MASTERARBEIT

VERTRAULICHE BERECHNUNGSMETHODEN IM RAHMEN DES EINSATZES VON CLOUD COMPUTING BEI DER DATENVERARBEITUNG

Eine szenarienbasierte Evaluierung der Praxisfähigkeit von

aktueller homomorpher Verschlüsselungstechnologie

ausgeführt am

Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: Philipp Grobelscheg, BSc BSc

Personenkennzeichen: 1710320003

Graz, am 14. Dezember 2018 .....

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

DANKSAGUNG

An dieser Stelle will ich mich bei allen bedanken, die mir in den letzten Jahren während des Studiums und im Speziellen während dieser Masterarbeit zur Seite gestanden sind.
Hierbei möchte ich mich bei meinem Betreuer Dipl.-Ing. (FH) Christian Schmid, MSc bedanken für die gute Betreuung und die freundlichen Gespräche. Ein besonderer Dank gilt meiner Freundin Anna, welche mir über die letzten Jahre, trotz der dreifachen Beanspruchung aus zwei Studien und meiner Tätigkeit als Applikationsentwickler immer zur Seite gestanden ist. Ein großer Dank gilt auch meiner Familie für die außerordentliche Unterstützung, welche mir immer viel bedeutet hat.

KURZFASSUNG

In dieser Arbeit werden mögliche Anwendungen des vertraulichen Cloud Computing im Zusammenhang mit der Datenverarbeitung evaluiert. Dazu wird eine experimentelle Umgebung, in der vertrauliche homomorphe Verschlüsselung für verschiedene Szenarien angewendet und deren praktische Anwendbarkeit bewertet wird, geschaffen. Cloud Computing hat wirtschaftlich ein großes Potenzial. Insbesondere im Kontext der Anwendung von Cloud Computing im Zusammenhang mit der Verarbeitung sensibler Daten besteht jedoch keine Garantie für den Erhalt des Datenschutzes. Die Wahrung der Vertraulichkeit von persistierten Daten wird zumeist durch bewährte Verschlüsselungsmethoden gewährleistet. Vor der Durchführung von Berechnungen müssen diese jedoch in Klartext entschlüsselt werden. Die homomorphe Verschlüsselung bietet eine eingeschränkte Möglichkeit diese Berechnungen auch mit verschlüsselten Daten durchzuführen. Da sich dieses Forschungsgebiet in den letzten zehn Jahren stark entwickelt hat, gibt es noch relativ wenig Anhaltspunkte dafür, inwiefern die Verwendung homomorpher Verschlüsselung in realen Anwendungen praktikabel sein kann. In dieser Arbeit werden in einem ersten Schritt Szenarien erstellt, welche eine Verbindung zwischen den Potenzialen der Technologie und den kommerziellen sowie politischen Erwartungen herstellen. Im nächsten Schritt wird durch eine Evaluierung verschiedener homomorpher Verschlüsselungsbibliotheken beurteilt, welche kryptographische Basis für alle erforderlichen mathematischen Operationen der entworfenen Szenarien geeignet ist. Anschließend werden die entworfenen Szenarien implementiert und hinsichtlich ihrer Ergebnisqualität, ihrer Speicherauslastung und ihrer Laufzeit ausgewertet. Abhängig von der technischen Ausrichtung des Szenarios wird ein hohes oder niedriges Implementierungspotenzial für reale Anwendungen bestimmt. Die Ergebnisse sollten jedoch insbesondere im Hinblick auf den zugrunde liegenden Kontext interpretiert werden. In Zukunft gilt es weitere praxisnahe Szenarien zu entwerfen, zu implementieren und ausführlich zu testen, um die Grenzen und das Potenzial dieser Technologie besser einschätzen zu können.

ABSTRACT

This thesis evaluates potential applications of confidential cloud computing in the context of data processing. The author creates an experimental environment that applies confidential homomorphic encryption to a variety of scenarios and assesses their practicality. Cloud computing has great potential economically. However, especially in the context of the application of cloud computing related to the processing of sensitive data, there is no guarantee of privacy. Permanently securing the confidentiality of data is mostly ensured by proven encryption methods, but especially prior to performing calculations, they must be decrypted into plain text. Homomorphic encryption provides a limited possibility to perform these calculations on encrypted data. Since this field has developed strongly in the last 10 years, there is still relatively little evidence of the practicality of using homomorphic encryption in real-world applications. In a first step, this thesis investigates the creation of scenarios which establish a connection between the potentials of the technology, and the commercial as well as political expectations. The next step assesses which cryptographic basis is suitable for all necessary mathematical operations of all designed scenarios by an evaluation of different homomorphic encryption libraries. Subsequently, the designed scenarios are implemented and evaluated on their quality of results, the memory load, and their running time, to gain insights into possible practical applications. Depending on the technical focus of the scenario, a high or low potential for implementation in real-world applications is determined. However, the results should be interpreted especially in terms of their underlying context. In the future, it will be necessary to design, implement, and extensively test further practice-like scenarios to better understand the limits and the potential of this technology.

INHALTSVERZEICHNIS

1. EINLEITUNG

Durch die stetig zunehmende Verwendung von Daten als Quelle für analytische Entscheidungsfindungen innerhalb der Wirtschaft und auch in anderen Gebieten, wie der Forschung und dem Staatswesen, nimmt der Berechnungsaufwand für einzelne Organisationen immer stärker zu. (Sarunski, 2016)

Dies ließ in den letzten Jahren Anbieter von Cloud Computing Kapazitäten auffallend stark wachsen, was wiederum den Einfluss dieser meist global agierenden Unternehmen ausweitete. Gleichzeitig werden auch die Themen Datensicherheit und Datenschutz in Wirtschaft und Gesellschaft immer wichtiger. (Bedner, 2013; Münzl, Pauly, & Reti, 2015)

Es gibt aktuell einige Methoden zur vertraulichen Speicherung von Daten, welche durch moderne Verschlüsselungsverfahren bewerkstelligt werden können. Auch der sichere Transport von Informationen kann auf vertraulichem Wege durchgeführt werden. In diesen Technologiebereichen gibt es zwar auch Fortschritte bzw. werden vermeintlich sichere Protokolle oder Verfahren als unsicher überführt, aber trotzdem beinhalten diese Technologien – im Gegensatz zum Thema dieser Masterarbeit – in der Praxis schon seit Jahren bewährt eingesetzte Methoden. (Bedner, 2013)

Dieses Themengebiet beschreibt den letzten Bereich von Vertraulichkeit beim Einsatz von Cloud Computing, welcher noch nicht erwähnt wurde. Wenn die verschlüsselten Daten in Datenbanken oder in anderer Form auf der Infrastruktur einer dritten Partei liegen, dann kann diese die Daten zwar nicht lesen, aber auch eine weitere Verarbeitung ist dabei in den meisten Fällen ausgeschlossen. Doch von den Vorteilen von Cloud Computing kann nur profitiert werden, wenn auch die aufwendigen Berechnungsschritte von der Skalierbarkeit der Infrastruktur des spezialisierten Cloud-Anbieters vollzogen werden. Dies führt zu einem Dilemma beim gewünschten Einsatz von Cloud Computing im Rahmen von gesetzlich oder auch wirtschaftsmachtpolitisch sensiblen Daten. Eine der Lösungen für diese Problematik ist der Einsatz von homomorphen Verschlüsselungsmethoden. (Bedner, 2013)

Diese Entwicklungen führen zu einer weiteren Vertiefung von diversen Methoden zur Berechnung großer Datenmengen durch Dritte, bei denen die berechnende Partei keine Informationen über die Daten erhalten soll. Aufgrund der vielseitigen Szenarien der Anwendung und der unterschiedlichen Ausprägungen der Vertraulichkeit von Daten, sowie der Berechnungen, sind hierfür in den letzten Jahrzehnten unterschiedlichste Verfahren mit eigenen Prioritäten, Voraussetzungen und Limitierungen entworfen worden. (Rountree & Castrillo, 2014)

Da in der Öffentlichkeit oft nach einer generellen Lösung für das Problem gesucht wird, kommt dabei die homomorphe Verschlüsselung verstärkt in die Schlagzeilen von wirtschaftlichen und technologischen Medien. (Brenner, 2016; Greenberg, 2009) Dort wird sie oft als eine allgemeine Lösung gesehen, doch zahlreiche Experten in wissenschaftlichen Fachjournalen haben ihre Bedenken für eine Verwendung als Generallösung ausgesprochen. (Dowlin et al., 2017;

Hoffstein, Pipher, & Silverman, 2014) Es gilt nun herauszufinden, welche Anwendungsszenarien in Bezug auf vertrauliches Cloud Computing im Kontext der Datenanalyse durch Dritte in der realen Wirtschaft und Gesellschaft existieren und wie die grundsätzliche Lösung dieser Berechnungen unter Abwägung der Vor- und Nachteile der Alternativen zur Wahrung der Vertraulichkeit der Daten bestmöglich gestaltet werden kann. (Brenner, Perl, & Smith, 2012)

Momentan herrscht zu dieser Problematik eine sehr breite Meinungsvielfalt innerhalb von Wirtschaft und Technik und allgemeine Vorhersagen, welche Rolle diese Berechnungsmethoden in der Zukunft spielen werden, sind mit Vorsicht zu betrachten. Diese Thematik wird in den nächsten Jahren noch sehr ambivalent diskutiert werden, vor allem da sich die Rahmenbedingungen dieses Spannungsfelds aus wirtschaftlicher Relevanz, legislativen Vorgaben und technologischem Fortschritt in kurzer Zeit ändern können. (Bowen, 2011; Ghorbel, Ghorbel, & Jmaiel, 2017)

Innerhalb dieser Arbeit soll nicht nur festgestellt werden, welche Anwendungsszenarien für bestimmte Organisationen von Relevanz wären, sondern auch welche dieser Szenarien beispielhaft für die Nutzung der zugrundeliegenden Berechnungsmethode sind. Es wurde dafür ein Set aus unterschiedlichen Szenarien, welche auf eine Analyse des Einsatzes von Cloud Computing in der Wirtschaft beruhen, konzeptioniert. Diese Szenarien wurden mittels derselben Bibliothek für homomorphe Verschlüsselung implementiert, aber mit unterschiedlichen technologischen und systematischen Schwerpunkten.

Mehrere Stunden an reiner Laufzeit wurden für die Auswertungen der Implementierungen mit unterschiedlichen Eingangswerten aufgewendet. Am Schluss ergab sich durch die Analyse der Auswertungen ein Bild, welches viele Grenzen der Technologie im Kontext der zugrundeliegenden Szenarien aufzeigt. Aber es wurden auch praktikable Möglichkeiten bei der Anwendung von homomorpher Verschlüsselung für weitere Versuche unter bestimmten Umständen erkannt. Mit den Resultaten soll weiteres Licht in die komplexe Materie gebracht werden.

2. FORSCHUNGSFRAGE

Innerhalb dieser Masterarbeit soll die folgende Forschungsfrage beantwortet werden:

„Welche Anwendungsmöglichkeiten ergeben sich durch den Einsatz von aktuellen Technologien zur homomorphen Verschlüsselung im Rahmen der Berechnung mit vertraulichen Daten durch Cloud Computing Infrastruktur?“

2.1 Wissenschaftliche Methodik

Diese Masterarbeit wird die Forschungsfrage mittels Erkenntnissen, welche durch die Kombination einer State-of-the-Art Analyse, dem Design von Untersuchungsszenarien, der Umsetzung dieser innerhalb einer Proof-of-Concept-Implementierung und der Analyse der Experimente gewonnen wurden, beantworten.

Zuvor wird eine Literaturanalyse über die bereits vorhandene Forschungsarbeit zu den Themen „Vertrauliches Cloud Computing“, „Homomorphe Verschlüsselung“ und „Praktische Anwendungsgebiete von vertraulichem Cloud Computing“ durchgeführt. Resultierend daraus, sollen technologische und wirtschaftliche Erkenntnisse die Grundlage für das darauffolgende Experiment und damit für die Versuchsimplementierungen darstellen.

Beim Experiment werden unterschiedliche Berechnungen in an die Praxis angelehnte Beispielprobleme implementiert. Dabei werden unterschiedliche Szenarien geschaffen, welche hinsichtlich ihrer Aufgabenstellung und ihrer Datenmenge Unterschiede aufweisen. Diese Szenarien werden auch ohne Wahrung der Vertraulichkeit bei der Berechnung durchgeführt.

Die abschließende statistische Analyse der Ergebnisse soll hervorbringen, wie groß der entstehende Leistungs- und somit auch ein resultierender Kostenmehraufwand der homomorphen Verschlüsselung sind. Dabei soll die unterschiedliche Einsetzbarkeit der technischen Möglichkeiten für die verwendeten Verfahren näher betrachtet werden. Ableitungen für unterschiedliche Datenmengen sollen näher gebracht werden.

2.2 Relevanz der Forschungsfrage

Die Relevanz dieser Forschungsarbeit kann grundsätzlich in zwei unterschiedliche Interessensgebiete unterteilt werden. Aus wirtschaftlicher und auch aus wissenschaftlicher Sicht ist die Beantwortung der Forschungsfrage von Bedeutung und kann Antworten und Erkenntnisse für Problemstellungen in beiden Bereichen liefern. Doch auch als verbindendes Element zwischen dem aktuellen Stand der Wissenschaft und den aktuellen Bedürfnissen und Rahmenbedingungen der Wirtschaft können die Ergebnisse dieser Masterarbeit gesehen werden.

