Fraunhofer FOKUS

ISPRAT

CASED

TECHNISCHE UNIVERSITÄT DARMSTADT

Der elektronische Safe als vertrauenswürdiger Cloud Service

ISPRAT Studie

J. Braun, E. Kieme, W. Wunderlich, L. Strick, A. Wiesmaier

Der elektronische Safe als vertrauenswürdiger Cloud Service

ISPRAT-Studie

September 2011

Autoren

Eric Klieme
eric.klieme@fokus.fraunhofer.de

Linda Strick
linda.strick@fokus.fraunhofer.de

Wolfgang Wunderlich
wolfgang.wunderlich@fokus.fraunhofer.de

Fraunhofer-Institut für Offene Kommunikationssysteme
Kaiserin-Augusta-Allee 31
10589 Berlin

Johannes Braun
jbraun@cdc.informatik.tu-darmstadt.de

Dr. Ing. Alexander Wiesmaier
wiesmaie@cdc.informatik.tu-darmstadt.de

Technische Universität Darmstadt
Kryptographie und Computeralgebra
Hochschulstraße 10
64289 Darmstadt

Inhaltsverzeichnis

1 Einleitung 7

1.1 Motivation 7

1.2 Gliederung der Studie 8

2 Ausgangssituation und Grundlagen 9

2.1 Der elektronische Safe 9

2.1.1 Safe-Eigentümer 10
2.1.2 Safe-Benutzer 10
2.1.3 Safe-Anbieter 10
2.1.4 Anwendungsfälle des elektronischen Safes 11

2.2 Anforderungen an einen Cloud-Safe 12

2.2.1 Vertraulichkeit 13
2.2.2 Verfügbarkeit 13
2.2.3 Integrität 14
2.2.4 Schlüsselmanagement 14

2.3 Architektur 15

2.3.1 Safe-Eigentümer Client 15
2.3.2 Safe-Benutzer Client 16
2.3.3 Safe-Anbieter 16
2.3.4 Speicher-Anbieter 16
2.3.5 Identitäts- und Berechtigungsmanagement 16
2.3.6 Logging Service 17
2.3.7 Notification Service 17

2.4 Grundlagen des Cloud Computings 17

2.4.1 Dienstklassen 17
2.4.2 Cloud-Betriebsmodelle 18
2.4.3 Cloud-Referenzarchitektur 18

3 Systeme mit Cloud-Safe-ähnlichen Konzepten 20

3.1.1 Kollaborationsumgebungen 20
3.1.2 Speicherplatzverschlüsselung 20
3.1.3 Online-Festplatten 20
3.1.4 De-Mail 21
3.1.5 E-Postbrief 22
3.1.6 Fazit 23

4 Der elektronische Safe als Cloud-Dienst 24

4.1 Identitätsmanagement-Anbieter 24

4.2 Cloud-Safe-Anbieter 24

4.3 Speicher-Anbieter ... 24
4.4 Konfigurationsvarianten ... 25
4.4.1 Standardvariante des Cloud-Safes ... 26
4.4.2 Komfortvariante des Cloud-Safes ... 27
4.4.3 Safe-Anbieter mit Signaturservice ... 28
4.5 Bewertung der Konfigurationen ... 28
4.5.1 Vertraulichkeit ... 28
4.5.2 Verfügbarkeit ... 29
4.5.3 Integrität ... 29
4.5.4 Fazit ... 30
5 Sicherung der Vertraulichkeit ... 31
5.1 Effiziente Realisierung ohne Redundanz ... 31
5.2 Realisierung mit Redundanz ... 32
5.2.1 Shamir's Secret Sharing Verfahren ... 32
5.2.2 Verschlüsselungsverfahren ... 33
5.3 Bewertung der Verfahren ... 34
5.4 Langfristige Sicherheit ... 34
5.5 Empfehlungen ... 35
6 Theoretischer Hintergrund und Sicherheitsanalyse des vorgeschlagenen Verschlüsselungsverfahrens ... 36
6.1 Einleitung ... 36
6.1.1 Motivation ... 36
6.1.2 Beitrag ... 36
6.1.3 Verwandte Arbeiten ... 37
6.2 Grundlagen ... 40
6.2.1 Perfektes Secret Sharing ... 40
6.2.2 Ideal sichere Systeme ... 41
6.3 GSSCC - General secret sharing based cipher combining ... 43
6.4 Sicherheit im Ideal Cipher-Modell ... 44
6.4.1 Erschöpfende Schlüsselsuche ... 45
6.4.2 Meet-in-the-Middle-Angriff ... 46
6.5 Instanziierung von GSSCC ... 47
6.5.1 Partielle ideale Sicherheit unter CPA2-Angriffen ... 48
6.5.2 Chosen-Ciphertext-Angriffe ... 50
6.5.3 Blocklänge von GSSCC ... 50
6.6 Diskussion ... 51
6.6.1 Über die Auswahl der Chiffren zur GSSCC-Instanziierung ... 51
6.6.2 Weitere Anmerkungen ... 52
6.7 Meet-in-the-Middle-sicherer Blockmodus ... 53
6.8 Fazit ... 54
7 Potentielle Angriffsszenarien ... 55
7.1 Sicherheitsrisiken und –aspekte ... 55
7.1.1 Angreifertypen ... 55

7.1.2 Passive Angreifer ... 55
7.1.3 Aktive Angreifer ... 55
7.1.4 Externe Angreifer ... 56
7.1.5 Interne Angreifer ... 56
7.1.6 Intention der Angreifer ... 56
7.2 Angreifermodelle ... 56
7.2.1 Externer Angreifer passiv ... 56
7.2.2 Externer Angreifer aktiv ... 57
7.2.3 Interner Angreifer mit partiellem Zugriff ... 57
7.2.4 Interner Angreifer mit vollem Zugriff ... 58
7.3 Integritätsschutz ... 58
7.3.1 Verfügbarkeitsschutz ... 58
7.4 Schutz der Vertraulichkeit ... 59
8 Realisierungskonzepte ... 60
8.1 Adaptoren ... 60
8.2 Schnittstellen ... 61
8.3 Marktanalyse angebotener Cloud-Service-Frameworks ... 63
8.3.1 IBM ... 63
8.3.2 Microsoft ... 64
8.3.3 Amazon ... 65
8.3.4 Google ... 66
8.3.5 JClouds ... 67
8.4 Standardisierung und Zertifizierung ... 68
8.5 Fazit ... 70
9 Anwendungsbeispiel – Der moderne Arbeitsplatz der Verwaltung ... 72
9.1 Grundvoraussetzung ... 72
9.2 Der Cloud-Safe der Verwaltung ... 74
10 Fazit und Ausblick ... 76
11 Literaturverzeichnis ... 77

1 Einleitung

1.1 Motivation

Cloud Computing ist zu einem der meistdiskutierten Schlagwörter der aktuellen Debatte im IT-Bereich geworden. Es verspricht die Konsolidierung von IT-Ressourcen in Pools, die automatisierte und bedarfsangepasste Skalierung von Ressourcen sowie die Bereitstellung mandantenfähiger Dienste nach dem Selbstbedienungsprinzip. Cloud-Computing hat das Potenzial, mittel- bis langfristig einen beträchtlichen Teil der traditionellen IT-Leistungsangebote in der öffentlichen Verwaltung zu ersetzen. Insbesondere in Hinblick auf die Forderungen nach einer verbesserten Interaktion zwischen Bürgern, Wirtschaftsbetrieben und den zuständigen Verwaltungen auf Basis elektronischer Medien bildet Cloud Computing eine Basis für neue Kooperationsformen, die sowohl die Interaktion zwischen Bürgern/Unternehmen und Behörden als auch die Kooperation zwischen Behörden maßgeblich vereinfacht. Allerdings bestehen rechtliche Vorbehalte gegenüber dem Cloud Computing, insbesondere in den Bereichen Sicherheit und – in enger Beziehung dazu – Datenschutz. Dies wird auch von der EU-Justizkommissarin Viviane Reding verdeutlicht: *„Behörden und Unternehmen sind ihren Mitarbeitern und Kunden den verantwortungsvollen Umgang mit persönlichen Daten schuldig. Die Geschäftsmodelle vieler Unternehmen heute – vor allem in der IT-Branche – basieren auf dem Vertrauen der Verbraucher. Bürger scheinen jedoch mehr und mehr um die Sicherheit ihrer Daten besorgt, und das nicht ohne Grund. Wir erleben derzeit geradezu eine Welle von Datenpannen, Datendiebstählen, Datenverlusten durch Softwarefehler oder auch kommerziell bewusst getätigter Datenverkäufe [Red2011]“.*

Insbesondere in Hinblick auf den elektronischen Rechtsverkehr müssen gewisse Voraussetzungen erfüllt sein. Rechtlich ist der stringente Abbau von Schriftformerfordernissen unabdingbar. Inhaltlich bedeutet dies die Bereitstellung von praxisrelevanten Internet-basierten Anwendungen, die höchsten Anforderungen an die Gewährleistung von Datenschutz und Datensicherheit gerecht werden müssen.

Es stellt sich darüber hinaus die Frage, wie der Bürger zukünftig mit elektronisch übersendeten Daten, Dokumenten und Urkunden umgeht. Diese müssen einerseits langfristig gespeichert und anderseits vor Missbrauch geschützt werden. Es ist offensichtlich, dass weite Teile der Bevölkerung sowie der kleinen und mittelständischen Unternehmen nicht in der Lage sind, diese Aufgaben im Rahmen ihrer IKT-Nutzung, z.B. über den häuslichen PC, sicherzustellen. Bereits heute gibt es viele Anbieter von zentralisierten Speicherkapazitäten, die eine angemessene Betriebssicherheit bieten. Diese Angebote sind aber hinsichtlich der oben angesprochenen Anforderungen an den Datenschutz und die Datensicherheit im Allgemeinen nicht ausreichend.

Die vorliegende Studie befasst sich mit der vertraulichen Aufbewahrung sensibler Daten und Dokumente in einem elektronischen Cloud-Safe. Unter Benutzung verschiedener Cloud-Anbieter wird eine Infrastruktur konzipiert, die Bedarfsträgern aus dem öffentlichen Sektor wie auch privaten Dienstleistern die Möglichkeit bietet, Bürgern und

Unternehmen bzw. Kunden eine mit dem Bedarf skalierende vertrauenswürdige Infrastruktur zur Verfügung zu stellen, ohne die notwendigen Ressourcen selbst vorhalten zu müssen. Unter diesem Gesichtspunkt werden auch Angreifermodelle untersucht. Außerdem erfordern die starken Unterschiede in der Bereitstellung von Cloud-Services die konzeptionelle Anbindung des Cloud-Safes an die Dienste der Cloud-Anbieter mittels Adaptoren.

Ergebnis ist eine neuartige Lösung, die langfristige Vertraulichkeit garantiert und die Schaffung kooperierender Verwaltungsprozesse am Beispiel der Behördenarbeitsplätze ermöglicht.

1.2 Gliederung der Studie

Die vorliegende Studie ist in 10 Kapitel gegliedert. Während Kapitel 1 die allgemeine Einleitung beschreibt, werden in Kapitel 2 die Grundlagen des elektronischen Safes beschrieben und die Anforderungen an einen elektronischen Safe als Cloud-Dienst gestellt, welcher der Einfachheit halber Cloud-Safe genannt wird. Relevant sind dabei die Sicherheitsanforderungen, auf denen Vertrauen aufgebaut werden kann. Das Kapitel schließt mit einem kurzen Überblick über Cloud Computing ab, um eine einheitliche Grundlage für die Cloud-Diskussion zu schaffen.

Kapitel 3 beschäftigt sich mit Konzepten, die Speichermöglichkeiten über das Internet erlauben, beziehungsweise Konzepte beinhalten, die dem Cloud-Safe ähnlich sind.

Kapitel 4 schlägt mögliche Konfigurationsvarianten für den Cloud-Safe vor und bewertet diese Varianten bezüglich der Vertraulichkeit, Verfügbarkeit und Integrität.

Während Kapitel 5 ausführlich Verfahren zur Sicherung der Vertrauenswürdigkeit diskutiert, werden in Kapitel 6 die theoretischen Grundlagen dazu vorgestellt.

In Kapitel 7 werden die Sicherheitsrisiken und Angreifermodelle vorgestellt und Verfahren zum Schutz von Vertraulichkeit und der Integrität dargestellt.

Kapitel 8 schließlich beschäftigt sich mit den Realisierungskonzepten mit Blick auf Anpassungen über Adaptoren und Grundlagen für interoperable Cloud-Service-Plattformen und der Notwendigkeit, offene Schnittstellen und Standards zu schaffen.

In Kapitel 9 wird der moderne Arbeitsplatz als Beispiel für eine Cloud-Safe-Integration bei IT-Dienstleistern der Verwaltung vorgeschlagen.

Kapitel 10 enthält das Fazit und einen Ausblick.

2 Ausgangssituation und Grundlagen

2.1 Der elektronische Safe

Elektronische Safes zur Ablage und Freigabe digitaler Unterlagen (Daten und Dokumente) bieten eine Infrastruktur, um den Austausch zwischen Verwaltung, Wirtschaft und Bürgern zu vereinfachen und zu beschleunigen. Mit Hilfe des eSafes können die verschiedenen Akteure einer umfassend vernetzten digitalen Welt ihre vertraulichen Informationen austauschen [HSP2010].

Obwohl in dieser Studie der Schwerpunkt auf der sicheren Aufbewahrung und nicht in dem Austausch von Dokumenten liegt, soll der Vollständigkeit halber das gesamte Bild eines elektronischen Safes hier dargestellt werden.

Der elektronische Safe dient, in Analogie zum traditionellen Tresor (Safe), als Speicher für digitalisierte Dokumente und Daten und bietet neben der Ablage auch die Möglichkeit einzelne Dokumente oder Daten auszuwählen, um diese Dritten auf Verlangen vorzeigen zu können. Dies kann allerdings nur mit Einwilligung des Eigentümers erfolgen. Gewährleistung der Vertraulichkeit ist eine wesentliche Anforderung für die Ablage von Dokumenten im Netz, die eine Ende-zu-Ende-Verschlüsselung zwischen den Kommunikationspartnern erfordert. Beispielhaft ist in **Abbildung *1*** die Freigabe von Daten und Dokumenten dargestellt. Dazu erzeugt der Safe-Eigentümer einen Freigabebereich in seinem Safe, die sogenannte Transportbox, und hinterlegt eine Kopie des elektronischen Dokuments. Nach Erteilung einer Berechtigung für einen Dritten (hier die öffentliche Verwaltung) in Schritt 2, kann dieser in Schritt 3 die Dokumente entsprechend der Freigaben abholen.

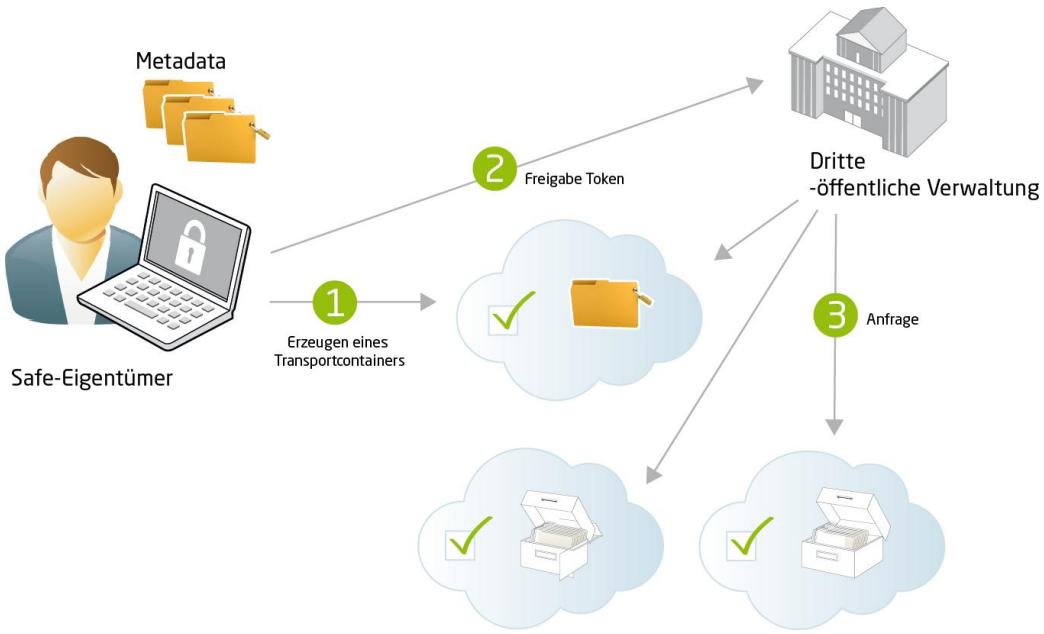

Abbildung 1: Freigabe eines Dokuments an Dritte

Der elektronische Safe kann sowohl von Privatpersonen als auch Unternehmen als sichere und vertrauenswürdige Ablage genutzt werden.

Zur Klärung der Begrifflichkeiten werden zunächst die Akteure beschrieben, gefolgt von dedizierten Anwendungsfällen.

2.1.1 Safe-Eigentümer

Der elektronische Safe bietet dem Eigentümer neben der sicheren Verwahrung von Daten und Dokumenten auch die Verwaltung der gespeicherten Dokumente an. Basierend auf dem Lebenszyklus eines elektronischen Safes, der die Phasen von der Erstellung des Safes bis zur Löschung des Safes durchläuft [HKP2010], hat der Safe-Eigentümer dabei über die gesamte Zeit die Kontrolle über seinen Safe. Dazu gehören sowohl die Verwaltung des Safes hinsichtlich der Zugangsdaten, als auch die gesamte Rechteverwaltung. Alle Anwendungsfälle des Safes werden dementsprechend durch den Eigentümer gesteuert und kontrolliert. Der Safe-Eigentümer kann dabei durch eine juristische (Unternehmen) oder natürliche Person vertreten sein. Der Eigentümer kann Dritten dediziert Zugriff auf Inhalte gewähren.

2.1.2 Safe-Benutzer

Der Safe-Benutzer ist der Akteur, der basierend auf einer gezielten Freigabe seitens des Safe-Eigentümers Zugriff auf bestimmte Daten oder Dokumente hat. Der Safe-Benutzer kann eine aktive oder reaktive Rolle einnehmen. Werden Daten für einen Verwaltungsvorgang benötigt, kann der Safe-Benutzer aktiv eine Anfrage an einen Safe-Eigentümer über bestimmte Datensätze oder Dokumente, beispielsweise per Email (z.B. De-Mail), stellen. Je nachdem, ob der Safe-Eigentümer diese Anfrage akzeptiert, wird dem Safe-Benutzer Zugriff gewährt. Diese Autorisierung erfolgt allerdings für einen begrenzten Zeitraum und nur auf ausgewählte Dokumente oder Daten. Gleichzeitig kann ein Safe-Benutzer auch reaktiv in die Prozesse im Safe-Kontext eingebunden sein. Stellt ein Safe-Eigentümer eine Anfrage im Rahmen eines Verwaltungsvorgangs und kann er gleichzeitig einen bestimmten Datensatz freigeben, so ist der Safe-Benutzer von Beginn an autorisiert, die entsprechenden Daten zu nutzen.

Ein Safe-Benutzer kann entweder eine juristische oder natürliche Person sein.

2.1.3 Safe-Anbieter

Der Safe-Anbieter übernimmt die Bereitstellung des „elektronischen Safes“. Er schließt mit dem Safe-Eigentümer einen Vertrag, der die Einhaltung gesetzlicher Aufbewahrungsfristen und technischer Dienstgütevereinbarungen (*Service Level Agreements*) festlegt. Weiterhin erfolgt über den Safe-Anbieter die Verwaltung der Verteilungsinformation der abgelegten Dokumente und Daten. Mit Hilfe dieser Informationen kann ein Benutzer den Inhalt seines elektronischen Safes abrufen.

2.1.4 Anwendungsfälle des elektronischen Safes

Für eine kurze Beschreibung der Anwendungsfälle eines elektronischen Safes werden folgende Funktionen näher erläutert:

- Registrierung
- Anmeldung
- Hochladen von Dateien
- Herunterladen von Dateien
- Löschen von Dateien
- Freigabeverwaltung
- Ordner- und Dateiverwaltung

Registrierung

Zunächst muss sich ein Safe-Eigentümer beim Safe-Anbieter registrieren. Dies kann beispielsweise mit dem neuen Personalausweis geschehen. Im Allgemeinen sollte es sich um ein sicheres Registrierungsverfahren handeln. Nach erfolgreicher Registrierung wird dem Safe-Eigentümer die benötigte clientseitige Anwendung zur Verfügung gestellt.

Anmeldung

Nachdem sich ein Safe-Eigentümer bei einem Safe-Anbieter registriert hat, kann er mit Hilfe der clientseitigen Anwendung eine Verbindung zum Safe-Service des Anbieters herstellen. Dazu ist für jede Nutzung eine sichere Authentifizierung notwendig, die gegebenenfalls über den neuen Personalausweis realisiert werden kann. Ist die Authentifizierung erfolgt, wird dem Safe-Eigentümer der Inhalt seines elektronischen Safes mit allen verfügbaren Funktionen präsentiert.

Hochladen von Dateien

Beim Hochladen von Dateien kann das Sicherheitsniveau anhand von geeigneten Parametern vom Safe-Eigentümer selbst festgelegt werden. Um Daten sicher zu speichern, wird das *Perfect Secret Sharing* Verfahren angewendet, das auf dem Teilen von Geheimnissen in sogenannte *Shares* beruht. Details zu dem Verfahren und warum gerade dieses Verfahren besonders sicher ist, werden in Kapitel 5.2.1 beschrieben. Im Rahmen von Synchronisierungsvorgängen – etwa beim Schließen des Safes – werden die *Shares* auf die verschiedenen Speicher-Anbieter verteilt. Der Safe-Eigentümer kann angeben, auf wie viele Speicher die Teile (Shares) verteilt werden.

Herunterladen von Dateien

Nach der erfolgreichen Authentisierung hat der Safe-Eigentümer vollen Zugriff auf alle im Safe gespeicherten Daten und Dokumente und kann die Dokumente herunterladen. Dabei werden die einzelnen Teile (Shares) des Dokuments anhand der beim Safe-Anbieter gespeicherten Metadaten dem Safe-Eigentümer bereitgestellt.

Verschieben und Umbenennen von Dateien, Verwaltung der Ordnerstruktur

Diese Operationen betreffen lediglich die Metadaten der Dateien. Die Organisation der Dateien wird beim Safe-Anbieter bzw. innerhalb der clientseitigen Komponente angepasst.

Löschen von Dateien

Soll eine Datei gelöscht werden, wird eine Löschanforderung an alle assoziierten Speichereinheiten des elektronischen Safe-Anbieters gesandt. Nach Bestätigung müssen auch die Metadaten aus der Datenbank des Safe-Anbieters gelöscht werden.[1]

Freigabe von Dateien

Bereits gespeicherte Dateien kann ein Safe-Eigentümer für andere Nutzer freigeben. Der Safe-Eigentümer bestimmt über die Dauer der Gültigkeit für den Zugriff und wie oft der Zugriff erlaubt werden soll. Zugriffsrechte können auch während des Freigabezeitraumes widerrufen werden. Freigabe bezieht sich in diesem Zusammenhang immer nur auf Leserechte.

Für eine detaillierte Darstellung aller Use-Cases sei auf [HKP2010] verwiesen.

2.2 Anforderungen an einen Cloud-Safe

Der Cloud-Safe soll es dem Safe-Eigentümer (aus Sicht des Cloud-Safe-Anbieters der Nutzer des Cloud-Safes) ermöglichen, beliebige elektronische Dokumente vertrauenswürdig in der Cloud zu speichern. Die Speicherung ist dabei auf Sicherheit und Langfristigkeit ausgelegt. Idee hierbei ist, durch das Angebot eines Cloud-Safes, der Vertrauenswürdigkeit und Sicherheit gewährleistet, die Cloud als sicheren und langfristigen Speicherort zu promoten. Ob dieser Cloud-Safe nun direkt von einem Cloud-Anbieter oder einem externen Anbieter, der Cloud-Dienste nutzt, bereitgestellt wird, ist in diesem Fall nicht relevant.

Gerade in Hinblick auf die Bedenken bezüglich Sicherheit bei Cloud Computing spielt die Gewährleistung der Informationssicherheit eine zentrale Rolle [BIT2010]. Dies bedeutet in technischer Hinsicht für den Betrieb einer Cloud die Einhaltung der Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit. Diese werden in Kapitel 2.2 näher erläutert. Für den Betrieb eines Cloud-Safes bedeutet das, dass entsprechende Parametrisierung erforderlich ist, die es dem Safe-Eigentümer erlaubt, entsprechende Schutzbedarfe auszusuchen. Diese Unterschiede werden in verschiedenen Varianten des Cloud-Safes in Kapitel 4.4 dargestellt.

[1] Dass die Datei damit tatsächlich vollständig aus der Cloud gelöscht wurde, kann nicht garantiert werden. Daher muss mit geeigneten Mitteln sichergestellt werden, dass die Daten über sehr lange Zeiträume als vertraulich betrachtet werden können und ein Löschen des Schlüssels als ausreichend betrachtet werden kann.

Für die Nutzung eines Cloud-Safes, sei es der Eigentümer oder ein externen Nutzer (siehe Akteure in Kapitel 2.1) sind dedizierte Client-Anwendungen erforderlich, die den sicheren, vertrauenswürdigen Zugriff auf die im Cloud-Safe gespeicherten Dokumente von jedem Rechner aus ermöglichen.

Zusätzlich soll der Cloud-Safe dazu dienen, anderen Nutzern Zugriff auf bestimmte Dateien zu gewähren, ohne eine direkte Kommunikation zwischen den Beteiligten zu erfordern bzw. diese auf ein Minimum zu beschränken. Der Austausch von Dateien und vertraulichen Dokumenten wird dabei abhängig von der Konfiguration des Cloud-Safes realisiert.

Zusammenfassend sind die Grundfunktionalitäten des Cloud-Safes:

- Einheitlicher Zugriff auf einen elektronischen Safe für das Speichern und den Zugriff auf vertrauliche Dokumente und Dateien in der Cloud.
- Bereitstellung definierter Sicherheitsmechanismen, unabhängig von der darunterliegenden Cloud-Technologie.
- Austausch von Dateien zwischen Safe-Eigentümern und Safe-Nutzern und dedizierte Zugriffsrechteverwaltung.

Aus den in Kapitel 2.1.4 beschriebenen Funktionen resultiert eine Vielzahl von Anforderungen an eine Umsetzung des Cloud-Safes. Im folgenden Abschnitt werden sowohl die funktionalen als auch die nicht-funktionalen Anforderungen diskutiert.

Wesentlich ist, dass der Cloud-Safe mittels einheitlicher, offener Schnittstelle bedient werden kann. Dazu sind Varianten für Realisierungen in Kapitel 8 beschrieben.

2.2.1 Vertraulichkeit

Der Safe-Eigentümer hat eine starke Forderung nach Vertraulichkeit. Das heißt, dass der Inhalt der im Safe eingelagerten Daten nur dem Eigentümer selbst zugänglich sein darf bzw. nur den beteiligten Dritten, die durch den Eigentümer (partiell) autorisiert wurden. Letzteren soll der Zugriff nur auf die Informationen gewährt werden, die explizit und über einen begrenzten Zeitraum vom Eigentümer freigegeben wurden. Dies ist besonders relevant im Zusammenhang mit der Art der Einsichtnahme.

2.2.2 Verfügbarkeit

Die zweite Anforderung ist die hohe Verfügbarkeit bzw. Erreichbarkeit des Safes. Dabei ist der alltägliche Zugriff zum Speichern und Abrufen der Dokumente und Dateien wichtig, der 24 Stunden am Tag und an sieben Tagen der Woche verfügbar sein muss. Darüber hinaus zielt die Verfügbarkeit auch auf die langfristige Archivierung ab. Safe-Eigentümer müssen sich darauf verlassen können, dass ihre Dokumente und Daten über mehrere Jahre mit gleichbleibender Qualität archiviert und nach Ablauf auch nachhaltig gelöscht werden.

2.2.3 Integrität

Zudem ist die Integrität zu schützen. Dies bedeutet, dass alle Daten stets aktuell, korrekt und vollständig sind. Ein Safe-Eigentümer muss aus diesem Grund genau sehen können, wann welche Zugriffe innerhalb des Safes gemacht wurden. In diesem Zusammenhang muss ebenfalls genau erkennbar sein, wer diese Zugriffe durchgeführt hat. Im rechtlichen Rahmen muss eine Zurechenbarkeit der jeweiligen Transaktionen klar möglich sein, was für alle beteiligten Safe-Akteure wichtig ist. Auftretende Komplikationen können durch Einsicht in diese Protokolle gelöst werden.

2.2.4 Schlüsselmanagement

Vertraulichkeit bedeutet aber auch, dass Daten und Dokumente verschlüsselt werden, damit sie nicht von Externen gelesen werden können.

Die folgenden Best-Practice-Erfahrungen für die Schlüsselverwaltung sollen laut BSI umgesetzt werden [BSI2011]:

- *Die Schlüsselerzeugung sollte in sicherer Umgebung und unter Einsatz geeigneter Schlüsselgeneratoren erfolgen.*
- *Kryptographische Schlüssel sollten möglichst nur für einen Einsatzzweck dienen.*
- *Generell sollten Schlüssel nie in klarer Form, sondern grundsätzlich verschlüsselt im System gespeichert werden. Darüber hinaus muss die Speicherung stets redundant gesichert und wiederherstellbar sein, um einen Verlust eines Schlüssels zu vermeiden.*
- *Die Schlüssel müssen sicher (vertraulich, integer und authentisch) verteilt werden.*
- *Administratoren der Cloud dürfen keinen Zugriff auf Kundenschlüssel haben.*
- *Es müssen regelmäßig Schlüsselwechsel durchgeführt werden. Die verwendeten Schlüssel sollten regelmäßig auf ihre Aktualität überprüft werden.*
- *Der Zugang zu Schlüsselverwaltungsfunktionen sollte eine separate Authentisierung erfordern.*
- *Die Schlüssel sollten sicher archiviert werden.*
- *Nicht mehr benötigte Schlüssel (z. B. Schlüssel, deren Gültigkeitsdauer abgelaufen ist) sind auf sichere Art zu löschen bzw. zu vernichten. Ausnahme bei Langzeitarchivierung ist, dass die Daten zuvor umverschlüsselt werden müssen, um diese nicht mit zu vernichten.*

Detaillierte Informationen zur Verschlüsselung und entsprechenden Verfahren werden in Kapitel 5 ausführlich diskutiert.

