenden kryptografischen Verfahren und MA-11 Geeignete Erzeugung von Schlüsseln in der Backuplösung erreicht.

AF-15 **Zugang zu Quellcode der Anwendung** : Die Funktionsweise des Backupsystems und die eingesetzten Techniken sind durch MA-4 Verfügbarkeit des Sourcecodes in einem öffentlich verfügbaren Repository verifizierbar.

Folgend werden alle erarbeiteten Maßnahmen sowie ihre Auswirkungen bzw. Umsetzung im Backupsystem aufgelistet. Erkenntnisse zur Nichterfüllbarkeit von Maßnahmen sind angeführt.

MA-1 **Review des Konzepts der Backuplösung** : Diese Diplomarbeit wird während der Verfassung regelmäßig durch Kollegen aus dem universitären Umfeld geprüft. Das Konzept wird im Rahmen dessen ebenfalls mehrfach einem Review unterzogen. Bei Abschluss der Diplomarbeit wird ein Peer-Review durchgeführt.

MA-2 **Einsatz von Analysetools zur Erkennung von Software-Konzeptionsfehlern und -Schwachstellen** : *Kapitel 5.2.4.1* - Die eingesetzten Bibliotheken der Backuplösung werden durch OWASP auf Schwachstellen überprüft. Der OWASP-Check meldet Sicherheitslücken im Nextcloud-Client der Backuplösung. Dabei handelt es sich jedoch False-Positives, da sich die Sicherheitslücken auf den Quellcode des Nextcloud-Servers beziehen. Des Weiteren werden Sicherheitslücken im genutzten Spring-Framework gemeldet. Diese sind ebenfalls False-Positives, da diese für die eingesetzte Spring-Shell-Bibliothek gemeldet werden. Die entsprechenden Meldungen wurden als False-Positives markiert und zukünftig unterdrückt[2]. Der Nextcloud-Server wurde auf Version 20.0.14 upgedatet. Diese ist von den gemeldeten Schwachstellen nicht mehr betroffen. Durch den OWASP-Check wurden während der gesamten Entwicklungszeit Sicherheitslücken in den verwendeten Bibliotheken identifiziert. Ein Upgrade der entsprechenden Bibliotheken konnte jede Sicherheitslücke beheben. Die letzte Überprüfung wurde am 14.04.2022 durchgeführt.

Kapitel 5.2.4.2 - Der Quellcode der Anwendung wird durch das statische Analysetool SonarQube untersucht. Der Großteil der Fehler mit der Klassifizierung *Blocker*, *Critical* und *Major* wurden behoben. Nicht behobene Fehler wurden entsprechend kommentiert und exportiert[3]. Die letzte Überprüfung wurde am 14.04.2022 durchgeführt.

[2] `https://raw.githubusercontent.com/lucnygr/p2pbackup_public/development/p2pbackup/owasp-suppression.xml` - besucht am 14.04.2022

[3] `https://raw.githubusercontent.com/lucnygr/p2pbackup_public/development/docker-p2pbackup/sonar-issues.json` - besucht am 14.04.2022

MA-3 **Empfohlener Einsatz von Sicherheitsmaßnahmen gegen Schadprogramme** : Dem Anwender wird der Einsatz von Sicherheitsmaßnahmen gegen Schadprogramme empfohlen. Dies wird von der Backuplösung nicht überprüft.

MA-4 **Verfügbarkeit des Sourcecodes in einem öffentlich verfügbaren Repository** : Der Quellcode der Backupanwendung ist unter `https://github.com/lucnygr/p2pbackup_public.git` verfügbar.

MA-5 **Verifikation der Applikationsdateien per Hash-Wert** : *Kapitel 5.2.3.1*: Zur Verifikation der Applikationsdateien wird mit dem Quellcode bei jeder Release ein SHA-512-Hash[4] zur Verfügung gestellt.

MA-6 **Verwendung von als derzeit sicher geltenden kryptografischen Verfahren** : *Kapitel 5.2.3* - Bei der Auswahl von kryptografischen Verfahren wird auf Empfehlungen des BSI und anderer Institute zurückgegriffen. Die Empfehlungen des BSI gelten bis Ende 2027.

MA-7 **Verschlüsselung der Konfigurations- und Metadaten** : *Kapitel 5.2.3.8* - Lokale Konfigurations- und Metadaten werden symmetrisch per AES im CBC-Modus mit PKCS5-Padding verschlüsselt. Die Verschlüsselung der HyperSQL-DB entspricht jedoch nicht den Empfehlungen des BSI, da der verwendete IV nicht getauscht und bei jeder Persistierung der Datenbank verwendet wird. Es wurden OpenSource-Alternativen für eingebettete Datenbanken unter Java untersucht. Die H2-DB unterstützt AES-128-Verschlüsselung, die Angabe eines bestimmten Betriebsmodus oder Änderung des IV ist aber nicht möglich [33]. Derby unterstützt AES mit Schlüsseln bis zu 256 Bit im CBC-Modus und ermöglicht die Änderung des Schlüssels, erlaubt dafür aber keine Angabe eines Paddings zur Erfüllung der Empfehlung [81]. Keine dieser OpenSource-Datenbanken unterstützt die Verschlüsselung nach BSI-Empfehlung vollständig. Die korrekte Implementierung würde den Umfang dieser Diplomarbeit überschreiten, daher ist die vollständige Umsetzung dieser Maßnahme nicht möglich.

MA-8 **Integritätsprüfung der Konfigurations- und Metadaten** : *Kapitel 5.2.3.9* beschreibt die Integritätssicherung der lokalen Metadaten per HMAC. Dieser wird manuell aus den Datenbankdateien der HyperSQL-DB errechnet und geprüft. Die Verwendung eines Betriebsmodus wie GCM mit eingebauter Integritätssicherung wird von der HyperSQL-DB nicht unterstützt. Andere Datenbanken wie die H2-DB [33] und Derby [81] unterstützen diese ebenfalls nicht, daher wurde die Berechnung des HMAC als Alternative gewählt.