2.2.1 Wirtschaftliche und gesellschaftliche Relevanz der Forschungsfrage

Aus wirtschaftlicher Sicht ist die Weiterentwicklung von vertraulichem Cloud Computing von hohem Interesse. Wie in einer aktuellen Studie des deutschen Bundesverbands Bitkom gemeinsam mit der Wirtschaftsprüfungsgesellschaft KPMG ersichtlich wurde, nutzt in Deutschland schon eine Mehrheit der Unternehmen auch Angebote einer öffentlichen Cloud für unkritische Anwendungen. Großunternehmen vertrauen jedoch öffentlichen Clouds auch bei kritischen Anwendungen immer mehr. Gleichzeitig werden Sicherheitsbedenken als zentrale Hürde für die Nutzung von Cloud-Diensten gesehen. (Bitkom & KPMG, 2018a)

Dieses Spannungsfeld führt zu einer immer wichtigeren Abwägung zwischen Datensicherheit, Kostenvorteilen und Innovationsfähigkeit durch Investitionsflexibilität. Da – wie in den folgenden Kapiteln näher erläutert – die unterschiedlichen Formen von Datensicherheit eine sehr komplexe und heterogene Lösungslandschaft besitzen, muss man sich genau mit dem individuellen Nutzungsszenario auseinandersetzen um eine technologisch, aber auch wirtschaftlich sinnvolle Entscheidung treffen zu können. (Münzl et al., 2015)

Wenn es die Rahmenbedingungen für eine Verarbeitung von vertraulichen Daten in einer öffentlichen oder gemeinschaftlich genutzten Cloud zulassen, werden auch Wirtschaft und Organisationen des Staatswesens, wie Forschungseinrichtungen, sehr stark an einer Weiterentwicklung von vertraulichem Cloud Computing interessiert sein. Diese Forschungsarbeit soll die zwei Welten des technologischen Fortschritts und der wirtschaftlichen Bedürfnisse gleichzeitig betrachten, um auf die Frage nach den potentiellen Anwendungsgebieten außerhalb der Wissenschaft Antworten liefern zu können. (Münzl et al., 2015)

2.2.2 Wissenschaftliche und technologische Relevanz der Forschungsfrage

Die Wissenschaft von vertraulichem Cloud Computing ist sehr stark von Erfolgen der letzten zehn Jahre geprägt. Dies macht die Thematik zu einem sehr innovativen und schnell veränderbarem Wissenschaftsbereich. Dabei teilen sich das Interesse an diesem Forschungsgebiet Wissenschaftlerinnen und Wissenschaftler mit diversen Spezialisierungen. Diese Spezialisierungen reichen von der Kryptographie, über die Informatik, bis hin zur anwendungsorientierten Wirtschaftsinformatik. (Bedner, 2013)

Das Zusammenspiel dieser Parteien Bedarf noch viel grenzüberschreitender Literatur im Sinne dieser Forschungsbereiche. Zum Beispiel sind die Vorstellungen einiger Wissenschaftler der Wirtschaftsinformatik von möglichen Einsatzgebieten der homomorphen Verschlüsselung oft nicht fehlerfrei, was vermehrt zu Missverständnissen in Bezug auf die Anwendungsszenarien in der realen Wirtschaft führt. (Chirgwin, 2018)

Gleichzeitig wird von vielen Wissenschaftlerinnen und Wissenschaftlern eine ausgiebige Anwendung mit unterschiedlichen Experimentalumgebungen ihrer Implementierungen gewünscht. (Armknecht et al., 2015; Chen, Han, Huang, Jalali, & Laine, 2018) Zahlreiche praktische Forschungsarbeiten – welche in den folgenden Kapiteln näher erläutert werden –

konnten schon unterschiedliche Erkenntnisse bei der Anwendung auf realen Datenbasen feststellen.

Die Barriere für Wissenschaftlerinnen und Wissenschaftler, aber auch für Studierende und Forschungspersonal aus der Wirtschaft, bei der Weiterführung von bisherigen wissenschaftlichen Fortschritten wird oft aufgrund der Komplexität der Thematik, sowie durch fehlende und fehlerhafte Literatur erhöht. Eine weitere Forschungsarbeit, welche die aktuellen Technologiefortschritte in Verbindung mit geeigneten Anwendungsszenarien bringt, wird deswegen von diversen Seiten begrüßt. (Armknecht et al., 2015)

Weiters veränderte sich das Forschungsgebiet in den letzten Jahrzehnten stark. Dadurch wird eine State-of-the-Art Betrachtung der aktuellen Erkenntnisse meist jedes Jahr anders aussehen und kann aufgrund dieser Aktualität einen Beitrag zur Vervollständigung der wissenschaftlichen Literatur in besagtem Forschungsgebiet liefern. (Armknecht et al., 2015)

3. ANWENDUNGSGEBIETE VON VERTRAULICHEM CLOUD COMPUTING

In diesem Kapitel sollen die technologischen Grundlagen und Definitionsabgrenzungen von Cloud Computing zusammenfassend erläutert werden. Es gilt weiters die Einschränkungen im Bereich der Verarbeitung von sensiblen Daten aufzuzeigen und diese aufgrund von legislativen, wirtschaftlichen und gesellschaftlichen Voraussetzungen entstehenden Nutzungszenarien exemplarisch vorzustellen. Im nächsten Kapitel werden die unterschiedlichen technologischen Möglichkeiten zur Umsetzung eines vertrauenswürdigenden Cloud Computing erklärt.

3.1 Cloud Computing

Das Konzept von Cloud Computing beschreibt eine Gruppe von Dienstleistungen, welche in unterschiedlichen Ausprägungen angeboten werden können, meist in Verbindung durch eine Sammlung von verschiedenen Technologien. Es gibt unterschiedliche Formen sogenannter „Clouds“ in Bezug auf ihre Bereitstellung. Öffentliche (engl. public) Clouds werden von externen Anbietern bereitgestellt und durch unterschiedliche Kunden genutzt. Private (engl. private) Clouds werden intern von der eigenen Organisation verwaltet und auch nur durch diese genutzt. Gemeinschaftliche (engl. community) Clouds sind grundsätzlich öffentliche Clouds, welche aber nur durch eine geschlossene Gemeinschaft genutzt werden. Eine Vereinigung von mehreren eigenständigen Clouds zur besseren Portabilität der Daten wird auch als hybride Cloud bezeichnet. (Barton, 2014; Bowen, 2011; Rountree & Castrillo, 2014)

Auch die Integrationsebene des Anbieters hat einen Einfluss auf die Definition von Cloud Computing. Man unterscheidet grundsätzlich in Infrastruktur als Dienstleistung (engl. Infrastructure as a Service - IaaS), Plattformlösungen als Dienstleistung (engl. Platform as a Service - PaaS) und Software als Dienstleistung (engl. Software as a Service - SaaS). Innerhalb dieser drei Kategorien verschiebt sich die Intensität der Bereitstellung von physischer oder virtueller Infrastruktur, Betriebssystemen, Datenbanken und Entwicklungsplattformen sowie von Anwendungssoftware zu bestimmten Einsatzgebieten. (Barton, 2014)

3.1.1 Eigenschaften, Vorteile und Hemmnisse von Cloud Computing

Eigenschaften von Cloud Computing aus Sicht der nutzenden Organisation, welche den Unterschied zu anderen Paradigmen von Computersystemen ausmachen, sind: (Furht & Escalante, 2010)

1. Skalierbarkeit und Dienstleistungen auf Abruf
2. Nutzerorientierte Schnittstellen (z.B. Web Services über REST)
3. Garantierte Bereitstellung (z.B. in Bezug auf Leistung und Speicherkapazität)
4. Autonomes System
5. Variable Bezahlmodalitäten

Diese Eigenschaften spielen eine große Rolle in den unterschiedlichen Motivationen zur Nutzung von Cloud Computing oder Cloud-Diensten und bestimmen somit die unterschiedlichen Nutzungsszenarien von Cloud Computing. Vor allem die Agilität und Skalierbarkeit in Bezug auf eine variable Nutzungsintensität, ist für viele Organisationen von großem Vorteil. Die variable Kostenstruktur ermöglicht auch kleineren Organisationen die schnelle Adaption von Infrastruktur und somit auch die Nutzung von kurzfristigen Potentialen aus höherer Rechenleistung und Speicher. (Rountree & Castrillo, 2014)

Vorteile für die Anbieter von Cloud Diensten entstehen vor allem durch das Konzept der Skaleneffekte. Diese werden durch die sinkenden marginalen Kosten für eine Vergrößerung der Infrastruktur, zum Beispiel im Bereich von Speicherkapazitäten oder Rechenleistung, erreicht. Außerdem sind hierbei auch die Verwaltungs- und Wartungskosten zu beachten, welche meist einen hohen Fixkostenanteil beinhalten. (Rountree & Castrillo, 2014)

Organisationen meiden aber aufgrund von unterschiedlichen Hemmnissen noch die Nutzung von Cloud Computing im Einzelnen oder im generellen Ausmaß. Die Abhängigkeit vom Anbieter kann zwar durch sogenannte Service-Level-Agreements (SLA) kontrolliert werden, aber es bleibt ein Risiko der Nichteinhaltung von Vertragsbestandteilen durch den Anbieter bestehen. Im Übrigen ist auch die fehlende Transparenz über die Veränderung der zukünftigen Kosten, aufgrund von bisherigen kurzfristigen Anpassungen, ein Grund nicht auf Dienste von Dritten im Sinne einer öffentlichen Cloud zurückzugreifen. Außerdem spielen Datenschutz, Datensicherheit und Compliance eine bedeutende Rolle in den Bedenken von Verantwortlichen. Organisationen mit sensiblen Daten haben eine größere Hemmschwelle Cloud-Dienste zu nutzen. Diese Thematik beschreibt das folgende Unterkapitel näher. (Barton, 2014)

3.1.2 Informationssicherheit und Datenschutz im Kontext von Cloud Computing

Im Besonderen im Bereich der Nutzung von öffentlichen Cloud-Diensten sind die legislativen, wirtschaftlichen und politischen Rahmenbedingungen im Kontext von Sicherheit, Datenschutz und Compliance von großer Bedeutung. Da dies vor allem die Anwendung von Daten in öffentlichen und gemeinschaftlich genutzten Cloud-Umgebungen betrifft, wird in diesem Unterkapitel näher gebracht, wo die Unternehmerlandschaft Bedenken bei der Nutzung von Cloud-Umgebungen sieht und wie diese zurzeit genutzt werden. (Bedner, 2013; Ghorbel et al., 2017)

Im Jahr 2017 hat eine Studie – durchgeführt von Bitkom – beispielsweise ergeben, dass Konformität mit der Datenschutzgesetzgebung sowie eine transparente Sicherheitsarchitektur die bedeutendsten Kriterien im Rahmen der Adoption von Cloud-Diensten darstellen. (Bitkom & KPMG, 2018a)

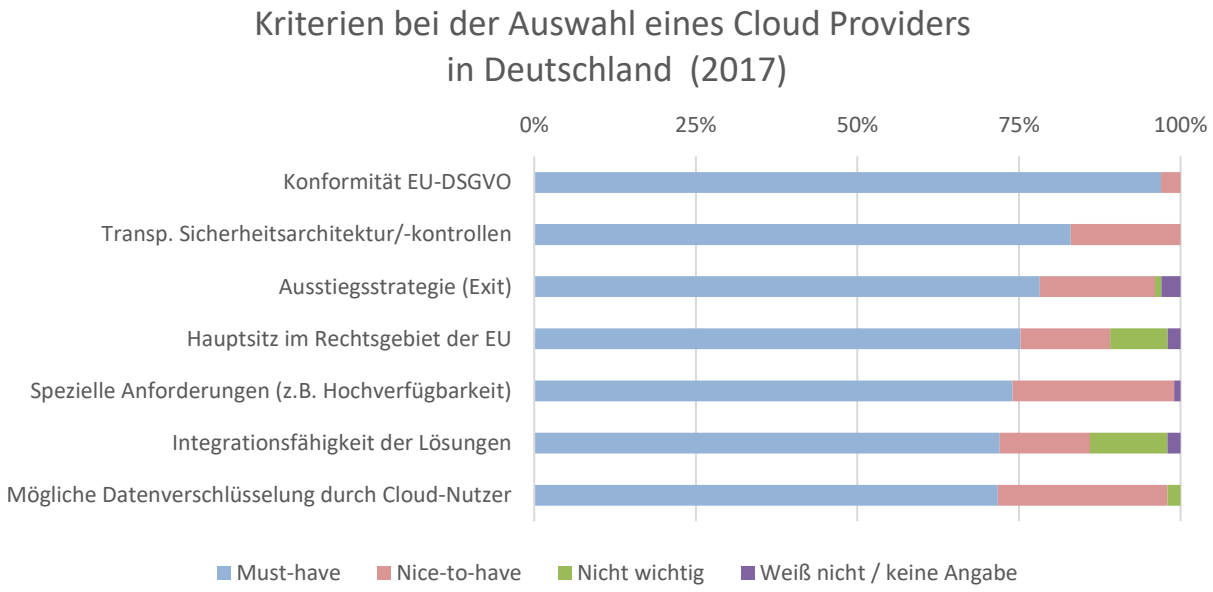

Abbildung 1: Kriterien bei der Auswahl eines Cloud-Providers.
In Anlehnung an: (Bitkom & KPMG, 2018a)

Dies lässt Rückschlüsse zu, dass Unternehmen vorsichtig beim Umgang mit dem Verlagern von sensiblen Daten in die Cloud sind. Dieselbe Studie bestätigt dieses Bild aber nur in Teilen. Mit der Frage nach der Art der Daten, welche in öffentlichen Cloud-Diensten gespeichert werden, lässt sich dieses Problemfeld gut erkennen. Ganze 30 Prozent der Unternehmen, welche bei der Studie in Deutschland im Jahr 2017 teilgenommen haben, geben an, auch kritische Businessinformationen in öffentlichen Cloud-Diensten speichern zu lassen. 38 Prozent dieser bejahen jedoch weiters die Frage in Bezug auf Kundendaten oder andere personenbezogene Daten. (Bitkom & KPMG, 2018b)

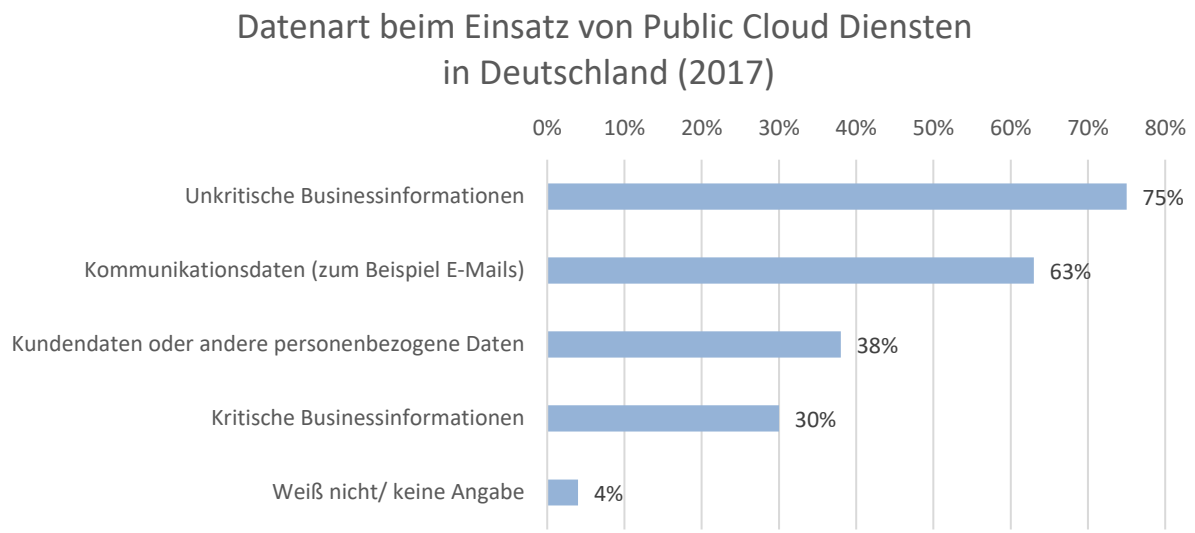

Abbildung 2: Datenart beim Einsatz von Public Cloud-Diensten
In Anlehnung an: (Bitkom & KPMG, 2018b)