Wesentlich ist auch die staatliche Unterstützung hinsichtlich der Vertrauensbildung in elektronische Cloud-Safes. Dies könnte durch entsprechende gesetzliche Vorgaben entstehen. Näheres dazu ist in Kapitel 3 zu finden.

Das Vertrauen von Bürgern in elektronische Safes kann insbesondere dadurch erhöht werden, dass der Staat Gesetze zum Umgang mit elektronischen Dokumenten erlässt, nach

denen sich Cloud-Safes richten und sich sogar zertifizieren lassen können. Ein so zertifizierter Cloud-Safe legt den Grundstein zur Vertrauenswürdigkeit.

2.3 Architektur

Im folgenden Abschnitt erfolgt eine Darstellung der verschiedenen Komponenten, die die erläuterten Funktionen des Safes bereitstellen. **Abbildung *1*** stellt einen Überblick der jeweiligen Komponenten dar, die anschließend näher erläutert werden.

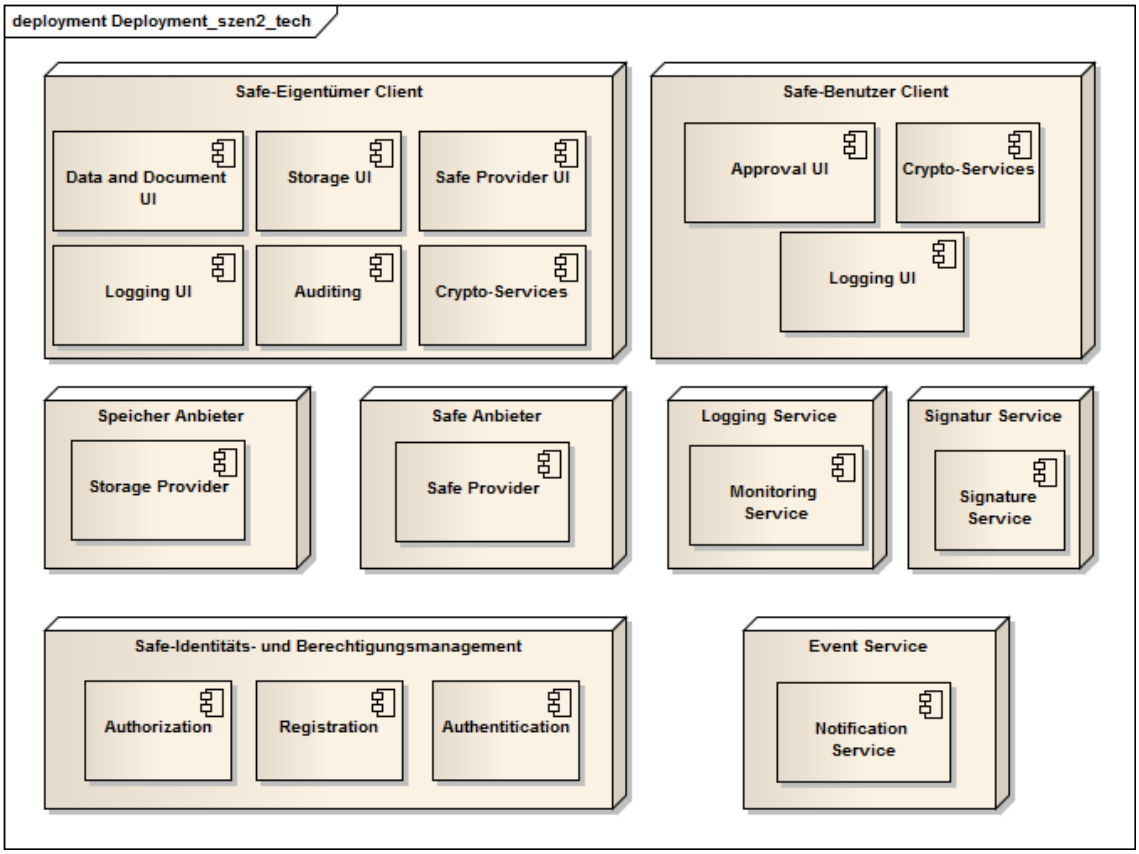

Abbildung 2: Komponenten des Cloud-Safes

2.3.1 Safe-Eigentümer Client

Die Client-Anwendung realisiert die Schnittstelle des Safe-Eigentümers mit dem System des elektronischen Safes. Diese Anwendung wird vom Eigentümer genutzt, um die in 2.1.4 beschriebenen Funktionen ausführen zu können, und bietet dementsprechend die transparente Visualisierung der im Folgenden beschriebenen Funktionen an.

Der Client zeigt die Darstellung aller im Safe abgelegten Dateien an, sowie eine Übersicht aller erfolgten oder noch nicht bearbeiteten Freigaben. Weiterhin werden durch den Safe-Eigentümer Client die Ver- bzw. Entschlüsselung aller im Safe abgelegten Dateien oder Dokumente durchgeführt (Crypto Services), so dass diese nur beim Eigentümer temporär unverschlüsselt und im Klartext ersichtlich vorliegen. Zusätzlich wird durch weitere

Softwaremodule ein umfangreiches Logging und Auditing durchgeführt, um die Nachvollziehbarkeit während der Durchführung verschiedener Prozesse durch deren Protokollierung zu gewährleisten.

Außerdem erfolgt eine Darstellung der verwendeten (Safe-)Speicher-Anbieter (Storage UI) sowie des Safe-Anbieters (Safe Provider UI).

2.3.2 Safe-Benutzer Client

Der Client des Safe-Benutzers ähnelt der Software, die der Safe-Eigentümer benutzt. Der Safe-Benutzer hat nur einen verminderten Funktionsumfang, also nur eingeschränkte Einsicht in Dokumente. Somit entfallen alle Funktionen, die mit dem Einstellen und Freigeben von Daten oder Dokumenten zusammenhängen. Er hat auch keine Logging- bzw. Auditingfunktionen, die nur dem Safe-Eigentümer zustehen. Die wesentlichen Funktionen dienen zur Freigabeverwaltung und der Einsicht freigegebener Dokumente (Approval UI).

2.3.3 Safe-Anbieter

Der Safe-Service-Anbieter übernimmt die Bereitstellung des „elektronischen Safes“. Er schließt mit dem Safe-Eigentümer einen Vertrag, der die Einhaltung gesetzlicher Aufbewahrungsfristen und technischer Dienstgütevereinbarungen (Service Level Agreements) festlegt. Der Safe-Anbieter ist auch der Aufbewahrungsort für die Metadaten, die der Safe-Eigentümer generiert.

2.3.4 Speicher-Anbieter

Der (Safe-)Speicher-Anbieter ist die auf die Speicherung von Safe-Inhalten spezialisierte Komponente. Sie ist durch eine hohe Verfügbarkeit und Vertraulichkeit charakterisiert. Zudem beinhaltet sie Authentisierungs- und Autorisierungsmechanismen sowie Logging- und Auditingkomponenten, die zur Sicherstellung von Vertraulichkeit und Integrität genutzt werden.

2.3.5 Identitäts- und Berechtigungsmanagement

Wesentlicher Bestandteil des elektronischen Safes ist die vertrauenswürdige Nutzung. In diesem Zusammenhang ist es notwendig, sich beim Safe-Anbieter zu registrieren, und des Weiteren als Eigentümer oder Nutzer authentisiert bzw. für bestimmte Aktionen autorisiert zu werden. In der beschriebenen Architektur übernimmt eine Identitäts- und Berechtigungsmanagement-Komponente diese Funktionen. Sie ist in diesem Fall für die Authentisierung von Safe-Eigentümern und -Nutzern verantwortlich. Weiterhin erfolgt auf Basis der festgestellten Identität eine Autorisierung, die Zugriffsrechte für abgelegte oder freigegebene Daten überprüft.

2.3.6 Logging Service

Um im System des elektronischen Safes eine Nachvollziehbarkeit aller durchgeführten Aktionen zu sichern, bedarf es eines umfangreichen, globalen Loggings. Diese Daten müssen weiterhin durch Monitoringsysteme kontrolliert werden, so dass Missbrauch oder Angriffe schnell festgestellt werden können (siehe Kapitel 7).

2.3.7 Notification Service

Der Notification Service wird genutzt, um Safe-Eigentümer als auch Safe-Benutzer über für sie wichtige Ereignisse (Events) zu informieren. Beispielhaft kann hierfür etwa eine stattgefundene Freigabe oder eine Freigabeaufforderung gesehen werden.

2.4 Grundlagen des Cloud Computings

Das Konzept von Cloud Computing basiert auf der Bereitstellung von IT-Infrastrukturen, Software-Plattformen und Anwendungen, die über das Internet bezogen werden können. Dabei werden die Leistungen für den Nutzer in standardisierter Form bedarfsgerecht und flexibel konditioniert. Eigenschaften wie Mandantenfähigkeit, Zahlung nach Verbrauch, Selbstbedienung (Self-Service) und nachfrageorientierte Bereitstellung von IT-Ressourcen zeichnen Cloud-Angebote aus.

2.4.1 Dienstklassen

Bei Angeboten von Cloud Computing Dienstleistungen wird grundsätzlich zwischen den folgenden drei Dienstklassen unterschieden [DPS2010]:

Software als Dienst (Software as a Service: SaaS)

Das Angebot des Anbieters besteht in Anwendungen, d.h. der Nutzer kann „remote“ auf Software, die auf den Servern des Anbieters läuft, zugreifen. Die Nutzung der Anwendungen durch den Kunden wird vertraglich geregelt und beinhaltet die benötigten Hard- und Softwarelizenzen, sowie die Vereinbarung für die Wartung und den Betrieb der angebotenen Software. Die Administration der Software und der darunterliegenden Cloud Infrastruktur, des Netzwerks, der Server, den Betriebssystemen und des Speichers obliegt dem Cloud-Anbieter, der Kunde muss hierfür keine Ressourcen aufwenden. Auch Upgrades und Updates werden durch den Anbieter vorgenommen.

Plattform als Dienst (Platform as a Service: PaaS)

PaaS bietet dem Cloud-Kunden die Möglichkeit, eigene Anwendungen innerhalb einer Entwicklungsumgebung zu erstellen und in, durch den Cloud-Anbieter zur Verfügung gestellten, Laufzeitumgebungen auszuführen. Auch hier werden alle Administrationsaufgaben durch den Anbieter wahrgenommen. Anders als bei SaaS hat der Kunde hier allerdings die Kontrolle über die Anwendung, da er sie selbst entwickelt und konfiguriert hat.

Infrastruktur als Dienst (Infrastucture as a Service: IaaS)

In dieser Variante stellt der Anbieter Prozessleistung, Speicher, Netzwerkzugang und Rechenleistungen zur Verfügung, so dass der Kunde die Möglichkeit hat, eigene Software auf nativen Plattformen (die vom Anbieter virtualisiert werden) auszuführen. Für den Kunden entsteht dadurch die Illusion, über einen Pool oder ein Netzwerk von physikalischen Servern zu verfügen.

2.4.2 Cloud-Betriebsmodelle

Cloud-Dienste werden in der Regel über das Internet bzw. ein IP-basiertes Netzwerk bezogen. Betriebs-, Eigentums- und Organisationsaspekte können dazu führen, dass die Nutzung der Cloud nur einer eingeschränkten Anzahl von Nutzern gewährt werden kann. Abhängig von diesen Aspekten ergeben sich damit verschiedene Arten von Clouds, die sich zunächst grob in *öffentliche (public)* und *geschlossene* (*private*) Clouds unterteilen lassen. Eine Mischform stellen die *hybriden* Clouds dar, die aus einem geschlossenen Teil und einer offenen Cloud bestehen.

Akteure

Der Cloud-Anbieter, der dem Kunden die benötigte Dienstleistung anbietet.

Der Cloud-Kunde, der die Dienstleistung des Cloud-Anbieters nutzt.

2.4.3 Cloud-Referenzarchitektur

Als Grundlage für eine Referenzarchitektur wird hier auf das Eckpunktpapier des BSI verwiesen [BSI-Cloud2010].

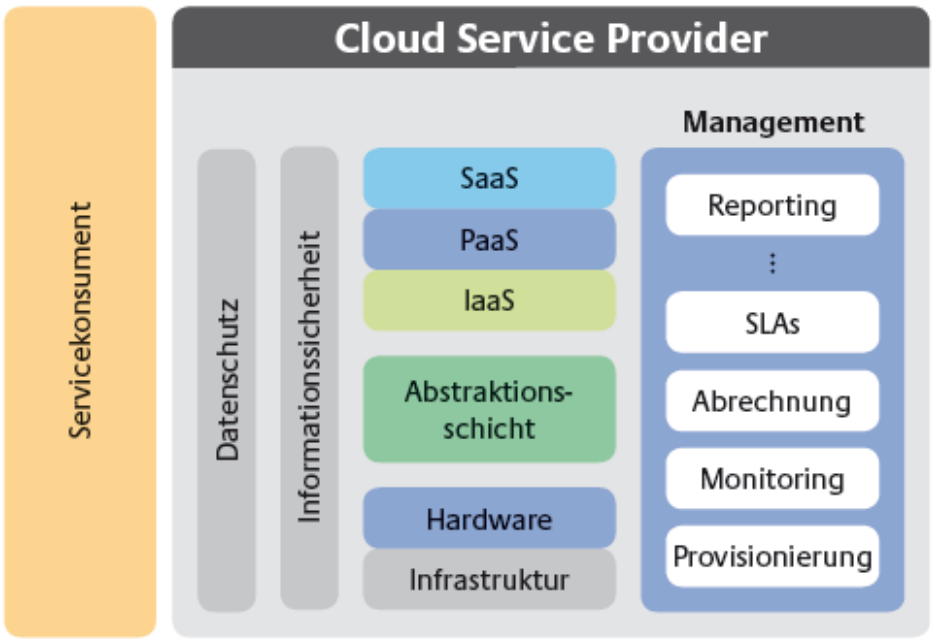

Abbildung 3– Referenzarchitektur für Cloud Computing Plattformen[2]

Betrachtet man die Referenzarchitektur, stellt man fest, dass neben den Diensten wie IaaS, PaaS und SaaS eine Reihe zusätzlicher Funktionalitäten Bestandteil des Cloud-Angebots sind, die sich insbesondere auf das Management und die Sicherheit, im Speziellen auf Datenschutz und Informationssicherheit, beziehen.

[2] vgl. BSI-Cloud2010, S. 21

3 Systeme mit Cloud-Safe-ähnlichen Konzepten

Werden die funktionalen und nicht funktionalen Anforderungen an elektronische Safes betrachtet, so stellt sich die Frage, wie sich eine Safe-Lösung von anderen Systemen abgrenzt, die ähnliche Funktionen anbieten. Im Folgenden werden einige Konzepte bzw. Systeme mit Safe-Lösungen verglichen.

3.1.1 Kollaborationsumgebungen

Kollaborationsumgebungen dienen der gemeinsamen Bearbeitung von Dokumenten durch mehrere räumlich oder zeitlich entfernte Teilnehmer. Die Bearbeitung kann zeitgleich oder zeitversetzt erfolgen.

Der Zugriff auf die gemeinsamen Dokumente wird durch Authentisierungs- und Autorisierungsverfahren gesteuert, die je nach Produkt variieren. Es wird protokolliert, welche Zugriffe und Operationen auf den Dokumenten erfolgt sind. Diese Funktionen ähneln Protokollierungsfunktionen für elektronische Safes.

Unterschiedlich ist jedoch die Zielsetzung der Anwendung, denn in der Kollaborationsumgebung ist die gleichberechtigte Zusammenarbeit wesentlich und Vertraulichkeit auf Gruppenebene wird durch Zugriffskontrollmechanismen gewährleistet. Im Gegensatz dazu ist der elektronische Safe eine persönliche, sichere und vertrauliche Ablage für Dokumente, für die nur der Safe-Eigentümer Zugriffsrechte für Dritte vergeben kann und in der das Bearbeiten von Dokumenten nicht oder nur sehr eingeschränkt möglich ist. Vertraulichkeit im e-Safe wird durch weitere Mechanismen, wie zum Beispiel Verschlüsselung, erreicht.

3.1.2 Speicherplatzverschlüsselung

Für die Verschlüsselung von Festplatten oder von ausgezeichneten Bereichen auf Datenträgern gibt es verschiedene, auf dem Markt existierende und etablierte Lösungen. Das Schutzziel Vertraulichkeit, das durch die Verschlüsselung von Daten erreicht werden soll, ist auch ein Kerngedanke elektronischer Safes. Der Schutz der Daten wird bei Festplattenverschlüsselung durch Passwörter oder Chipkarten unterstützt.

Eine Freigabe der verschlüsselten Daten für Dritte ist jedoch nicht möglich, da diese die Daten nicht entschlüsseln können. Weiterhin müssen die Eigentümer der Daten beachten, dass ohne zusätzliche Zugriffs- oder Sicherungsmechanismen ein technischer Defekt der Festplatten sehr schnell zu Datenverlust führt.

3.1.3 Online-Festplatten

Online-Festplatten, wie sie beispielsweise durch die Website Dropbox.com angeboten werden, erhalten immer stärkere Zusprüche. Das zentrale Prinzip ist die Ablage von Daten, meist Musik oder Fotos, auf persönlichen Festplatten bzw. ausgezeichneten Bereichen mit begrenzter Größe im Internet. Dabei ist nicht näher definiert, wo sich der

Speicherplatz befindet. Meist ist nur ein kostenloses Benutzerkonto nötig, um den Dienst in Anspruch nehmen zu können. Die kostenlose Nutzung unterstützt normalerweise eine geringe Festplattengröße, wobei einige Gigabyte ausreichen, um die wichtigsten persönlichen Daten zu speichern. Gegen ein monatliches Entgelt ist die Nutzung größerer Bereiche möglich.

Die Daten werden auf der Anbieterseite oft bei spezialisierten Storage-Hostern verschlüsselt gespeichert und dienen somit hauptsächlich einer Online-Datensicherung. Mit entsprechender Client-Software ist es möglich, von beliebigen anderen Endgeräten auf die Daten zuzugreifen. Zum Teil ist auch die Freigabe einer solchen Online-Festplatte für andere Nutzer der Plattform möglich. Umfangreiches Tracking oder Monitoring der Dateizugriffe, wie sie durch Kollaborationssoftware gegeben sind, existieren hier nicht. Zudem ist eine feingranulare Einstellung von Berechtigungen für Dritte (zum Beispiel der Zeitraum für den Zugriff) für die Daten nicht möglich.

3.1.4 De-Mail

De-Mail-Dienste unterstützen einen sicheren, vertraulichen und nachweisbaren Geschäftsverkehr für Bürger, Unternehmen und Verwaltungen im Internet. Die rechtliche Grundlage für De-Mail wird durch das De-Mail-Gesetz „Gesetz zur Regelung von De-Mail-Diensten und zur Änderung weiterer Vorschriften“[Bund2011] geschaffen. Die Funktionsweise von De-Mail wird durch die Technischen Richtlinien des [BSI-Mail] definiert.

De-Mail-Dienste umfassen den Postfach- und Versanddienst für sichere elektronische Post und optional einen Identitätsbestätigungsdienst und eine Dokumentenablage.

Nur akkreditierte Dienstanbieter können De-Mail-Dienste bereitstellen (derzeit befinden sich in der Akkreditierung die Deutsche Telekom, Deutsche Post, United Internet mit GMX und Web.de sowie die Francotyp-Postalia (Mentana Claimsoft)). Einem Benutzer wird dabei ein De-Mail-Konto zur persönlichen Verfügung bereitgestellt und eine De-Mail-Adresse zugeordnet. Die De-Mails werden zwischen den beteiligten De-Mail-Dienstanbietern verschlüsselt übermittelt. Eine Ende-zu-Ende Verschlüsselung zwischen Sender und Empfänger ist optional ebenfalls möglich. Versand- und Empfangsbestätigungen können ausgestellt werden. Die De-Mail-Adresse sowie Angaben zum Nutzer und die für die Verschlüsselung notwendigen Informationen können in einem Verzeichnisdienst des Dienstanbieters bereitgestellt werden. De-Mail ist ein geschlossenes Kommunikationssystem.

Die De-Mail-Dokumentenablage (vormals auch als De-Safe bezeichnet) ist eine optionale Komponente, die von einem De-Mail Dienstanbieter angeboten werden kann. Diese Dokumentenablage ist einem Benutzer persönlich zugeordnet. Dokumente können eingestellt (aus dem Postfach oder vom lokalen Rechner des Nutzers) und ausgelesen, kategorisiert und gesucht werden. Die Dokumente werden verschlüsselt abgelegt. Protokolle über die Aktivitäten bezüglich der Dokumentenablage werden vom De-Mail-Dienstanbieter bei Bedarf signiert bereitgestellt. Die Autorisierung für den Zugang zur

Dokumentenablage erfolgt über die Anmeldung an dem De-Mail-Konto mit dem Authentifizierungsniveau „normal“ oder „hoch“. Basierend auf dem Authentifizierungsniveau können Sichtbarkeit und Zugriff auf die eigenen Dokumente in der Dokumentenablage eingeschränkt sein. Die Dokumentenablage ist nur in Verbindung mit De-Mail nutzbar.

Derzeit werden die technischen Richtlinien des BSI für die Dokumentenablage überarbeitet. Die Dokumentenablage wird dabei so erweitert, dass Dokumente für Dritte freigegeben und von Dritten eingestellt werden können. Für die Authentisierung von Dritten und den Transport der Dokumente zwischen den kommunizierenden De-Mail-Nutzern werden auch zukünftig die De-Mail-Dienste verwendet. Ebenfalls ist es geplant, eine vertrauenswürdige Langzeitspeicherung zu ermöglichen.

Die De-Mail-Dokumentenablage und ein elektronischer Safe unterstützen beide eine sichere Ablage von Dokumenten. Im Gegensatz zum elektronischen Safe ist die De-Mail-Dokumentenablage jedoch nur im De-Mail-Kontext nutzbar. Eine Einbindung in beliebige Prozesse ist zwar angedacht, erfordert allerdings immer die Anbindung der Prozesse an das De-Mail-System. Ebenfalls ist die De-Mail-Dokumentenablage anbieterspezifisch implementiert. Ein Wechsel zwischen verschiedenen De-Mail-Dienstanbietern erfordert das Exportieren bzw. Importieren der gesamten Dokumentenablage.

Der Start des De-Mail-Dienstes ist für Ende 2011 geplant.

3.1.5 E-Postbrief

Der E-Postbrief der Deutschen Post AG [Post2011] bietet ähnliche Funktionen wie der De-Mail Dienst und soll eine verbindliche, verlässliche und vertrauliche e-Mail Kommunikation zwischen zwei Parteien (Privatkunden und Geschäftskunden) ermöglichen. Dem Nutzer wird ein elektronischer Briefkasten (Postfach) zum Empfangen und Versenden von E-Postbriefen, e-Mails und Faxen zur Verfügung gestellt. Der verfügbare Speicherplatz ist auf 100MB für jeden Bereich begrenzt.

Der Versand aller E-Postbriefe erfolgt nur zwischen E-Postbrief Adressen der registrierten und identifizierten Nutzer und ist daher ein geschlossenes Kommunikationssystem. E-Postbriefe können auch als klassische Briefe auf dem Postweg zugestellt werden. Der Absender kann zwischen verschiedenen Zusatzleistungen wählen: Einschreiben Einwurf, Einschreiben mit Empfangsbestätigung, mit hohem Absender Ident-Nachweis, persönlich verschlüsselt und persönlich signiert. Die eingesetzten Signaturverfahren erfüllen nicht die Voraussetzungen einer qualifizierten elektronischen Signatur nach dem Signaturgesetz (SigG), so dass gesetzlich vorgesehene Formerfordernisse nicht erfüllt werden.

Die Identifizierung des Kunden erfolgt mit dem PostIdent-Verfahren. Die Authentifizierung am E-Postbrief-Portal kann in verschiedenen Sicherheitsniveaus erfolgten: „normal“ mit E-Postbrief Adresse und Passwort und „hoch“ mit HandyTAN. E-Postbriefe werden verschlüsselt verschickt (Portalverschlüsselung). Die

Verbindungsstrecke vom Anwender zum E-Postbrief-System wird durch TLS (Transport Layer Security) gesichert. Auch die Ablage im Postfach erfolgt verschlüsselt.

Eine Dokumentenablage wie sie optional im De-Mail-Kontext vorgesehen ist, wird nicht angeboten. Der E-Postbrief-Dienst ist seit Juli 2010 verfügbar. Die Deutsche Post wird auch De-Mail-Dienste anbieten.

3.1.6 Fazit

Teilweise werden e-Safe-ähnliche Funktionen auch von anderen Anwendungen bzw. Produkten unterstützt.

Die hohe Verfügbarkeit eines Cloud-Safes kann man mit der Verfügbarkeit von Online-Festplatten vergleichen, die sich auch in einer Cloud befinden können. Unterschiedlich ist jedoch die hohe Vertraulichkeit der Dokumente im e-Safe im Gegensatz zu Online-Festplatten, deren Schutzniveau nicht vertraglich zugesichert wird und deren Authentifizierungslevel niedrig ist (meist Username/Passwort).

Ein wesentliches Schutzziel des elektronischen Safes ist Vertraulichkeit. Die Vertraulichkeit von Daten bzw. Dokumenten wird aber auch durch Speicherplatzverschlüsselung unterstützt, wobei der verschlüsselte Speicherbereich sich auch in der Cloud befinden könnte. Bei verschlüsselten Speicherbereichen muss der Eigentümer seine kryptografischen Verschlüsselungsschlüssel so sichern, dass auch im Fehlerfall eine Entschlüsselung möglich ist. Im Gegensatz zum e-Safe unterstützt Speicherplatzverschlüsselung jedoch nicht die Freigabe für Dritte und die Einbindung der verschlüsselten Daten in Prozesse.

Die gemeinsame Bearbeitung von Dokumenten wird durch Kollaborationsumgebungen unterstützt. Diese Funktion ist jedoch für den Cloud-Safe nicht gewünscht. Ähnlich sind hier jedoch die Protokollierungsfunktionen, die die Nachvollziehbarkeit von Aktivitäten auf den Dokumenten unterstützen.

Die sichere und nachweisbare Übermittlung von Dokumenten wird durch De-Mail und den E-Postbrief unterstützt. Dabei ist das Kommunikationssystem vorgegeben. Der Cloud-Safe gestattet die Freigabe von Dokumenten für Dritte; das Kommunikationssystem für die Übermittlung der Dokumente ist jedoch nicht vorab festgelegt, sondern basiert auf den Schnittstellen, die der Cloud-Safe unterstützt.

Die Dokumentenablage im De-Mail-System ist mit einem Cloud-Safe vergleichbar, ist jedoch auf die Nutzung im geschlossenen De-Mail-System eingeschränkt.

4 Der elektronische Safe als Cloud-Dienst

Ein elektronischer Safe kann als Cloud Dienst entweder direkt von einem Cloud-Anbieter oder von jedem beliebigen SaaS-Anbieter angeboten werden. Zur besseren Verfügbarkeit mag ein Cloud-Safe-Anbieter sich verschiedener Speicher-Anbieter bedienen, die entweder in einer oder mehreren Clouds angeboten werden.

Safe-Eigentümer und Safe-Benutzer sind die Kunden des Cloud-Safe-Anbieters, wie schon detailliert in 2.1 beschrieben. Der Kunde kann eine Person oder eine Organisation (z.B. eine Behörde) sein, die den Cloud-Safe in Anspruch nimmt.

4.1 Identitätsmanagement-Anbieter

Identitätsmanagement für die Registrierung, Authentisierung und Autorisierung der Safe-Eigentümer und Safe-Benutzer kann durch den Anbieter von Cloud-Safes selbst realisiert, von externen Anbietern als Dienst genutzt oder vom Cloud-Anbieter in der Cloud bereitgestellt werden (Teil des Sicherheits-Bausteines der Referenzarchitektur in 2.4.3). Werden mehrere Cloud-Anbieter in Anspruch genommen, könnten die verschieden Cloud-Anbieter in unterschiedlichen Sicherheitsdomänen liegen. In diesem Fall müssen die Identitäts-Anbieter kooperieren (föderiertes Identitätsmanagement) und es kann in Betracht gezogen werden, einen externen Identitätsmanagementanbieter zu nutzen.

4.2 Cloud-Safe-Anbieter

Der Cloud-Safe-Anbieter bietet den Cloud-Safe als Software-as-a-Service (SaaS) an, also als vollständigen Dienst. Die gesamte Kommunikation mit den Kunden wird über die Cloud-Safe-Komponente abgewickelt. Der Cloud-Safe-Anbieter benötigt Speicher-Anbieter (*Storage Provider*), um die verschlüsselten Teildaten (*Secret Shares*) an verschiedenen Speicherorten abzulegen. Dazu bedient er sich eines oder mehrerer Speicheranbieter, die entweder innerhalb einer Cloud oder in verschieden Clouds liegen. Der Cloud-Safe selbst wird unter Benutzung der Cloud Infrastrukturdienste (IaaS) und der Cloud-Plattform-Dienste (PaaS) realisiert.

4.3 Speicher-Anbieter

Der Speicherplatz für die Teildaten wird von einem Speicher-Anbieter als Cloud-Dienst angeboten und kann von mehreren Cloud-Kunden genutzt werden. Kunden sind in diesem Fall die Vertragspartner, also entweder der Cloud-Safe-Anbieter oder der Safe-Eigentümer. Der muss eine langfristige Ablage der Teildaten ermöglichen, also von einem Cloud-Anbieter bereitgestellt werden, der u.U. zertifiziert ist und über eine solide Firmengröße entsprechend langfristig auf dem Markt sein wird. Wichtige Voraussetzung, die ein Speicher-Anbieter mitbringen muss, ist dass er offene Schnittstellen - Standardkonform zu den Richtlinien (TR-Richtlinien) des BSI - und darüber hinaus eine

Export- bzw. Importfunktion zur Verfügung stellt, die es dem Nutzer erlaubt, den Speicher-Anbieter zu wechseln.