MA-9 **Verschlüsselung der Kommunikation mit anderen Teilnehmern** : *Kapitel 5.2.3.6* - Die Kommunikation zwischen Teilnehmern ist mittels TLS 1.3 abgesichert.

[4] `https://github.com/lucnygr/p2pbackup_public/releases/tag/1.0.1` - besucht am 24.04.2022

MA-10 **Gesicherte Ablage von kryptografischen Schlüsseln** : *Kapitel 5.2.3.4* - Das Schlüsselpaar des Anwenders wird mittels AES-Verschlüsselung und abgeleitetem Geheimnis gespeichert.

MA-11 **Geeignete Erzeugung von Schlüsseln in der Backuplösung** : *Kapitel 5.2.3.2* - Die Erzeugung von Zufallszahlen unter Java ist vom genutzten Betriebssystem, der Java-Version und den lokalen Sicherheitseinstellungen abhängig. Die Standardkonfiguration unter Java 17 entspricht nicht den Empfehlungen. Der Zufallszahlen-Generator wird daher manuell konfiguriert. Kryptografische Schlüssel werden damit mit empfohlenen Zufallszahlen-Generatoren erzeugt.

Kapitel 5.1.2.1 - Standardmäßig wird das Public-/Private-Key-Paar eines Benutzers von der Backupanwendung erzeugt. *Kapitel 5.2.3.3* - Das Public-/Private-Key-Paar wird mit als sicher erachteten Parametern erzeugt. *Kapitel 5.2.3.7* - Symmetrische Schlüssel werden mit als sicher erachteten Parametern von der Backupanwendung erzeugt.

MA-12 **Entzug der Kommunikationsrechte von anderen Teilnehmern** : *Kapitel 5.1.2.1* - Durch die gegenseitige Authentifizierung mittels Zertifikat und Private-Key ist durch Entfernen des Zertifikats eines Teilnehmers die Unterbrechung der Kommunikation jederzeit möglich. *Kapitel 5.3.3.3* beschreibt das Vorgehen zum Tausch des eigenen Schlüsselpaares.

MA-13 **Bestätigung des öffentlichen Schlüssels anderer Teilnehmer** : *Kapitel 5.2.3.5* - Beim Hinzufügen eines Teilnehmers muss die Korrektheit des SHA-Hashes des Zertifikats vom Benutzer überprüft und bestätigt werden.

MA-14 **Erzeugung von Sicherungsdaten unabhängig von Verteilung bei anderen Teilnehmern** : *Kapitel 5.3.1.1* - Die Erzeugung von Sicherungsdaten geschieht parallel zur Verteilung. Es wird nicht darauf gewartet, dass ein Sicherungsblock verteilt wurde bevor der nächste generiert wird.

MA-15 **Mehrere Replikate der Sicherungsdaten** : *Kapitel 5.3.1.3* erläutert den Vorgang zur Verteilung von mindestens drei Replikaten von Sicherungsdaten an andere Teilnehmer.

MA-16 **Regelmäßige Verifikation der Sicherungsdaten bei anderen Teilnehmern** : *Kapitel 5.2.3.12* erläutert den Vorgang zur Verifikation von Sicherungsdaten bei anderen Teilnehmern.

MA-17 **Verschlüsselung der Sicherungsdaten mit derzeit als sicher geltenden symmetrischem Verfahren** : *Kapitel 5.2.3.10* - Sicherungsdaten werden symmetrisch mittels AES verschlüsselt.

MA-18 **Integritätsprüfung der Sicherungsdaten** : *Kapitel 5.2.3.10* - Die Integritätsprüfung von Sicherungsdaten bei Wiederherstellung erfolgt durch den Einsatz von AES im Betriebsmodus GCM.

MA-19 **Verzögerte Löschung der lokalen Sicherungsdaten** : *Kapitel 5.3.1.2* erläutert den Vorgang des Cloud-Uploads der Sicherungsdaten inkl. der verzögerten Löschung der lokalen Sicherungsdaten. *Kapitel 5.3.1.3*: Erläutert den Vorgang der Verteilung der Sicherungsdaten sowie den Zeitpunkt zur Löschung der lokalen Sicherungsdaten.

MA-20 **Integritätsprüfung der Cloud-Sicherungsdaten** : *Kapitel 5.2.3.11*: Sicherungsdaten, die sichernde Teilnehmer vom Cloud-Anbieter herunterladen, werden anhand eines HMAC auf deren Integrität geprüft.

MA-21 **Direkte Teilnehmer-Kommunikation bei fehlenden/nicht integren Daten im Cloud-Speicher** : *Kapitel 5.3.1.3* erläutert die Verteilung von Sicherungsdaten zu anderen Teilnehmern inkl. der Kommunikation bei nicht integren Sicherungsdaten. Teilnehmer kommunizieren direkt miteinander.

MA-22 **Minimale Anforderungen an den Cloud-Speicher** : Die Anforderungen an den Cloud-Speicher wurden auf ein Minimum beschränkt und in *Kapitel 5.1.2.2* dokumentiert.

MA-23 **Verwendung mehrerer Cloud-Speicher** : *Kapitel 5.1.2.2* definiert minimale Anforderungen zur Anbindung mehrerer Cloud-Speicher. *Kapitel 5.2.1*: Es werden die beiden Cloud-Anbieter Google Drive und Nextcloud Files eingesetzt. Wie in *Kapitel 6.1.1* festgestellt eignet sich die Anbindung an Google Drive in der verwendeten Form jedoch nicht für den geplanten Einsatz im Backupsystem, da Downloads über die öffentlichen URLs nach kurzer Zeit durch die Google Cloud-Infrastruktur blockiert werden.