Dabei muss man vor allem zwischen unterschiedlichen Definitionen von Sicherheit sowie Datenschutz unterscheiden, denn ein sicheres System muss keinesfalls ein datenschutzkonformes System darstellen und umgekehrt. (Hansen, 2012)

Besonders bei der Verarbeitung von personenbezogenen Daten hat die Legislative der Europäischen Union mit der Datenschutzgrundverordnung (DSGVO) eine gesetzliche Basis geschaffen, welche bei Verstoß auch mit hohen Sanktionen durchgesetzt werden kann. (Feiler & Horn, 2018) Auch die Vereinigten Staaten von Amerika haben für unterschiedliche Branchen Gesetze zum Schutz von personenbezogenen Daten erlassen. Und vor allem die oft fragliche Zuordnung der anzuwendenden Gesetzeslage und des Gerichtsstands im Hinblick auf global agierende Anbieter von Cloud Computing Dienstleistungen, wird von Rechtswissenschaftlern als ein großes Problem der Zukunft in Hinblick auf den Einsatz von Cloud Computing gesehen. (Bowen, 2011)

Neben personenbezogenen Daten haben auch vertrauliche Daten, aufgrund einer hohen wirtschaftlichen Relevanz für ein Unternehmen oder ihrer machtpolitischer Bedeutung einen besonderen Stellenwert bei der Datenanalyse durch Dritte. (Bedner, 2013; Bowen, 2011)

Die besondere Bedeutung von Datenschutz und Datensicherheit in der Wirtschaft wird im folgenden Kapitel zur Evaluierung praktischer Anwendungsszenarien verstärkt berücksichtigt.

3.2 Potentielle Anwendungsgebiete

Aufgrund der unterschiedlichen Vorrausetzungen bei der Integration von Cloud-Diensten innerhalb einer Organisation, gilt es immer klar abzugrenzen, welche Art von Daten verarbeitet wird und in welcher Form des Cloud Computing dies geschieht. (Bedner, 2013) Zur Veranschaulichung der bisherigen theoretischen Definitionsbereiche dient die folgende Tabelle. Die grundsätzlich für den Entwurf von Experimentierszenarien relevanten Kategorien innerhalb dieser Arbeit wurden farblich hinterlegt.

Daten	Legislativ sensibel	Wirtschaftlich sensibel	Nicht sensibel
Bereitstellung der Cloud	Öffentlich	Privat	Gemeinschaftlich	Hybrid
Angebot der Dienstleistung	SaaS	PaaS	IaaS

Tabelle 1: Darstellung der theoretischen Definitionsbereiche von Cloud Computing
In Anlehnung an: (Bowen, 2011)

Innerhalb dieser Forschungsarbeit sollen Anwendungsgebiete mit den folgenden Eigenschaften gefunden werden:

1. Eine Organisation besitzt sensible Daten in großer Menge.
2. Diese Daten sollen von einer dritten Partei in einer öffentlichen oder gemeinschaftlich genutzten Cloud Computing Umgebung statistisch analysiert oder auf sonstige Wege mathematisch verarbeitet werden. Hybride Lösungen können möglich sein, sollen aber aufgrund der individuellen Komplexität nicht weiter berücksichtigt werden.
3. Die Vertraulichkeit der Daten darf nicht durch den Vorgang der Berechnung verletzt werden.
4. Die dritte Partei soll ehrlich, aber neugierig im kryptographischen Sinne (engl. honest, but curious) sein. Das bedeutet sie handelt nicht bösartig und hält sich an das vereinbarte Protokoll. (Brenner, 2012)

3.2.1 Branchenintensität der Nutzung von Cloud Computing

Dass Cloud Computing in der Wirtschaft eine immer stärkere Rolle spielt, wird von vielen Statistiken belegt. Da die vorhin definierten Unterschiede je nach Branche zu einer heterogenen Intensität der Nutzung führen, sollen in dieser Forschungsarbeit beispielhafte Szenarien mit besonderem Fokus auf Branchen mit hoher Nutzung von Cloud Computing erstellt werden. Nach einer Befragung von Führungspersonen in deutschen Unternehmen im Jahr 2016, ist die Nutzungsintensität besonders bei Unternehmen der Energie, Chemie- und Pharmaindustrie, der IT und Telekommunikation und der Automobilbauindustrie hoch. (Bitkom & KPMG, 2016)

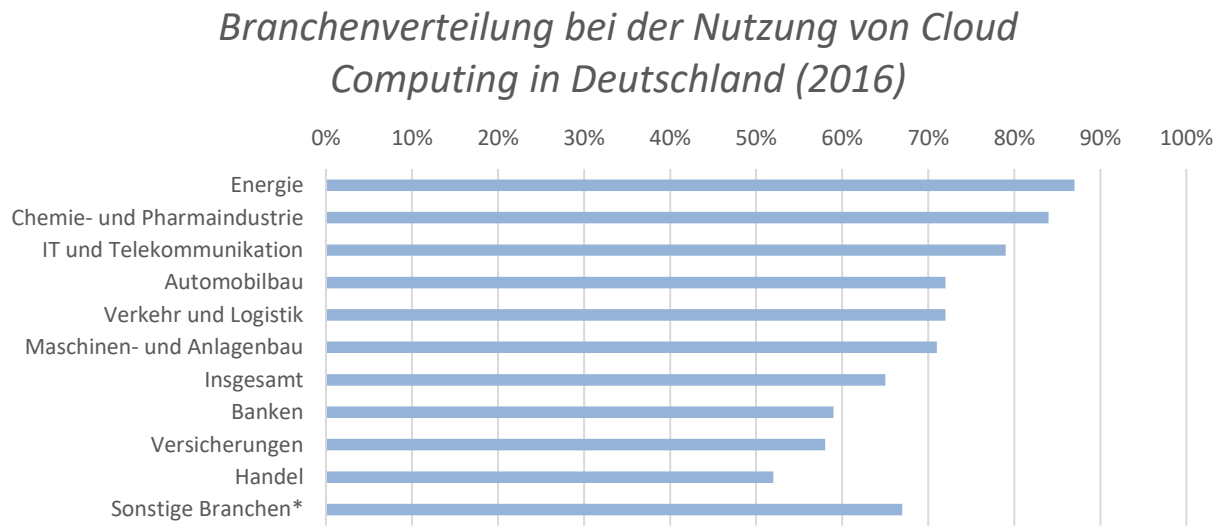

Abbildung 3: Branchenverteilung bei der Nutzung von Cloud Computing in Deutschland 2016
In Anlehnung an: (Bitkom & KPMG, 2016)

3.2.2 Cloud Computing in der Energiebranche

Die Energiebranche ist schon seit jeher an einer umfassenden Vernetzung ihrer Infrastruktur sowie der Nutzung der daraus entstehenden Daten interessiert. Doch vor allem die technologischen Entwicklungen der letzten Zeit haben die Unternehmen der besagten Branche stärker an ihrem Geschäftsmodell arbeiten lassen. (Doleski, 2017)

„(…) Daten avancierten zum Treibstoff energiewirtschaftlicher Geschäftsmodelle."

(Doleski, 2017, S. 18)

Es gilt den Herausforderungen der Zukunft – auch durch die zunehmende Bedeutung von datenbasierten Lösungskonzepten – aktiv zu begegnen. Besonders durch die zunehmende Bedeutung von intelligenten Heimautomatisierungsgeräten (engl. Smart Home) in Kombination mit der Einführung digital vernetzter Stromzähler (engl. Smart Meter), werden Cloud-Dienste im Bereich der Energiewirtschaft eine immer größere Rolle spielen. Die Frage, wie diese Cloud-

Umgebungen aufgebaut sind und sein werden, stellt für die Führungskräfte der Unternehmen eine Problematik mit hoher Relevanz dar. (Doleski, 2017)

Neben der Analyse von Verbrauchsdaten von Energieabnehmern gibt es auch andere Betätigungsfelder von neuen, digitalen Technologien innerhalb dieser Branche. Dazu zählen unter anderem Anwendungen in Zusammenhang mit intelligenten Automatisierungssystemen in privaten Haushalten (engl. Smart Home). (Doleski, 2017)

Die rein deskriptiven Ablesedaten, welche zuvor manuell gesammelt wurden, ermöglichen aufgrund ihrer Granularität und ihrer Verfügbarkeit in Echtzeit auch neue Anwendungsgebiete im Bereich von prognostischer Versorgungsoptimierung, sowie von präskriptiven Anwendungen zur rechtzeitigen Information von Endverbrauchern. (Kaiser, 2017)

Ein intelligentes Stromnetz (engl. Smart Grid) kann wesentlich bei der Optimierung der Erzeugung und des Verbrauchs von Energie helfen. Dabei kommt besonders den wechselseitigen Kommunikationsbeziehungen eine hohe Bedeutung zu. Diese Kommunikation und die daraus resultierende Anreicherung von Daten unterliegen besonderen rechtlichen Beschränkungen. (Spiecker, 2017)

Um die Effizienz der Energiewirtschaft aus ökonomischer und ökologischer Sicht zu verbessern, gibt es nationale Maßnahmen zur verpflichtenden Einführung von intelligenten Stromzählern. Andererseits werden aber Richtlinien und Verordnungen im Bereich des Datenschutzes ausgeweitet. Die Frage, ob die gesammelten Daten personenbezogen sind, hat essentielle Bedeutung im Rahmen der datenschutzrechtlichen Betrachtung. (Spiecker, 2017)

Im Bereich der Messdaten wird von Experten ein Personenbezug gesehen. Besonders interessant ist die datenschutzrechtliche Betrachtung von Messdaten, da hierbei die herkömmlichen Strategien zur Vermeidung der Identifizierbarkeit, wie eine Anonymisierung nur begrenzt wirken. Durch die Analyse der Messdaten könnten sich Nutzungsprofile ergeben, welche eine individuelle Identifizierung zulassen würden. (Spiecker, 2017)

Diese technologischen, wirtschaftlichen und rechtlichen Rahmenbedingungen führen zu einer Vielzahl an möglichen Einsatzszenarien von Methoden des vertraulichem Cloud Computing.

3.2.3 Cloud Computing in der Pharmaindustrie

Die Pharmaindustrie ist eng mit dem Gesundheitssektor verknüpft. Innerhalb dieses Sektors bieten die Vorteile des Einsatzes von Cloud Computing eine Bandbreite von Möglichkeiten zur Kostenreduzierung oder auch zur Qualitätssicherung. (Kuo, 2011)

In der Erforschung von Medikamenten werden oft aufgrund von aufwendigen Analysen Berechnungen mit hoher Komplexität durchgeführt. Diese Operationen müssen oft in ähnlicher Form wiederholt an unterschiedlichen Datensätzen aus Patientendatenbanken durchgeführt werden. (Nigam & Bhatia, 2016)

Besonders die Skalierbarkeit der Ressourcen spielt in dieser Branche in Zusammenhang mit der Einführung von Cloud Computing in Kontext der technologischen Aspekte eine bedeutende Rolle.

Die Herausforderungen hierbei erstrecken sich von Abhängigkeiten gegenüber dem Anbieter bis hin zu datenschutzrechtlichen Problemstellungen. (Ghorbel et al., 2017; Kuo, 2011)

Vor allem kleinere Spitäler, medizinische Praxen und Labore haben keine eigene interne IT-Abteilung. Dabei stellt die Beschaffung von Infrastruktur oftmals eine hohe finanzielle Investition für die einzelne Organisation dar. Außerdem entstehen Risiken, da kein Personal mit ausreichenden Fachkenntnissen vorhanden ist. (Kuo, 2011)

Im Übrigen wird der Einsatz von Cloud Computing-Infrastruktur von manchen Organisationen auch als umweltfreundlichere Alternative gesehen. Dadurch dass die Rechenlast besser verteilt und somit die Ressourcen effizienter genutzt werden, soll die Verwendung von Cloud Computing Systemen zu einer Ressourcenschonung führen. (Kuo, 2011)

Die besondere Problematik zeigt sich in der Sensibilität der Daten. Gesundheitsdaten, dazu zählen etwa Krankheitsinformationen, Diagnosen, medizinisches Bildmaterial, Verschreibungen oder medizinische Aufzeichnungen gewisser Werte unterliegen meist strengen Datenschutzvorschriften. (Ghorbel et al., 2017; Griebel et al., 2015) Die Bedenken dabei betreffen auch die besonders interessanten Themengebiete wie ressourcenintensive medizinische bildgebende Diagnostik. (Griebel et al., 2015)

Die Durchführung von Szenarien, welche die Verarbeitung von Gesundheitsdaten beinhaltet, ist unter anderem in der Wissenschaft von Interesse, da die Datenanalyse mithilfe vertraulicher Berechnungen eine Möglichkeit bietet, um Gesundheitsdaten zu analysieren, welche bisher nicht in diesem Umfang verfügbar waren. (Carpov, Nguyen, Sirdey, Constantino, & Martinelli, 2016)

Ein mögliches Szenario ist zum Beispiel die Sammlung von Gesundheitsdaten über tragbare oder mobile Geräte einer Patientin oder eines Patienten, welche die Daten lokal verschlüsseln. Diese dann in die Cloud-Umgebung hochladen, damit dort ein gewisser Risikofaktor errechnet werden kann. Das Ergebnis wird verschlüsselt wieder retourniert, ohne dass die rechnende Partei jemals die Daten tatsächlich gesehen hat. (Carpov et al., 2016)

Weitere mögliche Anwendungsgebiete wurden schon in wissenschaftlichen Arbeiten näher betrachtet. Hierzu werden im Kapitel 4.3.2 Beispiele für die Anwendung auf genetische Erbinformationen erläutert.