4.4 Konfigurationsvarianten

Der Cloud-Safe-Anbieter tritt gegenüber dem Nutzer als vertrauenswürdige Instanz auf. Dabei kann der Cloud-Safe als unabhängige, vertrauenswürdige Instanz beispielsweise von einer Behörde angeboten werden oder von einem privatwirtschaftlichen Unternehmen. Wichtig ist, dass die Infrastruktur des Cloud-Anbieters die entsprechenden Sicherheitsanforderungen erfüllt; also sicherheitsrelevante Komponenten, wie sie in der Referenzarchitektur (Abbildung 3) dargestellt sind, realisiert hat. Dies sind insbesondere die Komponenten der Informationssicherheit und des Managements.

Die Bausteine des Cloud-Safes sind in Kapitel 2.3 beschrieben und können in verschiedenen Varianten als Kombination der technischen Grundbausteine konfiguriert werden. Grundsätzlich wird unterschieden zwischen der Standardvariante des Cloud-Safes, der die minimal notwendige Funktionalität anbietet, und der Komfortvariante mit zusätzlicher Funktionalität. Ein Cloud-Safe kann darüber hinaus eine weitere Komponente anbieten, die insbesondere für die Beweiswerterhaltung von Dokumenten von Bedeutung ist, die Signaturkomponente. Kriterien für die Auswahl der Standard- oder Komfortvariante sind abhängig von (1) den persönlichen Sicherheitsanforderungen (2) dem Grad des Vertrauens zum Safe-Anbieter und (3) den Kosten. Die Bausteine des Cloud-Safes sind in Kapitel 2.3 beschrieben und können zu den oben beschriebenen Varianten konfiguriert werden.

4.4.1 Standardvariante des Cloud-Safes

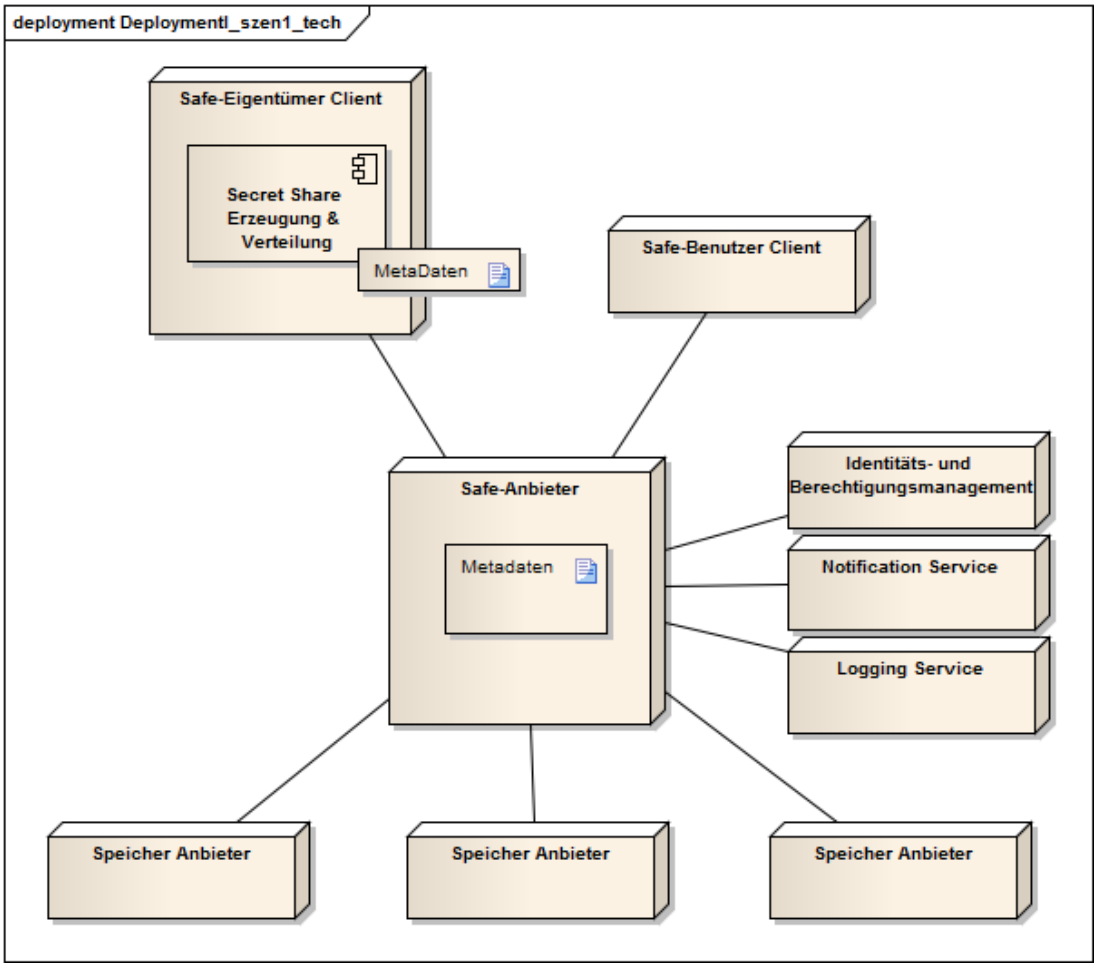

Abbildung 4: Komponenten der Standardvariante des Cloud-Safes

Bei dieser Variante übernimmt der Safe-Eigentümer selbst die Verteilung der *Secret Shares*. Das zentrale Prinzip ist hier die Freiheit des Safe-Eigentümers, beliebige Speicheranbieter zu nutzen und nach eigenen Vorstellungen zu kombinieren und zu koordinieren. Die Safe-Client Komponente muss hier die Auswahl und Verwaltung der Speicher-Anbieter unterstützen. Der Cloud-Safe Anbieter muss demzufolge dem Safe-Eigentümer eine Clientanwendung bereitstellen, die sowohl Erzeugung und Verteilung der *Secret Shares* übernimmt.

Der Safe Anbieter selbst wird insoweit als vertrauenswürdig eingestuft, dass die bei ihm gespeicherten Metadaten als sicher zu betrachten sind, wobei die Metadaten, die u. a. Informationen über die einzelnen Speicheranbieter enthalten, verschlüsselt beim Safe-Anbieter vorliegen.

Darüber hinaus bietet der Safe-Anbieter eine Freigabeverwaltung in der Form an, dass berechtigten Nutzern die notwendigen Metadaten bzw. verschlüsselten Dateien über entsprechende Schnittstellen zur Verfügung gestellt werden können.

Der Vorteil dieser Konfiguration besteht darin, dass der Nutzer, in diesem Fall der Safe-Eigentümer, die volle Kontrolle hat und die Geheimnisteile (*Secret Shares*) selbst an die vom Safe-Anbieter zur Verfügung gestellten Speicheranbieter verteilt. Dazu benötigt der Client jedoch eine umfangreiche Clientsoftware, welche die Verschlüsselung und Aufteilung sowie die Entschlüsselung und Rekonstruktion der Dateien ermöglicht. Das Schlüsselmanagement wird dabei in der Client-Anwendung bzw. vom Nutzer (Safe-Eigentümer) übernommen.

4.4.2 Komfortvariante des Cloud-Safes

In dieser Konfiguration hat der Nutzer (Safe-Eigentümer) des Cloud-Safes großes Vertrauen in den Anbieter und überlässt dem Anbieter die Verschlüsselung und die Verteilung. Verschlüsselungs- und Aufteilungsalgorithmen werden im Cloud-Safe implementiert und auch das Schlüsselmanagement liegt im Verantwortungsbereich des Cloud-Safes. In diesem Fall sind sehr hohe Sicherheitsanforderungen an die Einsatzumgebung sowie den Betrieb des Cloud-Safes zu stellen.

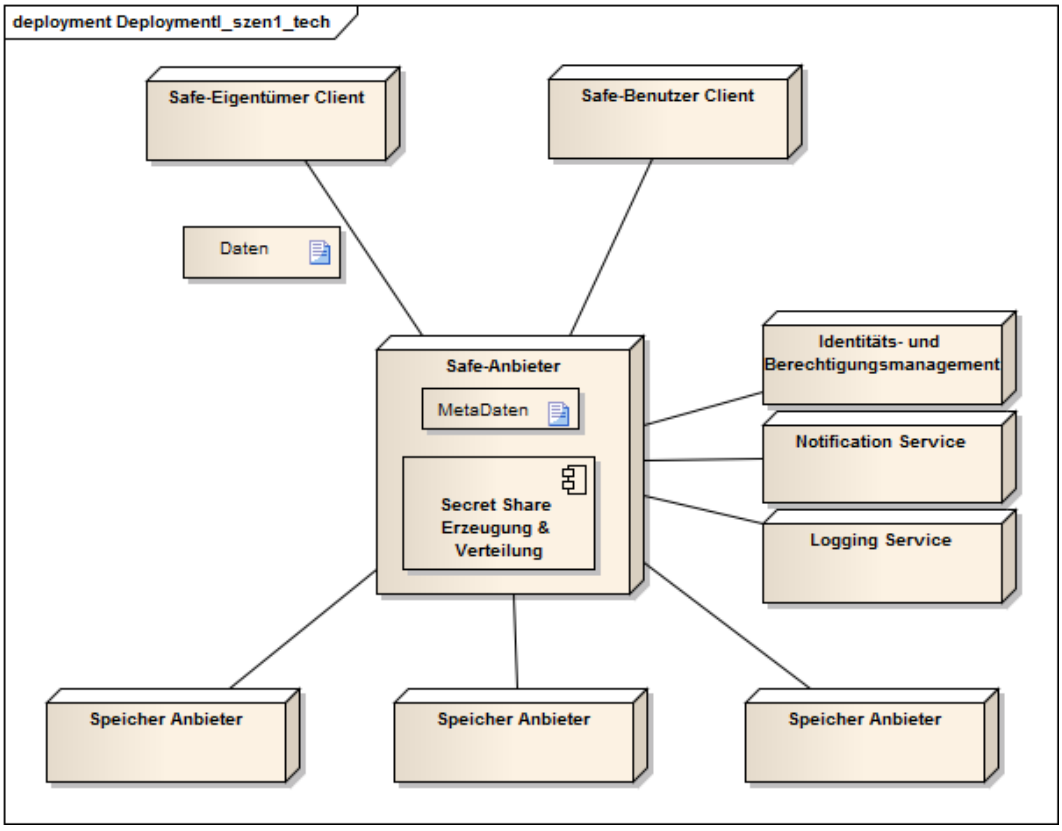

Abbildung 5: Komponenten der Komfortvariante des Cloud-Safes

Der Vorteil dieser Konfiguration ist, dass die Client-Anwendung auf ein Minimum reduziert werden kann. Hier müssen keine besonderen kryptografischen Funktionen implementiert werden.

Der Nachteil dieser Konfiguration ist darin zu sehen, dass die Übertragung der Daten zum Cloud-Safe zusätzlich gesichert werden muss, und die Daten beim Cloud-Safe Anbieter im Klartext vorliegen.

Diese Konfiguration ist beispielsweise für einen unternehmensinternen Betrieb vorstellbar, bei dem Daten lediglich über das unternehmenseigene Netzwerk an den Cloud-Safe übertragen werden.

4.4.3 Safe-Anbieter mit Signaturservice

Als Zusatzfunktionalität kann der Cloud-Safe-Anbieter einen Service für die Beweiswerterhaltung von elektronischen Signaturen anbieten. Dieser umfasst eine Neusignierung gemäß §17 Signaturverordnung und bietet die Signaturerneuerung durch Übersignatur gemäß BSI TR-03125 „Beweiswerterhaltung kryptographisch signierter Dokumente" [BSI2011]. Die Integration kann gemäß [HKL2009], wie für den elektronischen Safe I [LW2009] realisiert, durchgeführt werden.

Im Falle der Komfortvariante des Cloud-Safes werden die Dokumente bei Bedarf übersigniert.

Handelt es sich lediglich um eine Standardvariante des Cloud-Safes, übernimmt dieser die Benachrichtigung der Nutzer.

4.5 Bewertung der Konfigurationen

4.5.1 Vertraulichkeit

Vertraulichkeit kann in zwei Dimensionen betrachtet werden. Einerseits sollen die Inhalte der Kommunikation vertraulich bleiben, anderseits müssen die Kommunikationswege sicher sein. Um die Vertraulichkeit zwischen den Komponenten zu gewährleisten, muss eine Verschlüsselung der genannten Kommunikationskanäle erfolgen. Dies bedeutet auch, dass zwischen Safe-Eigentümer, Cloud-Safe und Benutzer ein sicherer Transportweg zur Verfügung stehen muss.

Zur Sicherstellung der Vertraulichkeit beim Cloud-Safe-Anbieter ist eine Kombination aus perfektem *Secret Sharing* und Verschlüsselung notwendig, wie in Kapitel 5 ausführlich diskutiert wurde. Dazu muss der Cloud-Safe-Anbieter die Sicherheitsparameter k, n sowie die Verschlüsselungsverfahren dem Safe-Eigentümer zur freien Auswahl zur Verfügung stellen. Der Cloud-Safe-Anbieter kann dabei Standardwerte vorgegeben, die der Safe-Eigentümer gemäß seinen Präferenzen anpassen kann. Hier ist ein klarer Trade-off zwischen zu speichernder Datenmenge bzw. Performanz und Sicherheit zu sehen. Dabei ist anzumerken, dass durch Anheben von k die Sicherheit, durch Anheben von n die Redundanz und damit die Verfügbarkeit verbessert wird (siehe auch Abschnitt 6.6). Um eine möglichst effiziente Nutzung des Cloud-Safes zu ermöglichen, muss der Safe-Eigentümer die Dateien anhand ihrer Vertraulichkeit einstufen und die Parameter entsprechend wählen.

Bei der Standardvariante des Cloud-Safes kann der Safe-Eigentümer einen oder mehrere Speicher-Anbieter gemäß seiner Sicherheitsanforderung in Relation zu den Kosten auswählen um die entsprechenden „Shares“ zu verteilen. Die Datei wird dann in der Clientanwendung geteilt und verschlüsselt und anschließend an die Speicheranbieter übertragen. Dies bedeutet, dass der Safe-Eigentümer eine umfangreiche Clientanwendung benötigt.

Im Fall der Standardvariante ist zu beachten, dass der Safe-Eigentümer selbst für die Verwaltung der Schlüssel zuständig ist. Das ist insbesondere dann von Bedeutung, wenn es um Langzeitarchivierung geht. Der Safe-Eigentümer muss selbst sicherstellen, dass seine Schlüssel entsprechend lange gültig sind. Im Gegensatz dazu übernimmt bei der Komfortvariante der Cloud-Safe-Anbieter diese Rolle.

Das heißt, dass im Fall der Langzeitarchivierung zwar der Transportweg zwischen Safe-Eigentümer und Cloud-Safe-Anbieter verschlüsselt erfolgt, der Cloud-Safe-Anbieter jedoch das Schlüsselmanagement übernimmt und die Daten mit Sitzungsschlüsseln verschlüsselt, um sie an die entsprechenden Speicheranbieter weiterzugeben. Das Umverschlüsseln erfolgt direkt beim Cloud-Safe-Anbieter. In der Standardvariante müsste der Safe-Anbieter eine erweiterte Clientanwendung zur Verfügung stellen, die eine Umverschlüsselung, falls erforderlich, vornimmt.

In beiden Varianten muss der Cloud-Safe-Anbieter für die Freigabe an berechtigte Nutzer Zugriff für einen vom Safe-Eigentümer festgelegten Zeitraum ermöglichen. Die für die Freigabe genutzten Schlüssel können vorab vereinbart sein oder müssen dem zugriffsberechtigten Nutzer übergeben werden. Bei der Standardvariante muss der Eigentümer (Clientanwendung des Eigentümers) über den Cloud-Safe-Anbieter den öffentlichen Schlüssel austauschen. In der Komfortvariante kann das vom Cloud-Safe-Anbieter übernommen werden.

4.5.2 Verfügbarkeit

Unabhängig von der Variante stellt der Cloud-Safe immer einen „Single Point of Failure“ dar, falls die Cloud, und damit auch der Cloud-Safe, nicht erreichbar ist.

Nicht so kritisch anzusehen ist es, wenn ein Speicheranbieter ausfällt, da durch die redundante Speicherung der *Secret Shares* der Ausfall einzelner Speicheranbieter nicht ins Gewicht fällt.

4.5.3 Integrität

Um sicherzustellen, dass die Daten vollständig und unverändert sein sollen, wird zum einen ein Schutz von Übertragungsfehlern verlangt, zum anderen ein Schutz vor vorsätzlicher Veränderung. Während Betriebssicherheit in dieser Studie vernachlässigt werden kann, werden zum Schutz vor vorsätzlicher Veränderung in Kapitel 7.2 potentielle Angriffsszenarien besprochen

Unabhängig davon wird vom Cloud-Safe Anbieter erwartet, dass in der Cloud selbst entsprechende Dienste zur Verfügung stehen, die beispielsweise Zugriffsversuche Dritter auf Dateien protokolliert und den Safe-Eigentümer auch entsprechend informiert. So sind Logging-Dienste und Identity-Management-Dienste Grundvoraussetzung eines Cloud-Safe Angebots.

4.5.4 Fazit

Die Diskussion der verschiedenen Schutzziele hat gezeigt, dass es Vor- und Nachteile des ersten vorgeschlagenen Szenarios gibt, welches den kombinierten Safe-Provider vorsieht. Dieser beinhaltet in einer organisatorischen Einheit alle relevanten Safe-Services, die Nutzung der Speicher-Anbieter, die verschiedenen Security-Services sowie die Safe-Provider-Komponente, die entsprechende Anfragen kapselt bzw. koordiniert.

Gerade die einheitliche Kontrolle erleichtert es, Identity-Management, Integritätsstandards sowie Logging-Funktionalitäten und Ausfallsicherheit einfacher durchzusetzen, da diese über alle Safe-Services einheitlich gestaltet und integriert werden können. Andersrum muss durch Zurechenbarkeit im Bereich der Safe-Transaktionen sichergestellt werden, dass rechtliche Gültigkeit nachgewiesen werden kann, wodurch die Feststellung der Identität grundlegender Bestandteil ist.

Es ist nötig, alle Safe-Services so umzusetzen, dass ein beliebiger Wechsel erfolgen kann.

Ein weiteres Problem stellt die Sicherung der Verfügbarkeit dar. So kann der Ausfall eines Safe-Services innerhalb des Safe-Providers bereits dazu führen, dass die gesamte Funktionalität eingeschränkt ist. Ein Arbeiten mit einer Menge von verschiedenen Safe-Providern sowie dem Einbau verschiedener Redundanz bzw. Secret Sharing Algorithmen ist dadurch unumgänglich.

Ein großer Vorteil bietet sich jedoch für den Nutzer des Safe-Providers. Gemessen an den nötigen Kommunikationsbeziehungen muss nur eine Kommunikation mit der Safe-Provider Komponente erfolgen, um den elektronischen Safe in Anspruch nehmen zu können. Der Aufwand für Einrichtung bzw. Wartung der beteiligten Services liegt damit bei dem Safe-Provider. Abschließend ist also zu sagen, dass durch die Abgabe der Verantwortung in den Bereich des Safe-Providers die Nutzung für den Safe-Eigentümer stark vereinfacht werden kann. Gleichzeitig steigt jedoch die Abhängigkeit, da bereits Teile der Safe-Services zum Ausfall des gesamten Safe-Providers führen können und es zudem Probleme im Datenschutzbereich geben kann.

5 Sicherung der Vertraulichkeit

Um die langfristige Vertraulichkeit von im Cloud-Safe gespeicherten Dateien sicherzustellen, wird ein von Schneier [Sch1996] empfohlenes Verfahren zur Kombination von Blockchiffren implementiert. Dabei handelt es sich um eine Kombination aus perfektem Secret Sharing und Verschlüsselung. Dabei wird eine Datei in sogenannte Shares zerlegt. Für die Rekonstruktion der Datei wird dabei eine Mindestanzahl von k Shares benötigt. Für weniger als k Shares ist die perfekte Sicherheit garantiert. Nun wird jedes Share zusätzlich mit einem eigenen Verschlüsselungsalgorithmus und einem unabhängig gewählten Verschlüsselungsschlüssel verschlüsselt. Damit ist garantiert, dass der Bruch von weniger als k Verschlüsselungsalgorithmen die Vertraulichkeit der Daten nicht gefährdet.

In diesem Kapitel beschreiben wir das Verfahren in vereinfachter Form. In Kapitel 6 folgt eine ausführliche theoretische Betrachtung und Sicherheitsanalyse.

5.1 Effiziente Realisierung ohne Redundanz

Im Folgenden wird ein sehr effizientes Verfahren vorgestellt, welches direkt auf [Sch1996] zurückgeht und auf der perfekten Sicherheit des One-Time-Pad [Ver1926, Sha1949] Verschlüsselungsverfahrens beruht. In diesem Fall werden $k-1$ Bitstrings derselben Länge wie die geheime Nachricht zufällig und gleichverteilt gewählt und bilden die ersten $k-1$ Shares. Das $k$te Share ist dann die binäre Addition (XOR) mit der geheimen Nachricht. Die binäre Addition jeder beliebigen Untermenge von weniger als k Shares ist damit eine OTP Verschlüsselung der geheimen Nachricht und damit informationstheoretisch sicher.

Sei o.B.d.A l die Länge der zu verschlüsselnden Datei s in Bitrepräsentation. Weiter sei $\Sigma^l = \{0,1\}^l$ die Menge aller Bitstrings der Länge l. $b \overset{\$}{\leftarrow} \Sigma^l$ bedeutet, dass b uniform und gleichverteilt zufällig aus der Menge Σ^l gewählt wird. $\mathcal{C}_i$ bezeichne ein Verschlüsselungsverfahren bestehend aus einer Verschlüsselungsfunktion $\mathrm{E}_{i,key_i}(*)$ und einer Entschlüsselungsfunktion $\mathrm{D}_{i,key_i}(*)$ für den Schlüssel key_i der Länge l_2 mit $\mathrm{D}_{i,key_i}\left(\mathrm{E}_{i,key_i}(x)\right) = x$. „$\oplus$“ bezeichnet die binäre Addition XOR. Dann kann die Konstruktion der Shares wie folgt beschrieben werden:

Verschlüsselung:

- $\{b_1, b_2, \dots, b_{k-1}\} \overset{\$}{\leftarrow} \Sigma^l$
- $b_k = s \oplus b_1 \oplus b_2 \oplus \dots \oplus b_{k-1}$
- $\{key_1, key_2, \dots, key_k\} \overset{\$}{\leftarrow} \Sigma^{l_2}$
- Wähle k verschiedene Verschlüsselungsverfahren $\mathcal{C}_i, i = 1, \dots, k$

- $c_i = \mathrm{E}_{i,key_i}(b_i),\ i = 1, \dots, k$

Entschlüsselung:

- $b_i = \mathrm{D}_{i,key_i}(c_i),\ i = 1, \dots, k$
- $s = b_1 \oplus b_2 \oplus \dots \oplus b_k$

5.2 Realisierung mit Redundanz

Das Verfahren lässt sich ebenso mit einem sogenannten Threshold Secret Sharing Verfahren realisieren. Bei einem Threshold Verfahren werden insgesamt n Shares erzeugt, wobei jede beliebige Menge von k Shares eine Rekonstruktion ermöglicht. Für weniger als k Shares ist jedoch die perfekte Sicherheit garantiert. Im Folgenden wird Shamir's Secret Sharing Verfahren zur Konstruktion der Shares verwendet, wie es für den eSafe [LW2009] genutzt wurde.

5.2.1 Shamir's Secret Sharing Verfahren

Zunächst wird Schamir's Secret Sharing Verfahren [Sha1979] kurz dargestellt. Bezeichne $\boldsymbol{S} = \{\mathrm{Share}(*), \mathrm{Recover}(*)\}$ Schamir's Secret Sharing Verfahren mit einer Funktion $\mathrm{Share}(*)$ zur Konstruktion der Shares und einer Funktion $\mathrm{Recover}(*)$ zur Rekonstruktion des Geheimnisses, dann sind die Funktionen wie folgt beschrieben:

Share$(*)$**:**
Sei F ein algebraischer Körper, $s \in F$ das Geheimnis, $k, n \in \mathbb{N}$ mit $k < n$.

Wähle $k-1$ geheime, unabhängig und gleichverteilte Zufallszahlen $a_i \in F, i = 1,2, \dots, k-1$ mit $a_0 = s$. Diese bilden die Koeffizienten des Polynoms

$$f(x) = \sum_{i=0}^{k-1} a_i\, x^i = s + \sum_{i=1}^{k-1} a_i\, x^i,$$

d.h. das Geheimnis ist der konstante Term des Polynoms.

Wähle n paarweise verschiedene IDs $x_i \in F, i = 1,2, \dots, n$. Durch Auswertung $f(x_i)$ des Polynoms an den Stellen x_i erhält man n Paare $(x_i f(x_i)) = (x_i, y_i)$ für $i = 1,2, \dots, n$. Dabei ist y_i das Share zur ID x_i.[3] Die x_i sind keine geheime Information und können daher für den Cloud-Safe einmal fest gewählt werden.

Um eine Datei beliebiger Länge zu teilen, wird diese wie von Miyamoto et al. [MDN2006] vorgeschlagen in b Blöcke der Länge m Bits aufgeteilt. Jeder Block wird einzeln als

[3] Hier ist anzumerken, dass es ebenso möglich ist, $k-1$ Shares (wie bei der OTP Variante) zufällig zu wählen und mit diesen und dem Geheimnis s das entsprechende Polynom durch Interpolation zu bestimmen. Damit können dann die fehlenden $n-k+1$ Shares berechnet werden.

Geheimnis s betrachtet. Mit obiger Gleichung berechnet man in b Iterationen für feststehende IDs die zugehörigen Geheimnisteile. Die Geheimnisteile mit selber ID werden daraufhin in einer Datei zusammengefasst und bilden ein sogenanntes Share Y_i zur ID x_i. Insgesamt gibt es also n Shares (bestehend aus je b Geheimnisteilen).

Recover$(*)$:
Um eine Datei zu rekonstruieren, müssen alle b Blöcke einzeln rekonstruiert werden, in die die Datei zuvor aufgeteilt wurde. Jeder dieser Blöcke bildet jeweils ein Geheimnis s.
Um s zu rekonstruieren, benötigt man k Paare (x_i, y_i), $\mathrm{i} = 1, \dots, k$ aus ID und zugehörigem Geheimnisteil.

Das Geheimnis kann nun durch Lagrange-Interpolation rekonstruiert werden:

$$s = f(0) = \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{y_i} \prod_{\mathrm{l}=1, \mathrm{l}\neq\mathrm{i}}^{\mathrm{k}} \frac{\mathrm{x_l}}{\mathrm{x_l} - \mathrm{x_i}}$$

5.2.2 Verschlüsselungsverfahren

Nun kann das Verschlüsselungsverfahren wie folgt beschrieben werden:

Verschlüsselung:

- $\{Y_1, Y_2, \dots, Y_n\} = \mathrm{Share(s)}$
- $\{key_1, key_2, \dots, key_n\} \overset{\$}{\leftarrow} \Sigma^{l_2}$
- Wähle n verschiedene Verschlüsselungsverfahren $\boldsymbol{C_i}, i = 1, \dots, n$
- $c_i = \mathrm{E}_{i,key_i}(Y_i), i = 1, \dots, n$

Die c_i bilden die Schlüsseltexte.

Entschlüsselung:

- Wähle k paarweise verschiedene $c_i, \dots c_k$ aus den n verfügbaren Schlüsseltexten aus.
- $Y_i = \mathrm{D}_{i,key_i}(c_i), i = 1, \dots, k$
- $s = \mathrm{Recover}(Y_1, \dots, Y_k)$

5.3 Bewertung der Verfahren

Die Hauptunterschiede der vorgeschlagenen Verfahren liegen in der Erzeugung der Geheimnisteile. Die zusätzliche mögliche Redundanz des zweiten Verfahrens erfordert einen erhöhten Rechenaufwand bei Zerteilung und Rekonstruktion. Jedoch bleibt eine Datei auch rekonstruierbar solange maximal $n - k$ Shares verloren gehen, und verlorene Shares können wiederhergestellt werden. Beide Verfahren haben gemein, dass die zu speichernde Datenmenge vervielfacht wird, wobei gewünschte Redundanz eine zusätzliche Vervielfachung bedeutet.

Bezüglich der Vertraulichkeit sei hier auf Kapitel 6 verwiesen. Zusammengefasst bieten beide Verfahren perfekte Sicherheit gegen Angreifer, welche nur in Besitz von weniger als k Geheimnisteile gelangen können. Damit eignet sich das Verfahren besonders für verteilte Umgebungen wie dies beim Cloud Computing der Fall ist. Ist ein Angreifer im vollen Besitz aller Geheimnisteile, so ist die Vertraulichkeit sichergestellt, solange weniger als k Verschlüsselungsverfahren gebrochen sind. Darüber hinaus schützt das Verfahren durch die Randomisierung vor Angriffsmöglichkeiten gegen einzelne Verschlüsselungsverfahren. Die Sicherheit gegen Brute-Force Angriffe kann proportional zum Sicherheitsparameter k gesteigert werden.

Mit dem Verfahren, basierend auf Shamir‘s Secret Sharing, kann zusätzlich zu den Verfügbarkeits-Mechanismen auch Speicher-Anbieter Redundanz eingebracht werden. Der Verlust von $n - k$ Shares ist unkritisch. Es bleibt daher noch Zeit, die Daten zu sichern sollte der Verlust einzelner Shares registriert werden. Darüber hinaus wird die Verfügbarkeit verbessert, sollte zum Zeitpunkt des Abrufes ein Teil des Cloud-Netzwerkes nicht erreichbar sein.