MA-24 **Sorgfältige Auswahl des Cloud-Anbieters** : *Kapitel 5.2.1.1* - Google Drive wird durch Audits überprüft und ist ISO/IEC 27001 zertifiziert. *Kapitel 5.2.1.2* - Die Verwendung von Nextcloud Files erlaubt die Nutzung von beliebigen geprüften oder selbst betriebenen Servern.

MA-25 **Leitfaden für Tausch des privaten Schlüssels** : *Kapitel 5.3.3.3* - Für den Fall der Kompromittierung des Schlüsselpaares wurde eine Anleitung über das notwendige Vorgehen erstellt. Andere Teilnehmer müssen das Zertifikat des kompromittierten Benutzers austauschen. Im Zuge des Tausches werden relevante Sicherungsstände wiederhergestellt und müssen vom kompromittierten Benutzer in Eigenverantwortung gesichert werden.

KAPITEL 7

Zusammenfassung und Ausblick

Die Durchführung von Backups ist essentiell um sich vor Datenverlust zu schützen. Die verfügbaren Lösungen reichen von rein lokalen Systemen mit externen Festplatten zu Server-basierten Systemen mit Nutzung von Cloud-Komponenten. Die Lösungen unterschieden sich in Bezug auf Benutzerfreundlichkeit und Kosten, haben Vor- und Nachteile aufgrund ihrer Architektur und gewährleisten Schutzziele in unterschiedlichem Ausmaß. Meistens kann das eingesetzte Produkt nicht auf dessen korrekte Funktionsweise überprüft werden. Kommerzielle Produkte haben oftmals den Nachteil des Vendor-Lock-Ins. Zusätzliche Herausforderungen ergeben sich für den Schutz der Privatsphäre in Bezug auf einen Datenzugriff des Anbieters. Bei Server-basierten Lösungen spielt der Standort des Rechenzentrums und beispielsweise die Datenschutzgesetze des Landes eine Rolle.

Diese Arbeit schlägt als Alternative zu bestehenden Lösungen ein Backupsystem vor, dass auf echten sozialen Netzwerken bekannter Personen aufbaut und eine Kombination aus Peer-to-Peer- und Cloud-Komponenten nutzt. Ziel der Arbeit war die Erhebung der Anforderungen an dieses Backupsystem, eine strukturierte Sicherheitsanalyse der geplanten Lösung, um notwendige Schutzziele bestmöglich zu erfüllen sowie die Proof-of-Concept-Implementierung der Lösung. Die Anforderungsliste enthält neben offensichtlichen Funktionen wie Datensicherung und -wiederherstellung wichtige Anforderungen wie die Speicherung von Offsite-Backups oder die Minimierung der Sicherungszeit. Die Sicherheitsanalyse schlüsselt Bedrohungen an ein Backupsystem im Peer-to-Peer- und Cloud-Umfeld auf. Das Ergebnis der Sicherheitsanalyse ist eine Liste von Maßnahmen um diese Bedrohungen zu verhindern oder bestmöglich zu mitigieren.

Mit dieser Diplomarbeit sollten im Rahmen der Proof-of-Concept-Implementierung aktuelle konzeptionelle oder technologische Mittel zur Umsetzung der Maßnahmen ermittelt werden. Es wird gezeigt, dass eine Kombination aus Peer-to-Peer- und Cloud-Komponenten genutzt werden kann, um Schwächen beider Architekturen zu verringern. Für die Wahl von kryptografischen Verfahren kann auf Empfehlungen von Institutionen wie dem BSI zurückgegriffen werden. Empfehlungen für kryptografische Verfahren werden jedoch nur

für die Zeit bis Ende 2027 abgegeben und müssen regelmäßig neu evaluiert und die eingesetzten kryptografischen Verfahren angepasst werden. Aus der Analyse geht hervor, dass unter Java 17 die Generierung von guten Zufallszahlen von mehreren Faktoren wie dem Betriebssystem und den lokalen Sicherheitseinstellungen abhängt und standardmäßig nicht von der Generierung von guten Zufallszahlen ausgegangen werden kann. Es wurde festgestellt, dass die eingebettete HyperSQL-Datenbank nicht nach aktuellen kryptografischen Empfehlungen des BSI verschlüsselt. Andere verschlüsselnde Open-Source-Datenbanken wie H2DB und Derby unterstützen diese Empfehlungen ebenfalls nicht. Zusätzlich unterstützen die Datenbanken keinen Betriebsmodus wie GCM mit Integritätscheck, weshalb diese Prüfung manuell durchgeführt werden muss. Zukünftige Arbeiten könnten dafür Lösungen und Alternativen untersuchen. Dies kann die korrekte Weiterentwicklung der Verschlüsselung bei bestehenden Datenbanken oder die Nutzung eines alternativen Datenspeichers wie einer NoSQL-Datenbank sein. Alle weiteren Maßnahmen können mit dem gewählten Technologiestack und dem gewählten Design umgesetzt werden.

Die Ergebnisse dieser Diplomarbeit können als Ausgangspunkt für zukünftige kombinierte Sicherungslösungen verwendet werden. Je nach Zieldesign lassen sich die getroffenen Maßnahmen für gleiche oder ähnliche Bedrohungen anwenden. Die Lösungsidee, eingesetzte Konzepte und Empfehlungen für kryptografische Verfahren lassen sich an zukünftige Entwicklungen anpassen und wiederverwenden.