Doch auch das politische und gesellschaftliche Interesse kann ein hohes Ausmaß besitzen. Eine elektronische Patientenakte muss durch diverse Technologien eine sichere Speicherung, Übertragung und auch Verarbeitung von sensiblen Gesundheitsdaten bewerkstelligen. (Sadeghi & Schneider, 2010)

Zudem haben auch Unternehmen, welche ursprünglich aus anderen Bereichen kommen, ein Interesse an der Bereitstellung diverser Dienste. Große Technologieunternehmen wie Google oder Microsoft haben Lösungen für die Informationsverwaltung im Gesundheitsbereich entwickelt. (Sadeghi & Schneider, 2010)

Auch in der chemischen Produktion sind oft viele Produktionsanlagen miteinander verbunden. Zur Steuerung dieser werden vernetzte Systeme eingesetzt, welche durch eine Segmentierung der Netzwerke innerhalb eines Produktionsbereiches abgesichert werden. Um weitere

Möglichkeiten der Effizienzsteigerung zu erreichen, sollen diese abgeschirmten Bereiche entlang der kompletten Wertschöpfungskette, einschließlich der Systeme von Lieferanten und Kunden, kurz- und langfristig miteinander vernetzt werden. (Bundesministerium für Wirtschaft und Energie, 2016)

3.2.4 Cloud Computing in der Automobilindustrie

Die Automobilindustrie steht vor mehreren großen Umbrüchen. Neben den Veränderungen in der Nachfrage nach unterschiedlichen Antriebstechnologien, wie zum Beispiel dem Elektroantrieb, wird vor allem durch die Digitalisierung geprägte Technologien eine Reihe von disruptiven Entwicklungen voranstehen. (Winkelhake, 2017)

Fortschritte im Bereich des autonomen Fahrens und auch die Optimierung der Wertschöpfungskette, vom Fahrzeughersteller hin zum Mobilitätsdienstleister, lösen die herkömmliche Betrachtung von digitalen Daten und deren Verarbeitung in Automobilkonzernen durch neue Technologien ab. Im Besonderen stehen die Prozessoptimierungen im Zusammenhang mit dem Trend „Industrie 4.0“ und den notwendigen modernen IT-Strukturen in unterschiedlichen Formen von Cloud-Architekturen im Fokus der Branche. (Winkelhake, 2017)

Die IT-Sicherheit sowie die Vorgehensweise mit personenbezogenen Daten werden unter anderem in Deutschland sehr sensitiv behandelt. Diese stellen auch oft Hemmnisse für Digitalisierungsprojekte dar. Dabei sollen in der Industrie verbreitete Regelungen, wie die ISO 2700x-Normenreihe, eine Basis für einen sicheren Umgang mit vertraulichen Daten darstellen. Außerdem existieren Spezialnormen wie die DIN EN 50600, welche einen im Kontext der Datensicherheit korrekten Aufbau der Infrastruktur von Rechenzentren standardisieren. (Winkelhake, 2017)

Im Rahmen der Verarbeitung von personenbezogenen Daten ist vor allem bei stärkerer Zusammenarbeit mit anderen Unternehmen der Austausch dieser Daten genauer zu betrachten. Grenzüberschreitende Wertschöpfungsketten stellen die Industrie vor komplexe juristische Fragen. (Winkelhake, 2017)

Im Kontext der Auswertung von großen Datenmengen stellen unter anderem folgenden Bereiche interessante Anwendungsmöglichkeiten dar: (Winkelhake, 2017)

- Gewährleistungsmuster erkennen
- Kundeninteressen segmentieren
- Erkenntnisse aus Service- und Wartungsarbeiten zur Qualitätsverbesserung

In einer Studie des deutschen Bundesministeriums für Wirtschaft und Energie werden im Rahmen von Fallbeispielen mögliche IT-Sicherheitsproblemstellungen im Kontext von „Industrie 4.0“ aufgezeigt. Eines dieser Fallbeispiele behandelt die starke Automatisierung des Produktionsprozesses eines Automobilherstellers. Dabei wird die Vernetzung von unterschiedlichen Standorten eines Unternehmens und einer Ausnahmensituation zur

Anpassung der Produktionsstände näher erläutert. (Bundesministerium für Wirtschaft und Energie, 2016)

Das deutsche Bundesministerium für Wirtschaft und Energie beschreibt auch näher mögliche Risiken, welche durch eine fehlerhafte Ausführung solcher Szenarien entstehen können: (Bundesministerium für Wirtschaft und Energie, 2016)

- Erkenntnisse über die Produktionsanlagen für Fremde
- Manipulation der Funktionsweise von verketteten Anlagen
- Einschränkung der Betriebssicherheit der Anlage

Doch unabhängig von den Risiken, welche durch den verstärkten Einsatz der Datenverarbeitung in Cloud Computing Systemen entstehen, werden allgemein ausgehend vom Wandel des Geschäftsmodells der Automobilindustrie, hin zu einem dienstleistungsorientierten Mobilitätsangebot, vernetzte Verkehrssysteme zum Ausgleich von Mobilitätsangebot und -nachfrage im Zusammenhang mit Elementen des Cloud Computing eine wichtige Rolle spielen. (Flügge, 2016)

Die Automobilbranche muss sich aufgrund der gesellschaftlichen Entwicklungen an eine Vielzahl von Einschränkungen halten, welche aufgrund von Luftverschmutzungsvermeidung oder Klimazielen aufgestellt wurden. Diese vormals am Prüfstand unter Laborbedingungen geprüften Emissionswerte, werden nun in unterschiedlicher Intensität mittels portablen Messeinheiten von Kontrollorganisationen verlangt. Dies soll die reale Emittierung dieser Luftschadstoffe besser darstellen und deren Auswirkungen auf die Umwelt leichter abschätzbar machen. (Winkelhake, 2017)

Einige Ansätze zur Emissionskontrolle nutzen schon Elemente von Cloud Computing Systemen. Eine Vernetzung der Daten von mehreren Fahrzeugen könnte einen großen Vorteil für die wirtschaftlichen Ziele der Unternehmen und die ökologischen Ziele der Gesellschaft darstellen. (Flügge, 2016; Ning, Wubulihairen, & Yang, 2012)

4. METHODEN VON VERTRAULICHEM CLOUD COMPUTING

Bei vertraulichem Cloud Computing ist grundsätzlich zwischen drei Phasen, bei denen das Schützen der Vertraulichkeit von Relevanz ist, zu unterscheiden: (Ghorbel et al., 2017)

1. Daten im Ruhezustand (engl. in rest - z.B. persistiert auf einer Festplatte)
2. Daten beim Transport (engl. in transit – z.B. bei einem FTP-Upload)
3. Daten während ihrer Nutzung (eng. in use – z.B. bei der Berechnung einer Statistik)

Die Sicherung der Vertraulichkeit innerhalb der ersten zwei Phasen kann schon sehr gut mittels heutiger Technologie gewährleistet werden. Methoden für die vertrauliche Berechnung werden – angepasst an ihr Nutzungsszenario – in abgeschwächter Form eingesetzt. Eine baldige Lösung zur vollständigen und kompromisslosen Erfüllung der Wahrung der Vertraulichkeit während der Berechnung wird von vielen Experten bezweifelt. Doch vor allem die steigende Bedeutung der statistischen Datenanalyse von großen Datenmengen lässt neben der Wissenschaft auch die Wirtschaft wieder verstärkt in die Weiterentwicklung derartiger Verfahren investieren. (Chatterjee & Sengupta, 2015; Dowlin et al., 2017)

4.1 Grundlagen zur homomorphen Verschlüsselung

Unter homomorpher Verschlüsselung versteht man eine Verschlüsselung, welche eine Berechnung mit verschlüsselten Daten zulässt. Das Ergebnis der Berechnung der unverschlüsselten Daten sowie das entschlüsselte Ergebnis der Berechnung der verschlüsselten Daten sind ident. Die in diesem Kapitel vorgestellten, unterschiedlichen homomorphen Schemata entsprechen keiner vollständigen Aufzählung, sondern sollen nur dem besserem Verständnis der Problemstellungen von homomorphen Verschlüsselungsmethoden dienen. (Brenner, 2012; Sadeghi & Schneider, 2010)

Man unterscheidet zwischen den folgenden Untergruppen der homomorphen Verschlüsselung: (Yasuda, Shimoyama, & Kogure, 2014)

1. Partielle homomorphe Verschlüsselung (engl. partial homomorphic encryption - PHE)
2. Annähernd homomorphe Verschlüsselung (engl. somewhat homomorphic encryption - SHE)
3. Vollständige homomorphe Verschlüsselung (engl. fully homomorphic encryption - FHE)

Der Unterschied zwischen der annähernd homomorphen Verschlüsselung und der vollständigen homomorphen Verschlüsselung liegt in der Restriktion auf die Anwendung von Polynomen niedriger Ordnung. Mittels nivellierter homomorpher Verschlüsselung (engl. leveled) kann dabei die Performance und Speichernutzung von Berechnungen optimiert werden. Dabei beschränkt

man sich auf eine zuvor schon bekannte Anzahl und Art von Berechnungen, welche durchgeführt werden sollen. Dies ermöglicht eine effizientere Auswahl der Verschlüsselungsparameter. Diese wird in der Literatur auch als praktische homomorphe Verschlüsselung (engl. practical homomorphic encryption) bezeichnet. (Dowlin et al., 2017; Dyer, Dyer, & Xu, 2017)

4.1.1 Partielle homomorphe Verschlüsselung

Es gibt zahlreiche unterschiedliche homomorphe Schemata. Viele davon weisen aber eine gewisse Begrenzung auf. Diese ist zum Beispiel die Restriktion, dass nur ein mathematischer Operationstyp angewendet werden kann. Die betreffenden Schemata werden der Gruppe der partiellen homomorphen Verschlüsselungen zugeordnet. (Brenner, 2012)

4.1.1.1 Verfahren von Rivest, Shamir und Adleman (RSA)

Seit dem Jahr 1978 ist bekannt, dass man entweder eine Addition oder Multiplikation auf verschlüsselte Daten mit speziellen Verfahren anwenden kann. Eines der bekanntesten Verfahren, welches multiplikative Berechnungen auf den öffentlichen Schlüsseltext zulässt, ist RSA. Vereinfacht dargestellt ist folgende Berechnung mit dem Schlüsseltext von RSA möglich: (Fontaine & Galand, 2007; Hoffstein et al., 2014)

$$c_1 * c_2 \equiv m_1^e * m_2^e \equiv (m_1 * m_2)^e \ (mod\ N)$$

4.1.1.2 Paillier-Kryptosystem

Das kryptographische System von Pascal Paillier, welches 1999 im Rahmen der Eurocrypt vorgestellt wurde, ermöglicht eine homomorphe Addition und Subtraktion von Schlüsseltexten. Außerdem werden auch andere gemischt-homomorphe Rechenoperationen – Berechnungen von Klartext mit Schlüsseltext – unterstützt. (Paillier, 1999)

4.1.2 Vollständige homomorphe Verschlüsselung

Wenn zwei Rechenoperationen möglich sind – im besonderen Fall also Addition und Multiplikation – spricht man von einem algebraischen Homomorphismus. Bei der Anwendung auf boolesche Schaltkreise erweitert sich der Umfang dieser zwei Operationen, da eine Addition modulo 2 (entspricht einem „AND“-Schalter) und eine Mutliplikation modulo 2 (entspricht einem „XOR“-Schalter) jede weitere beliebige Funktion ermöglichen. (Brenner, 2012)

Die hier vorgestellten homomorphen Verschlüsselungsschemata basieren auf der Ringtheorie. Ringe sind hierbei algebraische Strukturen in welchen eine Addition und Multiplikation möglich ist. Die bekannteste Form eines solchen Ringes stellt der Zahlenraum der ganzen Zahlen dar.

Eine Verschlüsselung der Werte eines Ringes entspricht mathematisch der Aussage, dass ein Wert aus dem Ring R auf einen Wert aus dem Ring S abgebildet werden kann. (Hoffstein et al., 2014)

$$\varphi: R \rightarrow S$$

Wenn nun ein weiterer Zahlenraum – wie hier eine verschlüsselte Instanz der Klartexte – die Anforderung eines Ringes erfüllt und ebenso die unten angeführte Anforderung erfüllt wird, dann spricht man von einem Ringhomomorphismus. (Hoffstein et al., 2014)

$$\varphi(x + y) = \varphi(x) + \varphi(y) \; und \; \varphi(x * y) = \varphi(x) * \varphi(y) \; \forall \, x, y \in R$$

Doch es ist wichtig anzumerken, dass auch wenn es grundsätzlich möglich wäre, alle Operationen auf Bit-Ebene durchzuführen, eine Division im ganzzahligen Raum mit den aktuellen Implementierungen nur in sehr stark eingeschränkter Weise möglich ist. Dies stellt also für praktikable Anwendungen eine noch zu überwindende Hürde dar. (Lauter, López-Alt, & Naehrig, 2014)

Da man im Zuge der Recherche nach Implementierungen immer auf die unterschiedlichsten Schemata und deren Kurzbezeichnungen trifft, werden in den folgenden Subkapiteln die wichtigsten vollständig homomorphen Schemata vorgestellt.