5.4 Langfristige Sicherheit

Im Allgemeinen wird von symmetrischen Verschlüsselungsverfahren angenommen, dass diese resistent gegen Quantencomputer sind [Ber2009]. Es gibt zwar Grovers Algorithmus [Gro1997], jedoch kann dessen Effekt durch Verdopplung der Schlüssellänge kompensiert werden [Ber2009]. Der Schlüsselraum unseres Verfahrens kann beliebig groß gewählt werden, um den Anstieg der Rechenleistung, welche zukünftigen Angreifern zur Verfügung steht, zu antizipieren. Bis heute sind Sicherheitsniveaus von 256 Bit weit außerhalb der Reichweite von praktikablen Angriffen und 512 Bit oder mehr werden für die nächsten Jahrzehnte als sicher betrachtet [LV2001]. Die hier dargestellten Verfahren sind darüber hinaus resistent gegen Angriffe auf einzelne Algorithmen. Damit sind Angriffe auf die Verfahren weit komplexer als auf einzelne Algorithmen, da diese mehrere verschiedene Algorithmen gleichzeitig berücksichtigen müssen und es bleibt ein offenes Problem, ob solche Angriffe überhaupt existieren (siehe Abschnitt 6.5.1). Die Vermeidung von CCA-Angriffen kann darüber hinaus auch für langfristige Anwendungen garantiert werden, da Signaturen erneuert werden können, wenn diese im Begriff sind, unsicher zu werden.

Die Verfahren sind für die langfristig sichere Aufbewahrung besonders geeignet, da die Sicherheit nicht allein von einem Kryptosystem abhängt, dessen Sicherheit über die Zeit stark abnehmen kann. Es werden die Stärken verschiedener Kryptosysteme kombiniert, ohne deren mögliche Schwachpunkte auf die anderen Kryptosysteme zu übertragen.

Wird im Zeitverlauf eines der eingesetzten Verschlüsselungsverfahren durch Bekanntwerden von Angriffen unsicher, so lassen sich die Angriffe im Allgemeinen nicht direkt auf die hier vorgeschlagenen Verfahren übertragen. Nichtsdestotrotz können präventive Maßnahmen unternommen werden, indem der mit dem unsicheren Verfahren verschlüsselte Geheimnisteil gelöscht und durch ein neues, mit einem als sicher geltenden Verfahren verschlüsselten Geheimnisteil ersetzt wird. Auch wenn das gelöschte Geheimnisteil noch einige Zeit in Backups bestehen bleibt, wird es im Laufe der Zeit gelöscht und die Zugreifbarkeit durch einen Angreifer stark eingeschränkt. Die Vertraulichkeit ist während dieser Übergangszeit durch die anderen Verfahren sichergestellt.

Durch die perfekte Sicherheit gegenüber der Gruppe von Angreifern, die nicht in der Lage sind k zusammengehörige Schlüsseltexte zu identifizieren, ist die langfristige Vertraulichkeit gegenüber dieser Gruppe garantiert. Diese Eigenschaft ist unabhängig von zukünftigen Entwicklungen in der Kryptoanalyse oder Leistungssteigerung von Computersystemen.

5.5 Empfehlungen

Die Sicherheitsparameter k, n sowie die Verschlüsselungsverfahren sollten vom Nutzer wählbar sein. Dem Benutzer werden Standardwerte vorgegeben, die er gemäß seinen Präferenzen anpassen kann. Hier ist ein klarer Trade-off zwischen zu speichernder Datenmenge bzw. Performanz und Sicherheit zu sehen. Dabei ist anzumerken, dass durch Anheben von k die Sicherheit, durch Anheben von n die Redundanz und damit Verfügbarkeit verbessert wird (siehe auch Abschnitt 6.6). Um eine möglichst effiziente Nutzung des Cloud-Safes zu ermöglichen, sollten die Dateien anhand ihrer Vertraulichkeit eingestuft und die Parameter entsprechend gewählt werden. Als Verschlüsselungsverfahren werden aktuelle Blockchiffren wie AES und TwoFish mit Schlüssellängen von 256 Bit empfohlen.

Die Daten sollten möglichst unabhängig voneinander in der Cloud verteilt werden. Dadurch wird es einem Angreifer erschwert, zusammengehörige Schlüsseltexte zu identifizieren, insofern er nicht direkten Zugriff auf die Verwaltungsdaten des Cloud bzw. Safe-Anbieters hat. Werden die Schlüsseltexte über mehrere Clouds hinweg verteilt, so kann die Sicherheit weiter erhöht werden, da ein Angreifer mehrere Cloud-Systeme kompromittieren müsste, um die Dateien angreifen zu können.

Eine möglichst große Heterogenität bezüglich der Storage-Server sollte angestrebt werden.

6 Theoretischer Hintergrund und Sicherheitsanalyse des vorgeschlagenen Verschlüsselungsverfahrens

6.1 Einleitung

6.1.1 Motivation

Bestimmte elektronische Daten müssen für eine sehr lange Zeit geheim gehalten werden. Beispiele hierfür sind elektronische Gesundheitsdaten und elektronische Wahlzettel, welche während der gesamten Lebenszeit der betreffenden Person oder sogar über den Tod hinaus vertraulich zu behandeln sind. Das Problem ist, dass alle praktischen, heute bekannten Verschlüsselungsverfahren für diese Anwendung ungeeignet sind. Deren Sicherheit nimmt aufgrund von Fortschritten in der Kryptoanalyse oder steigender Ressourcenverfügbarkeit (Rechenkapazität, Speicherplatz, etc.) im Zeitverlauf ab. Darüber hinaus besteht die Möglichkeit, dass die Algorithmen ohne Vorwarnung gebrochen werden könnten, was zu einem plötzlichen Verlust der Vertraulichkeit führen würde.

Sobald ein Schlüsseltext einem möglichen Angreifer bekannt wird, kann die Verschlüsselung im Gegensatz zu elektronischen Signaturen nicht mehr verlängert werden, wenn der verwendete Verschlüsselungsalgorithmus oder der Schlüssel kompromittiert wird oder unsicher zu werden droht. Ein Angreifer, welcher einmal im Besitz eines bestimmten Schlüsseltextes ist, kann diesen theoretisch solange speichern, bis er in der Lage ist, den Verschlüsselungsalgorithmus zu brechen. Ein komplettes und sicheres Entfernen eines Schlüsseltextes auf Bestreben des Eigentümers kann nicht mehr garantiert werden, sobald ein Schlüsseltext die vom Eigentümer kontrollierte Umgebung verlassen hat. Die Möglichkeit der Umverschlüsselung mit einem neuen stärkeren Algorithmus besteht theoretisch nur, solange der Schlüsseltext zum Zeitpunkt der Umverschlüsselung nicht bereits von einem Angreifer kopiert worden ist und wenn die Möglichkeit besteht, den alten Schlüsseltext komplett und nicht wiederherstellbar zu löschen. Diese Anforderungen sind im Allgemeinen jedoch nicht gegeben oder können nicht garantiert werden, vor allem in verteilten Infrastrukturen, bei denen Daten über offene Netzwerke übertragen und in verteilten Server-Infrastrukturen gespeichert werden.

Damit besteht eine dringende Notwendigkeit, praktische Verfahren bereitzustellen, die es erlauben, vertrauliche Daten über sehr lange Zeiträume vertraulich zu halten und die nicht durch Fortschritte in der Kryptoanalyse oder die Entwicklung neuer mächtiger Ressourcen bedroht werden.

6.1.2 Beitrag

Im Folgenden generalisieren wir Schneiers auf dem One-Time-Pad (OTP) basierendes Verfahren verschiedene Blockchiffren zu kombinieren, welches wir OTPCC (siehe Abschnitt 6.1.3) nennen, zu einem Verfahren, welches mit beliebigen (k, n)-Threshold Secret-Sharing-Verfahren funktioniert. Dieses neue Verfahren nennen wir GSSCC

(Generalized Secret Sharing based Cipher Combining). Es handelt sich dabei ebenso um ein (k, n)-Threshold Verfahren, bei dem nur $k \leq n$ der n kombinierten Verschlüsselungsalgorithmen zur Entschlüsselung benötigt werden, was die Flexibilität und Verfügbarkeit erhöht.

Wir zeigen, dass es keine notwendige Bedingung ist, verschiedene Verschlüsselungsverfahren zu kombinieren, um eine gesteigerte Sicherheit zu erreichen. Die mehrfache Verwendung des gleichen Verfahrens mit unterschiedlichen, unabhängig gewählten Schlüsseln ist dafür ausreichend, wenn das Verschlüsselungsverfahren einige zusätzliche Anforderungen erfüllt.

Wir stellen eine genaue Sicherheitsabschätzung von GSSCC bereit. Die effektive Bitsicherheit ergibt sich aus der Summe der k kürzesten der verwendeten Schlüssel, insofern einem potentiellen Angreifer alle korrespondierenden Geheimnisteile bekannt sind, wohingegen das Verfahren perfekte Sicherheit bietet, solange ein Angreifer nicht in Besitz von mindestens k der Geheimnisteile gelangt.

Das Verfahren bietet Schutz gegen Adaptive Chosen Plaintext (CPA2), Angriffe gegen die verwendeten Algorithmen. Die Sicherheit gegen CPA2 Angriffe verbleibt auf dem initialen Niveau, bis mindestens k der Verschlüsselungsverfahren gebrochen sind, insofern ein spezieller Blockmodus (siehe Abschnitt 6.7) angewendet wird.

Das Sicherheitsniveau gegen Adaptive Chosen Ciphertext (CCA2) Angriffe sinkt jeweils entsprechend dem Sicherheitsverlust des kompromittierten Verschlüsselungsalgorithmus.[4]

Erste informationstheoretische Betrachtungen deuten darauf hin, dass bei Einsatz von GSSCC ein bestimmtes Sicherheitsniveau erhalten werden kann, selbst wenn alle Verschlüsselungsverfahren kompromittiert sind. Das bedarf weiterer Nachforschungen.

6.1.3 Verwandte Arbeiten

Der einzige heute bekannte Verschlüsselungsalgorithmus, der beweisbar sicher ist und perfekte Vertraulichkeit garantiert, ist das *One-Time-Pad* (OTP) und wurde im Jahre 1926 von Vernam [Ver1926] erfunden. Die informationstheoretische oder perfekte Sicherheit dieses Verfahrens wurde 1949 von C. E. Shannon [Sha1949] bewiesen.

Das OTP ist jedoch für die Datenspeicherung nicht praktikabel. Um perfekte Sicherheit zu erreichen, muss ein Schlüssel verwendet werden, der mindestens genau so lang wie die zu verschlüsselnde Nachricht ist [Sha1949]. Darüber hinaus darf ein Schlüssel lediglich einmal verwendet werden und muss jeweils zufällig und gleichverteilt erzeugt werden. Dies wiederrum bedeutet, dass bei der Verwendung des OTP dieselbe Menge an Schlüsselmaterial anfällt wie die zu verschlüsselnde Datenmenge. Die Schlüssel müssen

[4] Anzumerken ist hier, dass Standardmechanismen wie Signaturen der entsprechenden Klar- oder Schlüsseltexte auch bei GSSCC als Schutz gegen Chosen Ciphertext Angriffe angewendet werden können.

wiederrum vertraulich gespeichert werden, wodurch der Nutzen für die Datenspeicherung zunichte gemacht wird.

Betrachtet man den Datentransport mittels des OTP erhält man ein ähnliches Bild, wobei der Schlüsselaustausch das Problem darstellt. Um dies zu realisieren und die perfekte Sicherheit nicht zu gefährden, benötigt man spezielle Schlüsselaustauschprotokolle. Es gibt einige Ansätze für solche Systeme, jedoch sind alle mit eigenen Praktikabilitätsproblemen verbunden [BC1996, Mau1993b, CM1997, Mau1993a, Rab2005, Rab2006].

Der Einsatz von *Perfekten Secret Sharing Verfahren* (PSS) [Sti1992] ist eine weitere Methode, Daten so zu speichern, dass informationstheoretische Sicherheit garantiert werden kann (siehe auch Abschnitt 6.2.1). PSS Verfahren basieren nicht auf Verschlüsselung, sondern teilen ein Geheimnis in sogenannte *Shares* auf. Das geschieht auf eine Weise, dass der Besitz von weniger als einer bestimmten Anzahl von k Shares absolut keine Informationsgewinnung über das Geheimnis erlaubt [Sha1979], k Shares zusammen jedoch die vollständige Rekonstruktion des Geheimnisses erlauben. Der Parameter k kann dabei vom Nutzer gewählt werden. Wenn einige Shares kompromittiert werden, müssen diese ungültig gemacht werden, bevor ein Angreifer in den Besitz von insgesamt k Shares gelangen kann. In [HJK1995, SB2005] werden beispielsweise solche sogenannten *Proaktiven Secret Sharing Verfahren* vorgeschlagen. Sie basieren im Allgemeinen auf Shareerneuerung. Eine Grundvoraussetzung für den Sicherheitserhalt ist, dass die bis dahin nicht kompromittierten Shares nach der Shareerneuerung sicher gelöscht werden und damit für einen Angreifer nicht mehr erreichbar sind. Wenn das nicht garantiert werden kann, besteht die Möglichkeit, dass ein Angreifer zu einem späteren Zeitpunkt in den Besitz eines oder mehrerer der alten Shares gelangt und er diese dann gemeinsam mit den zuvor kompromittierten Shares zur Rekonstruktion des Geheimnisses verwenden kann. Die Möglichkeit zur sicheren Löschung kann jedoch in verteilten Speicherinfrastrukturen nicht immer garantiert werden. Außerdem müssen die sogenannten *Share Holder*, welche die einzelnen Shares speichern soweit vertrauenswürdig sein, dass sie nicht unautorisiert zusammenarbeiten. Letztendlich besteht dazu noch das Problem, die Shares sicher zu den Share Holdern zu transportieren.

In [Sha1949] beweist Shannon einige sehr starke Sicherheitseigenschaften von Bijektionsfamilien, wenn bestimmte Anforderungen erfüllt sind. In einfachen Worten und bezogen auf unseren Anwendungsbereich zeigt er, dass ein Angreifer unabhängig von dessen Rechenkraft und der Menge der abgehörten Schlüsseltexte, keine andere Möglichkeit hat als den Schlüssel zu raten, wenn ausschließlich gleichverteilte und unabhängige Zufallszahlen mit dem jeweiligen Verschlüsselungsverfahren verschlüsselt werden. Hierfür sind lediglich die inhärenten bijektiven Eigenschaften des Verschlüsselungsverfahrens notwendig, jedoch nicht die Verschlüsselungsstärke. Die Schwierigkeit hier ist, zu garantieren, dass diese Anforderungen erfüllt werden. Für weitere Details sei hier auf Abschnitt 6.2.2 verwiesen.

Multiple Verschlüsselung ist ein gängiger Weg, um stärkere Verschlüsselungsalgorithmen aus schwächeren zu generieren. Multiple Verschlüsselung bedeutet, einen Klartext mehrfach mit demselben Algorithmus, jedoch mit verschiedenen Schlüsseln zu verschlüsseln. Multiple Verschlüsselung wird auf verschiedene Arten angewendet, wie beispielsweise Doppel- oder Dreifachverschlüsselung mit zwei bzw. drei Schlüsseln und angewendet in verschiedenen Modi. Eine Übersicht ist in [Sch1996] zu finden, wobei 3DES [NIST1999] die wohl bekannteste Anwendung ist. Multiple Verschlüsselung kann tatsächlich die Schlüssellänge vergrößern, insofern der eingesetzte Algorithmus keine Gruppe bildet [Sch1996], was für DES in [CW1993] bewiesen wurde. Der Einsatz von n verschiedenen und unabhängigen Schlüsseln führt dabei jedoch nicht zwangsläufig zu einer Vergrößerung der Schlüssellänge um den Faktor n. Gaži und Maurer erweitern in [GM2009] die Arbeit von Bellare und Rogaway [BR2006] und zeigen, dass $n = 3$ das kleinstmögliche n ist, so dass ein wesentlicher Sicherheitsgewinn gegenüber einfacher Verschlüsselung erzielt werden kann. Der Sicherheitsgewinn bei einer Wahl von $n > 3$ wird dabei offen gelassen. Meet-in-the-Middle-Angriffe [MH1981] bringen den effektiven Gewinn an Schlüssellänge deutlich unter den Faktor drei, was wiederrum deutlich macht, dass es nicht einfach ist, das Sicherheitsniveau von Multiplen Verschlüsselungsverfahren zu bestimmen.

Sogenanntes *Kaskadieren* ist multipler Verschlüsslung sehr ähnlich. Dabei werden jedoch verschiedene Verschlüsselungsalgorithmen eingesetzt. Kaskadieren hat ähnliche Nachteile wie multiple Verschlüsselung. Es gibt beispielsweise keine Garantie, dass das Kombinieren verschiedener Algorithmen die Sicherheit erhöht. Es existieren jedoch Beweise, dass Kaskadieren mindestens so sicher ist wie der erste Algorithmus der in der Kaskade angewendet wird [MM1993], solange unabhängige Schlüssel verwendet werden. Wenn die Algorithmen kommutativ sind,[5] ist Kaskadieren mindestens so sicher wie der stärkste der eingesetzten Algorithmen [Sch1996, EG1985].

In [Sch1996] schlägt Schneier das oben erwähnte OTPCC vor, um verschiedene Blockchiffren zu kombinieren. OTPCC funktioniert folgendermaßen:

1. Ein Geheimnis wird mittels des auf OTP basierenden Secret Sharing Verfahrens aufgeteilt (siehe Abschnitt 2.1).
2. Jedes Share wird mit einer anderen Blockchiffre und einem zufällig, unabhängig und gleichverteilt gewählten Schlüssel verschlüsselt.

Aus dem Einsatz von PSS folgt, dass die Kenntnis von $k - 1$ Shares nichts über das Geheimnis verrät. Insofern das OTP zusammen mit den eingesetzten Chiffren keine Homomorphieeigenschaften (siehe Abschnitt 6.5.1) besitzt, ist es eindeutig, dass jede Chiffre gebrochen werden muss, um das Geheimnis zu entschlüsseln. Damit ist das Verfahren garantiert mindestens so sicher wie alle nicht kompromittierten Verschlüsselungsverfahren. Ein solcher Ansatz ist für verteilte Umgebungen besonders geeignet, da perfekte Sicherheit garantiert werden kann, solange ein Angreifer nicht in den

[5] Zwei Verschlüsselungsalgorithmen C, Q sind kommutativ, wenn $C_i(Q_j) = Q_l(C_m)$ für jedes i, j und entsprechende l, m gilt [34].

Besitz aller Geheimnisteile gelangt. Gelingt das trotzdem, bietet das Verfahren immer einen sehr hohen Schutz der Vertraulichkeit.

6.2 Grundlagen

6.2.1 Perfektes Secret Sharing

Perfekte Secret Sharing-Verfahren sind informationstheoretisch sicher und wurden erstmals im Jahre 1979 unabhängig voneinander von Shamir [Sha1979] und Blakley [Bla1979] vorgestellt. Während Shamir die Eigenschaften von Polynomen zur Konstruktion des Verfahrens ausnutzt, verwendet Blakley Schnitte von nicht parallelen Hyperebenen. Es ist wohlbekannt, dass bei einem informationstheoretisch sicheren Secret Sharing-Verfahren jedes Share mindestens so groß sein, muss wie das Geheimnis selbst [Kra1994]. Bezüglich Threshold Verfahren, beziehen wir uns im Folgenden auf Shamir‘s Secret Sharing-Verfahren (SSSS). Dieses wird *ideal* genannt [DD1994, BD1989], da die Shares exakt die Größe des Geheimnisses haben, was für ein PSS-Verfahren offensichtlich optimal ist. Darüber hinaus betrachten wir eine einfache geradlinige Konstruktion basierend auf dem OTP.

Zunächst geben wir eine formale Definition von (k, n)-Secret Sharing-Verfahren mit den Parametern $n, k \in \mathbb{N}$, $k \leq n$ an. $|A|$ bezeichnet dabei die Kardinalität der Menge A, also die Anzahl der darin enthaltenen Elemente. Zufallsvariablen bezeichnen wir im Folgenden mit Großbuchstaben mit Dach, bspw. $\widehat{M}$, und deren Domain mit Großbuchstaben, bspw. M. Elemente einer Domain $m \in M$ werden mit Kleinbuchstaben bezeichnet.

Im Folgenden bezeichnen $\widehat{M}, \widehat{Y}_i$ für $i = 1, \dots, n$ und $\widehat{Y}(m)$ Zufallsvariablen mit den Domains M, Y_i für $i = 1, \dots, n$ und $Y(m)$. M ist die Menge aller möglichen Klartexte und Y_i bezeichnet die Menge aller möglichen Shares mit Index i. $Y\ (m)$ ist die Menge aller möglichen gültigen Sharesets der Kardinalität n für einen Klartext $m \in M$, wobei ein Shareset von der Form $(y_1, \dots, y_n),\ y_i \in Y_i$ ist. Wir definieren $Y = \bigcup_{m \in M} Y(m)$ als die Menge aller gültigen Sharesets der Kardinalität n für beliebige Klartexte $m \in M$.

Zusätzlich definieren wir den Untermengenoperator "⊆" auf Sharesets a, b so dass $a = (y_{j_1}, \dots, y_{j_k}) \subseteq (y_1, \dots, y_n) = b$ gdw. $\{j_1, \dots, j_k\} \subseteq \{1, \dots, n\}$ und wir nennen zwei Sharesets a, b äquivalent $a \sim b$ gdw. $(|a|, |b| \geq k) \wedge (a \subseteq c) \wedge (b \subseteq c), c \in Y$ gilt. $|a|$ bezeichnet die Kardinalität, also die Anzahl von Shares in einem Shareset a.

Bei einem (k, n)-Secret Sharing-Verfahren wird für den Input $m \in M$ und Parameter n, k ein Set sogenannter Shares $(y_1, \dots, y_n),\ y_i \in Y_i$ auf eine Weise zufällig erzeugt, so dass m aus jeder Untermenge von mindestens k Shares rekonstruiert werden kann. Solch eine Menge nennen wir *gültiges Shareset*. Wenn weniger als k Shares keine Information über das Geheimnis $m \in M$ preisgeben, wird das Verfahren *perfekt* genannt. Wenn $k < n$ gilt, handelt es sich um ein (k, n)-Threshold Verfahren.

Definition 1 ((k, n)-Secret Sharing-Verfahren (SSS))**.** *Ein Secret Sharing-Verfahren* ***S*** *ist ein Tupel* $(M, Y,$ Share, Recover$)$ *mit den folgenden Eigenschaften:*

- M bezeichnet die Menge aller möglichen geheimen Klartexte.
- *Y bezeichnet die Menge aller möglichen gültigen Sharesets.*
- Share (m, n, k) *gibt für die Eingabe* $m \in M$ *und Parameter* $n, k \in \mathbb{N}, k \leq n$ *ein zufällig gewähltes Set von Shares* $y(m) = (y_1, \dots, y_n) \in Y(m)$ *für den geheimen Klartext* m *aus.*
- Recover $(y_{j_1}, \dots, y_{j_k})$ *gibt* m' *aus, mit* $m' = m$ *gdw.* $(y_{j_1}, \dots, y_{j_k}) \subseteq y(m), y(m) \in Y(m)$.

Perfektes Secret Sharing ist damit wie folgt definiert:

Definition 2 (Perfektes Secret Sharing (PSS)). *A (k, n)-Secret Sharing-Verfahren* $\boldsymbol{S} = (M, Y, \text{Share}, \text{Recover})$ *ist perfekt gdw.* $\forall y \in Y$ *und* $\hat{A} = (\hat{Y}_{j_1}, \dots, \hat{Y}_{j_v}) \subseteq (\hat{Y}_1, \dots, \hat{Y}_n)$, $a = (y_{j_1}, \dots, y_{j_v}) \subseteq y$ *gilt:*

- $\forall \{j_1, \dots j_v\} \subseteq \{1, \dots, n\}$ *mit* $v \geq k$ *gibt es ein eindeutiges* $m \in M$ *sodass*

$$Pr(\hat{M} = m | \hat{A} = a) = 1 \text{ und}$$

- $\forall \{j_1, \dots j_v\} \subseteq \{1, \dots, n\}$ *mit* $v < k, \forall m \in M$

$$Pr(\hat{M} = m | \hat{A} = a) = Pr(\hat{M} = m)$$

Auf dem OTP basierendes Secret Sharing wie in Abschnitt 5.1 ist ein triviales Beispiel für ein perfektes (k, k)-Threshold-Verfahren.

Wie bereits angemerkt, werden bei Shamir's Secret Sharing-Verfahren (SSSS) [Sha1979] die Eigenschaften von Polynomen, die über endlichen Körpern definiert sind, ausgenutzt wie in Abschnitt 5.2.1. SSSS ist ein (k, n)-Threshold-Verfahren. Die von Miyamoto et al. [MDN2006] vorgeschlagene Methode (Abschnitt 5.2.1) erlaubt eine effiziente Implementierung für beliebig große Geheimnisse.

Für SSSS kann das folgende Theorem bewiesen werden [Aza2009, TW1988, Sha1979], das aus der Konstruktion auch direkt für die OTP Variante folgt.

Theorem 1. *Seien* $y(m) = (y_1, \dots, y_n) \in Y(m)$ *die Shares für Nachricht* m. *(Aufgeteilt nach SSSS oder dem auf OTP basierenden SS). Dann sind die Shares in jeder Teilmenge* $a \subset y(m)$ *mit* $|a| < k$ *zufällig gleichverteilt und paarweise unabhängig.*

6.2.2 Ideal sichere Systeme

Wir erklären *ideal sichere Systeme* (ideal secrecy systems) [Sha1949, Sch1996] als Basis für spätere Sicherheitsbetrachtungen. Wir nehmen an, dass der Leser mit Shannons Entropiedefinition [Sha1949] vertraut ist. Zunächst sei M die Menge aller möglichen Klartexte bzw. Nachrichten und K die Menge aller möglichen Schlüssel. $m \xleftarrow{\$} M$ bedeutet dass m zufällig und gleichverteilt aus M gewählt wird.

Shannon definiert ideal sichere Systeme als solche Verschlüsselungssysteme, bei denen jeder Ciphertext Only-Angriff (selbst erschöpfende Schlüsselsuche) zu vielen gleichwahrscheinlichen Entschlüsselungen führt [Gef1965] und damit gleichzeitig zu vielen gleichwahrscheinlichen möglichen Schlüsselkandidaten. Das ist dabei unabhängig von der Menge an bekannten Schlüsseltexten. Er definiert die beiden Maße Mehrdeutigkeit der Nachricht (equivocation of message), die mit $H_E(M)$ bezeichnet wird, und Mehrdeutigkeit des Schlüssels (equivocation of key), die mit $H_E(K)$, für die durchschnittliche Anzahl sinnvoller Entschlüsselungen bzw. Schlüssel (abhängig von der Menge bekannter Schlüsseltexte) bezeichnet wird. Zu beachten ist dabei, dass $H_E(K) \geq H_E(M)$ [Sha1949] gilt, da die Entschlüsselung eines Schlüsseltextes mit verschiedenen Schlüsseln gleich sein kann.

Stark ideal sichere Systeme (strongly ideal secrecy systems) sind solche Verschlüsselungssysteme, bei denen $H_E(K)$ konstant beim Anfangswert $H(K)$, der Entropie des Verschlüsselungssystems bleibt. $H(K)$ ist ein Maß für die Größe des Schlüsselraumes. Wenn $k \overset{\$}{\leftarrow} K$, dann gilt [Sch1996]

$$H(K) = log_2|K|$$

In einfachen Worten bleibt bei einem stark ideal sicheren System selbst nach einer erschöpfenden Schlüsselsuche jeder Schlüssel $k \in K$ gleichwahrscheinlich, egal wie viele Schlüsseltexte für die Analyse zur Verfügung stehen. Dadurch ist das Beste, was ein Angreifer erreichen kann, zufällig einen der Schlüssel aus dem gesamten Schlüsselraum zu wählen. Allerdings kann es durchaus möglich sein, einen eindeutigen Schlüssel mit einem gegebenen Klartext-Schlüsseltext Paar zu bestimmen. Für weitere Details und formale Definitionen sei hier auf [Sha1949] verwiesen.

Darüber hinaus benutzt Shannon den Begriff abgeschlossenes Verschlüsselungsverfahren (closed cipher). Damit wird ein Verschlüsselungsverfahren bezeichnet, bei dem für jeden möglichen Klartext und für jeden Schlüssel genau ein valider Schlüsseltext existiert und umgekehrt.

Definition 3 (Chiffre)**.** *Ein Verschlüsselungsverfahren* $\boldsymbol{\mathcal{C}} = (P, C, K, \mathrm{E}, \mathrm{D})$ *besteht aus den Mengen* P, C, K *und einem Tupel von Algorithmen* E *und* D*:*

- P *ist die Menge aller möglichen Klartexte.*
- C *ist die Menge aller möglichen Schlüsseltexte.*
- K *ist die Menge aller möglichen Schlüssel.*
- $\mathrm{E}_k(m)$ *gibt für die Eingabe* $m \in P$ *und den Schlüssel* $k \in K$ *die Verschlüsselung* $c \in C$ *aus.*
- $\mathrm{D}_k(c)$ *gibt für die Eingabe* $c \in C$ *und den Schlüssel* $k \in K$ *die Entschlüsselung* $m \in P$ *aus.*
- $\forall k \in K,\ m \in P$ *gilt* $\mathrm{D}_k(\mathrm{E}_k(m)) = m$*.*

Definition 4 (abgeschlossenes Verschlüsselungsverfahren)**.** *Ein Verschlüsselungsverfahren* $\boldsymbol{\mathcal{C}} = (P, C, K, \mathrm{E}, \mathrm{D})$ *ist abgeschlossen, wenn gilt:*

$$\forall k \in K, p \in P\colon \mathrm{E}_k(p) \in C \wedge \forall k \in K, c \in C\colon \mathrm{D}_k(c) \in P$$

Damit formuliert und beweist Shannon folgendes Theorem.