Der Proof-of-Concept des Backupsystems ermöglicht eine Weiterentwicklung bzw. Forschung in mehreren Bereichen. Die Einteilung in Datenblöcke fixer Größe kann durch eine dynamische Wahl der Blockgröße ersetzt werden. Dateien werden anhand ihres Inhalts in Blöcke variabler Größe geteilt [49, 87], wodurch eine Verschiebung von Blöcken innerhalb einer Datei erkennbar wird und nur geänderte Blöcke gesichert werden.

Für jede Datei wird ein Block mit Metadaten erzeugt und ebenfalls gesichert. Das Verhältnis von Nutzdaten zu Metadaten wurde nicht analysiert. Zukünftige Arbeiten können daran anknüpfen und untersuchen, wie sich eine Änderung der Blockgröße auf die Anzahl der Metadaten-Blöcke und die Sicherungsdaten auswirkt. Größere Blöcke führen zu geringeren Metadaten pro Datei, dafür muss bei einer Datenänderung ein größerer Block gesichert werden, was die gesamte gesicherte Datenmenge erhöht.

Die aktuelle Proof-of-Concept-Implementierung eignet sich nicht zur Zusammenarbeit mit Google Drive, da dessen Infrastruktur aufeinanderfolgende Aufrufe innerhalb kurzer Zeit als verdächtig bewertet und weitere Zugriffe blockiert. Es besteht daher Bedarf bei der Integration mit weiteren Cloud-Anbietern. Deren Anbindung ist zu implementieren und nach dem beschriebenen Testszenario zu testen. Dabei kann festgestellt werden, ob bei weiteren Cloud-Anbietern ein ähnliches Verhalten wie bei Google Drive oder andere Problemfälle eintreten. Gegebenenfalls sind Lösungsmöglichkeiten für die Integration der Cloud-Anbieter zu finden.

Die verwendeten kryptografischen Verfahren wurden aufgrund der Empfehlung des BSI zur Erfüllung der Maßnahmen und Schutzziele ausgewählt. Performance- oder Effizienz-

Überlegungen wurden nicht durchgeführt. In Hinblick auf Laufzeit und Speicherverbrauch kann die Nutzung verschiedener Hash-Algorithmen oder Verschlüsselungsverfahren untersucht werden.

Die konzeptionierte Backuplösung zeigt, dass Datensicherung mit einem kombinierten Ansatz zur Nutzung von Peer-to-Peer- und Cloud-Komponenten möglich ist. Die Kombination adressiert Probleme aus beiden Bereichen und bedient sich zusätzlich sozialer Netzwerke zur Unterstützung der Hauptanwendungsfälle. Die Lösung verwendet aus aktueller Sicht kryptografisch sichere Verfahren, die aufgrund der Verfügbarkeit des Quellcodes kontrolliert und angepasst werden können. Dies führt zu einer Backuplösung, die mit Änderungen bei kryptografischen Empfehlungen, bei Teilnehmern des Backupsystems oder bei den genutzten Cloud-Anbietern umgehen kann.

Abbildungsverzeichnis

Tabellenverzeichnis

Akronyme

AD Associated Data. 17

AE Authenticated Encryption. 17, 18

AEAD Authenticated Encryption with Associated Data. 17

AES Advanced Encryption Standard. 15, 18, 21, 52, 54, 77

ANSSI Agence nationale de la sécurité des systèmes d'information. 14, 18

BSI Bundesamt für Sicherheit in der Informationstechnik. 5, 9–12, 14, 15, 18, 20, 22, 40, 41, 43, 45, 46, 76–78

CBC Cipher Block Chaining. 16, 18, 23, 77

CBC-CS Cipher Block Chaining - Ciphertext Stealing. 16, 18

CCM Counter with Cipher Block Chaining Message Authentication. 18

CFB Cipher Feedback. 17, 18

CTR Counter Modus. 15, 17, 18

DES Data Encryption Standard. 18

DH Diffie-Hellman. 16

DHT Distributed Hash Table. 28, 60–62, 104

DIBS Distributed Internet Backup System. 55–57

DLIES Logarithm Integrated Encryption Scheme. 20

DoS Denial of Service. 8, 38, 64

DSA Digital Signature Algorithm. 24

DSGVO Datenschutz-Grundverordnung. 5, 7

DSS Digitaler Signaturstandard. 24

EAX encrypt-then-authenticate-then-translate. 18

ECB Electronic Code Book. 16

ECC Elliptic Curve Cryptography. 20

ECDSA Elliptic Curve Digital Signature Algorithm. 24

ECIES Elliptic Curve Integrated Encryption Scheme. 20

ENISA European Network and Information Security Agency. 14

GCM Galois/Counter-Mode. 17, 18, 81

GPG Gnu Privacy Guard. 55

HMAC Keyed-Hash Message Authentication Code. 23

IAD Information Assurance Directorate. 14, 15, 18, 20, 22, 77, 78

IEC International Electrotechnical Commission. 6

ISMS Informationssicherheitsmanagementsystem. 6, 7, 9, 12

ISO International Organization for Standardization. 6

IV Initialisierungsvektor. 16, 17

JNI Java Native Interface. 86

JRE Java Runtime Environment. 58, 74

JVM Java Virtual Machine. 58, 74

LAN Local Area Network. 43, 54

LT-Code Luby-Transform-Code. 56

MAC Message Authentication Code. 22, 23, 62

MD5 Message-Digest Algorithm 5. 55

NIST National Institute of Standards and Technology. 14, 18, 24

NSA National Security Agency. 14, 15, 18, 20, 22, 77, 78

OAEP Optimal Asymmetric Encryption Padding. 20, 76

OFB Output Feedback. 17, 18

RS-Code Reed-Solomon-Code. 55, 56

RSA Rivest, Shamir, Adleman. 20, 21, 24, 52, 53, 76

SCTP Stream Control Transmission Protocol. 76

SHA Secure Hash Algorithm. 22, 78, 85

SOG-IS Senior Officials Group Information Systems Security. 14, 15, 20, 22

SPBS Secure Peer-to-Peer Backup System. 67, 73

TCP Transmission Control Protocol. 76

TLS Transport Layer Security. 76, 81

UDP User Datagram Protocol. 76

UUID Universal Unique Identifier. 85, 86, 89, 92

VPN Virtual Private Network. 40

WLAN Wireless Local Area Network. 43

XML Extensible Markup Language. 75

XSD XML Schema Definition. 75

Literatur

[1] W. Barth. *Datensicherung unter Linux. Grundlage - Werkzeuge - Konzepte*. Open Source Press GmbH, 2004. 319 S. ISBN: 3-937514-00-7.