4.1.2.1 Gentry-Kryptosystem

Craig Gentry hat erstmals im Jahr 2009 mit seiner Veröffentlichung „A Fully Homomorphic Encryption Scheme“ eine vollständig homomorphe Verschlüsselungsmethode vorgestellt. Gentry nutzt hierfür die Gittertheorie, welche die Darstellung der Schlüsseltexte durch Positionen im geometrischen Raum in Relation zu anderen Positionen beeinhaltet. In der englischen Fachliteratur ist diese Theorie besser unter dem Namen „lattice-based cryptography“ bekannt. (Gentry, 2009; Hoffstein et al., 2014)

Im von Gentry beschriebenen Verfahren werden die Eingangsdaten jeweils per Bit verschlüsselt. Das bedeutet, dass ein eigener Schlüsseltext für jeden Bit in den Daten erstellt werden muss. Diese Variante führt zu einem hohen Leistungsaufwand. (Dowlin et al., 2017)

4.1.2.2 Smart-Vercauteren (SV)

Kurz darauf im Jahre 2010 haben zwei europäische Wissenschaftler den Ansatz von Gentry erweitert. Als wesentliche Verbesserung kann eine kleinere Nachrichtenerweiterung und

Schlüsselgröße im Vergleich zu Gentrys Schema angesehen werden. (Smart & Vercauteren, 2010)

4.1.2.3 Brakerski, Gentry und Vaikuntantathan (BGV)

Basis vieler aktueller Implementierungen ist das Schema von Brakerski, Gentry und Vaikuntantathan, welches 2011 vorgestellt wurde. Es war eines der ersten Schemen, das sich in realen Anwendungen als praktisch erwies. Es führte einige Erweiterungen und Optimierungen zu den bisherigen Schemata hinzu. (Brakerski, Gentry, & Vaikuntanathan, 2011)

4.1.3 Sicherheitskritik

Die homomorphe Veschlüsselung bietet Lösungen für einige, aber nicht alle Problemszenarien in der Realität. Es ist möglich Protokolle mit öffentlichem (engl. public key) und privatem Schlüssel (engl. private key) und mit symmetrischem privaten Schlüssel zu entwerfen. (Rothblum, 2011)

Doch die homomorphe Verschlüsselung hat einige Rahmenbedingungen, welche ihre Sicherheit einschränken. Um unterschiedliche kryptographische Verfahren miteinander zu vergleichen werden folgende Robustheitsgrade untersucht: (Bellare & Rogaway, 2005; Brenner, 2012)

1. Nicht-Unterscheidbarkeit (IND – engl. Indistinguishability)
2. Angriff mit wählbarem Klartext (CPA – engl. Chosen Plaintext Attack)
3. Angriff mit wählbarem Schlüsseltext (CCA1 – engl. Chosen Ciphertext Attack)
4. Angriff mit adaptiv wählbarem Schlüsseltext (CCA2 – engl. Adaptive Chosen Ciphertext Attack)

Es ist naheliegend, dass die Anforderung IND-CCA2 mittels homomorpher Verschlüsselung nicht erreicht werden kann, da eine Multiplikation eines Klartextes mit einem Schlüsseltext einer Konstanten beliebig oft durchgeführt werden kann. Dies führt dazu, dass der potentielle Angreifer an unterscheidbare Schlüsseltexte, welche mit derselben Konstante multipliziert wurden, gelangt. (Brenner, 2012)

In weiterer Folge liefert auch die Parameterwahl bei nivellierender homomorpher Verschlüsselung einen die Datensicherheit beschränkenden Faktor. Hier wird der Ansatz gewählt, dass eine möglichst effiziente Lösung gefunden wird, welche die Geheimhaltung und die fehlerfreie Entschlüsselung noch sicherstellt. (Dowlin et al., 2017)

4.2 Implementierungen von homomorpher Verschlüsselung

In diesem Unterkapitel sollen einige der aktuellen und in der Forschung genutzten, frei verfügbaren Bibliotheken zur Umsetzung von homomorpher Verschlüsselung beschrieben werden. Die ausgewählten Bibliotheken werden aufgrund ihrer Bedeutung in der Wissenschaft näher erläutert.

4.2.1 hcrypt

Das von der Leibniz Universität Hannover geförderte Projekt hcrypt ist ein früher Ansatz zur Implementierung des Smart-Vercauteren Schematas. Zur Zeit der Entwicklung von hcrypt im Jahr 2011 war keine frei verfügbare Implementierung des weiterentwickelten Gentry Schemas vorhanden und die von den Forschern der hannoverischen Universität untersuchten privaten Bibliotheken waren fehlerbehaftet. (Brenner, 2012)

Mit der Veröffentlichung der Bibliothek wollten die Wissenschaftler eine funktionierende Versuchsumgebung für weitere Studien und damit eine größere Zahl von Versuchsexperimenten möglich machen. (Brenner, 2012)

Zur besseren Übersicht der Unterschiede werden folgend die gesammelten Informationen der einzelnen Bibliotheken in tabellarischer Ansicht veranschaulicht. Die letzte Änderung bezieht sich auf die letzte frei veröffentlichte Version der entsprechenden Implementierung.

Name:	**hcrypt**
Repository/Quellcode:	https://github.com/hcrypt-project
Entwickler:	Michael Brenner (Leibniz Universität Hannover)
Lizenz:	Selbst verfasst (siehe Repository-Link)
Verwendete Technologien:	Smart-Vercauteren Schema
Letzte Änderungen:	26.08.2015

Tabelle 2: hcrypt – gesammelte Informationen
In Anlehnung an: (Brenner, 2012).

4.2.2 HElib

HELib ist eine Bibliothek zur Verwendung von vollständiger homomorpher Verschlüsselung (FHE). Diese Bibliothek wird in einer Vielzahl von wissenschaftlichen Versuchsimplementierungen benützt und wurde unter der Führung von Shai Halevi, einem Wissenschaftler des MIT, welcher am Thomas J. Watson Research Center von IBM forscht, entwickelt. (Halevi & Shoup, 2014; Xu, Chen, Wu, & Feng, 2016)

Sie unterstützt in ihrer derzeitigen Form die Addition, Subtraktion, Multiplikation und Skalarprodukte über Schlüsseltexte von ganzzahligen Werten (engl. integer). Außerdem werden boolesche Operationen (u.a. AND, OR, NOT und XOR) auf binäre Schlüsseltexte unterstützt. (Ibarrondo, 2018)

Zurzeit wird das Schema von Brakerski, Gentry und Vaikuntanathan (BGV) verwendet. Zusätzlich wurden einige Verbesserungen in Zusammenhang mit der homomorphen Evaluierung beschleunigt. Dies wurde insbesonders durch den effizienten Einsatz der Smart-Vercauteren Schlüsseltext-Paketierung und den Optimierungen von Gentry, Halevi und Smart erreicht. Die Bibliothek unterstützt auch einen beschleunigten Initialisierungsprozess. (engl. bootstrapping) (Halevi, 2018; Halevi & Shoup, 2014)

In den letzten Änderungen dieses Jahres wurden einige Verbesserungen im Bereich der linearen Transformation implementiert, die eine schnellere Berechnung ermöglichen. (Chirgwin, 2018; Halevi & Shoup, 2018)

Name:	**HElib**
Repository:	https://github.com/shaih/HElib
Entwickler:	Shai Halevi (IBM, MIT)
Lizenz:	Apache 2.0
Verwendete Technologien:	BGV Schema, Smart-Vercauteren Paketierung, Gentry-Halevi-Smart Optimierungen
Letzte Änderungen:	19.06.2018

Tabelle 3: HElib – gesammelte Informationen
In Anlehnung an: (Halevi, 2018).

4.2.3 SEAL - Simple Encrypted Arithmetic Library 2.3.1

Die Simple Enrypted Arithmetic Library („SEAL“) ist eine von der Cryptography Research Group bei Microsoft Research unter der Führung von Kim Laine entwickelte Bibliothek für homomorphe Verschlüsselung. Sie ist in C++ programmiert, besitzt aber auch .NET-Umhüllungen (engl. wrapper). Auch ein Python-Wrapper von Lab41 einem US-amerikanischen Forschungslabor, ist verfügbar. (Dowlin et al., 2017; Lab41, 2017/2018)

Auch diese Bibliothek wird in einigen Forschungsarbeiten benützt. Dazu muss man anmerken, dass die Implementierung unter der Microsoft Research License steht. Dies bedeutet, dass damit Projekte zu wissenschaftlichen, nicht-kommerziellen Zwecken umgesetzt werden dürfen. Im Gegensatz dazu verwenden andere, hier vorgestellte Umsetzungen eine freiere Lizenz. Der Programmcode ist zwar frei zugänglich, wird aber nicht aktiv auf einem öffentlichen Repository weiterentwickelt. (Microsoft Research, 2018)

Da SEAL eine sogenannte Nivellierung unterstützt, ist die Leistung und Speichernutzung sehr stark von der Parameterwahl abhängig. Diese Wahl kann unter stark eingeschränkten Rahmenbedingungen mithilfe eines zusätzlichen Moduls auch automatisiert getroffen werden. Für die praktische Nutzung ist diese automatisierte Parameterwahl, jedoch oft nicht direkt anwendbar. (Dowlin et al., 2017)

Name:	**SEAL** – Simple Encrypted Arithmetic Library
Repository/Quellcode:	https://www.microsoft.com/en-us/download/details.aspx?id=56202 Python-Wrapper: https://github.com/Lab41/PySEAL
Entwickler:	Kim Laine (Microsoft Research)
Lizenz:	Microsoft Research License Agreement
Verwendete Technologien:	Brakerski/Fan-Vercauteren (B/FV) Schema, Smart-Vercauteren Paketierung, Gentry-Halevi-Smart Optimierungen
Letzte Änderungen:	19.06.2018

Tabelle 4: SEAL – gesammelte Informationen
In Anlehnung an: (Microsoft Research, 2018).

4.2.4 FHEW

Die Bibliothek FHEW stellt eine symmetrische Verschlüsselung für einzelne Bit bereit, welche die homomorphe Evaluierung von arbiträren booleschen Verbindungen auf Schlüsseltexte ermöglicht. Entstanden ist das Projekt durch eine Kooperation von Léo Ducas vom Amsterdamer Centrum Wiskunde & Informatica und Daniele Micciancio von der University of California. (Ducas & Micciancio, 2014)

Dabei haben sie sich auf die Optimierung des Reinitialisierungsverfahrens von Schlüsseltexten zur kontinuierlichen Berechnung der selbigen spezialisiert. In der Ergebnisinterpretation orientierten sie sich an der Laufzeit von HElib und konnten diese erheblich unterbieten. (Ducas & Micciancio, 2014)

Name:	**FHEW** – A Fully Homomorphic Encryption Library
Repository/Quellcode:	https://github.com/lducas/FHEW
Entwickler:	Léo Ducas (CWI Amsterdam)
Lizenz:	GNU GPL v2
Verwendete Technologien:	Erweitertes Reinitialisierungsverfahren
Letzte Änderungen:	30.05.2017

Tabelle 5: FHEW – gesammelte Informationen
In Anlehnung an: (Ducas & Micciancio, 2014).

4.2.5 cuHE/cuFHE

Diese Implementierungen haben sich als Ziel gesetzt vor allem im Bereich der Geschwindigkeit Vorteile gegenüber anderen Bibliotheken zu haben. Der Name steht für CUDA (Fully) Homomorphic Encryption Library und bezieht sich auf die Verwendung der Technologie CUDA von Nvidia zur beschleunigten Abarbeitung von Programmteilen durch den Grafikprozessor. (Dai, 2018; Dai & Sunar, 2015)

Die Implementierung cuHE verwendet das Schema von Doröz, Hu und Sunar und wurde im Rahmen der Vernom Group des Worcester Polytechnic Institute entwickelt. Beide Umsetzungen sind dabei unter der Führung von Wei Dai entstanden. Die Bibliothek zur vollständigen homomorphen Verschlüsselung befindet sich noch in einer Beta Phase, soll aber auch erhebliche Geschwindigkeitsvorteile besitzen. (Dai, 2018; Dai & Sunar, 2015; Vernamlab, 2016/2018)

Name:	**cuHE/cuFHE – CUDA (Fully) Homomorphic Encryption Library**
Repository/Quellcode:	https://github.com/vernamlab/cuHE https://github.com/WeiDaiWD/cuFHE
Entwickler:	Vernam Group, Wei Dai
Lizenz:	MIT
Verwendete Technologien:	cuHE – Doröz-Hu-Sunar (DHS) cuFHE
Letzte Änderungen:	cuHE - 08.06.2017 cuFHE - 21.05.2018

Tabelle 6: cuHE/cuFHE – gesammelte Informationen
In Anlehnung an: (Dai, 2018; Vernamlab, 2016/2018).

4.2.6 python-paillier

Diese Bibliothek wurde von Data 61 entwickelt. Data 61 ist Australiens führendes Innovationszentrum für Datenanalyse, welches 2016 aus dem Zusammenschluss der National ICT Australia Ltd (NICTA) und der „Digital Productivity“ Sparte von CSIRO entstanden ist. (CSIRO Data 61, 2018b)

Die Implementierung benützt das Schema von Pascal Paillier, welches in dieser Arbeit schon kurz erläutert wurde. Diese Bibliothek besitzt somit nur eine eingeschränkte universelle Verwendbarkeit und hat durch den Fokus auf die Anwendung von stark vereinfachten Algorithmen für die Datenanalyse ein bestimmtes Ziel. (CSIRO Data 61, 2018a)

Name:	**python-paillier**
Repository:	https://github.com/n1analytics/python-paillier
Entwickler:	CSIRO Data 61
Lizenz:	GNU GPL v3.0
Verwendete Technologien:	Paillier Schema
Letzte Änderungen:	19.06.2018

Tabelle 7: python-paillier – gesammelte Informationen

In Anlehnung an: (CSIRO Data 61, 2018a).

4.2.7 tfhe - Faster Fully Homomorphic Encryption

Dieses quelloffene Projekt wurde im Zusammenhang mit der CRYPTO-COMP Initiative umgesetzt. Der Fokus der Implementierung lag in der Generalisierung von LWE und GSW homomorphen Verschlüsselungsschemas. Es wurde die Ausführungszeit des Aktualisierungsprozess der Schlüsseltexte zur Wahrung der Entschlüsselungsfähigkeit nach mehreren Berechnungen verkürzt.

Weiters wurden ein Beweis der Umsetzung des Konzepts und eine Sicherheitsanalyse durchgeführt. Trotz der hohen Geschwindigkeit der Implementierung des fortschrittlichen Konzepts, ist sie nicht vollständig für reale Anwendungen geeignet. Diese Einschränkung besteht vor allem aufgrund des hohen Expansionsfaktors – dieser liegt bei zirka 400.000 - der verschlüsselten Daten, welche auch unter einer geringen Stapelfähigkeit leiden. (Chillotti, Gama, Georgieva, & Izabachène, 2016, 2017)

Name:	**tfhe-Faster Fully Homomorphic Encryption**
Repository:	https://github.com/tfhe/tfhe
Lizenz:	Apache 2.0
Verwendete Schemen:	LWE, GSW
Letzte Änderungen:	19.06.2018

Tabelle 8: tfhe – gesammelte Informationen
In Anlehnung an: (Chillotti et al., 2016, 2017).

4.3 Forschungsprojekte

In diesem Unterkapitel sollen beispielhaft bereits durchgeführte Experimente oder Forschungsprojekte, welche homomorphe Verschlüsselungstechnologien verwendet haben, näher beleuchtet werden. Dies soll einerseits die derzeitigen Möglichkeiten und andererseits durch die aufgetretenen Problemstellungen die Grenzen der Anwendung aufzeigen.