Theorem 2. *Wenn $\boldsymbol{\mathcal{C}}$ abgeschlossen ist und jedes $p \in P$ mit derselben Wahrscheinlichkeit vorkommt, dann ist $\boldsymbol{\mathcal{C}}$ stark ideal.*

Im Allgemeinen kann die Ideal-Eigenschaft für jede natürliche Sprache approximiert werden. Für die Reduktion von Redundanzen werden dabei Kompressionsmethoden auf den Klartext angewendet [Sch1996]. Jedoch gibt es dabei verschiedene Nachteile. Ideale Systeme werden schnell sehr komplex und haben eine schlechte Fehlerfortpflanzungseigenschaft. Das jeweilige Verschlüsselungsverfahren muss genau auf die jeweilige Sprache abgestimmt sein, was aufwendige Studien erfordert. Selbst kleine Fehler oder Änderungen in der statistischen Struktur machen solche Verfahren anfällig für die Kryptoanalyse. Außerdem ist es nicht immer möglich ein ideales System mit endlicher Komplexität zu erzeugen [Sha1949] und ein einziges Klartext-Schlüsseltext Paar kann den verwendeten Schlüssel aufdecken und die ideale Sicherheit zunichtemachen.

Im Folgenden zeigen wir, wie die ideale Sicherheitseigenschaft für beliebige Sprachen genutzt werden kann, ohne Kenntnisse über die jeweilige Sprache oder hochkomplexe Transformationen des Klartextes zu erfordern. Zusätzlich verhindert das hier gezeigte Verfahren das Bekanntwerden von Klartext-Schlüsseltext Paaren von vornherein durch dessen inneren Aufbau.

6.3 GSSCC - General secret sharing based cipher combining

Bei GSSCC handelt es sich um eine generische auf Secret Sharing basierende Methode, verschiedene Verschlüsselungsalgorithmen zu einem neuen Verfahren mit bestimmten gesteigerten Sicherheitseigenschaften zu kombinieren. Für die Konstruktion wenden wir Schneiers Idee an, verschiedene Blockchiffren mittels des OTP zu kombinieren und generalisieren diese für beliebige perfekte (Threshold) Secret Sharing Verfahren.

Zunächst beschreiben wir die Verschlüsselung und Entschlüsselung des GSSCC-Verfahrens. Dafür seien $\boldsymbol{\mathcal{C}}_i = (P_i, C_i, K_i, \mathrm{E}_i, \mathrm{D}_i)$ abgeschlossene Verschlüsselungsverfahren gemäß Definition 4. $\boldsymbol{S} = (M, Y, \mathrm{Share}, \mathrm{Recover})$ sei ein beliebiges perfektes Secret Sharing-Verfahren gemäß Definition 2. Dann ist *GSSCC* wie folgt definiert.

Definition 5 (GSSCC). *General secret sharing cipher combining GSSCC ist ein Tupel* $(M, \Gamma, \Gamma^*, \Pi, \boldsymbol{S}, \boldsymbol{\mathcal{C}}, \mathrm{Enc}, \mathrm{Dec})$ *mit folgenden Eigenschaften:*

- *M ist die Menge aller möglichen Klartexte.*
- *Γ ist die Menge aller möglichen Schlüsseltexte, $\Gamma = C_1 \times \ldots \times C_n$, wobei C_i der Schlüsseltextraum des Verschlüsselungsverfahrens $\boldsymbol{\mathcal{C}}_i$ ist.*
- *Γ^* ist die Menge aller möglichen k-Teilmengen der Schlüsseltexte, $\Gamma^* = \{C_{j_1} \times \ldots \times C_{j_k} | \{j_1, \ldots, j_k\} \subseteq \{1, \ldots, n\}\}$, wobei C_i der Schlüsseltextraum des Verschlüsselungsverfahrens $\boldsymbol{\mathcal{C}}_i$ ist.*

- *Π ist die Menge aller möglichen Schlüssel, $\Pi = K_1 \times \ldots \times K_n$, wobei K_i der Schlüsselraum des Verschlüsselungsverfahrens $\boldsymbol{\mathcal{C}}_i$ ist.*
- *$\boldsymbol{S}$ ist das verwendete Secret Sharing-Verfahren.*
- *$\boldsymbol{\mathcal{C}}$ ist die Menge aller eingesetzten Verschlüsselungsverfahren $\boldsymbol{\mathcal{C}}_i$, $i \in \{1, \ldots, n\}$.*
- *$\mathrm{Enc}_\pi(m, n, k, \boldsymbol{S}, \boldsymbol{\mathcal{C}})$ gibt für die Eingabe $m \in M$ und Parameter $k, n \in \mathbb{N}, k \leq n$, das perfekte Secret Sharing Verfahren $\boldsymbol{S}$, eine Menge von Verschlüsselungsverfahren $\boldsymbol{\mathcal{C}}$ und den Schlüssel $\pi \in \Pi$ den Schlüsseltext $c(m) \in \Gamma$ für den Klartext m aus.*
- *$\mathrm{Dec}_\pi(c(m)^*, \boldsymbol{S}, \boldsymbol{\mathcal{C}})$ gibt für den (Teil-)Schlüsseltext $c(m)^* \in \Gamma^*, \pi, \boldsymbol{S}, \boldsymbol{\mathcal{C}}$ den Klartext $m' \in M$ mit $m' = m$, wenn $c(m)^* \subseteq c(m)$ gilt, aus.*

Für eine Instanziierung von GSSCC sind $n, k, \boldsymbol{S}$ und $\boldsymbol{\mathcal{C}}$ fix, daher schreiben wir der Übersichtlichkeit halber im Folgenden nur $\mathrm{Enc}_\pi(m)$ und $\mathrm{Dec}_\pi(c(m)^*)$.

Die Verschlüsselungsfunktion Enc: $M \xrightarrow{share} Y \xrightarrow{encrypt} \Gamma$ verwendet den Algorithmus Share von $\boldsymbol{S}$ und die Verschlüsselungsalgorithmen E_i von $\boldsymbol{\mathcal{C}}_i$ für $\boldsymbol{\mathcal{C}}_i \in \boldsymbol{\mathcal{C}}$ als Subroutinen und funktioniert folgendermaßen:

1. $y(m) = (y_1, \ldots, y_n) = \mathrm{Share}(m, n, k)$.
2. $c(m) = (c_1, \ldots, c_n) = (\mathrm{E}_{1,k_1}(y_1), \ldots, \mathrm{E}_{n,k_n}(y_n))$.

Die Entschlüsselungsfunktion Dec: $\Gamma^* \xrightarrow{decrypt} Y^* \xrightarrow{recover} M$ (mit $Y^* = \{Y_{j_1} \times \ldots \times Y_{j_k} | \{j_1, \ldots, j_k\} \subseteq \{1, \ldots, n\}\}$) verwendet den Algorithmus Recover von $\boldsymbol{S}$ und die Entschlüsselungsalgorithmen D_i von $\boldsymbol{\mathcal{C}}_i$ für $\boldsymbol{\mathcal{C}}_i \in \boldsymbol{\mathcal{C}}$ als Subroutinen und funktioniert für $\{j_1, \ldots, j_k\} \subseteq \{1, \ldots, n\}$ folgendermaßen:

1. $y(m)^* = (y_{j_1}, \ldots, y_{j_k}) = (\mathrm{D}_{j_1,k_{j_1}}(c_{j_1}), \ldots, \mathrm{D}_{j_k,k_{j_k}}(c_{j_k}))$.
2. $m = \mathrm{Recover}(y_{j_1}, \ldots, y_{j_k})$.

Die Korrektheit des Verfahrens folgt aus der Korrektheit der angewendeten Secret Sharing- und Verschlüsselungsverfahren.

6.4 Sicherheit im Ideal Cipher-Modell

In diesem Abschnitt betrachten wir die Sicherheit der generischen GSSCC-Konstruktion im Ideal Cipher-Modell. Für die Betrachtungen bezüglich einer konkreten Instanziierung von GSSCC sei auf Abschnitt 6.5 verwiesen.

Im Ideal Cipher-Modell werden ideale Verschlüsselungsverfahren angenommen, d.h. dass sich die Verschlüsselungsverfahren verhalten wie zufällige Permutationen [BFK2009]. Es ist möglich, dass ideale Verschlüsselungsverfahren in der Praxis nicht existieren [CGH2004, CPS2008]. Die Sicherheit im Ideal Cipher-Modell zeigt jedoch, dass es notwendig ist, Schwachstellen eines konkreten Primitives auszunutzen, um das Verfahren zu brechen und somit nicht das Verfahren selbst inhärente Schwachstellen aufweist.

Wegen des vorausgesetzten Ideal Cipher-Modells ist die erschöpfende Schlüsselsuche der einzig mögliche Angriff auf die verwendeten Verschlüsselungsverfahren. Der Einfachheit halber nehmen wir für die Analyse an, dass alle eingesetzten Verschlüsselungsverfahren denselben Schlüsselraum K haben und damit die gleiche Schlüssellänge $l = log_2(|K|)$. Unterschiede für unterschiedliche Schlüssellängen behandeln wir im Anschluss.

Wir nehmen an, dass alle Schlüssel gleichverteilt zufällig und unabhängig gewählt werden und die Verschlüsselungsverfahren abgeschlossen sind. Aufgrund des Ideal Cipher-Modells schließen wir Kollisionen der Entschlüsselungen aus und zwar $\mathrm{D}_k(c) = \mathrm{D}_{k^*}(c)$ für $k \neq k^*$ und $c \in C$. Die Wahrscheinlichkeit einer solchen Kollision ist durch die Geburtstagsschranke auf höchstens $1/2\frac{|K|^2}{|P|}$ beschränkt. Daher reduzieren solche Kollisionen die Anzahl der möglichen Entschlüsselungen $\mathrm{D}_k(c) \in P, k \in K$ nicht maßgeblich, wenn $|P|$ signifikant größer als $|K|^2$ ist. Bspw. führen $|P| = 2^{1024}$ und $|K| = 2^{256}$ zu einer Kollisionswahrscheinlichkeit von $1/2^{513}$.

Da jedes Verschlüsselungsverfahren ideal ist und die Schlüssellänge l hat und aufgrund der Eigenschaft von Perfekten Secret Sharing-Verfahren, dass jede Teilmenge von mindestens k Shares zur eindeutigen Rekonstruktion des Geheimnisses verwendet werden kann, erhält ein Angreifer keine zusätzlichen Informationen von zusätzlichen Shares. Daher setzen wir o.B.d.A. $n = k$ für die weitere Analyse.

6.4.1 Erschöpfende Schlüsselsuche

Zunächst betrachten wir einen Angreifer, der nur Schlüsseltexte, also Sets von verschlüsselten Shares $c(m) = (c_1, \dots, c_k)$, kennt. Damit ist der Schlüssel für das Verfahren insgesamt von der Länge $\tilde{l} = k * l$. Der einfachste Angriff ist erschöpfende Schlüsselsuche, indem in jedem Durchlauf ein Gesamtschlüssel gewählt wird und die Shares entschlüsselt und kombiniert werden. Dieser benötigt im Durchschnitt $2^{\tilde{l}-1}$ Versuche um den Schlüssel zu finden. Im Detail bedeutet das $2^{\tilde{l}-1} * k$ einzelne Entschlüsselungen, $2^{\tilde{l}-1}$ Anwendungen des Recover-Algorithmus und $O(1)$ Speicherplatz.

Um wiederholte Entschlüsselungen eines Shares mit demselben Schlüssel zu vermeiden, kann ein Angreifer für jedes Share eine Liste anlegen, welche alle möglichen Entschlüsselungen enthält. Das bedeutet 2^l Entschlüsselungen für jedes Share und damit $2^l * k$ Entschlüsselungen insgesamt, sowie $O(2^l * k)$ Speicherplatz, um diese Listen zu speichern. Daraus ergeben sich wiederrum $2^{\tilde{l}}$ mögliche Kombinationen von entschlüsselten Shares aus den Listen als Eingabe für den Recover Algorithmus des Secret Sharing Verfahrens. Dies führt im Durchschnitt zu einem Aufwand von $2^{\tilde{l}-1}$ Anwendungen des Recover-Algorithmus zum Auffinden des Schlüssels.[6]

[6] Hier ist anzumerken, dass ein Angreifer in beiden Szenarien entscheiden muss, welcher der erzeugten Klartextkandidaten ein sinnvoller Klartext ist, was zusätzlichen Aufwand oder mehrere verschiedene Schlüsseltexte zur Verifikation erfordert.

Für $k < n$ und unterschiedliche Schlüssellängen $l_1, \dots, l_n$ für die Verschlüsselungsverfahren reduziert sich die effektive Schlüssellänge auf die Summe der k kürzesten Schlüssellängen.

6.4.2 Meet-in-the-Middle-Angriff

Ist mindestens ein Klartext-Schlüsseltext Paar $(m, c(m))$ für eine GSSCC-Instanziierung bekannt, so kann dieses für einen Meet-in-the-Middle Angriff ausgenutzt werden, um den Rechenaufwand für die erschöpfende Schlüsselsuche zu reduzieren. Wenn das Secret Sharing-Verfahren eine partielle Rekonstruktion erlaubt,[7] ist ein Time-Memory Trade-Off möglich. Sei dafür $(y_1, \dots, y_k)$ ein Set von k Shares, $a \subset (y_1, \dots, y_k)$ und $b = (y_1, \dots, y_k) \backslash a$, dann bedeutet partielle Rekonstruktion, dass die Shares in a und b jeweils separat kombiniert werden können, wodurch man m_1 und m_2 erhält. m kann dann aus diesen Teilergebnissen rekonstruiert werden.[8] Es ist leicht zu sehen, dass es auch möglich ist, die Shares in mehr als zwei Teilmengen aufzuteilen. Im Folgenden bezeichnen wir die partielle Rekonstruktion mit $\text{partRecover}(y_{j_1}, \dots, y_{j_u})$, für $\{j_1, \dots, j_u\} \subset \{1, \dots, k\}$. Der Angriff funktioniert dann wie folgt (beispielhaft für gerades k und Verschlüsselungsverfahren mit Schlüssellänge l):[9]

1. Teile die verschlüsselten Shares in zwei Teilmengen $g_1 = (c_1, \dots, c_{k/2})$ und $g_2 = (c_{k/2+1}, \dots, c_k)$ auf. Somit bezieht sich die Hälfte des Schlüsselmaterials auf die erste und die andere Hälfte auf die zweite Teilmenge.
2. Sei $\sigma = (\sigma_1, \dots, \sigma_{k/2}) \in \Sigma = K^{k/2}$. Für jedes $\sigma \in \Sigma$ berechne $y_\sigma = (\mathrm{D}_{1,\sigma_1}(c_1), \dots, \mathrm{D}_{k/2,\sigma_{k/2}}(c_{k/2}))$ und $m_{1,\sigma} = \text{partRecover}(y_\sigma)$. Speichere die Paare $(m_{1,\sigma}, \sigma)$ in einer Liste L_1.[10]Dann ersetze jeden Listeneintrag $m_{1,\sigma}$ durch $m \bullet m_{1,\sigma}$. "•" wird dabei gemäß der entsprechenden Operation des verwendeten Secret Sharing Verfahrens gewählt.[11]
3. Berechne sequenziell alle möglichen partiellen Rekonstruktionen $m_{2,\sigma'}$ mit $\sigma' \in \Sigma$ unter Verwendung der Shares aus g_2 wie für g_1 beschrieben, wobei der Schritt der Kombination mit m ausgelassen wird.
4. Prüfe auf eine Kollision mit den Werten in L_1, d.h. ob $m_{2,\sigma'} \in^? L_1$. Wenn eine Kollision gefunden wird, ist der zugehörige Schlüsselteil aus L_1 verknüpft mit dem aktuellen Schlüsselteil σ' ist ein Kandidat für den Gesamtschlüssel. Ein Kandidat kann mit zusätzlichen Klartext-Schlüsseltext-Paaren verifiziert werden.

---

[7] Dies ist für Blakeley's, Shamir's und OTP basierte Secret Sharing-Verfahren der Fall.
[8] Partielle Rekonstruktion ist das XOR von einer Teilmenge an Shares bei OTP basiertem SS oder die Summe einer Teilmenge von Shares multipliziert mit den jeweiligen Lagrange Multiplikatoren bei SSSS.
[9] Ein ungerades k führt zu Teimengen der Größe $\lceil k/2 \rceil$ und $\lfloor k/2 \rfloor$ mit den entsprechenden Laufzeit- und Speicherplatzbedarfen für den Angriff. Für unterschiedliche Schlüssellängen können die Shares so auf die Teilmengen verteilt werden, dass der Schlüssel annähernd halbiert wird.
[10] Das Ergebnis der partiellen Rekonstruktion kann dabei für verschiedene Schlüssel gleich sein. Das impliziert jedoch nur verschiedene Schlüsselkandidaten, die verifiziert werden müssen.
[11] "•" ist bspw. das bitweise XOR für OTP basiertes SS, für SSSS ist es die Subtraktion über dem verwendeten endlichen Körper F.

Der Angriff erfordert $O(2^{\tilde{l}/2})$ Zeit und $O(2^{\tilde{l}/2})$ Speicherplatz verglichen zu $O(2^{\tilde{l}})$ Zeit für die triviale erschöpfende Schlüsselsuche. Das ist der beste bisher bekannte generische Angriff und wir sehen die Unterteilung des Schlüssels in zwei Hälften als den bestmöglich erreichbaren Angriff bezüglich des Rechenaufwands. Dies folgt, da die Einteilung der verschlüsselten Shares und damit verbundenen Schlüsselteile in zwei Teilmengen unterschiedlicher Größe impliziert, dass entweder für die erste oder die zweite Teilmenge $t \geq \tilde{l}/2 + l$ Kombinationsmöglichkeiten bestehen. Das bedeutet direkt eine Erfordernis von $O(2^t)$ Zeit, um entweder die Liste aufzubauen oder um nach Kollisionen zu suchen. Eine Aufteilung in mehr als zwei Teilmengen scheint keinen Vorteil zu bringen, da aufgrund der perfekten Sicherheit (siehe Definition 2) in jede Verifikation k Shares eingebracht werden müssen. Das bedeutet, dass keine Verifikation mit Teilmengen von weniger als k Shares möglich ist und daher die partiellen Rekonstruktionsergebnisse mehrerer Teilmengen verknüpft werden müssen, bis nur noch zwei partielle Rekonstruktionsergebnisse übrig sind um obige Verifikation zuzulassen.

In Abschnitt 6.7 präsentieren wir einen Chaining-Mode, der solche Meet-in-the-Middle-Angriffe grundlegend verhindert.

6.5 Instanziierung von GSSCC

Es ist klar, dass solange mindestens $n - k + 1$ der verwendeten Verschlüsselungsverfahren als sicher anzusehen sind, das GSSCC-Verfahren sicher ist. Ein Angreifer muss mindestens k Verschlüsselungsverfahren brechen, da wegen der Eigenschaften perfekter Secret Sharing-Verfahren mindestens k entschlüsselte Shares benötigt werden, um Informationen über das Geheimnis abzuleiten [Sch1996]. Die Zufälligkeit und Unabhängigkeit der Shares bietet jedoch zusätzliche Sicherheit, da dadurch Angriffe auf einzelne Verschlüsselungsverfahren grundlegend verhindert werden.

Wie in obiger Analyse gezeigt, kann, wenn man das Ideal Cipher-Modell zugrunde legt, GSSCC als Combiner verwendet werden, um die Schlüssellänge beliebig (abhängig von k) zu erhöhen. Damit ist es möglich, ein Verschlüsselungsverfahren zu erzeugen, was Angreifern mit enormer Rechenkraft widerstehen kann. Es ist jedoch nicht klar, ob ideale Verschlüsselungsverfahren existieren. Deswegen analysieren wir GSSCC, instanziiert mit Shamir‘s Secret Sharing-Verfahren und aktuellen Blockchiffren, um zu zeigen, dass die Konstruktion Schutz vor verschiedenen potentiellen Schwachstellen der eingesetzten Verschlüsselungsverfahren bietet.

Für SSSS als Instanziierung von $\boldsymbol{S}$ gilt $M = Y_i = F \; \forall i \in \{1, .., n\}$, wobei F der endliche Körper ist, der für SSSS verwendet wird (für Details siehe Abschnitt 2.1 und [Sha1979]). Indem man $F = GF(2^r)$ wie in [Rab1989] wählt, arbeitet SSSS auf natürliche Weise auf beliebigen Bitstrings $b \in \{0,1\}^r$, da es ein bijektives Mapping von F auf die Bitstrings der Länge r gibt. O.B.d.A. betrachten wir im Folgenden nur Nachrichten $m \in M = \{0,1\}^r$, da dies durch Padding von kürzeren Nachrichten oder Aufteilen längerer Nachrichten in Blöcke immer erreicht werden kann.

Weiter verwenden wir aktuelle Blockchiffren $\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_n$ wie AES und Twofish. Der Übersichtlichkeit halber habe jede Blockchiffre die gleiche Blocklänge bs, die gleiche Schlüssellänge l und Schlüsselraum K und sei r ein Vielfaches der Blocklänge, d.h. $r = v * bs$, $v \in \mathbb{N}$. Damit gilt unter Verwendung eines beliebigen Verschlüsselungsmodus für die Blockchiffren, bspw. dem CBC-Modus, auf natürliche Weise $P_i = \{0,1\}^r$, $i \in \{1, .., n\}$.

Wir nehmen an, dass die Entschlüsselungen $\mathrm{D}_{i,k}(c)$ für verschiedene Schlüssel $k \in K_i, i \in \{1, \ldots, n\}$ gleichverteilt über $\{0,1\}^r$ sind und zufällig erscheinen, was wir für aktuelle Blockchiffren als angebracht betrachten, da für diese im Allgemeinen eine hohe Diffusion als Designkriterium gilt [Knu1998]. Damit gilt $\mathrm{D}_{i,k}(c) \neq \mathrm{D}_{i,k^*}(c)$, für $k \neq k^*$ im Allgemeinen aufgrund der Geburtstagsschranke, wenn r signifikant größer ist als $log_2|K| = l$, bspw. $r = 2 * k * l$.

Anzumerken ist hier, dass die Verschlüsselungsverfahren keine Homomorphieeigenschaften bezüglich der Kombinationsoperation des verwendeten Secret Sharing-Verfahrens haben dürfen. Damit ist gemeint, dass es nicht möglich sein darf, die verschlüsselten Shares zunächst zu kombinieren und dann das Ergebnis mit einem bestimmten Verschlüsselungsverfahren und einem passenden Schlüssel zum richtigen Klartext zu entschlüsseln.[12] Für unterschiedliche Schlüssel repräsentieren aktuelle Blockchiffren verschiedene Permutationen, daher nehmen wir an, dass dies im Allgemeinen solche Homomorphieeigenschaften verhindert.

6.5.1 Partielle ideale Sicherheit unter CPA2-Angriffen

Nun betrachten wir einen Angreifer, der Zugriff auf ein Verschlüsselungsorakel hat und Klartext-Schlüsseltext Paare $(m, c(m))$ für beliebige, adaptiv gewählte $m \in M$ erhalten kann. Mit *partieller idealer Sicherheit* bezeichnen wir die Eigenschaft, das jedes Verschlüsselungsverfahren in einer beliebigen Teilmenge der Verfahren der Kardinalität von maximal $k - 1$ starke ideale Sicherheit bietet, da die Verschlüsselungsverfahren abgeschlossen sind und der Input gleichverteilt und zufällig ist. Wir betrachten hier nur die k schwächsten Verschlüsselungsverfahren und zeigen, dass die Sicherheitslevel, die sich aus der Analyse im Ideal Cipher-Modell ableiten, auch für eine bestimmte Instanziierung gelten.

Zunächst sei gesagt, dass ein Angreifer niemals in der Lage ist, die Eingabe für einzelne Verschlüsselungsalgorithmen zu erhalten, d.h. er ist nicht in der Lage, die Shares $(y_1, \ldots, y_k)$ zu bestimmen, da es für $F = GF(2^r)$ $2^{r*(k-1)}$ gleichwahrscheinliche Ausgaben des Secret Sharing-Verfahrens gibt (siehe Theorem 1). Daher werden CPA-Angriffe auf einzelne Blockchiffren durch die Konstruktion verhindert. Darüber hinaus sind die Shares unterschiedlicher Verschlüsselungsdurchläufe von GSSCC nach Konstruktion unabhängig. Dadurch sind die Eingaben der $i$ten Blockchiffre zufällig, gleichverteilt und paarweise unabhängig, wodurch man starke ideale Sicherheit erhält.

[12] Ein triviales Beispiel hierfür wäre die Instanziierung von GSSCC mit OTP basiertem SS und OTP Verschlüsselung der Shares.

Das wiederrum bedeutet, dass die Shares von unterschiedlichen Verschlüsselungsdurchläufen von GSSCC nicht verwendet werden können, um die einzelnen Chiffren anzugreifen.

Da die Shares in jeder Teilmenge mit maximaler Kardinalität $k-1$ zufällig gleichverteilt und unabhängig sind (siehe Theorem 1), kann ein Angreifer eine Chiffre nur in Abhängigkeit der $k-1$ anderen Chiffren angreifen. Das bedeutet, dass solange er nicht $k-1$ der Shares fixiert, seine Sicht auf das $k$te Share zufällig gleichverteilt ist, und es damit durch die stark ideale Sicherheit geschützt ist. Die ideale Sicherheit kann nur durch Kenntnis irgendeiner Funktion der Eingaben zu den anderen Chiffren unterlaufen werden.

Wir modellieren die Information, die ein Angreifer aus $k-1$ verschlüsselten Shares und einem gegebenen zugehörigen Klartext m ableiten kann, als die Funktion $\text{info}(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_k, m)$. Nun kann ein Angreifer $\text{info}(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_k, m)$ als zusätzliche Eingabe für den Angriff nutzen. Da jedoch weniger als k Shares einbezogen werden, gilt ideale Sicherheit für jede der einbezogenen Chiffren und Gleichwahrscheinlichkeit für jeden möglichen Schlüssel. Das bedeutet, dass $\text{info}(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_k, m)$ bestenfalls die Werte von $2^{l*(k-1)}$ möglichen partiellen Rekonstruktionen annehmen kann, was zur selben Menge von Kandidaten für das $i$te Share führt. Nach der Annahme der Gleichverteilung möglicher Entschlüsselungen unter verschiedenen Schlüsseln für die einzelnen Chiffren, sind auch die partiellen Rekonstruktionen gleichverteilt. Durch die Geburtstagsschranke ergibt sich damit für bspw. $r = 2 * k * l$ die Wahrscheinlichkeit einer Kollision unter den partiellen Rekonstruktionen von maximal $\frac{1}{2}\frac{(2^{(k-1)*l})^2}{2^{2*k*l}} = \frac{1}{2}\frac{1}{2^{2*l}}$, was vernachlässigbar ist. Daher schließen wir solche Kollisionen aus den Betrachtungen aus.

Da wir Gleichverteilung und Pseudozufälligkeit für die Entschlüsselungen unter Anwendung verschiedener Schlüssel für die einzelnen Chiffren annehmen, ist es nicht möglich, die Ergebnisse der verschiedenen partiellen Rekonstruktionen auf bspw. einen bestimmten Bereich des Klartextraumes einzuschränken. Daher ist die Wahrscheinlichkeit, eine korrekte Annahme über den Klartext des $i$ten Shares zu treffen, und damit die Wahrscheinlichkeit für einen möglichen erfolgreichen Angriff auf die $i$te Chiffre, gleich $1/2^{l*(k-1)}$.

Aus diesen Betrachtungen folgern wir, dass mögliche existierende Angriffe auf einzelne Chiffren die Sicherheit von GSSCC nicht reduzieren können.

Es ist jedoch ein offenes Problem, ob effizient kombinierte Angriffe auf aktuelle Blockchiffren[13] existieren, bei denen nur die zugehörigen Schlüsseltexte $(c_1, \dots, c_k)$ und eine Funktion $f(y_1, \dots, y_k)$ der zugehörigen Klartexte bekannt ist. Dabei ist f dergestalt, dass $|M|^{k-1}$ gleichwahrscheinliche Eingaben existieren, für die f dasselbe Ergebnis ausgibt.

[13] Angriffe welche mehrere Blockchiffren gleichzeitig angreifen.

Angenommen, solch ein Angriff würde existieren, dann muss dieser mindestens $k/2$ der für die GSSCC-Instanziierung verwendeten Chiffren beinhalten, um bezogen auf den Rechenaufwand effizienter als der generische Meet-in-the-Middle Angriff zu sein. Das folgt aus der Anzahl der möglichen partiellen Rekonstruktionen, die sich aus einer erschöpfenden Schlüsselsuche auf die nicht im Angriff inbegriffenen Chiffren ergibt.

Wir folgern aufgrund der partiellen idealen Sicherheit und der gleichverteilten, pseudozufälligen Entschlüsselungen unter verschiedenen Schlüsseln, dass jeder bekannte Angriff auf einzelne Verschlüsselungsverfahren die Sicherheit von GSSCC nicht reduziert. Das Sicherheitslevel, welches durch den *Meet-in-the-Middle*-Angriff impliziert wird, gilt solange kein kombinierter Angriff auf mehr als $k/2$ der Verschlüsselungsverfahren gefunden wird.

6.5.2 Chosen-Ciphertext-Angriffe

Da die Rekonstruktion eines Geheimnisses eine deterministische Funktion ist, kann die partielle ideale Sicherheit von GSSCC unterlaufen werden, wenn ein Angreifer die Möglichkeit hat, Klartexte zu selbst gewählten Schlüsseltexten zu erhalten. Angenommen, ein Angreifer kann gewählte Schlüsseltext- (CCA) oder adaptiv gewählte Schlüsseltext-Angriffe (CCA2) auf GSSCC ausführen, so kann er Klartext-Schlüsseltext-Paare erzeugen, bei denen die Schlüsseltexte bis auf ein einziges verschlüsseltes Share gleich sind. Das kann bspw. erreicht werden, indem der Angreifer die Schlüsseltexte als $(c_1, \dots, c_{k-1}, c_k)$ und $(c_1, \dots, c_{k-1}, c'_k)$ wählt. Durch den Vergleich der resultierenden Klartexte, kann er die Unterschiede unter der Bedingung analysieren, dass der Unterschied nur durch eines der angewendeten Verschlüsselungsverfahren zustande kommt. Das könnte Angriffsmöglichkeiten auf das Schlüsselmaterial dieses Verfahrens bieten. Daher sollte diese Art von Angriffen ausgeschlossen werden, was durch die Verwendung von Signaturen möglich ist [KL2007]. Da Signaturen mit dem Zeitverlauf erneuert werden können, reicht es aus, Signaturverfahren zu verwenden, welche derzeit als sicher zu betrachten sind und diese wenn nötig zu erneuern. Ohne CCA2-Angriffe verhindernde Maßnahmen sinkt das Sicherheitsniveau jedoch lediglich maximal um die entsprechende Schlüssellänge des jeweils kompromittierten Verschlüsselungsverfahrens.