[2] M. Bellare, R. Canetti und H. Krawczyk. „Keying hash functions for message authentication“. In: Springer-Verlag, 1996, S. 1–15.

[3] M. Bellare, R. Canetti und H. Krawczyk. „Message Authentication using Hash Functions- The HMAC Construction“. In: *CryptoBytes* 2 (1996).

[4] G. Bertoni, J. Daemen, M. Peeters und G. Assche. *The Keccak sponge function family*. 2012. URL: http://keccak.noekeon.org/ (besucht am 24. 04. 2022).

[5] Bibliographisches Institut GmbH. *Duden. Kryptografie*. URL: https://www.duden.de/rechtschreibung/Kryptografie (besucht am 24. 04. 2022).

[6] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Standard 200-1: Managementsysteme für Informationssicherheit (ISMS)*. Version 1.0. 15. 11. 2017. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/BSI_Standards/standard_200_1.html?nn=128578 (besucht am 24. 04. 2022).

[7] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Standard 200-2: IT-Grundschutz-Methodik*. Version 1.0. 15. 11. 2017. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/BSI_Standards/standard_200_2.html?nn=128640 (besucht am 24. 04. 2022).

[8] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Standard 200-3: Risikoanalyse auf der Basis von IT-Grundschutz*. Version 1.0. 15. 11. 2017. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/BSI_Standards/standard_200_3.html?nn=128620 (besucht am 24. 04. 2022).

[9] Bundesamt für Sicherheit in der Informationstechnik. *Informationssicherheit und IT-Grundschutz. BSI-Standards 200-1, 200-2, 200-3*. Bundesanzeiger Verlag, 2017. ISBN: 978-3-8462-0815-1.

[10] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Kompendium*. Reguvis Fachmedien GmbH, 01. 02. 2021. ISBN: 978-3-8462-0906-6. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2021.html?nn=128542 (besucht am 24. 04. 2022).

[11] Bundesamt für Sicherheit in der Informationstechnik. *Kryptographische Verfahren: Empfehlungen und Schlüssellängen.* TR-02102-1. Version 2021-01. 19. 11. 2021. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.html (besucht am 31. 12. 2021).

[12] Bundesamt für Sicherheit in der Informationstechnik. *Kryptographische Verfahren: Empfehlungen und Schlüssellängen. Teil 2 – Verwendung von Transport Layer Security (TLS).* Technische Richtlinie TR-02102-2. Version 2022-01. 11. 02. 2022. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.html (besucht am 24. 04. 2022).

[13] Bundesnetzagentur. *Empfehlungen zur technischen Umsetzung von Signaturdiensten.* 19. 12. 2017. URL: https://www.bundesnetzagentur.de/EVD/SharedDocuments/Downloads/QES/Algorithmen/Empfehlungen2018.html (besucht am 24. 04. 2022).

[14] S. Cass. *Top Programming Languages 2021. Python dominates as the de facto platform for new technologies.* en. IEEE. 24. 08. 2021. URL: https://spectrum.ieee.org/top-programming-languages-2021 (besucht am 25. 04. 2022).

[15] Code 42 Software, Inc. *Code42 Privacy Statement.* 02. 08. 2021. URL: https://www.code42.com/privacy-statement/ (besucht am 24. 04. 2022).

[16] Code 42 Software, Inc. *CrashPlan for Small Business encryption information.* 05. 03. 2021. URL: https://support.code42.com/Small_Business/Get_Started/Encryption_information (besucht am 24. 04. 2022).

[17] Code 42 Software, Inc. *How CrashPlan for Small Business backup works.* 25. 10. 2021. URL: https://support.code42.com/Small_Business/Configuring/How_backup_works (besucht am 24. 04. 2022).

[18] *Datenschutz-Grundverordnung (EU) 2016/679. Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG (Datenschutz-Grundverordnung) (Text von Bedeutung für den EWR).* 27. 04. 2016. URL: https://eur-lex.europa.eu/legal-content/DE/ALL/?uri=celex:32016R0679 (besucht am 24. 04. 2022).

[19] M. Dell'Amico, P. Michiardi, L. Toka und P. Cataldi. „Adaptive redundancy management for durable P2P backup". In: *Computer Networks* 83 (06/2015), S. 136–148. DOI: 10.1016/j.comnet.2015.03.006.

[20] DIN Deutsches Institut für Normung e. V. *Informationstechnik – Sicherheitsverfahren – Informationssicherheitsmanagementsysteme – Anforderungen (ISO/IEC 27001:2013 einschließlich Cor1:2014 und Cor2:2015); Deutsche Fassung EN ISO/IEC 27001:2017.* 2017.

[21] DIN Deutsches Institut für Normung e. V. *Informationstechnik – Sicherheitsverfahren – Informationssicherheitsmanagementsysteme – Überblick und Terminologie (ISO/IEC 27000:2018);Deutsche Fassung EN ISO/IEC 27000:2020*. 2020.