Eine geeignete Testumgebung für die Prüfung der Funktion und Korrektheit von homomorphen Verschlüsselungsbibliotheken wurde von Wissenschaftlern des MIT zur Verfügung gestellt. Mit „HEtest“ lassen sich beispielsweise Kennzahlen zur Schlüsselgenerierung berechnen, welche zum Vergleich dienen können. (Varia, Yakoubov, & Yang, 2015)

4.3.1 Helios Voting

Eine schon früh in theoretischen Forschungsarbeiten vorgestellte Anwendung von homomorpher Verschlüsselungstechnik, ist die Abhaltung von sicheren Wahlen. Dafür wurden bis 2008 schon zahlreiche Protokolle, aber nur eine geringe Anzahl von Implementierungen, vorgestellt. Ben Adida von der Harvard University hat 2008 nun diesen weit verbreiteten Anwendungsfall im Zuge eines Forschungsprojektes ausgearbeitet. Er hat seine Implementierung namens Helios im Rahmen des USENIX Security Symposium vorgestellt. (Adida, 2008)

Helios ist ein webbasiertes Anwendungssystem zur Abhaltung von Wahlen mit einem sogenannten offenen Audit. Dies bedeutet, dass jede abgegebene Stimme nachvollziehbar in die Auszählung der Stimmen mit einbezogen wurde. Der Nachweis geschieht über einen sogenannten Stimmzettel-Tracker (engl. ballot tracker), welcher der Wählerin oder dem Wähler nach Abgabe seiner Stimme gezeigt wird. Gleichzeitig ist es auch möglich, seine Stimme, beziehungsweise die homomorphe Verschlüsselung seiner Stimme, zu überprüfen. Dies passiert über einen eigenen Dienst zur Verifizierung für einzelne Stimmzettel. Dieser überprüft die Signatur der Wahl, in Zusammenhang mit dem individuellen Stimmzettel-Tracker, auf die korrekte Verschlüsselung der Stimme. In der digitalen Beilage dieser Masterarbeit ist eine beispielhafte Auditinformation im JSON-Datenformat angefügt. Die Auditinformation wurde für die im Rahmen dieser Masterarbeit zu Testzwecken erstellte und bis Ende 2018 frei verfügbare Abstimmung unter „https://vote.heliosvoting.org/helios/e/TimeEurope“ erzeugt. (Adida, 2008)

Das Projekt soll für Wahlen von kleinen Organisationen, Gemeinschaften im Internet und für Universitäten dienen. Helios Voting kann einerseits selbst in einer eigenen Serverumgebung für andere Clients zur Verfügung gestellt werden oder man nutzt die bereits zur Verfügung gestellte Plattform der Organisation Helios unter der Führung von Ben Adida. (Adida, 2018; Helios Voting, 2018)

Helios basiert dabei auf Benaloh's Simple Verifiable Voting protocol – einer im Grunde partiell homomorphen Verschlüsselungstechnologie. Auch am Veröffentlichungsdatum von Helios ist dies zu erkennen, da das vollständig homomorphe Schema von Gentry erst 2009 vorgestellt wurde und zu dieser Zeit auch nur sehr eingeschränkt praxisfähig war. Gleichzeitig benötigt man für eine simple Wahl grundsätzlich nur die Addition als Rechenoperation. Die Beschränkung auf ein partiell homomorphes Verschlüsselungssystem erlaubt auch eine bessere Leistung und geringeren Speicherverbrauch. (Adida, 2008, 2018)

4.3.2 Analyse von genetischen Daten mittels homomorpher Verschlüsselung

In diesem Bereich sind in den letzten fünf Jahren einige interessante Forschungsarbeiten publiziert worden. Die Motivation hinter diesen durchgeführten Versuchsimplementierungen bestand darin, dass eine Vielzahl an Datenbanken mit genetischen Informationen auf der Welt einen hohen Wert für die medizinische Wissenschaft hat. Diese Daten werden meist freiwillig von Patientinnen und Patienten für Forschungszwecke zur Verfügung gestellt und auch wenn diese anonymisiert abgespeichert sind, können sie in Verbindung mit anderen gesundheitlich relevanten Daten eine Gefahr für die Privatsphäre der Patientinnen und Patienten darstellen. Um diese Privatsphäre sicherzustellen, sollen homomorphe Verschlüsselungstechnologien verwendet werden. (Çetin et al., 2017; Kim & Lauter, 2015; Lauter et al., 2014; Wang et al., 2016)

In einer Forschungsarbeit, welche im Rahmen der iDASH Secure Genome Analysis Competition 2016 in Zusammenarbeit mit der Cryptography Research Group von Microsoft Research entstanden ist, wurde die Umsetzbarkeit von Privatsphäre erhaltenden Methoden zur Abfrage genetischer Daten demonstriert. (Çetin et al., 2017)

Die homomorph verschlüsselten Informationen über Genome sollen also sicher in der Cloud gelagert werden und ebenfalls sicher vom Eigentümer der Daten nach bestimmten Eigenschaften abgefragt werden können, ohne Erkenntnisse für den Betreiber der Cloud über die Daten entstehen zu lassen. Dafür wurde die zuvor schon erläuterte Bibliothek SEAL in der Version 2.1 verwendet. (Çetin et al., 2017)

In den Szenarien der Wissenschaftler wird angenommen, dass die Daten in einer nicht vertrauenswürdigen Cloud gespeichert und verarbeitet werden. Die Nutzer und Eigentümer der Daten sind vertrauenswürdig und besitzen denselben Schlüssel. Als realistisches Beispiel wird eine Allianz von staatlichen Organisationen, Forschungsinstituten oder Krankenhäusern welche sich durch die größere Datenmenge bessere Erkenntnisse erwarten, genannt. Die Lösung mit der Speicherung und Verarbeitung durch einen Cloud Anbieter hat Vorteile in Bezug auf Kosten,

Skalierbarkeit und eventuell auch auf die Sicherheit durch den Einsatz einer professionellen Lösung. (Çetin et al., 2017; Lauter et al., 2014)

Als Ergebnis der Analyse der Versuchsimplementierung kann eine praktikable Verwendung von homomorpher Verschlüsselungstechnologie im Kontext der untersuchten Anwendung in Zukunft erreicht werden. Eine detailliertere Betrachtung der sicherheitsrelevanten Schwachstellen wird jedoch auch angeraten. (Çetin et al., 2017)

Im Übrigen ist das Forschungsprojekt HEALER für die statistische Analyse von Informationen von Genomen zu erwähnen. Dieses ermöglicht durch die Nutzung der HElib Bibliothek die Analyse von homomorph verschlüsselten Daten. Die generelle Anwendung wird jedoch durch einige Limitierungen begrenzt. (Wang et al., 2016)

4.3.3 NEXUS: Non Exposure User location privacy System

Die Motivation hinter dem Projekt, welches unter der Führung des Deutschen Forschungsinstituts für Künstliche Intelligenz (DFKI) entstanden ist, war das Spannungsfeld zwischen dem wachsenden Trend von ortsbezogenen Dienstleistungen und der verstärkten Abneigung dieser aufgrund des Eingriffes in die Privatsphäre. (Guldner, Spieldenner, & Schubotz, 2018)

Anonymisierungstechnologien und andere Technologien, welche die Privatsphäre schützen sollen, führen oft zu einer Verschlechterung der Servicequalität und können anfällig für Angriffe sein. Die folgende Forschungsarbeit demonstriert einen Ansatz, „Geo-Fencing“ im Zusammenhang mit homomorphen Verschlüsselungstechnologien zu implementieren. Das vorgestellte Protokoll soll die Privatsphäre schützen und gleichzeitig exakte Resultate für die Lokalisierung liefern. (Guldner et al., 2018)

Die Wissenschaftler benützten hierfür die zuvor vorgestellte Bibliothek python-paillier, welche das partiell homomorphe Verschlüsselungsschema von Pascal Paillier implementiert hat. Die Ergebnisse zeigten, dass eine realistische und praxisnahe Anwendung umsetzbar ist. Gleichzeitig wird von den Autoren auch angemerkt, dass Einschränkungen – wie die ausschließliche Betrachtung von rechteckigen Geo-fence-Formen – in zukünftigen Arbeiten eliminiert werden sollen. (Guldner et al., 2018)

4.3.4 Statistische Methoden für Machine Learning

Im Bereich der Umsetzung von Implementierungen für Machine Learning Algorithmen im Kontext von vertraulichem Cloud Computing ist in den letzten fünf bis zehn Jahren eine Vielzahl an theoretischen Beiträgen in wissenschaftlichen Journalen erschienen. (Aslett, Esperança, & Holmes, 2015a)

Die tatsächlichen Implementierungen unterscheiden sich aber grundlegend in ihren Rahmenbedingungen. Teilweise werden Protokolle verwendet, welche eine Einbindung von mehreren Parteien während dem Berechnungsvorgang voraussetzen. Außerdem muss man zwischen Umsetzungen mittels vollständig homomorpher und partiell homomorpher Schemata

unterscheiden. Ein großer Unterschied ist auch die Trennung vom Anlernen des Modells und der Vorhersage. Aufgrund der heterogenen Struktur werden in diesem Subkapitel in kurzer Form unterschiedliche Forschungsprojekte vorgestellt.

4.3.4.1 PySyft

PySyft ist eine Bibliothek, welche die Nutzung von Optimierungsmethoden künstlicher neuronaler Netze unter Schützung der Privatsphäre ermöglicht. Nach Eigenangabe befindet sie sich noch in einem Prä-Alpha-Status. PySyft nutzt zur Mehr-Parteien-Berechnung – besser bekannt unter dem englischen Begriff Multi-Party-Computation – die Bibliothek PyTorch und tensorflow. (*PySyft*, 2017/2018)

4.3.4.2 EncryptedStats R Package

Aufbauend auf dem Homomorphic Encryption R Package wurde eine Implementierung von zwei statistischen Methoden auf verschlüsselte Daten im Rahmen einer Forschungsarbeit der University of Oxford umgesetzt. Beide Erweiterungen der statistikorientierten Programmierumgebung R entstanden unter Führung von Louis J. M. Aslett. (Aslett, Esperança, & Holmes, 2015b; Aslett et al., 2015a)

Die Methoden Naive Bayes Klassifikation und Random Forest – welche auf dem Prinzip eines Entscheidungsbaumes basiert – wurden mithilfe des Schemas von Fan und Vercauteren (FV) zur vollständig homomorphen Verschlüsselung umgesetzt. Nach eigenen Angaben soll diese Implementierung zur Zeit der Veröffentlichung die erste sein, welche das Anpassen eines Modells (engl. fitting) sowie die Vorhersage von Werten unterstützt. (Aslett et al., 2015a, 2015b)

5. UNTERSUCHUNGSDESIGN

Wie zuvor schon angeführt, soll diese Masterarbeit die wirtschaftlichen Anforderungen mit den technologischen Lösungen zusammenführen. Um diese Brücke zwischen den zwei Welten zu schlagen, sollen die folgenden Szenarien dienen. Ihre Komplexität im Rahmen der Berechnung reicht von einer simplen Aggregation und Durchschnittsberechnung, hin zu einer Methodik zur linearen Regression.

Diese unterschiedliche Komplexität soll auch eine gewisse Repräsentation der Ergebnisse für diverse Arten von Berechnungen mit homomorpher Verschlüsselung bieten. Gleichzeitig ist auch ein mögliches unterschiedliches Verhalten beim Anstieg der Datenmenge von Interesse.

Die Szenarien orientieren sich an den tatsächlichen wirtschaftlichen und politischen Begebenheiten. Neben den konkreten Rahmenbedingungen, welche den Kontext vorgeben, ist vor allem die Wahl der Branche und des Anwendungsgebietes aufgrund der umfangreichen Literaturanalyse getroffen worden.

5.1 Aufstellung der Untersuchungsszenarien

Aufgrund der Analyse der wirtschaftlichen Betrachtung der Thematik werden Untersuchungsszenarien geschaffen. Hierfür werden im folgenden Kapitel Hypothesen aufgestellt. Die einzelnen Auswahlgründe sollen die wirtschaftlichen, gesellschaftlichen oder politischen Notwendigkeiten mit den aktuellen technischen Möglichkeiten homomorpher Verschlüsselung zusammenführen.

5.1.1 Vorgehensweise der Erstellung

Die Erstellung der Szenarien wird in mehreren Teilgebieten durchgeführt. Die dazu notwendigen Schritte sind die folgenden:

1. Vorstellung der Auswahlgründe
2. Ablauf der vertraulichen Datenverarbeitung
3. Kategorisierung der Szenarieneigenschaften
4. Darstellung der Datengrundlage
5. Vorstellung der Hypothesen

5.1.2 Technologische Schwerpunkte der Szenarien

Jedes Szenario besitzt auch einen eigenen Schwerpunkt im Zuge seiner Berechnung. Im ersten Szenario werden unterschiedliche mathematische Operationen ausgeführt, welche sich besonders auf den Leistungsverbrauch in Bezug auf eine korrekte Parameterwahl auswirken. (Chen et al., 2018)

Im zweiten Szenario wird aufgrund von bisherigen Veröffentlichungen durch die Wissenschaft eine Umsetzung einer linearen Regression auf vertrauliche Daten ermöglicht. (Du, Gustafson, Huang, & Peterson, 2017) Dieses Szenario wird aufgrund seiner Komplexität nur mit einem beschränkten Datensatz getestet.

Das dritte Szenario versucht eine relativ einfache mathematische Berechnung für eine sehr große Anzahl von Datensätzen – im Vergleich zu den anderen Szenarien – auf besonders effiziente Weise durchzuführen. Dies wird durch die Anwendung von speziellen technologischen Verfahren der Stapelverarbeitung ermöglicht. (Chen et al., 2018)

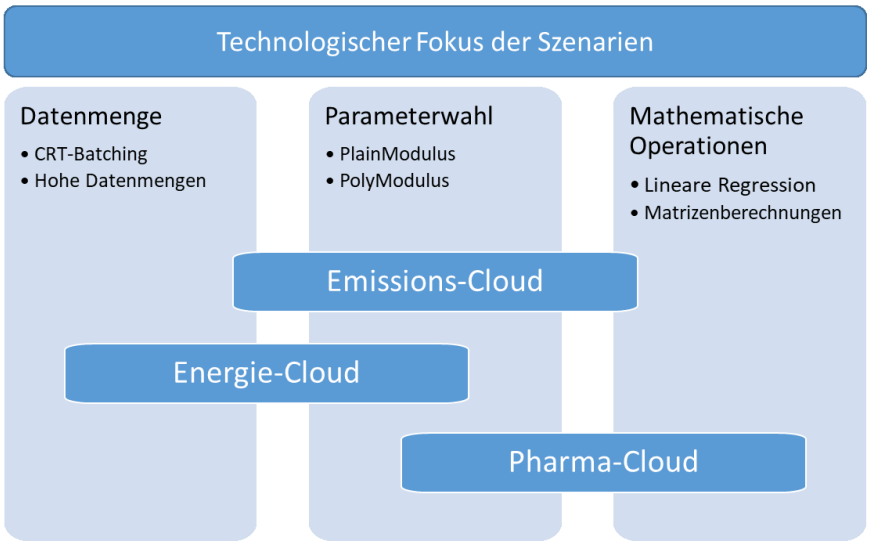

Abbildung 4: Technologischer Fokus der Szenarien
Eigene Darstellung.