6.5.3 Blocklänge von GSSCC

Wenn Shamir‘s Secret Sharing-Verfahren zur Instanziierung von GSSCC verwendet wird, ist die Blocklänge des Verfahrens durch die Größe der Elemente des endlichen Körpers F bestimmt. Für $F = GF(2^r)$ verschlüsselt GSSCC Nachrichtenblöcke der Länge r Bit bei einer Ausführung. Dies gilt auch für die Entschlüsselung. Die eingesetzten Blockchiffren können unterschiedliche Blocklängen haben. Um die Rekonstruktion eines Nachrichtenblockes zu ermöglichen, müssen jedoch jeweils Shares der Größe r Bit komplett entschlüsselt werden.[14]

[14] Bei einer Instanziierung mittels OTP basiertem Secret Sharing hängt die Blocklänge dagegen von der Blocklänge der verwendeten Chiffren ab.

Um Nachrichten beliebiger Länge zu verschlüsseln, können kürzere Nachrichten gepaddet werden. Nachrichten, die länger als r Bit sind, können nach dem Verfahren von Miyamoto et al. [MDN2006] (siehe auch Abschnitt 6.2.1) aufgeteilt werden. Der resultierende Vektor aus Shares kann dann mittels einer Blockchiffre in einem beliebigen Verschlüsselungsmodus verschlüsselt werden.

Eine weitere Methode, Nachrichten beliebiger Länge zu verschlüsseln, ist GSSCC selbst in einem Verschlüsselungsmodus anzuwenden, welcher auf die Struktur von GSSCC angepasst ist. Als Beispiel geben wir hier die Beschreibung eines CBC-Modus für GSSCC an.

CBC-Modus für GSSCC:
Sei $m = m_1 m_2, \dots, m_t \in M$ eine Sequenz von t Blöcken der Größe r und GSSCC instanziiert mit SSSS wie am Anfang dieses Abschnittes beschrieben mit dem Schlüssel π. Sei $IV \xleftarrow{\$} \{0,1\}^r$ der Initialisierungsvektor, dann:

$$c_1 = (c_{1,1}, \dots, c_{1,k}) = \mathrm{Enc}_\pi(m_1 \oplus IV),$$

$$c_j = (c_{j,1}, \dots, c_{j,k}) = \mathrm{Enc}_\pi(m_j \oplus c_{j-1,1} \oplus \dots \oplus c_{j-1,k}), \qquad 2 \le j \le t$$

um die Schlüsseltexte $c_0, \dots, c_t = IV, (c_{1,1}, \dots, c_{1,k}), \dots, (c_{t,1}, \dots, c_{t,k})$ zu erhalten. Um mit dem Schlüssel π zu entschlüsseln, geht man wie folgt vor:

$$m_1 = \mathrm{Dec}_\pi(c_1) \oplus IV,$$

$$m_j = \mathrm{Dec}_\pi(c_j) \oplus c_{j-1,1} \oplus \dots \oplus c_{j-1,k}, \qquad 2 \le j \le t$$

Anzumerken ist hier, dass für diesen Verschlüsselungsmodus alle Shares zur Entschlüsselung benötigt werden und nicht nur eine Teilmenge von k Shares, daher sollte $n = k$ für die Instanziierung gewählt werden.[15] Im Falle $r > bs$ müssen die Chiffren, die als Subroutinen der GSSCC-Verschlüsselung verwendet werden, selbst in einem beliebigen Verschlüsselungsmodus angewendet werden.

6.6 Diskussion

6.6.1 Über die Auswahl der Chiffren zur GSSCC-Instanziierung

Wir haben Schneiers Vorgehen k Blockchiffren unter Verwendung des OTP basierten Secret Sharing-Verfahrens auf beliebige perfekte Secret Sharing-Verfahren generalisiert. Die Generalisierung auf beliebige (k,k)-PSSS ist geradlinig, da ein Angreifer, aufgrund der informationstheoretischen Sicherheit bei weniger als k bekannten Shares (siehe Abschnitt 6.2.1), keine Informationen ableiten kann, wenn er nicht alle k Verschlüsselungsverfahren bricht.

[15] Ein (k,n)-Threshold kann durch die wiederholte Anwendung der Verschlüsselung mit allen gültigen k-Kombinationen aus n Shares erreicht werden.

In beiden Fällen, (k,k)-und (k,n)-Verfahren, führt das Anheben von k zu einem Anstieg der Sicherheit. Einerseits wird die effektive Schlüssellänge vergrößert, andererseits müssen die Eigenschaften der zusätzlichen Chiffre bei einem Angriff auf GSSCC berücksichtigt werden. Jedoch sind für die Sicherheit von (k,n)-Verfahren nur die k schwächsten Chiffren relevant. Daher führt das alleinige Anheben von n nicht zu einer Steigerung der Sicherheit, sondern kann diese sogar reduzieren.

Um den Schlüsselraum zu vergrößern, müssen selbstverständlich alle Shares, die zu einem Geheimnis gehören, mit unterschiedlichen, zufällig, gleichverteilt und unabhängig gewählten Schlüsseln verschlüsselt werden. Bisher haben wir die Anwendung von verschiedenen Chiffren zur Instanziierung von GSSCC betrachtet. Es könnte jedoch wünschenswert sein, GSSCC nur mit einem Verschlüsselungsverfahren oder mit weniger als k verschiedenen Schlüsseln zu verwenden, um die Sicherheit gegen erschöpfende Schlüsselsuche oder Known-Plaintext Angriffe zu erhöhen oder um ein Verschlüsselungsverfahren mit vergrößerter Blockgröße zu erhalten.

Wenn allerdings nur ein Verschlüsselungsverfahren eingesetzt wird, scheint es wahrscheinlicher, dass aufgrund von speziellen Eigenschaften des Verfahrens Homomorphieeigenschaften oder ein Angriff auf ein Set von Chiffretexten, bei dem nur eine Funktion der Klartexte bekannt ist (wie dies als offenes Problem in Abschnitt 6.5.1 dargestellt wurde), existieren. Bisher ist uns allerdings kein solcher Angriff bei aktuellen Blockchiffren bekannt.

Angenommen, dass aktuelle Blockchiffren gute pseudozufällige Permutationen sind, dann schützen diese vor Homomorphieeigenschaften und erhalten die Vorteile aus der partiellen idealen Sicherheit von GSSCC. Jedoch haben bisher unbekannte Schwachstellen potentiell höhere Auswirkungen, wenn nur ein Verschlüsselungsverfahren für die Instanziierung verwendet wird. Die Verwendung von verschiedenen Verfahren steigert die Sicherheit darüber hinaus durch deren inhärente Stärken.

6.6.2 Weitere Anmerkungen

Die Verwendung von (k,n)-Threshold Secret Sharing-Verfahren hat Vorteile gegenüber der Verwendung von (k,k)-Secret Sharing-Verfahren. (k,n)-Threshold-Verfahren erlauben Fehlerresistenz, Möglichkeiten zum Lastenausgleich und Flexibilität bezüglich der verwendeten Verschlüsselungsverfahren, da ein Geheimnis auch auf Geräten rekonstruiert werden kann, die nicht alle, sondern nur eine k-Teilmenge der Verfahren implementieren.

Betrachtet man Instanziierungen von GSSCC, so ist die OTP basierte Variante sehr effizient bezüglich des Rechenaufwands für das Aufteilen und Rekombinieren. Soll ein (k,k)-Sharing zum Einsatz kommen, könnte die OTP Variante bevorzugt werden. Für ein (k,n)-Sharing ist jedoch jedes (k,n)-Threshold Verfahren wie bspw. SSSS die bessere Wahl bezüglich des benötigten Speicherplatzes, da die Shares die Optimalgröße behalten (siehe Abschnitt 6.2.1). Darüber hinaus hat SSSS den zusätzlichen Vorteil, beliebige Blockgrößen

für GSSCC zuzulassen (siehe Abschnitt 6.5.3), wohingegen die Blockgröße bei der OTP Variante von den eingesetzten Verschlüsselungsverfahren abhängt.

Der Nachteil von GSSCC ist darin zu sehen, dass um k Verschlüsselungsverfahren zu kombinieren, aufgrund der unteren Schranke der Sharegröße bei perfekten Secret Sharing Verfahren, ein um mindestens das k-fache vergrößerter Schlüsseltext resultiert.

6.7 Meet-in-the-Middle-sicherer Blockmodus

Wir schlagen einen neuen, speziell für GSSCC entworfenen Blockmodus vor. Wir nennen den Modus MSB-Modus. Er ist, unter Anwendung von verschiedenen Meet-in-the-Middle Angriffen abwehrenden Maßnahmen, von dem oben vorgestellten CBC-Modus abgeleitet.

Sei $m = m_1 m_2, \dots, m_t \in M$ eine Sequenz von t Blöcken der Größe r und GSSCC instanziiert mit SSSS und dem Schlüssel π. y_i bezeichnet wieder das $i$te Share, welches von dem Secret Sharing Verfahren ausgegeben wird, also $y_i = \mathrm{D}_{\mathrm{i,k_i}}(c_i)$. Sei $IV \xleftarrow{\$} \{0,1\}^r$ der Initialisierungsvektor, dann:

$$c_0 = (c_{0,1}, \dots, c_{0,k}) = \mathrm{Enc}_\pi(IV),$$

$$c_1 = (c_{1,1}, \dots, c_{1,k}) = \mathrm{Enc}_\pi(m_1 \oplus IV),$$

$$c_j = (c_{j,1}, \dots, c_{j,k}) = \mathrm{Enc}_\pi(m_j \oplus y_{j-1,1} \oplus \dots \oplus y_{j-1,k-1}), \qquad 2 \le j \le t$$

um die Schlüsseltexte $c_0, \dots, c_t = (c_{0,1}, \dots, c_{0,k}), \dots, (c_{t,1}, \dots, c_{t,k})$ zu erhalten. Um mit dem Schlüssel π zu entschlüsseln geht man folgendermaßen vor:

$$IV = \mathrm{Dec}_\pi(c_o),$$

$$m_1 = \mathrm{Dec}_\pi(c_1) \oplus IV,$$

$$m_j = \mathrm{Dec}_\pi(c_j) \oplus y_{j-1,1} \oplus \dots \oplus y_{j-1,k-1}, \qquad 2 \le j \le t$$

Die Hauptunterschiede zum CBC-Modus sind, dass der IV nicht im Klartext bereitgestellt wird, und außerdem das Chaining mit den Entschlüsselungen von $k-1$ Shares durchgeführt wird. Diese Shares sind zufällig, gleichverteilt und unabhängig (Theorem 1). Damit ist selbst für einen bekannten Klartext m die Eingabe zu den GSSCC Ausführungen unbekannt für einen Angreifer, was vor dem Meet-in-the-Middle Angriff schützt. Dadurch ist die effektive Bitsicherheit des Verfahrens die Summe der Bitlängen der k Schlüssel.

Da der IV zufällig und gleichverteilt gewählt wird, impliziert das stark ideale Sicherheit. Zusätzlich gilt, dass die Sicht auf die Eingabe für GSSCC für den aktuellen Block, ohne den vorangehenden Block entschlüsselt zu haben (bzw. $k-1$ Shares des vorangehenden Blocks), gleichverteilt und zufällig erscheint und stark ideale Sicherheit für jeden Block impliziert.

6.8 Fazit

Wir haben gesehen, dass perfektes Secret Sharing anwendbar ist, um Combiner von Verschlüsselungsverfahren zu erzeugen, welche die Stärken von verschiedenen Verfahren zusammenführen und darüber hinaus sicher gegen CPA2-Angriffe auf die einzelnen Algorithmen sind. GSSCC kann außerdem verwendet werden, um die effektive Schlüssellänge einzelner Verfahren zu vergrößern. Insbesondere die Zufälligkeit und Gleichverteilung von $k-1$ Shares zusammengenommen mit Shannon's ideal sicheren Systemen schafft sehr starke Sicherheitseigenschaften.

Das präsentierte GSSCC-Verfahren, angewendet im hier vorgeschlagenen MSB-Modus, resultiert in einer Bitsicherheit, die der Summe der kürzesten k Schlüssellängen der eingesetzten Verschlüsselungsverfahren entspricht. Bezüglich CPA2-Angriffen verbleibt die Sicherheit auf diesem Niveau, bis mindestens k der Verschlüsselungsverfahren gebrochen sind. Es gibt Anzeichen dafür, dass selbst ein substantielles Sicherheitslevel erhalten bleibt, wenn k oder mehr der Verschlüsselungsverfahren gebrochen sind. Bezüglich CCA2-Angriffen sinkt das Sicherheitsniveau mit jedem kompromittierten Verfahren höchstens um dessen Schlüssellänge. Gegenmaßnahmen gegen CCA2-Angriffe, wie bspw. Signaturen, sind für GSSCC einsetzbar. Ein weiterer Ansatz zur Vermeidung von CCA2-Angriffen ist die kaskadierte Anwendung von GSSCC, bspw. die Anwendung in einer Baumstruktur.

Ein offenes Problem ist die mögliche Existenz von Homomorphieeigenschaften der unterschiedlichen Verschlüsselungsverfahren unter dem PSS-Verfahren, was die Existenz effizient kombinierter Angriffe auf GSSCC ermöglichen würde. Da aktuelle Blockchiffren unterschiedliche Permutationen für jeden Schlüssel repräsentieren, scheinen Homomorphieeigenschaften in diesem Fall unwahrscheinlich.

7 Potentielle Angriffsszenarien

7.1 Sicherheitsrisiken und –aspekte

Beim Cloud Computing handelt es sich um eine Auslagerung von Daten, Prozessen und Anwendungen zu einem externen Cloud-Anbieter. Die IT-Sicherheit dieser Komponenten stellt daher eines der wichtigsten Problemfelder im Zusammenhang mit Cloud-Services dar [SR2009, Lea2009]. Die Sicherheits- und Verfügbarkeitsrisiken für die Nutzer werden durch Cloud Computing zunehmend intransparent und durch den hohen Automatisierungsgrad geht die Auslagerung mit einem Kontrollverlust einher. Das bedeutet, dass der Nutzer beispielsweise nur einen geringen bzw. keinen Einfluss auf den geographischen Ort seiner Daten hat [SR2009, Str2009]. Durch die Verbreitung von Cloud-Diensten treten neue Schwachstellen und Bedrohungen auf. In [RTS2009] zeigt Ristenpart et al. wie es beispielsweise möglich ist, eine bestimmte virtuelle Maschine (VM) eines Kunden zu identifizieren und eine eigene VM auf derselben Hardware-Ressource zu platzieren, um damit VM übergreifende Seitenkanalangriffe auf Kundendaten und Prozesse zu starten.

Für die genaue Beurteilung der Bedrohungen, müssen zunächst die einzelnen Schutzziele, die in verschiedenen Szenarien unterschiedlich gewichtet werden können, erörtert werden. Danach werden die Angreifer in verschiedene Typen eingeteilt und gemäß deren Fähigkeiten spezifiziert.

7.1.1 Angreifertypen

In diesem Abschnitt sollen die verschiedenen Angreifertypen dargestellt und erklärt werden. Die Einteilung hängt zum einen vom Verhalten der jeweiligen Angreifer ab, zum anderen auf welche Weise sie Zugang zu dem jeweiligen IT-System haben [Rup2010a].

7.1.2 Passive Angreifer

Ein passiver Angreifer greift nicht in das jeweilige IT-System ein. Er beschränkt sich auf das Mithören oder Mitschneiden von Kommunikation, welche über das Netzwerk abgewickelt wird und versucht so, an Informationen zu erlangen. Da ein solcher Angreifer nur passiv zuhört, ist er im Allgemeinen schwer bis gar nicht zu erkennen. Passive Angriffe bedrohen vor allem die Schutzziele Vertraulichkeit und Privatheit.

7.1.3 Aktive Angreifer

Ein aktiver Angreifer greift wie der Name sagt aktiv in die Prozesse eines IT-Systems ein. Er modifiziert oder löscht beispielsweise Daten, oder sabotiert unter Ausnutzung von Sicherheitslücken das System. Er kann auch durch aktive Maßnahmen die Verfügbarkeit von Diensten sabotieren. Allerdings ist ein solcher Angreifer leichter zu entdecken, da aktive Angriffe Spuren hinterlassen können. Oft werden mit aktiven Angriffen passive Angriffe vorbereitet und das Mithören ermöglicht. Aktive Angriffe gefährden auch die Verfügbarkeit, Integrität und Authentizität.

7.1.4 Externe Angreifer

Externe Angreifer sind im Zusammenhang mit Cloud Computing zunächst Angreifer, die die Cloud, also das Cloud Computing-Netzwerk von außen angreifen. Sie können dabei unter Ausnutzung von Schwachstellen in das Netzwerk eindringen, um an Informationen zu gelangen oder Zugriff auf Cloud-interne Netzwerke, virtuelle Maschinen oder gesicherte Komponenten erhalten. Des Weiteren fällt beispielsweise das Mitschneiden der Informationen, die zwischen Nutzer und Cloud-Anbieter über das Internet übertragen werden, in diesem Zusammenhang unter externe Angriffe.

7.1.5 Interne Angreifer

Interne Angreifer können auf unterschiedlichste Art und Weise in Erscheinung treten. Sie gehören dabei von vornherein dem Cloud-Netzwerk an und haben prinzipiell Zugriff auf bestimmte Komponenten. So können beispielsweise Kunden versuchen auf die virtuellen Maschinen anderer Kunden zuzugreifen oder deren Kommunikation mitzuhören. Ein Beispiel hierfür ist der in Abschnitt 7.1 genannte Seitenkanalangriff von Ristenpart et al. [RTS2009] Aber auch Angestellte des Kunden können beispielsweise als interne Angreifer auftreten. Darüber hinaus zählen zu den potentiellen internen Angreifern Mitarbeiter, Dienstleister, Kooperationspartner und Praktikanten des Cloud-Anbieters oder der Cloud-Anbieter selbst.

7.1.6 Intention der Angreifer

Eine weitere Unterscheidung von Angreifern kann unter Berücksichtigung der jeweiligen Intention und Motivation der Angreifer vorgenommen werden. Die Motivation von Angriffen ist dabei vielfältig. Unzufriedenheit, Verärgerung oder beispielsweise Erpressung sowie Bestechung können Gründe für Angriffe sein. Auch die Intentionen sind dabei vielfältig. So kann das Ziel, die Schädigung des Cloud-Anbieters oder dessen Kunden sein oder das reine Erlangen von Informationen zu verschiedenen Zwecken.

7.2 Angreifermodelle

Im Hinblick auf den Entwurf eines sicheren Datensafes in der Cloud gilt es, die Bedrohung aus verschiedenen Blickwinkeln zu untersuchen. Im Folgenden definieren wir verschiedene Angreifermodelle, abhängig von der Stärke des jeweiligen Angreifers. Im Hinblick auf das Anwendungsszenario Datensafe stehen die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit im Vordergrund.

7.2.1 Externer Angreifer passiv

In diesem Angriffsmodell wird der schwächste mögliche Angreifer angenommen. Er ist auf das passive Mithören der Kommunikation über öffentliche Netzwerke wie das Internet beschränkt. Daher beschränken sich die Aktivitäten dieses Angreifers auf den Zeitraum während ein Nutzer aktiv mit der Cloud oder dem Safe Provider kommuniziert, beispielsweise beim Datentransfer zwischen dem Client und dem Datenspeicher in der

Cloud. Ein solcher Angreifer gefährdet, wie bereits erwähnt, die Vertraulichkeit der Kommunikation. Auch die Privatheit des Nutzers kann durch einen solchen Angreifer gefährdet sein, da er durch das Mithören der Kommunikation die einzelnen Kommunikationspartner identifizieren kann. Wird die Kommunikation zwischen Safe Provider und Nutzer mitgeschnitten, so ist für den Angreifer eine Zuordnung von Daten zu einem bestimmten Nutzer möglich. Kann jedoch lediglich die Kommunikation zwischen Safe-Provider und Cloud-Anbieter verfolgt werden, ist eine eindeutige Zuordnung von Daten zum Nutzer nicht gegeben, da davon auszugehen ist, dass ein Safe-Provider seine Dienste einer großen Anzahl von Nutzern zur Verfügung stellt. Ein solcher Angreifer hat damit folgende Fähigkeiten:

- Mithören sämtlicher Kommunikation zwischen Client und Safe-Provider.
- Mithören sämtlicher Kommunikation zwischen Safe-Provider und Cloud-Anbieter.
- Kopieren aller über das Netzwerk übertragenen Daten.

7.2.2 Externer Angreifer aktiv

Im Falle eines aktiven externen Angreifers müssen darüber hinaus aktive Maßnahmen zur Erlangung von Zusatzinformationen in Betracht gezogen werden. Einen solchen Angreifer modellieren wir wie folgt:

- Alle Möglichkeiten des passiven Angreifers.
- Verändern, umleiten sämtlicher Kommunikation.
- Partieller Zugriff auf in der Cloud gespeicherte Daten.

7.2.3 Interner Angreifer mit partiellem Zugriff

Interne Angreifer stellen beim Cloud Computing eine besonders wichtige Gruppe von Angreifern dar. Da sie von vornherein Teilnehmer am Cloud-Netzwerk sind, müssen sie sich nicht zunächst Zugang verschaffen. Durch die parallele Nutzung von Ressourcen durch mehrere Nutzer, können Nutzer Zugriff auf die Daten anderer Nutzer erlangen. Dazu kommen Kooperationspartner des Cloud-Anbieters, welche Zugriff auf die von ihnen betriebenen Server haben. Es liegt daher nahe, Angreifer zu betrachten, die sich im Cloud Netzwerk befinden und Zugriff auf einzelne Ressourcen realisieren können. Dabei muss untersucht werden, wie stark dies die Vertraulichkeit der im elektronischen Safe abgelegten Daten beeinträchtigt.

Ein wichtiger Unterschied zu der Möglichkeit Informationen während der Kommunikation mitzuschneiden ist, dass die Daten in der Cloud dauerhaft vorliegen, und die Zugriffsmöglichkeit nicht auf den Zeitraum der tatsächlichen Kommunikation beschränkt ist. Allerdings ist anzumerken, dass ein Angreifer, welcher die Daten einmal kopiert hat, auf diese einen zeitlich unbeschränkten Zugriff erlangt.

7.2.4 Interner Angreifer mit vollem Zugriff

Handelt es sich bei dem Angreifer um den Safe-Anbieter selbst, so hat er prinzipiell Zugriff auf die Daten der Safe-Eigentümer, kann Log-Dateien einsehen und weitreichende Verknüpfungen zwischen Prozessen und Daten sowie Nutzern herstellen. Dies ist der stärkste mögliche Angreifer. Ein ähnlich starker Angreifer ist ein Cloud-Anbieter, wenn nur ein einziger Cloud-Anbieter für die Datenspeicherung aller Geheimnisteile genutzt wird. Safe Anbieter bzw. Cloud-Anbieter bezeichnen hierbei bspw. Mitarbeiter dieser, mit umfangreichen Rechten wie bspw. Administratoren.

7.3 Integritätsschutz

Für den Integritätsschutz müssen bei der Kommunikation und Speicherung entsprechende, integritätssichernde Maßnahmen (bspw. Signaturen, MAC) angewendet werden. Diese sind geeignet, um Integritätsschutz gegen aktive externe Angreifer zu gewährleisten. Dabei ist darauf zu achten, dass jeweils aktuell als sicher zu betrachtende Verfahren und Schlüssellängen eingesetzt werden. Der Integritätsschutz der gespeicherten Daten (Metadaten beim Safe-Provider und der Nutzer-Daten beim Cloud-Anbieter) liegt darüber hinaus in der Verantwortung des jeweiligen Speicheranbieters. Wird gemäß Kapitel 5 ein Verfahren mit Redundanz basierend auf Secret Sharing eingesetzt, so kann der Safe-Eigentümer selbst dann noch das Dokument rekonstruieren, wenn n-k der n Geheimnisteile manipuliert oder beschädigt wurden. Werden n verschiedene Speicher-Anbieter zur Ablage der Daten verwendet, ist der Integritätsschutz selbst noch gegeben, solange weniger als k der Speicher-Anbieter beschädigte Dateien liefern oder die Daten bösartig verändern.

7.3.1 Verfügbarkeitsschutz

Für die Gewährleistung der Verfügbarkeit gelten dieselben Annahmen wie für die Integrität. Durch ein Speicherverfahren mit Redundanz kann ein Ausfall von weniger als k Geheimnisteilen (und damit ein Ausfall von ebenso vielen Speicher-Anbietern) toleriert werden. Der Nutzer kann damit die Verfügbarkeit seiner Daten über die Parameterwahl beeinflussen. Ein einzelner Safe-Anbieter stellt hier jedoch ebenso ein Single Point of Failure dar, wie der Client selbst. Bei einer Konfiguration, bei der der Safe-Anbieter sämtliche Metadaten (Zugriffsadressen etc.) verwaltet, sind die Daten verloren, wenn diese Metadaten nicht mehr verfügbar sind. Daher muss der Safe-Anbieter eine hohe Verfügbarkeit durch Redundanz, Backups etc. gewährleisten.

Weiterhin ist das Schlüsselmanagement beim Client eine kritische Komponente. Durch Verlust von mehr als n-k Schlüsseln, werden die gespeicherten Daten unbrauchbar, da eine Entschlüsselung nicht mehr möglich ist. Daher sind entsprechende Schlüsselbackups einzusetzen. In der Komfortvariante Safe-Anbieter-Konfiguration (siehe Kapitel 4.4) wird dieses Problem zum Safe-Anbieter hin verlagert. In diesem Fall gefährdet ein interner Angreifer beim Safe-Anbieter jedoch zusätzlich zur Verfügbarkeit die Vertraulichkeit der Daten.

7.4 Schutz der Vertraulichkeit

Bezüglich des Vertraulichkeitsschutzes sei hier zunächst auf die Sicherheitsanalyse von GSSCC (Kapitel 6) verwiesen.

Aufgrund des Einsatzes von Secret Sharing kann die perfekte Sicherheit gegen interne und externe Angreifer garantiert werden, welche nicht Zugriff auf mindestens *k* Geheimnisteile haben. Durch die Verteilung der Daten auf mehrere Clouds wird der Zugriff durch unabhängige Systeme erschwert. Dazu kommt die Problematik der Zurechenbarkeit einzelner Geheimnisteile zu einer Datei, wenn die gesamte Kommunikation über den Safe Anbieter abgewickelt wird, da dieser potentiell eine sehr große Nutzerbasis bedienen kann.

Kennt ein Angreifer alle Geheimnisteile, so ist die Vertraulichkeit durch die eingesetzten Verschlüsselungsverfahren gewährleistet. Existierende Angriffe gegen die einzelnen Verfahren werden durch die Konstruktion von GSSCC verhindert. Die beschriebene *Meet-in-the Middle*-Attacke kann durch den vorgeschlagenen Blockmodus verhindert werden. Dadurch wird jedoch die Redundanz des Verfahrens eingebüßt, was bei der Wahl des Verschlüsselungsmodus zu berücksichtigen ist. Kritisch für die Vertraulichkeit ist damit die Clientanwendung bzw. das Schlüsselmanagement. Wird dieses als sicher angenommen, ist ein sehr hoher Schutz der Vertraulichkeit gegen interne wie externe Angreifer gegeben.

8 Realisierungskonzepte

Durch die Vielzahl von Cloud-Anbietern und deren angebotener Produkte liegt ein hoher Grad an Heterogenität im Cloud-Umfeld vor. So ist zwar der Zugang zu den Services über das Internet standardisiert, allerdings sind die Schnittstellen für den Zugriff auf die Cloud-Dienste unterschiedlich. So sieht beispielsweise ein Microsoft Azure Speicher-Dienst (Storage as a Service) andere Operationen vor, als ein Speicher-Dienst, der von Amazon angeboten wird. Diese können sich in Anzahl und Eigenschaft der durch *Application Programming Interfaces* (APIs) angebotenen Funktionen unterscheiden. Um Software-Komponenten in Lösungen für einen elektronischen Cloud-Safe integrieren zu können, ist daher die Konzeption einer geeigneten Abstraktionssicht notwendig.

8.1 Adaptoren

Die Transparenz gegenüber der Heterogenität der Cloud-Services der verschiedenen Cloud-Anbieter kann durch sogenannte Adaptoren erreicht werden. Diese Adaptoren stellen Abstraktionsbausteine für jeden Cloud-Service-Anbieter dar, die es ermöglichen, dass identische Software-Komponenten des elektronischen Safes sowohl auf Cloud-Systemen von Microsoft, IBM, Amazon oder Google aufsetzen können, ohne deren spezielle Eigenschaften bei der Implementierung mit einzubeziehen (siehe Abbildung 5). Damit wird erreicht, dass mit Hilfe der Adaptoren Speicheranbieter unterschiedlicher Hersteller genutzt werden können, um die Dokumententeile (*secret shares*) auf Speicherdienste von beispielsweise Microsoft, IBM oder Amazon zu verteilen.