[22] DIN Deutsches Institut für Normung e. V. *Informationstechnik – Sicherheitsverfahren – Leitfaden für Informationssicherheitsmaßnahmen (ISO/IEC 27002:2013 einschließlich Cor 1:2014 und Cor2:2015); Deutsche Fassung EN ISO/IEC 27002:2017*. 2017.

[23] *Duplicati - Fact Sheet*. 18. 03. 2016. URL: https://www.duplicati.com/articles/FactSheet/ (besucht am 24. 04. 2022).

[24] *Duplicati - How the backup process works*. 14. 10. 2016. URL: https://www.duplicati.com/articles/Backup-Process/ (besucht am 24. 04. 2022).

[25] M. Dworkin. *Recommendation for Block Cipher Modes of Operation. Methods and Techniques*. SP 800-38A. en. National Institute of Standards und Technology. 01. 12. 2001. DOI: https://doi.org/10.6028/NIST.SP.800-38A.

[26] C. Eckert. *IT-Sicherheit*. De Gruyter Oldenbourg, 21. 08. 2018. DOI: 10.1515/9783110563900.

[27] W. Ertel. *Angewandte Kryptographie*. ger. 6., aktualisierte Auflage. 2020. ISBN: 9783446463134.

[28] D. Giry. *BlueKrypt - Kryptographic Key Length Recommendation*. Version 32.3. BlueKrypt sprl. 24. 05. 2020. URL: https://www.keylength.com/en/compare/ (besucht am 24. 04. 2022).

[29] Google Ireland Limited. *Google Drive for Developers. API Reference*. 07. 04. 2022. URL: https://developers.google.com/drive/api/v3/reference (besucht am 24. 04. 2022).

[30] Google Ireland Limited. *So funktioniert der Google-Speicherplatz*. de. URL: https://support.google.com/drive/answer/9312312?hl=de (besucht am 24. 04. 2022).

[31] Google Ireland Limited. *Vertrauen und Sicherheit. Cloud-Compliance*. de. URL: https://cloud.google.com/security (besucht am 24. 04. 2022).

[32] R. Gracia-Tinedo, M. Sanchez Artigas und P. Garda Lopez. „Analysis of data availability in F2F storage systems: When correlations matter“. In: *Peer-to-Peer Computing (P2P), 2012 IEEE 12th International Conference on*. 2012, S. 225 –236. DOI: 10.1109/P2P.2012.6335803.

[33] *H2 Features. Database Files Encryption*. en. URL: https://www.h2database.com/html/features.html#file_encryption (besucht am 24. 04. 2022).

[34] IAD-NSA. *CNSA Suite and Quantum Computing FAQ*. 05. 01. 2016. URL: https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/cnsa-suite-and-quantum-computing-faq.cfm (besucht am 24. 04. 2022).

[35] IAD-NSA. *Commercial National Security Algorithm (CNSA) Suite Factsheet*. 30. 12. 2015. URL: https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/commercial-national-security-algorithm-suite-factsheet.cfm (besucht am 24. 04. 2022).

[36] IAD-NSA. *Commercial National Security Algorithm Suite*. 30. 12. 2015. URL: https://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm (besucht am 24. 04. 2022).

[37] A.-M. Kermarrec, E. L. Merrer, N. L. Scouarnec, R. Ludinard, P. Maillé, G. Straub und A. V. Kempen. „Performance evaluation of a peer-to-peer backup system using buffering at the edge". In: *Computer Communications* 52 (10/2014), S. 71–81. DOI: 10.1016/j.comcom.2014.06.002.

[38] H. Kersten und G. Klett. *Der IT Security Manager : Aktuelles Praxiswissen für IT Security Manager und IT-Sicherheitsbeauftragte in Unternehmen und Behörden*. ger. 4. Aufl. 2015. Edition kes. Wiesbaden: Springer Fachmedien Wiesbaden, 2015. ISBN: 3658099747. DOI: 10.1007/978-3-658-09974-9.

[39] A. Khanjani und R. Sulaiman. „The aspects of choosing open source versus closed source". In: *2011 IEEE Symposium on Computers Informatics*. 2011, S. 646–649. DOI: 10.1109/ISCI.2011.5958992.

[40] H. Krawczyk, M. Bellare und R. Canetti. *RFC2104: HMAC: Keyed-Hashing for Message Authentication*. IVM, UCSD, 02/1997.

[41] W. Kriha und R. Schmitz. *Internet-Security aus Software-Sicht : Ein Leitfaden zur Software-Erstellung für sicherheitskritische Bereiche*. ger. Xpert.press. Berlin, Heidelberg: Springer-Verlag, 2008. ISBN: 3540689060. DOI: 10.1007/978-3-540-68906-5.

[42] M. Landers, H. Zhang und K.-L. Tan. „PeerStore: better performance by relaxing in peer-to-peer backup". In: *Fourth International Conference on Peer-to-Peer Computing, 2004. Proceedings. Proceedings*. 2004, S. 72–79.

[43] X. Li, H. Wang, S. Yi, X. Yao, F. Zhu und L. Zhai. „Redundancy-Guaranteed and Receiving-Constrained Disaster Backup in Cloud Data Center Network". In: *IEEE Access* 6 (2018), S. 47666–47681. DOI: 10.1109/access.2018.2859427.

[44] C. Liu, Y. Lu, C. Shi, G. Lu, D. H. C. Du und D.-S. Wang. „ADMAD: Application-Driven Metadata Aware De-duplication Archival Storage System". In: *2008 Fifth IEEE International Workshop on Storage Network Architecture and Parallel I/Os*. 2008, S. 29–35. DOI: 10.1109/SNAPI.2008.11.

[45] A. Luntovskyy und D. Gütter. *Moderne Rechnernetze*. Springer Fachmedien Wiesbaden, 2020. DOI: 10.1007/978-3-658-25617-3.