5.1.3 Einschränkungen

Innerhalb eines Szenarios werden keinerlei technische Protokolle definiert und auf ihre Sicherheit geprüft. Es gilt anzunehmen, dass für die Umsetzung der möglichen Szenarien unter Betrachtung des Stands aktueller Sicherheitstechnik ein sicheres Protokoll des Schlüsselaustausches innerhalb der Parteien entworfen werden kann. Außerdem sollen eventuelle virtuelle Zugangspunkte und Hardware gegenüber bösartigen Angriffen geschützt werden. Im zentralen Mittelpunkt der Szenarien steht also nur der Einsatz der möglichen vertraulichen Berechnungen unter dem Einsatz von homomorpher Verschlüsselung.

5.2 Szenario 1 – Emissions-Cloud

Dieses Szenario beschäftigt sich mit einer Gruppe von Automobilherstellern und basiert auf konkreten Überlegungen aus der Wissenschaft zur Sammlung von Emissionsdaten, sowie aktuellen Ereignissen im Bereich der Automobilindustrie. (Flügge, 2016; Hilpert, Thoroe, & Schumann, 2011; Ning et al., 2012)

In diesem Szenario werden Datensätze über die realen, gemittelten Emissionswerte der Fahrzeuge im Betrieb von Fahrzeughalterinnen und -haltern gesammelt. Die Daten werden dabei herstellerübergreifend von einer unabhängigen Organisation gesammelt und temporär aufbewahrt und dabei ausgewertet. Aufgrund von legislativen Anforderungen und wirtschaftspolitischen Interessen ist eine Verarbeitung der Daten in unverschlüsselter Form nicht gewünscht.

5.2.1 Auswahlgründe

Dieses Szenario wurde in Hinblick auf eine Vielzahl von Gründen entworfen. Dabei lassen sich potentielle Beweggründe in unterschiedliche Kategorien einteilen. Diese basieren auf den Erkenntnissen zur Literaturanalyse. Das Spannungsfeld zwischen politischen, wirtschaftlichen und wissenschaftlichen Beweggründen ist komplex. Zum Beispiel stehen die etwaigen umweltpolitischen Vorteile durch die gewonnenen Erkenntnisse der gesammelten Daten in einem Konflikt mit dem Schutz der Privatsphäre. In den folgenden Unterkapiteln wird die Schnittstelle zwischen der durchgeführten Literaturanalyse und der praktischen Umsetzung näher erläutert.

5.2.1.1 Gesellschaftliche und politische Beweggründe

Die Sammlung der Daten könnte im Kontext dieses Szenarios von der Politik auf anonyme Weise vorgeschrieben sein, um mithilfe der Daten eine Möglichkeit zu schaffen, Überblick über regionale aktuelle Umweltbelastungen zu erlangen, um eventuelle Gegenmaßnahmen zu empfehlen. (z.B. Einfahrtsperren für ältere Fahrzeuge mit Dieselmotoren in Großstädte). (Achtnicht, Kesternich, & Sturm, 2018)

Die homomorphe Verschlüsselung ist außerdem ein Weg – wenn auch nicht der einzige – für eine anonyme Verarbeitung dieser gemeinschaftlich gesammelter Daten, um Datenschutzansprüchen gerecht zu werden. (Sadeghi & Schneider, 2010) Da hierbei auch eindeutig zuordnungsbare Daten – wie die Fahrzeug-Identifizierungsnummer – übermittelt werden können, kann dieser Aspekt durchaus eine Rolle spielen und würde vor allem bei der Ausweitung auf weitere individuelle Nutzungsdaten stark an Bedeutung gewinnen.

5.2.1.2 Wirtschaftliche Beweggründe

Personen- bzw. fahrzeugbezogene Daten gewinnen für die Automobilindustrie immer stärker an Bedeutung. Durch die steigende Vernetzung im Bereich der Kundinnen- und Kundenbeziehung mit den Herstellern und deren Vertriebs- und Servicepartnern, bieten sich für die einzelnen

Unternehmen neue Möglichkeiten zur stärkeren Kundenbindung sowie für etwaige Effizienzmaßnahmen. In diesem speziellen Szenario könnten z.B. etwaige Probleme bei der Abgasnachbehandlung schnell und effektiv ermittelt werden, ohne dabei die Anonymität grundsätzlich zu verlieren. Diese Informationen könnten genutzt werden um Besitzerinnen und Besitzer der problematischen Fahrzeuge zu informieren, dass eine Reparatur in einem Servicebetrieb notwendig ist. Bei der Sammlung von weiteren Fahrzeugdaten, wie Nutzungsprofilen und Wartungsdaten, eröffnen sich den Automobilherstellern weitere Möglichkeiten. (Flügge, 2016; Winkelhake, 2017)

Die Umsetzung dieser Problemstellung im Rahmen einer gemeinschaftlich genutzten Cloud-Infrastruktur scheint vor allem aufgrund der komplexen Datenquellen und der unterschiedlichen Nutzungsszenarien, Vorteile im Gegensatz zur individuellen Durchführung zu generieren. Die Grundlage für diese Erkenntnis ist das Ergebnis der Literaturanalyse im Kapitel Cloud Computing. (Furht & Escalante, 2010; Rountree & Castrillo, 2014)

5.2.1.3 Technologische und wissenschaftliche Beweggründe

Technologisch ist dieses Szenario unter anderem durch die Überschneidung von unterschiedlichen technischen Materien von Interesse. Beginnend bei der Erfassung und Übertragung von nahezu Echtzeitdaten eines einzelnen Fahrzeugs bis zur umwelttechnischen Analyse der Emissionswerte wurden ähnliche Szenarien schon in der Wissenschaft thematisiert. (Hilpert et al., 2011; Zalakeviciute, Rybarczyk, López-Villada, & Diaz Suarez, 2018)

Dieses Szenario würde auch für andere Bereiche der Wissenschaft eine interessante anonyme Datenbasis für etwaige Untersuchungen bieten. Diese wären vor allem im Zusammenhang mit den komplexen Systemen der lokalen Emissionsauswirkungen von Interesse. (Ning et al., 2012)

5.2.2 Ablauf der vertraulichen Datenverarbeitung

Die einzelnen Fahrzeuge der Hersteller übertragen ihre Daten über ein Mobilfunkmodul mit einer verschlüsselten Verbindung an die gemeinsame Emissions-Cloud. Vor der Übertragung werden die Emissionsdaten mit einem integrierten öffentlichen Schlüssel homomorph verschlüsselt.

Die Cloud-Infrastruktur bietet für die Übertragung eine Schnittstelle an. Um Missbrauch vorzubeugen wird vor der Übertragung die Fahrzeugidentifikationsnummer überprüft. Außerdem könnten zusätzlich Metadaten übertragen und gespeichert werden, welche die räumliche und zeitliche Klassifizierung des Datensatzes ermöglichen.

Die Überlegungen zu diesem Teil des Szenarios beruhen auf wissenschaftlichen Konzeptstudien und Anwendungen aus der Wirtschaft und wurden um den homomorph-kryptographischen Teil erweitert. (Flügge, 2016; Hilpert et al., 2011; Ning et al., 2012) In der nachfolgenden Grafik wurden die Übertragungen und die Zugriffe auf die Cloud-Infrastruktur überblicksmäßig dargestellt.

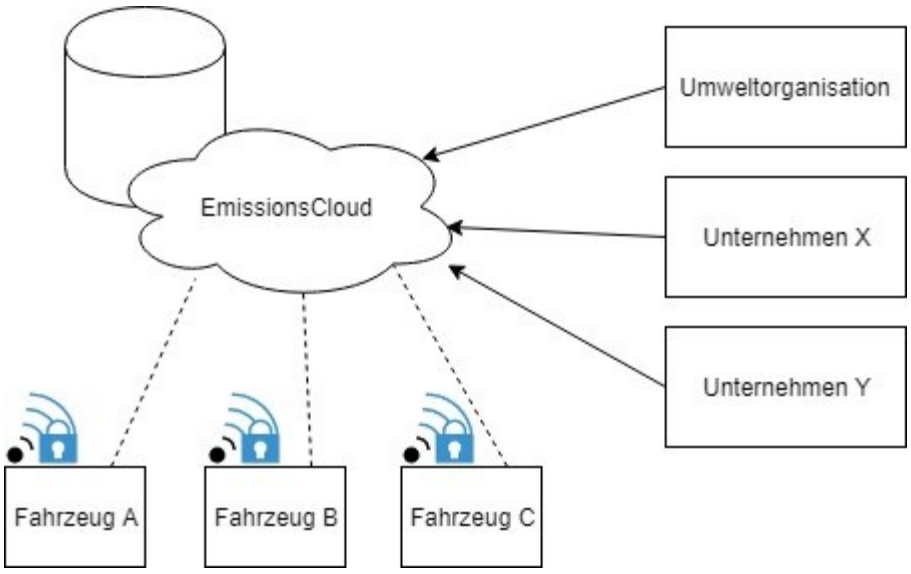

Abbildung 5: Systemdarstellung – Emissions-Cloud
Eigene Darstellung. Icon für sichere Datenübertragung unter CC-Lizenz. (DinosoftLabs, 2018)
Lizenz: https://creativecommons.org/licenses/by/3.0/

5.2.3 Kategorisierung der Szenarieneigenschaften

Durch die Auslagerung der Datenhaltung an eine gemeinschaftliche Organisation ist die Datenverarbeitung in Hinblick auf die gemeinsam genutzte Cloud-Infrastruktur als gemeinschaftliche Cloud zu betrachten. Das Angebot umfasst aus Nutzungssicht die Speicherung und Aggregation der Daten und die Möglichkeit über eine Schnittstelle Berechnungen auf den gesammelten Daten durchzuführen. Gleichzeitig können auch Analysen von den einzelnen Unternehmen angefordert und von der operierenden Organisation hinter der Cloud-Infrastruktur durchgeführt werden.

Daten	Legislativ sensibel	Wirtschaftlich sensibel	Nicht sensibel	
Bereitstellung der Cloud	Öffentlich	Privat	Gemeinschaftlich	Hybrid
Angebot der Dienstleistung	SaaS	PaaS	IaaS	

Tabelle 9: Kategorisierung – Szenario Emissions-Cloud
In Anlehnung an: (Bowen, 2011)

5.2.4 Datengrundlage

Zur Durchführung des Szenarios wurden 10.000 zufällige Datensätze erzeugt. Dies geschah mithilfe eines Dienstes zur teilautomatisierten Datensatzerzeugung („Mockaroo“) und einigen manuellen Nachbearbeitungen. Die Datensätze für die Emissionswerte wurden anhand von industrieüblichen Minimal- und Maximalwerten in Zusammenhang mit einem normalverteilten Zufall bestimmt und zur besseren Plausibilität manuell nachbearbeitet. Die folgenden Attribute wurden definiert: (Brocato, 2018)

Bezeichnung	**Beschreibung**	**Beispielwert**
Id	Identifikationsnummer	99
VIN	Fahrzeug-Identifizierungsnummer	5N1AN0NW6DN520517
car_maker	Fahrzeughersteller	Mazda
car_model	Fahrzeugmodell	CX-7
car_year	Herstellungsjahr	2010
CO2 (g/km)	Durchschnittlicher CO2-Wert pro Kilometer in Gramm	165,50
NOX (mg/km)	Durchschnittlicher NOx-Wert pro Kilometer in Milligramm	397,15

Tabelle 10: Datengrundlage – Szenario Emissions-Cloud
In Anlehnung an: (Brocato, 2018)

5.2.5 Hypothesen

Im ersten Schritt werden theoretisch offensichtliche Hypothesen kontrolliert. Doch vor allem deren eindeutige Beantwortung in Zusammenhang mit der Variation der Problemstellung hilft Antworten auf die Forschungsfrage zu finden.

H1.1: Auf die Qualität des Resultates der Berechnungen im Kontext des Szenarios „Emissions-Cloud“ hat die homomorphe Verschlüsselung keinen Einfluss.

H1.2: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Emissions-Cloud“ einen höheren Speicherverbrauch.

H1.3: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Emissions-Cloud“ einen höheren Leistungsverbrauch.

5.3 Szenario 2 – Pharma-Cloud

In diesem Szenario geht es um die Konzeption und den Ablauf eines Szenarios zur Verarbeitung von vertraulichen medizinischen Daten innerhalb einer Cloud-Infrastruktur. Im Speziellen soll eine Regressionsrechnung für einen Datensatz von Patienten durchgeführt werden und als Ergebnis eine für neue Patientendaten durchführbare Regressionsgleichung geliefert werden.