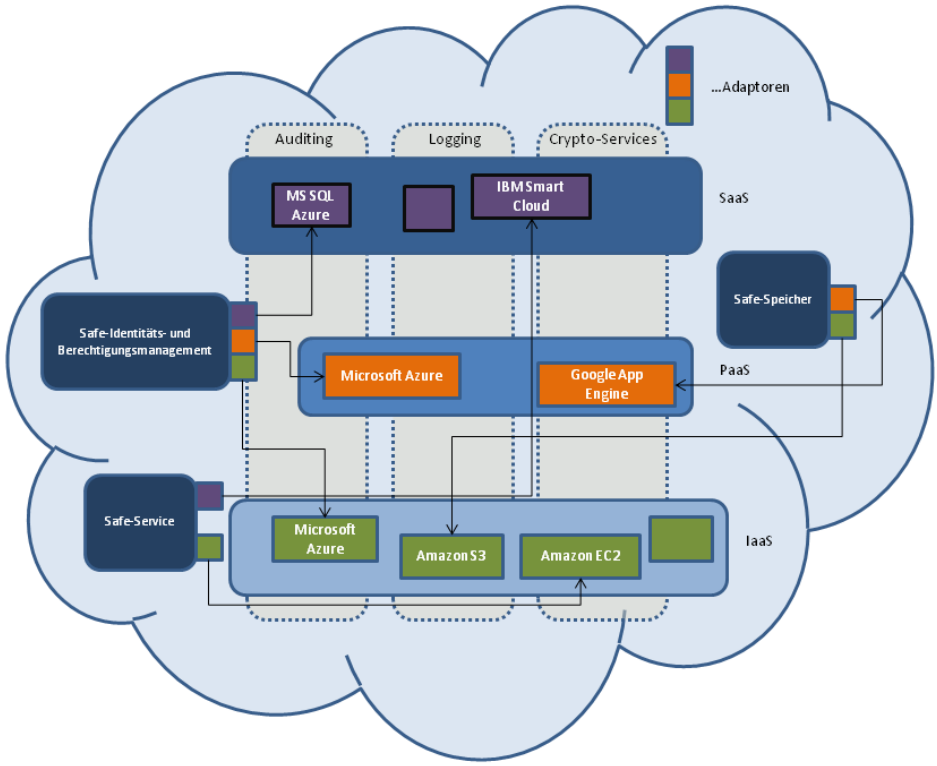

Abbildung 6- Nutzung von Adaptoren für die Integration

8.2 Schnittstellen

Eine mögliche Umsetzung beinhaltet also zum einen die Spezifikation der Schnittstellen zwischen Adaptoren und den Komponenten des elektronischen Safes oder Cloud-Safes, bestehend aus einer Anzahl standardisierter Funktionen, die ein Cloud-Safe nutzen kann, wodurch auch die Portierbarkeit auf beliebige Cloud-Anbieter unterstützt werden kann. So kann mit der Funktion „Teildatenstück ablegen“ über den Adaptor das Teildatenstück auf einen beliebigen (oder ausgewählten) Speicher-Anbieter abgelegt werden. Die Funktion enthält außerdem neben den reinen (Binär-) Daten auch Parameter zur eindeutigen Identifikation der Daten und verschieden Token, die für Autorisierung oder Authentifizierung genutzt werden.

Die zweite Gruppe der Schnittstellen bezieht sich auf die Funktionen, die von den Safe Services angeboten werden sollen. Bspw. benötigt der Cloud-Safe eindeutig definierte Schnittstellen, so dass bspw. beliebige Speicher-Anbieter durch die Safe-Service Komponente gesteuert werden können.

Die typischen Funktionen des Cloud-Safes stellen in ihren grundlegenden Eigenschaften Operationen mit Daten dar. Die Funktionen der Safe-Verwaltung, die zum Einstellen, Bearbeiten und Löschen der vertrauenswürdigen Dokumente dienen, gleichen einfachen Dateioperationen wie Schreiben, Löschen und Modifizieren. Analog kann das Identity

Management gesehen werden. Dort vertrauen Cloud-Safe Anbieter auf die Authentisierung bzw. Autorisierung durch (Safe-) Identitäts-, Access- und Berechtigungsmanagement und ermöglichen so den Zugriff auf bestimmte Daten oder das Ausführen definierter Funktionen. Dies geschieht eventuell in zusätzlichen Föderationen, wenn mehrere Speicher-Anbieter in unterschiedlichen Sicherheitsdomänen liegen und einer (Safe-) Identitäts- und Berechtigungskomponente vertrauen.

Die folgende mögliche Schnittstelle, die durch Adaptoren für Safe-Storage-Provider angeboten wird, sollte mindestens die folgenden drei Funktionen beinhalten:

- *Speichern eines Teildatenstücks in der Cloud*
  - **Parameter**: Cloud-Identifikation, Datenstück-Identifikation, Autorisierungs-Information, Authentifizierungs-Information, Teildaten
- *Der Zugriff auf Daten in der Cloud*
  - **Parameter**: Cloud-Identifikation, Datenstück-Identifikation, Autorisierungs-Information, Authentifizierungs-Information
- *Das Löschen von Daten aus der Cloud*
  - **Parameter**: Cloud-Identifikation, Datenstück-Identifikation, Autorisierungs-Information, Authentifizierungs-Information

Im Bereich der Identity bzw. Security Services sind folgende Bereiche zu betrachten:

- Die Integration von Identity-Management-Diensten in den Speicher- oder Zugriffsvorgang
- Die Integration von umfangreichen Logging- und Auditing-Diensten, die eine gründliche Protokollierung ermöglichen.

Die angegebene Menge bzw. Art der Parameter soll somit nur als Orientierung dienen, damit erkennbar ist, welche zusätzlichen Cloud Dienste (*Services*) auch auf Infrastrukturebene beachtet werden müssen. Analog stellen auch die Speicher-Anbieter für Cloud-Safes Schnittstellen bereit. Die folgende Abbildung stellt damit sowohl die zu definierenden Schnittstellen, als auch den konzeptionellen Aufbau der Komponenten inklusive Adaptoren dar. Dazu wurden exemplarisch die (Safe) Speicher-Anbieter weiter spezifiziert, so dass zum einen die *SafeStorageManager* erkennbar sind und zum anderen die *SafeStorageAdaptoren*. Beide implementieren einheitliche Interfaces, so dass in beide Richtungen standardisierte Schnittstellen vorliegen.

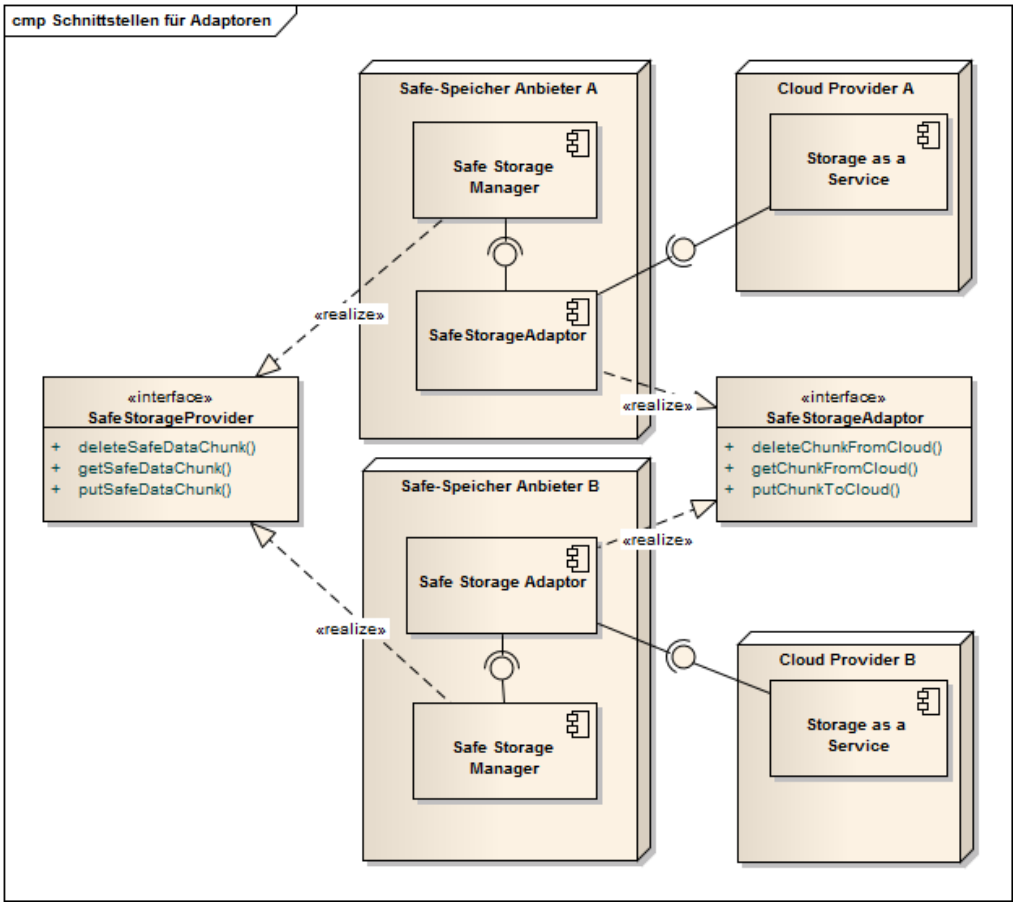

Abbildung 7 – Schnittstellen der Adaptoren

8.3 Marktanalyse angebotener Cloud-Service-Frameworks

Im Rahmen dieser Studie soll abschließend eine Analyse erfolgen, welche auf dem Markt vorhandenen Cloud Service Frameworks bereits geeignet sind, den elektronischen Safe als Cloud Service umzusetzen. Dazu sollen stellvertretend die angebotenen Cloud-Services von IBM, Microsoft, Amazon und Google klassifiziert und untersucht werden. Abschließend erfolgt die Darstellung des JCloud Frameworks, welches eine Open-Source Variante zur Standardisierung der Zugriffe auf verschiedene Clouds ermöglichen soll.

Basierend auf den Ergebnissen der Framework-Analyse sollen sich ergebende Herausforderungen im Bereich der Umsetzung diskutiert werden.

8.3.1 IBM

Die von IBM angebotenen Cloud-Services sind dadurch charakterisiert, dass sie ausschließlich die Bereitstellung von IBM Produkten in Form von Software as a Service vorsehen. Diese beinhalten bspw. Applikations- oder Webserver (IBM Websphere), Datenbanken (IBM DB2), Group Management Tools (IBM Lotus) oder Identity-Management Software (IBM Tivoli). Die Bereitstellung der Software wird dementsprechend in einem von vier verschiedenen Betriebsmodellen vorgenommen, die

Variationen des Modells der „Private Cloud“ entsprechen. Darin enthalten sind eine vollständig private Cloud (Kontrolle beim Kunden, IBM stellt Infrastruktur / Plattform), *managed* und *hosted* Private Clouds (IBM besitzt und managed Infrastruktur), sowie eine *Community* Cloud, in welcher mehrere Unternehmen eine private Cloud gemeinsam nutzen. Der Umfang der Überwachung und Steuerung der Dienste durch IBM bzw. das Level der Kontrolle durch den Kunden ist abhängig vom endgültig gewählten Betriebsmodell. Weiterhin gibt es die Möglichkeit, eine Public Cloud zu nutzen.

Im Kontext des elektronischen Safes stellen alle Angebote im Sinne einer *Private Cloud* eine Lösung dar, die genutzt werden kann, um einen Cloud-Safe Service bereitzustellen. Durch die von der IBM gelieferten Produkte kann weiterhin Software bereitgestellt werden, um Dienste, wie bspw. (Safe) Speicherdienste oder Safe Identitäts- oder Berechtigungskomponenten mit Hilfe der IBM Produkte zu implementieren. In Hinblick auf die Entwicklung von Adaptoren bzw. einer herstellerübergreifenden Standardisierung ist festzustellen, dass IBM durch ihr Produktportfolio eine Abdeckung für eine Vielzahl von Standards und Technologien anbietet. Dazu gehören bspw. relevante Standardisierungen in den Web-Service Technologien (WS-*, XML) oder im Identity-Management, die maßgeblich für eine Umsetzung sind. Da es jedoch keine direkte Möglichkeit gibt, die IBM Cloud bspw. in Form eines Software Development Kits, wie es eine PaaS bereitstellen würde, in zu entwickelnde Software mit einzubeziehen, muss die Entwicklung der Adaptoren in diesem Zusammenhang vor allem produktspezifisch erfolgen. Denn eine mögliche Umsetzung sieht durch ihre Produkteigenschaft vor, dass bspw. Safe-Anbieter Komponenten auf Websphere Application Servern installiert werden, Safe-Speicher-Anbieter, die IBM DB2 nutzen und das Identity Management durch Tivoli-Produkte umgesetzt werden. Durch die Produkt- und Konfigurationsvielfalt sowie die regelmäßigen Updates der Software kann dies in hohem Zusatzaufwand resultieren, da eine ständige spezifische Anpassung der Adaptoren durchgeführt werden muss.

8.3.2 Microsoft

Microsoft stellt Cloud-Services für Unternehmen über die Microsoft Azure Plattform (PaaS) bereit. So werden bspw. MS Server-Produkte wie MS Sharepoint oder Office Exchange Server als SaaS angeboten, aber auch SQL Azure, welche eine SaaS-Lösung des MS SQL Servers darstellt. Das .NET Programmiermodell bietet die Grundlage, um für die angebotene Azure Platform Software zu entwickeln, wobei ebenfalls die Programmiersprachen Java, Python oder Ruby genutzt werden können. Die entwickelte Software kann dann entweder als Web-Anwendung oder Hintergrundprozess innerhalb der Azure-Cloud installiert werden.

Für diese Software können ebenfalls verschiedene Speicher-Services der Infrastrukturebene (IaaS) genutzt werden, um große Binärdaten (Blob) oder relationale Daten (Tables) dauerhaft abzuspeichern. Das Datenmodell der Queues kann genutzt werden, um bspw. asynchrone Kommunikation zwischen einer Web-Anwendung und aufwendigen Hintergrundprozessen zu ermöglichen, indem Daten in einer Nachrichtenschlange abgelegt und abgearbeitet werden (bspw. Freigabeanforderungen).

Weiterhin bietet Microsoft über die Azure Plattform auch Möglichkeiten, Benutzerauthentisierung und –autorisierung durchzuführen (AppFabric Access Control), verteilte Anwendungen zu integrieren (AppFabric System Integration), private Overlay-Netzwerke zu implementieren oder automatisiertes Loadbalancing durchzuführen.

Ähnlich zu den angebotenen Lösungen der IBM eignen sich die Microsoft Cloud-Services somit für die Umsetzung eines elektronischen Safes, der die Ansprüche an die Vertrauenswürdigkeit umsetzen kann. Microsoft stellt dafür Infrastruktur und Software auf Basis etablierter Microsoft Produkte sowie ein Software-Developement Kit (SDK) bereit. Vor allem das SDK in Verbindung mit den Infrastrukturdiensten für Persistenzmechanismen eignen sich deshalb gut, um Adaptoren zu entwickeln, die die diskutierten Dateioperationen entsprechend kapseln und cloudspezifisch umsetzen. Autorisierung und Authentifizierung von Nutzern können über die Integration von Active Directory erfolgen.

Die Microsoft Azure Cloud wird ausschließlich als Public Cloud im Internet bereitgestellt. Zusätzlich bietet Microsoft die Möglichkeit, eine Infrastruktur als Private Cloud in einem firmeneigenen Rechenzentrum aufzubauen, ähnlich der IBM Lösung einer *hosted* Cloud. Dies geschieht unter der Verwendung des Microsoft System Center, das Teillösungen für Installation, Betrieb und Wartung einer eigenen Azure Private Cloud bietet.

8.3.3 Amazon

Amazon stellt eine Vielzahl von Cloud-Services als Amazon Web Services (AWS) bereit. Im Bereich *Storage as a Service* bietet Amazon die folgenden Dienste an: Amazon EC2 Elastic Block Storage, Amazon EC2 Local Instance Store, Amazon Simple Storage Service (S3), Amazon Simple Queue Service (SQS), Amazon SimpleDB, Amazon EC2 Relational Databases und Amazon Relational Database Service (RDS). Für eine Realisierung von (Safe) Speicher-Anbietern eignet sich Amazon S3 in Verbindung mit der Amazon SimpleDB (non-relational) am besten. Amazon S3 wird genutzt, um unstrukturierte Binärdaten (Blob) redundant und performant abzuspeichern, so dass sowohl Dateigrößen von 1Byte-5TB unterstützt werden, eine hohe Skalierbarkeit vorliegt und jede einzeln vorhandene Ressource, die sich in S3 befindet, über eine URL eindeutig identifizierbar ist. Zudem soll diese die beste und längste Verfügbarkeit innerhalb der Amazon Storage Services anbieten. Amazon SimpleDB übernimmt in diesem Fall die Zuordnung zwischen den reinen (Blob)Daten und Metadaten. In diesem Zusammenhang wäre Identifikationsinformationen für die Zuordnung der verteilt abgelegten Daten denkbar, zudem Informationen über verwendete Verschlüsselungs- oder Authentifizierungsmechanismen. Für den Zugriff auf die Amazon S3 Services existieren zudem SDKs für Java, C#, Perl und PHP, was analog zu Microsoft eine Adaptorenentwicklung begünstigt. Parallel dazu erweitern Web-Service Schnittstellen für SOAP und Rest den Umfang des Frameworks[16].

[16] http://d36cz9buwru1tt.cloudfront.net/AWS_Storage_Options.pdf

Für jeden der AWS bietet Amazon spezifische Sicherheitsdienste (Security Services) an. In dem angenommenen Fall von S3 und SimpleDB für einen Speicher-Anbieter (Storage Provider) würden die Identity- Access und Berechtigungsfunktionen wie folgt aussehen.

Die Daten auf Amazon S3 liegen in sogenannten Buckets und können dabei einerseits auf Bucket-Ebene als auch auf der darin liegenden Objektebene gesichert werden. Über Access-Control-Lists können Nutzer für den Zugriff auf ausgezeichnete Buckets oder Objekte autorisiert werden. Dies geschieht über den AWS Identity and Access Manager (IAM). Zudem werden SSL-Schnittstellen für S3 und Amazon SimpleDB bereitgestellt, Physische Sicherheit sowie Zertifizierungen nach bspw. ISO 27001 oder PCI liegen bei Amazon vor. Des Weiteren werden Logging Services für S3 angeboten, die sowohl den Zugriff, den Zugreifenden, und dessen IP-Adresse sowie Zeit und Datum speichern[17].

8.3.4 Google

Googles App Engine stellt eine Plattform as a Service (PaaS) Cloud Lösung dar. Sie bietet die Möglichkeit, auf Basis der vorgegebenen Programmiersprachen Java und Python vor allem Web-Applikationen zu entwickeln. Diese Applikationen werden dann entsprechend des Cloud Computing Paradigmas auf einer Infrastruktur betrieben, die durch Google bereitgestellt und gewartet wird.

Im Bereich der Datenspeicherung bietet Google zwei verschiedene Speichermodelle an: *High Replication Datastore (HRD)* und *Master/Slave Datastore (MSD).* HRD speichert die Daten automatisch redundant in mehreren Datenzentren, was zu einer sehr hohen Verfügbarkeit für Lese und Schreibprozesse führt. MSD speichert bzw. repliziert die Daten asynchron, so dass eine Datensicherung nicht zur gleichen Zeit durchgeführt wird, wie der Schreibprozess. Zudem wird nur ein Datencenter genutzt, um Schreibprozesse abzuwickeln. Daraus resultiert eine sehr hohe Konsistenz für Lese- und Abfrageprozesse, der alleinige Server für Schreibzugriffe schränkt jedoch die Verfügbarkeit für Schreibprozesse bspw. in Zusammenhang mit Wartungsarbeiten ein.

Parallel zu den Speicherarten ist die Struktur und Organisation der abgelegten Daten zu analysieren. Innerhalb der Google App Engine werden Daten als *Entities* abgespeichert, die ihrerseits *Properties* besitzen, welche in Form verschiedener Datentypen oder Referenzen zu anderen Entities vorliegen können. Zudem besitzt jede Entity einen *Key*, der diese eindeutig identifiziert. Der zur Speicherung genutzte Datastore liefert dazu eine API, die grundlegende Funktionen wie *get*, *put*, *delete* und *query* unterstützt. Jeder Zugriff in Form eines *get*, *put* oder *delete* geschieht zudem in Form einer Transaktion, sodass das entweder alle damit verbundenen Änderungen durchgeführt werden oder keine. Damit wird sichergestellt, dass die Integrität der Daten nicht durch einen konkurrierenden Zugriff während der Abarbeitung der eigentlichen Operationen beschädigt wurde[18].

[17] http://d36cz9buwru1tt.cloudfront.net/pdf/AWS_Security_Whitepaper.pdf
[18] http://code.google.com/intl/de-DE/appengine/docs/java/datastore/overview.html

Die vorgestellten Möglichkeiten, Daten dauerhaft abzuspeichern, ähneln SQL Datenbanken und sind damit nicht besonders dafür geeignet, unstrukturierte Binärdaten zu speichern. Aus diesem Grund existiert zusätzlich ein sogenannter „Blobstore“ (**B**inary **L**arge **Ob**ject), der zur Speicherung von Binärdaten vorgesehen ist, die die für normale Datastores erlaubte Größe übersteigen. Die Identifikation der Blobs erfolgt analog über die Zuordnung von *Keys.* Ähnlich zu Amazon könnte dementsprechend der HR Datastore genutzt werden, um strukturierte Informationen für jedes Teildatenstück abzuspeichern, wobei die reinen Nutzdaten über den Blob-Key referenziert werden[19].

Es ist festzustellen, dass die angebotene SDKs für Java und Python sowie das Dateimanagement gut geeignet sind, um Adaptoren zu entwickeln, die den Zugriff auf die Google App Engine entsprechend abstrahieren. Google bietet zusätzliche Services an, die genutzt werden können, um bspw. Authentisierung über Google Konten oder den OAuth Standard durchzuführen. Somit existieren zusätzliche Sicherheitsfunktionen, neben denen durch Java oder Python gegebenen prinzipiellen Sicherheitsmechanismen.

8.3.5 JClouds

Die Analyse der verschiedenen Cloud-Services hat gezeigt, dass es diverse Unterschiede zwischen den Anbietern gibt, was bspw. die angebotenen Funktionen, Anzahl der Cloud-Services oder unterstützte Programmiersprachen angeht. Das Open-Source Framework JClouds ist eine seit kurzem existierende API in Java, die es ermöglicht auf die Dienste von Cloud-Anbietern abstrahiert zuzugreifen. Zu den Cloud-Services, auf die zugegriffen werden kann, gehören bspw. die Amazon Web Services, Microsofts Azure Storage Service, die IBM Developer Cloud oder auch Eukalyptus. Anstatt Adaptoren für jede eigene Cloud zu entwickeln, könnte die gesamte Cloud Anbindung einheitlich über die Nutzung der JClouds API implementiert werden.

Dazu könnten bspw. die von den JClouds angebotenen Funktionen in den Bereichen „Compute API“ und „Blobstore API“ genutzt werden. Für die diskutierten Adaptoren bietet sich die Blobstore-API an, da diese auf die Speicherung von Key-Value Paaren spezialisiert ist.

Die Speicherung der Daten wird in JClouds durch die hierarchische Aufteilung in Container und Blobs realisiert, wobei der Container den Kontext der abzulegenden Daten definiert. Er entspricht bspw. den Buckets der Amazon Web Services und kann etwa durch die Zugangsdaten definiert sein. Das Blob entspricht den unstrukturierten Binärdaten, die in den Containern abgelegt werden sollen. Ein wesentlicher Vorteil in der Nutzung von JClouds würde sich bereits dadurch ergeben, dass diese einheitliche Hierarchie ein zu nutzendes Datenmodell vorgibt. Weiterhin bleibt die jeweilige Verbindung zu den Cloud-Services transparent für den Entwickler, so dass Authentifizierungs- oder Schreib- bzw. Lese-Prozesse einheitlich durchgeführt werden können, ohne die entsprechende produktspezifische Implementierung kennen zu müssen.

[19] http://code.google.com/intl/de-DE/appengine/docs/java/blobstore/overview.html

8.4 Standardisierung und Zertifizierung

Qualitäts-, Zuverlässigkeits- bzw. Sicherheitszertifikate unterstützen Anwender bei der Beurteilung und Auswahl von Produkten bzw. Produktkomponenten. Beim Cloud-Computing stehen die Anforderungen an die Dienste zunächst im Vordergrund. Darüber hinaus gilt es für den wachsenden Markt die Möglichkeiten von Produktzertifizierungen nach anerkannten Regeln vorzubereiten. Die *Common Criteria* sind ein anerkanntes Vorgehen. Ergänzend zum existierenden Minimalanforderungskatalog des BSI [BSI2011] sind unter Berücksichtigung existierender Richtlinien die Wege zur Definition von Schutzprofilen vorzubereiten. Die Herausforderung besteht dabei in der Ermittlung von zertifizierungsfähigen Komponenten, die für Anwender als auch Anbieter bedeutsam sind.

Ein Großteil des Erfolges eines elektronischen Safes als Cloud-Dienst hängt von dem Vertrauen ab, das einer derartigen Softwarelösung entgegen gebracht wird. Vertrauen basiert einerseits auf Sicherheit, wie in den Kapiteln 5 und 6 ausführlich diskutiert, aber es hängt auch von offenen Schnittstellen und Standards ab. Ein weiterer Aspekt ist die Zertifizierung, die ebenfalls Vertrauen stärkt.

Einerseits durch offene Cloud-Safe-Schnittstellen, die Prozesse, Anwendung und Implementierung einschließen, aber auch durch Prüfung der Cloud-Anbieter in den Bereichen Vertrag, Compliance, Sicherheit, Betrieb und Infrastruktur, kann ein von unabhängigen Institutionen durchgeführtes Audit zu einem vertrauenswürdigen Gütesiegel führen. Der Verband der Cloud-Services-Industrie in Deutschland, EuroCloud Deutschland_eco hat beispielsweise ein Programm zur Zertifizierung von SaaS aufgelegt, das auch auf den Cloud-Safe anwendbar wäre. Je nach Anforderung können unterschiedliche Zertifizierungstiefen erlangt werden. Als Ergebnis des Audits erhält der Anbieter einen bis fünf Sterne[20].

Die jetzige Spezifikation für den Bereich Software-as-a-Service muss auf weitere Bereiche wie Platform-as-a-Service und Infrastructure-as-a-Service ausgeweitet werden. Hierbei sind sehr unterschiedliche Bereitstellungsmodelle zu berücksichtigen, für die u.U. individuelle Kriterienansätze zu entwickeln sind. Eine Zusammenführung von Diensten der Cloud-Anbieter setzt entsprechende Schnittstellenfunktionen voraus, die wiederum Bestandteil von automatisierbaren Testverfahren sein können und sich in die grundlegenden Zertifizierungsprozesse integrieren lassen. Dadurch können auch unterbrechungsfreie Prüfverfahren durch automatisierte Funktionsprüfung von Systemschnittstellen und Funktionen erstellt werden.

Eine wesentliche Voraussetzung für den effektiven Einsatz von Cloud Angeboten ist die Vermeidbarkeit von „lock-in“ Situationen, d.h. die (ungewollte) Bindung an einen spezifischen Anbieter aufgrund von proprietären Technologien und Verfahren. Software, die in der Cloud betrieben werden soll, muss für beliebige Cloud Infrastrukturen portierbar sein. Interoperabilität zwischen Clouds sowie (Legacy) Anwendungen, die

[20] http://www.email-marketing-forum.de/Web/details/SasS-Guetesiegel-Die-Cloud-Zertifizierung-nimmt-Gestalt-an/27895

außerhalb der Cloud (etwa beim Kunden) betrieben werden, sorgt für eine Flexibilisierung von Nutzungsszenarien (z.B. „Cloud bursts“, d.h. die dynamische Nutzung von Drittangeboten bzw. „Cloud brookering“ als neuartiges Geschäftsmodell). Interoperabilität kann auf verschiedenen Ebenen betrachtet werden: Daten, Prozesse, System- bzw. Dienstintegration. Insbesondere für den öffentlichen Sektor existieren bereits Standards (XÖV und abgeleitete Formate) zur Repräsentation von Daten und Dokumenten, die auch eine Integration von Fachverfahren und behördlichen Prozessen in Cloud Infrastrukturen erlauben.

In diesem Zusammenhang wäre es denkbar, die genutzten Safe-Funktionalitäten zu standardisieren, um diese einfacher auf verschiedene Clouds portieren zu können. Grundlegende Safe-Funktionen wie Daten abspeichern, entnehmen oder Funktionen zum Transaktionsmanagement könnten als wiederverwendbare Komponenten vorliegen, was die Umsetzung in Form der vorgeschlagenen Szenarien erleichtert. Die vorgesehenen Security Service-Komponenten sind in diesem Fall mit einzubeziehen.

Zurzeit existieren Bestrebungen bekannter Gremien, deren Ziel die Identifikation von potentiellen Cloud-Standards ist. Da der übliche Standardisierungsprozess sehr langwierig ist, werden beispielsweise bei NIST[21][NIST], ISO[22], DMTF[23] und CCUSDC[24] Anwendungsfälle (*Use-Cases*) beschrieben um die potentiellen Standards schnell zu finden und entsprechend zu standardisieren.

Als Beispiel wird hier kurzer Überblick zu den NIST-Anwendungsfällen gegeben.

Die Einordnung der vorgeschlagenen *Use-Cases* erfolgt durch das NIST in die Gruppen:

- Cloud Management,
- Cloud Interoperability und
- Cloud Security.

Weiterhin gibt es eine Gruppe, welche die nach Ansicht des NISTs in naher Zukunft relevanten *Use-Cases* enthält. Folgende *Use-Cases* aus den jeweiligen Gruppen sind relevant im Kontext der Nutzung dieser zur Realisierung des elektronischen Safes:

[21] National Institute of Standards and Technology
[22] International Organization for Standardization
[23] Distributed Management Task Force
[24] die Cloud Computing Use Case Discussion Group

Cloud Management	Copy Data Objects Into a Cloud
	Copy Data Objects out of a Cloud
	Erase Data Objects In a Cloud
Cloud Interoperability	Copy Data Objects between Cloud Providers
Cloud Security	IdM – User Account Provisioning
	IdM – User Authentication in the Cloud
	IdM – Data Access Authorization Policy Management in the Cloud
	eDiscovery
	Security Monitoring
	Sharing of Access to Data in a Cloud
Future Use cases candidates	Transfer of ownership of data within a cloud

Tabelle 1. Relevante Kernfunktionen nach NIST

Vor allem die Use-Cases des Bereiches Cloud-Managements und des Identity-Managements (IdM) bilden Kernfunktionen des elektronischen Safes ab und könnten eine wichtige Rolle im Rahmen von dessen Umsetzung einnehmen. Sie enthalten das Kopieren von Daten in die Cloud und aus der Cloud heraus, das Löschen sowie das Verschieben zwischen den Clouds. Weiterhin enthalten diese das Bereitstellen von Benutzer-Accounts, die Authentifizierung der Benutzer sowie die Umsetzung von Autorisierungsregeln in der Cloud. Der *Use-Case* des Security-Monitorings kann einen maßgeblichen Beitrag zu geforderten Logging- und Auditing-Anforderungen darstellen, so dass bspw. die Zurechenbarkeit und Nachvollziehbarkeit im Bereich von Interaktionen mit Cloud-Safes oder Freigaben einfacher durchzusetzen sind. „*Sharing of Access to Data in a Cloud*“ ist ebenfalls als wichtiger Bestandteil für eine Umsetzung des Safes zu sehen, da ein Kernelement den Zugriff verschiedener Nutzer auf eine Ressource darstellt, selbst wenn dieser zeitlich oder durch andere Restriktionen begrenzt ist.