[46] P. Mahlmann und C. Schindelhauer. *Peer-to-Peer-Netzwerke*. Springer-Verlag Berlin Heidelberg, 2007.

[47] N. Mandagere, P. Zhou, M. A. Smith und S. Uttamchandani. „Demystifying data deduplication“. In: *Proceedings of the ACM/IFIP/USENIX Middleware '08 Conference Companion*. Companion '08. Leuven, Belgium: ACM, 2008, S. 12–17. ISBN: 978-1-60558-369-3. DOI: 10.1145/1462735.1462739.

[48] P. C. Mane, K. Ahuja und N. Krishnamurthy. „Stability, efficiency, and contentedness of social storage networks“. In: *Annals of Operations Research* 287.2 (08/2019), S. 811–842. DOI: 10.1007/s10479-019-03309-9.

[49] D. Meister und A. Brinkmann. „Multi-level comparison of data deduplication in a backup scenario“. In: *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*. SYSTOR '09. Haifa, Israel: ACM, 2009, 8:1–8:12. ISBN: 978-1-60558-623-6. DOI: 10.1145/1534530.1534541.

[50] D. T. Meyer und W. J. Bolosky. „A study of practical deduplication“. In: *Trans. Storage* 7.4 (02/2012), 14:1–14:20. ISSN: 1553-3077. DOI: 10.1145/2078861.2078864.

[51] S. Nelson. *Pro Data Backup and Recovery*. eng. Berkeley, CA: Apress, 2011. ISBN: 1430226633. URL: 10.1007/978-1-4302-2663-5 (besucht am 24. 04. 2022).

[52] Nextcloud GmbH. *Clients and Client APIs. Webdav*. en. URL: https://docs.nextcloud.com/server/latest/developer_manual/client_apis/WebDAV/index.html (besucht am 24. 04. 2022).

[53] Nextcloud GmbH. *Clients and Client APIs. OCS Share API*. en. URL: https://docs.nextcloud.com/server/latest/developer_manual/client_apis/OCS/ocs-share-api.html (besucht am 24. 04. 2022).

[54] Nextcloud GmbH. *Security and authentication. Nextcloud is designed to protect user datathrough multiple layers of protection.* en. URL: https://nextcloud.com/secure/ (besucht am 24. 04. 2022).

[55] Oracle. *Java Platform, Standard Edition & Java Development KitVersion 17 API Specification. Class DrbgParameters*. en. URL: https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/security/DrbgParameters.html (besucht am 24. 04. 2022).

[56] Oracle. *Security Developer's Guide. JDK Providers Documentation*. SecureRandom Implementations. en. URL: https://docs.oracle.com/en/java/javase/17/security/oracle-providers.html#GUID-9DC4ADD5-6D01-4B2E-9E85-B88E3BEE7453 (besucht am 24. 04. 2022).

[57] Oracle. *Security Developer's Guide. JDK Providers Documentation*. The SunJCE Provider. en. URL: https://docs.oracle.com/en/java/javase/17/security/oracle-providers.html#GUID-A47B1249-593C-4C38-A0D0-68FA7681E0A7 (besucht am 24. 04. 2022).

[58] OWASP Foundation, Inc. *OWASP Dependency-Check*. Version 6.5.0. URL: https://owasp.org/www-project-dependency-check/ (besucht am 24. 04. 2022).

[59] C. Paar und J. Pelzl. *Kryptografie verständlich : Ein Lehrbuch für Studierende und Anwender*. ger. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg Imprint: Springer Vieweg, 2016. ISBN: 3662492970. URL: 10.1007/978-3-662-49297-0 (besucht am 24. 04. 2022).

[60] J. M. Penha-Lopes. „Why Use an Open Source e-Voting System?“ In: *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. ITiCSE '05. Caparica, Portugal: ACM, 2005, S. 412–412. ISBN: 1-59593-024-8. DOI: 10.1145/1067445.1067616.

[61] W. C. Preston. *Backup and recovery*. eng. 1st edition. Beijing: O'Reilly, 2007. ISBN: 0596520387. URL: https://learning.oreilly.com/library/view/backup-recovery/0596102461/ (besucht am 24. 04. 2022).

[62] W. C. Preston. *Modern Data Protection*. eng. O'Reilly Media, Inc, 2021. ISBN: 9781492094043. URL: https://learning.oreilly.com/library/view/Modern-Data-Protection/9781492094043/?ar.

[63] A. Razaque, M. Almani und S. S. Rizvi. „Blackbox: Distributed peer to peer file storage and backup“. In: *2016 Annual Connecticut Conference on Industrial Electronics, Technology Automation (CT-IETA)*. 2016, S. 1–6. DOI: 10.1109/CT-IETA.2016.7868258.

[64] R. L. Rivest, A. Shamir und L. Adleman. „A Method for Obtaining Digital Signatures and Public-key Cryptosystems“. In: *Commun. ACM* 21.2 (02/1978), S. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342.

[65] C. Sanchez-Avila und R. Sanchez-Reillol. „The Rijndael block cipher (AES proposal) : a comparison with DES“. In: *Proceedings IEEE 35th Annual 2001 International Carnahan Conference on Security Technology (Cat. No.01CH37186)*. 2001, S. 229–234. DOI: 10.1109/.2001.962837.

[66] A. Schill und T. Springer. *Verteilte Systeme - Grundlagen und Basistechnologien*. Bd. 2. Springer-Verlag Berlin Heidelberg, 2012. ISBN: 978-3-642-25795-7. DOI: 10.1007/978-3-642-25796-4.

[67] K. Schmeh. *Kryptografie. Verfahren, Protokolle, Infrastrukturen*. 6., aktualisierte Aufl. iX-Edition. Heidelberg: dpunkt.Verlag GmbH, 2016. ISBN: 978-3-86490-356-4. URL: https://www.ebook.de/de/product/25636006/klaus_schmeh_kryptografie.html (besucht am 24. 04. 2022).

[68] G. Schryen und R. Kadura. „Open Source vs. Closed Source Software: Towards Measuring Security“. In: *Proceedings of the 2009 ACM Symposium on Applied Computing*. SAC '09. Honolulu, Hawaii: ACM, 2009, S. 2016–2023. ISBN: 978-1-60558-166-8. DOI: 10.1145/1529282.1529731.

[69] Senior Officials Group Information Systems Security. *SOG-IS Crypto Evaluation Scheme - Agreed Cryptographic Mechanisms*. Version 1.2. URL: https://www.sogis.eu/uk/supporting_doc_en.html (besucht am 24. 04. 2022).

[70] R. Sharma, A. Datta, M. DeH'Amico und P. Michiardi. „An empirical study of availability in friend-to-friend storage systems". In: *2011 IEEE International Conference on Peer-to-Peer Computing*. 2011, S. 348–351. DOI: 10.1109/P2P.2011.6038754.

[71] P. Silva, E. Monteiro und P. Simoes. „Privacy in the Cloud: A Survey of Existing Solutions and Research Challenges". In: *IEEE Access* 9 (2021), S. 10473–10497. DOI: 10.1109/access.2021.3049599.

[72] J. Song, R. Poovendran, J. Lee und T. Iwata. *RFC4493: The AES-CMAC Algorithm*. Techn. Ber. Internet Engineering Task Force. URL: https://tools.ietf.org/html/rfc4493 (besucht am 24. 04. 2022).

[73] S. Spitz, M. Pramateftakis und J. Swoboda. *Kryptographie und IT-Sicherheit : Grundlagen und Anwendungen*. ger. 2., überarbeitete Auflage. Wiesbaden: Vieweg+ Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2011. ISBN: 3834881201. URL: 10.1007/978-3-8348-8120-5.

[74] Statista. *Desktop PC operating system market share worldwide, from January 2013 to June 2021. Desktop PC OS market share 2013-2021*. en. 2021. URL: https://www.statista.com/statistics/218089/global-market-share-of-windows-7/ (besucht am 24. 04. 2022).

[75] Statista. *Mobile internet usage worldwide. Statista dossier about mobile internet usage worldwide*. en. 2020. URL: https://www.statista.com/study/21391/mobile-internet-usage-statista-dossier/ (besucht am 24. 04. 2022).

[76] Statista. *Share of the global server market by operating system in 2018 and 2019. Global server share by operating system 2018-2019*. en. 2020. URL: https://www.statista.com/statistics/915085/global-server-share-by-os/ (besucht am 24. 04. 2022).

[77] R. Steinmetz und K. Wehrle. *Peer-to-Peer Systems and Applications*. Springer-Verlag Berlin Heidelberg, 2005.

[78] P. Storz. *Bacula. Backup-Strategien und -Lösungen im Netzwerk*. Open Source Press, 2012. 447 S. ISBN: 978-3-941841-41-3.

[79] G.-Z. Sun, Y. Dong, D.-W. Chen und J. Wei. „Data Backup and Recovery Based on Data De-Duplication". In: *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*. Bd. 2. 2010, S. 379 –382. DOI: 10.1109/AICI.2010.200.

[80] A. S. Tanenbaum und M. Steen. *Verteilte Systeme: Prinzipien und Paradigmen*. 2. Auflage. Pearson Studium, 2008. ISBN: 978-3-8273-7293-2.

[81] The Apache Software Foundation. *Part Two: Configuring security for Derby*. ed. Version 10.15. URL: https://db.apache.org/derby/docs/10.15/security/cseccsecuree.html (besucht am 24. 04. 2022).

[82] TIOBE Software BV. *TIOBE Index for October 2021. October Headline: Python programming language number 1!* 2021. URL: http://www.tiobe.com/tiobe-index/ (besucht am 06. 11. 2021).

[83] L. Toka und G. Biczok. „On pricing online data backup“. In: *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2015, S. 564–569. DOI: 10.1109/infcomw.2015.7179445.

[84] F. Toussi. *HyperSQL User Guide. Chapter 11. System Management - Encrypted Databases*. en. The HSQL Development Group. 21. 04. 2022. URL: http://hsqldb.org/doc/2.0/guide/management-chapt.html#mtc_encrypted_database (besucht am 24. 04. 2022).

[85] D. N. Tran, F. Chiang und J. Li. „Friendstore: Cooperative Online Backup Using Trusted Nodes“. In: *Proceedings of the 1st Workshop on Social Network Systems*. SocialNets '08. Glasgow, Scotland: Association for Computing Machinery, 2008, 37–42. ISBN: 9781605581248. DOI: 10.1145/1435497.1435504.

[86] S. U. Zaman, R. Karim, M. S. Arefin und Y. Morimoto. „Distributed Multi Cloud Storage System to Improve Data Security with Hybrid Encryption“. In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 10/2019, S. 61–74. DOI: 10.1007/978-3-030-33585-4_6.

[87] Y. Zhang, H. Jiang, D. Feng, W. Xia, M. Fu, F. Huang und Y. Zhou. „AE: An Asymmetric Extremum content defined chunking algorithm for fast and bandwidth-efficient data deduplication“. In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 04/2015. DOI: 10.1109/infocom.2015.7218510.

[88] X. Zuo und A. Iamnitchi. „A Survey of Socially Aware Peer-to-Peer Systems“. In: *ACM Computing Surveys* 49.1 (07/2016), S. 1–28. DOI: 10.1145/2894761.