Jeder der aufgeführten *Use-Cases* beinhaltet dahingehend weitere Angaben zu Akteuren, Annahmen bzw. Vorbedingungen, Erfolgsszenarien, Fehlerbehandlung sowie Fehlerbedingungen[25].

Die Standardisierung durch eine dieser Institutionen würde es Cloud-Anbietern erlauben, standardisierte Dienste und Funktionen anzubieten und somit für mehr Transparenz zu sorgen, aber auch den Kunden den Wechsel erleichtern. Zusätzliche Funktionen zur Wahrung der Vertraulichkeit, wie Verschlüsslung und Signaturen, die eine PKI benötigen, müssten dann mittels Adaptoren angepasst werden. Weitere Adaptoren könnten notwendig sein für Proxy- oder Controller-Komponenten, oder für Safe-Agenten.

8.5 Fazit

Abschließend ist festzuhalten, dass eine technische Realisierung eines elektronischen Safes als Cloud Software-as-a-Service-Lösung technisch umsetzbar ist. Vor allem die

[25] http://www.nist.gov/itl/cloud/use-cases.cfm

Platform as a Service-Angebote, wie sie Microsoft Azure, Amazon AWS oder die Google App Engine anbieten, eignen sich gut, da diese Laufzeitumgebungen für industrierelevante Programmiersprachen und Standards bereitstellen. Weiterhin stellen sie Programmierbibliotheken bereit, die das performante und redundante Speichern, das Lesen und Löschen von Binärdaten auf skalierbarer und hoch verfügbarer Speicherinfrastruktur einfach möglich machen. Dies ergibt einen großen Vorteil, da viele der Safe-Anwendungsfälle das Arbeiten mit vertrauenswürdig abgelegten Daten, bspw. in Form von Modifikation, dem Löschen oder Hinzufügen von Daten, beinhaltet.

Zudem zeigt die Analyse der Anbieter IBM, Microsoft, Amazon und Google, dass das Übertragen der Prinzipien des elektronischen Safes auf Cloud-Services umsetzbar ist. Um die Authentizität sowie Integrität auch bei der Ortstransparenz von Cloud Computing durchzusetzen, bedarf es zudem an Möglichkeiten, Authentisierung und Authentifizierung sowie Logging- oder Auditing-Funktionen zu benutzen. Auch hier zeigt sich, dass in der Praxis die verwendbaren Programmiersprachen zwar einheitliche Konzepte umsetzbar machen, die vorgefertigten Lösungen jedoch stark variieren in der Art und Weise, wie sie eingebunden werden können. Während IBM und Microsoft auf die Nutzung ihrer Produkte (bspw. Active Directory) setzen, definiert Amazon einen eigenen Cloud-Support-Service, der für Integrity- und Access-Management zuständig ist. Google unterstützt als einzige der betrachteten Anbieter den OAuth Standard und stützt sich sonst vor allem auf die Google Mail Accounts. Eine eigenständige Identity- oder Logging-Komponente, wie sie bspw. bei Amazon vorliegt, existiert nicht.

Sowohl für die Vertrauensbildung als auch Integrität wird das in Kapitel 6.8 diskutierte GSSCC-Verfahren vorgeschlagen, dass zurzeit noch von keinem Cloud-Anbieter realisiert wurde. Dies ist eine Herausforderung für Cloud-Anbieter, die sich mit einer solchen Realisierung deutlich voneinander unterscheiden. Sie bieten damit den Vorteil einer wirklich sicheren Aufbewahrung sicherheitskritischer Informationen. Dadurch ist es sogar irrelevant, ob die Informationen (Daten und Dokumente) in einer Public oder Private Cloud liegen.

9 Anwendungsbeispiel – Der moderne Arbeitsplatz der Verwaltung

9.1 Grundvoraussetzung

„Informationen“ sind eine der wichtigsten Ressourcen, mit der die öffentliche Verwaltung arbeitet. Deshalb wird Dokumenten und Daten als Träger und zur Übermittlung von Informationen ein besonders hoher Stellenwert in der täglichen Arbeit zugeschrieben. Formulare, Anträge, Bescheide, Urkunden, Akten, Verzeichnisse, Datenbanken – um nur einige typische Aspekte dieses Verwaltungshandelns zu nennen. Dabei ist die öffentliche Verwaltung an besondere Vorgaben gebunden, wie das Schriftlichkeitsprinzip (z.B. zur Sicherung der Beweisführung) und das Aktenprinzip (z.B. zur transparenten und nachvollziehbaren Rechenschaftslegung). Eng damit verbunden sind gesetzliche Aufbewahrungspflichten, und -fristen, u.a. für Haushaltspläne, Personalakten, Gerichtsakten und Namensverzeichnisse. Aufbewahrungsfristen können bis zu 30 Jahre betragen, in Einzelfällen auch „ewig“ (beispielsweise bei Grundstücksinformationen).

Vor diesem Hintergrund und der Tatsache, dass digitale Dokumente und Daten die papiergebundenen Varianten sukzessive ersetzen, ergeben sich verschiedene Herausforderungen für die öffentliche Verwaltung. Denn Dokumente, die bisher als Papiervariante in Aktenmappen verteilt wurden, müssen in einer digitalisierten Verwaltung in elektronischer Form weiterhin den reibungslosen Ablauf und die Einhaltung der besonderen Vorgaben sicherstellen:

- Welche Voraussetzungen müssen erfüllt sein, damit digitale Dokumente und Daten auch nach jahrelanger Aufbewahrung noch geöffnet, ohne Qualitätsverlust vorgelegt (bspw. als Beweismittel in einem Gerichtsprozess) und korrekt weiterverarbeitet werden können? Wie stellt man die Lesbarkeit, Integrität und Authentizität der Dokumente als Voraussetzung für die digitale Aufbewahrung sicher?
- Bei der elektronischen Kommunikation zwischen Verwaltungen, Unternehmen und Bürgern muss die Rechtsverbindlichkeit von Dokumenten sichergestellt werden. Elektronische Signaturen müssen von verschiedenen Anwendungen erkannt und korrekt berücksichtigt werden und auch bei einer längeren Aufbewahrung ihre Rechtsgültigkeit behalten.

Aufbewahrungsfristen

„(…) Aufbewahrungsfristen sind Kategorien oder dokumentspezifisch geregelt und bewegen sich im Allgemeinen zwischen 3 und 10 Jahren. In seltenen Fällen sind Dokumente bis zu 30 Jahren, bei relevanten Angaben etwa über Grundstücke oder langfristige Rechtsverhältnisse auch „ewig“ aufzubewahren.“ [BMWi2007], siehe Tabelle 2.

Anwendung	Rechtsgrundlage	Aufbewahrungsdauer
Verwaltungsrecht	In Abhängigkeit der Dokumentart	Im allgemeinen nicht über 30 Jahre (beachte aber eine ggf. spätere Archivierungspflicht)
Medizinrecht	§ 10 Abs. 3 MBO-Ärzte § 28 Abs. 4 RöntgVO	Grundsätzlich 10 Jahre Bis zu 30 Jahren
HGB und Steuerrecht	§ 257 Abs. 4 HGB, § 147 Abs. 3 AO	Je nach Dokumenten- oder Datenart 6 oder 10 Jahre

Tabelle 2: Beispielhafte Aufbewahrungsfristen entsprechend des angegebenen Fachrechts

Weitere Aufbewahrungsfristen können sich allerdings z.B. aus längeren Verjährungsfristen oder aus der Berücksichtigung des Inhalts und des Aufbewahrungszwecks der Dokumente (z.B. zur Beweissicherung im Rahmen eines Gerichtsverfahrens) ergeben[BMWi2007].

Elektronische Aktenführung/Ersetzendes Scannen

Wesentliche Aspekte, die sowohl national (NEGS) als auch regional (z.B. E-Government und Organisationsgesetz der Senatsverwaltung für Inneres und Sport, vgl. [Rie2010]) diskutiert werden und in erheblichem Maße Auswirkungen auf die Art und Weise der zukünftigen Verwendung elektronischen Schriftguts in der Öffentlichen Verwaltung haben werden, sind die Förderung des Einsatzes elektronischer Aktenführung sowie das Ersetzende Scannen. Um Verwaltungsprozesse möglichst medienbruchfrei elektronisch abzubilden, kann das Ersetzende Scannen einen großen Beitrag leisten, weil Vorgänge dann vollständig elektronisch geführt werden können.

Im aktuellen Memorandum des IT-Planungsrates vom 30.06.2011 ([ITPa2011] heißt es, „Die IT-Sicherheit ist ein wichtiger Baustein für unsere erfolgreiche Modernisierung. Bund und Länder müssen IT-Verfahren den aktuellen Bedrohungen aus dem Internet stetig anpassen." Entsprechend müssen elektronische Dokumente in der Öffentlichen Verwaltung sicher aufbewahrt und gegen unbefugten Zugriff von außen geschützt werden. Dazu bietet sich der elektronische Safe als Cloud-Service an.

Eines der Ziele im Rahmen der Nationalen E-Government-Strategie (NEGS) ist, dass „alle geeigneten Verwaltungsangelegenheiten (…) sich über das Internet abschließend elektronisch erledigen [lassen]" [ITPa2011]. „Hierzu streben Bund und Länder im Rahmen ihrer Zuständigkeiten an:

- **elektronische und papiergebundene Kommunikation** rechtlich gleich zu stellen,
- **Schriftformerfordernisse und weitere Formvorschriften** zur Vereinfachung der elektronischen Kommunikation mit der Verwaltung, wo immer möglich abzubauen (…)."

Ferner ist es erklärtes Ziel der NEGS, dass die Zusammenarbeit von Bund, Ländern und Kommunen regelmäßig über Mittel der IKT erfolgt. Ein wesentliches Handlungsfeld dabei ist „die organisatorische und technische Ermöglichung des sicheren elektronischen Austauschs von Akten, Vorgängen und Dokumenten (…)“ [ITPa2010].

Informationelle Gewaltenteilung, das heißt „die Trennung zwischen den von verschiedenen Verwaltungsbereichen für unterschiedliche Zwecke erhobenen Daten“ [ITPa2010] erfordert, dass Daten nach entsprechenden Rollen und Nutzungskonzepten sicher in die Prozesse eingespeist werden müssen.

Im Rahmen der Diskussion zum E-Government Gesetz des Bundes, welches voraussichtlich im Herbst 2011-Ende 2012 Eingang ins Gesetzgebungsverfahren finden wird, werden zur Erreichung der Förderung des weiteren Ausbaus von E-Government durch Querschnittsnormen insbesondere die „Verpflichtung zur elektronischen Führung neuer Register“ als Beispiele für sogenannte „Motornomen“ genannt [Lai2010]. Auch hieraus lässt sich ein Trend hin zu einer zunehmenden Digitalisierung des Schriftguts in der öffentlichen Verwaltung ableiten. Zudem wird der Abbau bundesrechtlicher Hemmnisse angestrebt. Dazu gehören Bürokratieabbau, Verfahrensvereinfachung, Rechtsbereinigung, Öffnung für neue technische Verfahren etc. Gemäß des deutschen Signaturgesetzes erfordert die Abbildung der Schriftform in der elektronischen Welt die qualifizierte elektronische Signatur. Laier geht davon aus, dass die Masse der im Bundesrecht vorhandenen Schriftformerfordernisse teilweise historisch begründet und nicht immer tatsächlich alle Funktionen der Schriftform erforderlich seien. Vielmehr wird davon ausgegangen, dass andere technische Verfahren wesentliche Funktionen der Schriftform erfüllen würden [Lai2010].

Dies spiegelt sich auch im Vorhaben des IT-Planungsrates wieder, in der Zeit von 2011-2015 eine „gemeinsame (…) **eID-Strategie** [zu erarbeiten], um Bürgerinnen und Bürgern den sicheren Austausch mit der Verwaltung und der Wirtschaft über das Internet zu ermöglichen (…)“ [ITPa 2011].

9.2 Der Cloud-Safe der Verwaltung

Die Idee, Daten auf elektronischem Wege in Behördenprozesse zu integrieren, ist bereits seit mehreren Jahren ein relevantes Thema in Forschung und Wirtschaft. So existieren bisher verschiedene Ansätze, die versuchen, dem Bürger über das Internet durch Portalauftritte von Städten oder Kommunen die Kommunikation mit Behörden zu erleichtern. In Deutschland haben bspw. die Städte Münster und Hagen den neuen Personalausweis in ihre Online-Portale integriert, um das Ausfüllen von Dokumenten im Bereich von Kindergeld oder der Hundesteuer zu automatisieren. Durch einen Knopfdruck wird der Ausleseprozess der Daten aus dem neuen Personalausweis gestartet und anschließend in das Dokument gefüllt. Dies ist nur einer von vielen Anwendungsfällen, in denen die Verwaltung die Grundlage für eine elektronische Abwicklung von Prozessen zwischen Bürger, Wirtschaft und Verwaltung realisiert hat.

Durch die elektronische Abwicklung von Verwaltungsprozessen im Antrags- und Auskunftsverfahren entstehen zahlreiche elektronisch signierte Anträge, Bescheide, Urkunden usw., die in ihrem Beweiswert zu erhalten und einer Wiederverwendung zuzuführen sind.

Der Cloud-Safe kann hier als Schnittstelle zwischen Verwaltungen, Unternehmen und „Lebenslagen“ der Bürger dienen. Ein von der Verwaltung betriebener Cloud-Safe macht elektronische Dokumente und Daten für die Eigentümer der Daten permanent verfügbar - beispielsweise die Urkunden, die in verschiedenen Verwaltungsprozessen wiederverwendet werden müssen. Somit ist eine wiederholte prozessuale Einbindung von Daten in andere Verwaltungsprozesse möglich.

Der Cloud-Safe der Verwaltung bietet mandantenbezogene – also private Sicherheitsbereiche für die Verwaltung elektronischer Daten und Dokumente. Die Realisierung des Cloud-Safes deckt alle drei Ebenen des Cloud Computing ab.

Infrastructure-as-a-Service (IaaS): Die Bürger sind gleichzeitig die Besitzer ihres angemieteten privaten **Safes** und der darin enthaltenen Dokumente bzw. von der Verwaltung erzeugten Urkunden. Das Rechenzentrum eines Verwaltungsdienstleisters tritt als Cloud-Safe Anbieter auf und ist im besten Fall ein sicherheitszertifiziertes Rechenzentrum.

Platform-as-a-Service (PaaS): Es wäre sinnvoll, den Cloud-Safe mit etablierten Fachverfahren zu integrieren, wobei zu berücksichtigen ist, dass ein verbindliches Austauschformat für die Langzeitarchivierung, wie XAIP genutzt wird. Damit kann der Cloud-Safe als nutzergesteuerte Datenaustauschplattform dienen, die zudem noch die Beweiswerterhaltung der eingelagerten Daten gewährleistet.

Software-as-a-Service (SaaS): Als SaaS Cloud-Safe sind weitere Services einzubeziehen; wie der neue Personalausweises als Authentisierungsmittel, Entwicklung einer Client-Anwendung zur Aufteilung der zu sichernden Dokumente in Geheimnisteilchen und Verwaltungsfunktionen für den Cloud-Safe, ein Viewer zur Anzeige der abgelegten Daten, Bereitstellungs- und Löschfunktionen u.v.a. Mit dem Cloud-Safe eröffnet sich die Möglichkeit, elektronische Anträge in den privaten Cloud-Safe zu übernehmen, elektronische Nachweise aus dem privaten Cloud-Safe anzufügen und Bescheide in dem Cloud-Safe abzulegen.

10 Fazit und Ausblick

In der Studie haben wir gezeigt, dass der Cloud-Safe prinzipiell ein vertrauenswürdiger Ort für die Ablage und Verwaltung elektronischer Dokumente ist, sofern bestimmte Maßnahmen berücksichtigt werden.

Sowohl für die Vertrauensbildung, als auch für die Integrität wird das in Kapitel 6.8 diskutierte GSSCC-Verfahren vorgeschlagen, das allerdings zurzeit noch von keinem Cloud-Anbieter realisiert ist. Dies stellt eine Herausforderung an Cloud-Anbieter, die sich mit einer solchen Realisierung deutlich von der Masse hervorheben können. Mit diesem Verfahren bieten sie eine sichere Aufbewahrung sensibler Informationen an und können, wenn sie dann noch zusätzlich den Zertifizierungsprozess durch externe Auditoren durchlaufen, mit einem entsprechenden Gütesiegel als vertrauenswürdig angesehen werden.

Wird der Cloud-Safe in den Verwaltungsprozess eingebunden, so entsteht für die Verwaltung ein neues Dienstleistungsmodell, das es gestattet, den Bürgern einen Cloud-Safe zur sicheren Verwahrung und dem sicheren, rechtskonformen Austausch elektronischer Dokumente und Daten zur Verfügung zu stellen. Behördliche Prozesse können neu gestaltet und optimiert werden, indem diese Umgebung zu einer Plattform für den Austausch von Dokumenten und Daten wird. Dabei werden Zugriffe grundsätzlich durch den Eigentümer der Dokumente und Daten autorisiert, wobei bevorzugt der neue Personalausweis als Authentisierungsmittel zum Einsatz kommt. Der Cloud-Safe kann somit zur Schlüsseltechnologie für ein kooperatives eGovernment werden.

Mit dem Verfahren basierend auf Shamir‘s Secret Sharing kann zusätzlich zu den Verfügbarkeits-Mechanismen der Speicher-Anbieter Redundanz eingebracht werden. Der Verlust von $n - k$ Shares ist unkritisch. Es bleibt daher noch Zeit, die Daten zu sichern, sollte der Verlust einzelner Shares registriert werden. Darüber hinaus wird die Verfügbarkeit verbessert, sollte zum Zeitpunkt des Abrufes ein Teil des Cloud-Netzwerkes nicht erreichbar sein.

Zudem zeigt die Analyse der Cloud-Anbieter IBM, Microsoft, Amazon und Google, dass das Übertragen der Prinzipien des elektronischen Safes auf Cloud-Services umsetzbar ist und welche die meisten Dienste anbieten, um Authentifizierung, Autorisierung und Logging oder Auditing zu nutzen. Die Verschlüsselung und das S*ecret Sharing*-Verfahren muss der Cloud-Safe Anbieter selbst realisieren, da diese Funktionen in der Form nicht vorhanden sind.

11 Literaturverzeichnis

Aza2009 P. Azar. Secret sharing and applications. *Harvard College Mathematics Review*, 2009.

BC1996 G. Brassard and C. Crépeau. 25 years of quantum cryptography. *SIGACT News*, 27:13–24, September 1996.

BD19899 E. F. Brickell and D. M. Davenport. On the classification of ideal secret sharing schemes (extended abstract). In *CRYPTO '89: Proceedings on Advances in cryptology*, pages 278–285, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

Ber2009 D. J. Bernstein. Introduction to post-quantum cryptography. In D. J. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post-Quantum Cryptography*, pages 1–14. Springer Berlin Heidelberg, 2009.

BFK2009 J. Bender, M. Fischlin, and D. Kügler. Security analysis of the pace key-agreement protocol. In *Proceedings of the 12th International Conference on Information Security*, ISC '09, pages 33–48, Berlin, Heidelberg, 2009. Springer-Verlag.

BIT2010 Cloud Computing – Was Entscheider wissen müssen, Ein ganzheitlicher Blick über Technik hinaus. Positionierung, Vertragsrecht, Datenschutz, Informationssicherheit, Compliance. Leitfaden. Herausgegeben von BITKOM (Bundesverband Informationswirtschaft, Telekommunikation und neue Medizin e.V.))

BIT2010 Cloud Computing – Was Entscheider wissen müssen, Leitfaden BITKOM, 2010

Bla1979 G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, Monval, NJ, USA, 1979. AFIPS Press.

BMWi 2007 Handlungsleitfaden zur Aufbewahrung elektronischer und elektronisch signierter Dokumente. [Online] August 2007, S. 15. http://www.bmwi.de/BMWi/Navigation/Service/publikationen,did=218700.html. Dokumentation Nr. 564.

BR2006 M. Bellare and P. Rogaway. "Code-based game-playing proofs and the security of triple encryption". In *Eurocrypt 2006*, volume 4004, pages 409–426. Springer, 2006.

BSI2011 Bundesamt für Sicherheit in der Informationstechnik. Beweiswerterhaltung kryptographisch signierter Dokumente. BSI-TR-03125, Version 1.1, 2011.

HKP2010 C. Hoffmann, J. Klessmann, A. Penski, Dr. S. Schulz, T. Warnecke, ISPRAT Studie „Dienste auf Basis elektronischer Safes für Daten und Dokumente", 2010

BSI-Cloud Sicherheitsempfehlungen für Cloud Computing Anbieter, Eckpunktpapier, Bundesamt für Sicherheit in der Informationstechnik, 2011

BSI-Mail BSI, De-Mail Technische Richtlinien, TR-01201, https://www.bsi.bund.de/DE/Themen/EGovernment/DeMail/TechnischeRichtlinien/TechnRichtlinien_node.html

Bund2011 Bundesgesetzblatt, Jahrgang 2011 Teil I Nr. 19 ausgegeben zu Bonn am 2. Mai 2011

CGH2004 R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *J. ACM*, 51:557– 594, July 2004.

CM1997 C. Cachin and U. Maurer. Unconditional security against memory-bounded

adversaries. In B. Kaliski, editor, *Advances in Cryptology– CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 292–306. Springer Berlin / Heidelberg, 1997.

CPS2008 J.-S. Coron, J. Patarin, and Y. Seurin. The random oracle model and the ideal cipher model are equivalent. In D. Wagner, editor, *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin / Heidelberg, 2008.

CW1993 K. W. Campbell and M. J. Wiener. Des is not a group. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 512–520, London, UK, 1993. Springer-Verlag.

DD1994 E. Dawson and D. Donovan. The breadth of Shamirs secret-sharing scheme. *Comput. Secur.*, 13(1):69–78, 1994.

DSP2010 P. H. Deussen, L. Strick, and J. Peters. Cloud Computing für die öffentliche Verwaltung . ISPRAT Studie November 2010. Fraunhofer-Institut für Offene Kommunikationssysteme FOKUS, Electronic Government and Applications (ELAN), November 2010.

Eck2009 C. Eckert. IT-Sicherheit. Oldenbourg, 6. edition, 2009.

EG1985 S. Even and O. Goldreich. On the power of cascade ciphers. *ACM Trans. Comput. Syst.*, 3:108–116, May 1985.

EGEU2010 The Future Of Cloud Computing - Opportunities for European Cloud Computing Beyond 2010: Raport von Expert Group herausgegeben von Europäischen Union

Gef1965 P. R. Geffe. Secrecy systems approximating perfect and ideal secrecy. In *Proceedings of the IEEE*, volume 53, pages 1229–1230. IEEE Journals, 1965.

GM2009 P. Gaži and U. Maurer. Cascade encryption revisited. In *ASIACRYPT '09 Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*. Springer-Verlag Berlin, 2009.

Gro1997 L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *PHYS.REV.LETT.*, 79:325, 1997.

HJK1995 A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. pages 339–352. Springer-Verlag, 1995.

HKL2009 D. Hühnlein, U. Korte, L. Langer, and A. Wiesmaier. A comprehensive reference architecture for trustworthy long-term archiving of sensitive data. In Proceedings of the 3rd international conference on New technologies, mobility and security, NTMS'09, pages 48–52, Piscataway, NJ, USA, 2009. IEEE Press.

ITPa2011 IT-Planungsrat > Startseite > Memorandum des IT-Planungsrats. *www.it-planungsrat.de.* [Online] 30. Juni 2011. [Zitat vom: 4. Juli 2011.] http://www.it-planungsrat.de/SharedDocs/Pressemitteilungen/DE/2011/Memorandum%20des%20IT-Planungsrats.html?nn=1299634.

ITPb2011 IT-Planungsrat > Startseite > Memorandum des IT-Planungsrats. *www.it-planungsrat.de.* [Online] 30. Juni 2011. [Zitat vom: 4. Juli 2011.] http://www.it-planungsrat.de/SharedDocs/Pressemitteilungen/DE/2011/Memorandum%20des%20IT-Planungsrats.html?nn=1299634.

KL2007 J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

Knu1998 L. R. Knudsen. Contemporary block ciphers. In *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, pages 105–126, London, UK, 1999. Springer-Verlag.

Kra1994 H. Krawczyk. Secret sharing made short. In *CRYPTO '93: Proceedings of the 13th annual international cryptology conference on Advances in cryptology*, pages 136–146, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

Lea2009 N. Leavitt. Is cloud computing really ready for prime time? Computer, 42(1):15–20, jan. 2009.

Lai2010 Laier, Tanja. *E-Government-Gesetz des Bundes.* [PPT] Messe Moderner Staat 2010 : BMI, Oktober 2010.

LV2001 A. K. Lenstra and E. R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14:255– 293, 2001.

LW2009 L. Langer and A. Wiesmaier. Langfristige Sicherheit am Beispiel eines virtuellen Tresors. Technical Report. Technische Universität Darmstadt. 2009.

Mau1993a U. Maurer. Protocols for secret key agreement by public discussion based on common information. In E. Brickell, editor, *Advances in Cryptology – CRYPTO 92*, volume 740 of *Lecture Notes in Computer Science*, pages 461–470. Springer Berlin / Heidelberg, 1993.

Mau1993b U. M. Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39:733–742, 1993.

MDN2006 T. Miyamoto, S. Doi, H. Nogawa, and S. Kumagai. Autonomous distributed secret sharing storage system. *Syst. Comput. Japan*, 37(6):55–63, 2006.

MG2011 P. Mell and T. Grance. The nist definition of cloud computing (draft). Technical report, National Institute of Standards and Technology (NIST), 2011. Special Publication 800-145 (Draft).

MH1981 R. C. Merkle and M. E. Hellman. On the security of multiple encryption. *Commun. ACM*, 24:465– 467, July 1981.

MM1993 U. M. Maurer and J. L. Massey. Cascade ciphers: The importance of being first. *Journal of Cryptology*, 6:55–61, 1993.

NIST National Institute of Standards and Technology (NIST), Cloud Computing Program, http://www.nist.gov/itl/cloud

NIST1999 National Institute of Standards and Technology. Data encryption standard (DES). FIPS Publication 46-3, October 1999.

Post2011 Deutsche Post AG, Leistungsbeschreibung E-Postbrief, Stand 05/2011, http://service.epost.de/downloads/7/leistungsbeschreibung_e-postbrief.pdf

Rab1989 M. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*, 36:335–348, 1989.

Rab2005 M. Rabin. Provably unbreakable hyper-encryption in the limited access model. In *IEEE*, pages 34 – 37, 2005.

Rab2006 M. Rabin. Provably unbreakable hyper-encryption using distributed systems. In S. Dolev, editor, *Distributed Computing*, volume 4167 of *Lecture Notes in Computer Science*, pages 575–577. Springer Berlin / Heidelberg, 2006.

Red2011 Dr. Viviane Reding, Herausforderungen an den Datenschutz bis 2020: Eine europäische Perspektive. ZEITSCHRIFT FÜR DATENSCHUTZ, 1/2011, www.zd-beck.de

RTS2009 T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In ACM Conference on Computer and Communications Security, pages 199–212. ACM, 2009.

Rie2010 Rienaß, Udo. Das Berliner Vorhaben „E-Government- und Organisationsgesetz". *www.berlin.de.* [Online] Oktober 2010. [Zitat vom: 4. Juli 2011.] http://www.google.de/url?sa=t&source=web&cd=12&ved=0CBoQFjABOAo&url=http%3A%2F%2Fwww.berlin.de%2Fimperia%2Fmd%2Fcontent%2Fverwaltungsmodernisierung%2Fmodernerstaat2010%2F100929_riena__vortrag_zum_e_government_gesetz_final.pdf&rct=j&q=egovernment%20ersetz.

Rup2010a A. Ruppel. Angriffsarten und Angreifertypen in Cloud-Computing-Systemen. http://www.searchsecurity.de/themenbereiche/applikationssicherheit/web-applicationsecurity/articles/254228/, 2010.

Rup2010b A. Ruppel. Vertraulichkeit, Integrität und Verfügbarkeit beim Cloud Computing. http://www.searchsecurity.de/themenbereiche/applikationssicherheit/web-applicationsecurity/articles/254228/, 2010.

SB2005 A. Subbiah and D. Blough. Practical share renewal for large amounts of data, 2005.

Sch1996 B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.

Sha1949 C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, Oktober 1949.

Sha1979 A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

SR2009 W. Streitberger and A. Ruppel. Cloud Computing Sicherheit - Schutzziele. Taxonomie. Marktübersicht. Fraunhofer-Institut für Sichere Informations Technologie SIT, Sept. 2009.

Sti1992 D. R. Stinson. An explication of secret sharing schemes. *Des. Codes Cryptography*, 2(4):357–390, 1992.

Str2009 W. Streitberger. Risk management for Grid systems – An analysis of insurances on the basis of a simulated Grid economy. PhD thesis, Universität Bayreuth, 2009.

TW1988 M. Tompa and H. Woll. How to share a secret with cheaters. *J. Cryptol.*, 1(2):133–138, 1988.

UC2010 Cloud Computing Use Cases, White Paper version 4.0, produziert bei Cloud Computing Use Case Discussion Group, July 2010

VA2010 Vitako Aktuell. Cloud Computing – Entdecke die Möglichkeiten. 2010

Ver1926 G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *American Institute of Electrical Engineers*, XLV:109–115, 1926.

Win2010 M. Winkelmann. Cloud Computing: Sicherheit und Datenschutz. http://www.stiftungaktuell.de/files/cloudcomputing_winkelmann.pdf, 2010. Arbeitspapier für die Alcatel-Lucent Stiftung.