Dabei kann auf einige wissenschaftliche Publikationen verwiesen werden, welche im Bereich der Medizin die Anwendung von homomorpher Verschlüsselung erprobten oder zumindest konzeptionierten. (Çetin et al., 2017; Wang et al., 2016) Die besondere Sensibilität der Wissenschaft für die vertrauliche Datenverarbeitung in diesem Fachgebiet basiert auf der Schutzbedürftigkeit solch personenbezogener Gesundheitsdaten. (Griebel et al., 2015)

5.3.1 Auswahlgründe

Da sich, wie zuvor schon kurz erwähnt, einige Personen aus unterschiedlichen Bereichen der Forschung mit der vertraulichen Bearbeitung von medizinischen Daten beschäftigt haben, liegt es nahe, dass es viele Beweggründe für die Auswahl eines solchen Szenarios gibt. (Sadeghi & Schneider, 2010; Wang et al., 2016)

5.3.1.1 Gesellschaftliche und politische Beweggründe

Aus gesellschaftlicher und politischer Sicht gibt es viele Beweggründe um die Verarbeitung von personenbezogenen Daten im Gesundheitsbereich vertraulich zu halten. Die Sicherung der Privatsphäre von Patientinnen und Patienten ist von besonders hoher Bedeutung aus der Sicht des Datenschutzes. (Feiler & Horn, 2018)

Die steigende Anzahl an Anwendungsgebieten, welche im Bereich der Medizin evaluiert und konzeptioniert werden, könnte bei gesundheitspolitischen Maßnahmen, wie zum Beispiel einer elektronischen Gesundheitsakte, in der Realität einen Einfluss haben. (Sadeghi & Schneider, 2010)

5.3.1.2 Wirtschaftliche Beweggründe

Nicht nur medizinische Institutionen haben ein Interesse an der Durchführung von groß angelegten Studien. Auch Unternehmen der pharmazeutischen Branche benötigen Daten durch Studien an zahlreichen Patientinnen und Patienten, welche datenschutzrechtlich unbedenklich erhoben, übertragen, persistiert und auch verarbeitet worden sind. (Griebel et al., 2015; Sadeghi & Schneider, 2010)

5.3.1.3 Technologische und wissenschaftliche Beweggründe

In der Forschung gibt es einige Anwendungen von homomorpher Verschlüsselung im Kontext von menschlicher Gesundheit. Im Besonderen existieren einige Forschungsarbeiten über die vertrauliche Auswertung von Daten innerhalb medizinischer Studien, wie zum Beispiel im Bereich der Genetik. (siehe dazu auch Kapitel 4.3.2) (Çetin et al., 2017)

Bei Studien von Forschungseinrichtungen muss oft auf höchstpersönliche Daten zurückgegriffen werden, was große Studien oft vor hohe datenschutzrechtliche Hürden und Verantwortung stellen kann. (Griebel et al., 2015)

Zusätzlich bietet die Verarbeitung in Cloud-Infrastrukturumgebungen einen Vorteil für zeitlich und budgetär begrenzte, aber leistungsintensive Forschungsprojekte, da eine höhere Skalierbarkeit erreicht werden kann. (Rountree & Castrillo, 2014)

5.3.2 Ablauf der vertraulichen Datenverarbeitung

Anwendbare Szenarien für die vertrauliche Verarbeitung von Gesundheitsdaten wurden von unterschiedlichen Wissenschaftlerinnen und Wissenschaftlern schon in diversen Publikationen vorgestellt. Viele beziehen sich dabei auf den Informationsaustausch von Krankenhäusern, um das Wissen und die Skalierungseffekte einer gemeinschaftlichen Cloud-Infrastruktur nutzen zu können. (Du et al., 2017; Griebel et al., 2015)

In diesem Szenario werden Patientendaten (siehe Kapitel 5.3.4) von Krankenhäusern an die gemeinsame Pharma-Cloud übertragen. Diese führt Berechnungen zur Bestimmung einer Regressionsfunktion für den linearen Zusammenhang dieser Daten aus. Von den Ergebnissen, sowie von der Datenquelle an sich, können öffentliche und auch private Forschungsinstitute profitieren. (Du et al., 2017; Sadeghi & Schneider, 2010)

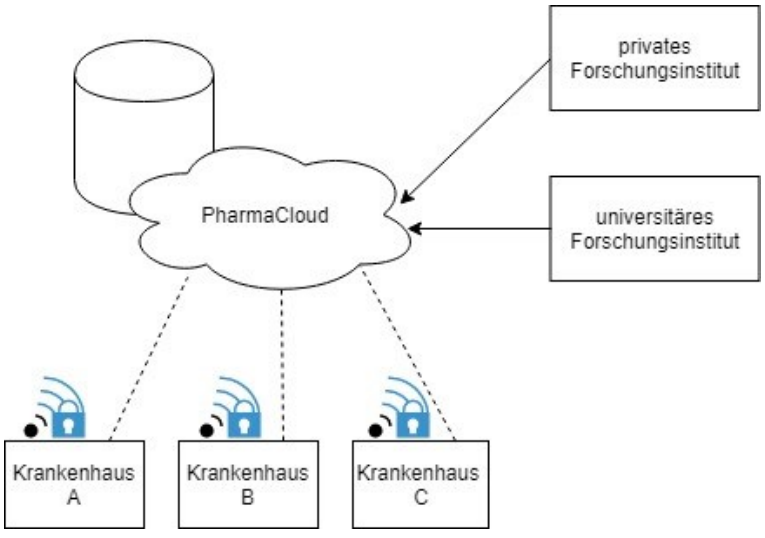

Abbildung 6: Systemdarstellung – Pharma-Cloud

Eigene Darstellung. Icon für sichere Datenübertragung unter CC-Lizenz. (DinosoftLabs, 2018)
Lizenz: https://creativecommons.org/licenses/by/3.0/

5.3.3 Kategorisierung der Szenarieneigenschaften

Die Kategorisierung der Eigenschaften des Einsatzes von Cloud Computing innerhalb des Szenarios wird in der folgenden Tabelle dargestellt. Die zugrundliegenden Daten besitzen aus legislativer Sicht schützenswerten Charakter. (Bowen, 2011; Ghorbel et al., 2017)

Die Nutzung der Cloud-Umgebung passiert auf gemeinschaftlichem Wege. Einerseits stellen die Krankenhäuser im Szenario ihre Daten bereit und andererseits bietet die Cloud unterschiedlichen Instituten die Verarbeitung der homomorph verschlüsselten Datensätze an.

Um dabei diese unterschiedlichen Funktionen zu erfüllen, fungiert das Angebot der Cloud als Plattform. Dadurch wird ein von individueller Software unabhängiger Zugang zu den verschlüsselten Daten für Institute ermöglicht.

Daten	Legislativ sensibel	Wirtschaftlich sensibel	Nicht sensibel
Bereitstellung der Cloud	Öffentlich	Privat	Gemeinschaftlich	Hybrid
Angebot der Dienstleistung	SaaS	PaaS	IaaS

Tabelle 11: Kategorisierung – Szenario Pharma-Cloud

In Anlehnung an: (Bowen, 2011)

5.3.4 Datengrundlage

Als Datengrundlage dient innerhalb dieses Beispiels der Datensatz einer wissenschaftlichen Publikation von Efron, Hastie, Johnstone und Tibshirani aus dem Jahr 2004 mit dem Titel „Least Angle Regression“. Diese Publikation beinhaltet 420 Datensätze über Diabetespatienten mit personenbezogenen Werten wie Alter, Geschlecht, Blutdruck und anderen Diagnosewerten. Zusätzlich ist ein Indikationswert über den Diabetesstatus gegeben („Y“), welcher das Ziel einer möglichen Regression darstellt. (Efron, Hastie, Johnstone, & Tibshirani, 2004)

Bezeichnung	**Beschreibung**	**Beispielwert**
Age	Alter	59
Sex	Geschlecht	2
BMI	Body-Mass-Index	32,1
BP	Blutdruck (Durchschnitt)	101
S1, S2, S3, S4, S5, S6	Unterschiedliche Diagnosewerte	157; 93,2; 38; 4; 4,8598; 87; 151
Y	Indikationswert	151

Tabelle 12: Datengrundlage – Szenario Pharma-Cloud

In Anlehnung an: (Efron et al., 2004)

5.3.5 Hypothesen

Wie zuvor werden wieder drei fundamentale Hypothesen im Zusammenhang mit Funktion und Ressourcenverbrauch aufgestellt. Die Analyse der Leistungsreserven wird in den Auswertungen eine spezielle Rolle spielen.

H2.1: Auf die Qualität des Resultates der Berechnungen im Kontext des Szenarios „Pharma-Cloud“ hat die homomorphe Verschlüsselung keinen Einfluss.

H2.2: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Pharma-Cloud“ einen höheren Speicherverbrauch.

H2.3: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Pharma-Cloud“ einen höheren Leistungsverbrauch.

5.4 Szenario 3 – Smart Meter Energie-Cloud

Wie zuvor schon im einführenden Kapitel angesprochen steht die Energiebranche vor großen Veränderungen. Im Vordergrund dieser Veränderungen steht die Steigerung der Effizienz. Dies impliziert außerdem eine geringere Umweltbelastung in Zusammenhang mit dem zusätzlichen Effekt von Kosteneinsparungen. Dies soll durch eine konstante, fein granulare Überwachung der einzelnen Haushalte und Energieunternehmen verwirklicht werden. Im Spannungsfeld von Datenschutz und Effizienzverteilung befindet sich die kontinuierliche Verarbeitung von Stromverbrauchsdaten einzelner Haushalte. (Doleski, 2017)

In diesem Szenario wird die Vertrauensbeziehung zwischen dem Stromanbieter und der Nutzerin oder des Nutzers im Mittelpunkt stehen. Die Nutzerin oder der Nutzer möchten über die aktuellen Kosten eines dynamisch auswählbaren Zeitpunkts in der Vergangenheit Bescheid wissen, aber den Stromanbieter dabei nicht den exakten Verbrauch der angesprochenen Smart Meter einsehen lassen.

5.4.1 Auswahlgründe

Aufgrund der Erkenntnisse aus der Literaturanalyse lassen sich einige Beweggründe für die Umsetzung eines solchen beispielhaften Szenarios feststellen. Diese Gründe lassen sich nicht eindeutig auf unterschiedliche Kategorien einteilen, doch wird dies in den folgenden Unterkapiteln zur besseren Übersicht getan, um die unterschiedlichen Sichtweisen besser zu erkennen.

5.4.1.1 Gesellschaftliche und politische Beweggründe

Dieses Szenario spielt vor allem aufgrund der komplexen Thematik von personenbezogenen Daten eine interessante Rolle. Da die geltende Rechtsmeinung nicht aggregierten Verbrauchsdaten eine mögliche Identifizierbarkeit der Person unterstellt, spielen solche Szenarien vor allem durch das Inkrafttreten von strengeren Datenschutzrichtlinien eine große Rolle. (Feiler & Horn, 2018)

5.4.1.2 Wirtschaftliche Beweggründe

Die wirtschaftlichen Vorteile von Cloud Computing in Zusammenhang mit der Möglichkeit dies auch auf Bereiche ausweiten zu können, bei welchen es vorher aufgrund unterschiedlicher Gründe noch nicht möglich war, wurden in der Literaturanalyse schon ausführlich aufgezählt. Im Speziellen könnte dieses Szenario einerseits eine Möglichkeit sein Datensätze, welche zuvor aus rechtlichen Gründen nicht in verwertbarer Form langfristig gespeichert werden konnten, aufzubewahren und bei etwaigen späteren expliziten Befähigungen auch im Nachhinein zu analysieren. (Doleski, 2017)

5.4.1.3 Technologische und wissenschaftliche Beweggründe

Im Bereich der Vernetzung von intelligenten Stromzählern und der Verarbeitung ihrer Daten werden in den letzten Jahren verstärkt wissenschaftliche Publikationen erstellt. Hierbei steht oft – wie bereits bei den politischen Beweggründen angeführt – die Sicherheit der Informationen im Vordergrund. Unterschiedliche Anonymisierungsverfahren werden hierzu evaluiert. (Bos, Castryck, Iliashenko, & Vercauteren, 2016; Spiecker, 2017)

Die Wissenschaft und die Wirtschaft müssen erst beweisen, ob sich Technologien zum vertraulichen Cloud Computing im Zusammenhang mit homomorphen Verschlüsselungstechniken eignen dieser Problematik entgegenzuwirken.

5.4.2 Ablaufbeschreibung der vertraulichen Datenverarbeitung

In dieser Versuchsimplementierung sollen Werte, welche durch einzelne Stromzähler in einem Haushalt erfasst werden, vertraulich von einem zentralen Rechner aggregiert werden. Ein wichtiger Punkt dabei ist, dass die einzelnen Daten der Smart Meter nicht unverschlüsselt beim Stromanbieter persistiert sein dürfen. Der private Schlüssel zur Verschlüsselung soll auch nur der Nutzerin oder dem Nutzer bekannt sein.

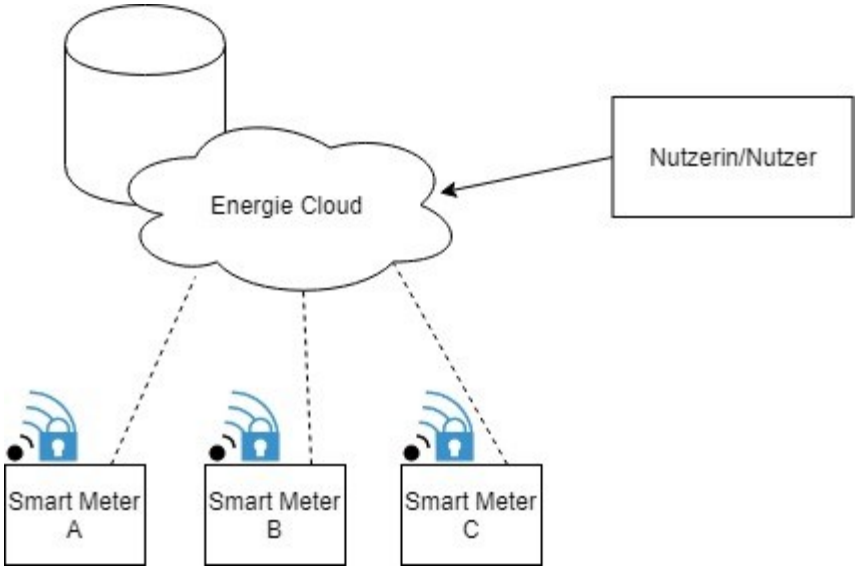

Abbildung 7: Systemdarstellung – Energie-Cloud
Eigene Darstellung. Icon für sichere Datenübertragung unter CC-Lizenz (DinosoftLabs, 2018)
Lizenz: https://creativecommons.org/licenses/by/3.0/

Diese vertraulichen Informationen sollen also der rechnenden Partei nicht offengelegt werden. Dies ermöglicht eine gegenseitige Vertrauensbeziehung. Außerdem soll es in potentiellen Ausbaustufen des Szenarios – welche in dieser Implementierung nicht vollständig durchgeführt werden – möglich sein gewisse Kennzahlen über die verschlüsselten Daten zu berechnen, ohne die Vertraulichkeit der Daten zu verletzen.

Dieses Szenario soll, um den Anreiz der Nutzerin oder des Nutzers an der persistenten Speicherung seiner minutengenauen Verbrauchswerte darzustellen, den exakten Kostenbetrag der einzelnen Smart Meter berechnen und ihn für die Kundin oder den Kunden bereitstellen. Dieser Kostenbetrag wird vom Stromanbieter dynamisch stundengenau festgelegt und bewegt sich in einem vorher definierten Rahmen. Dadurch entsteht für den Stromanbieter die Möglichkeit einen Anreiz zu schaffen, um in Zeiträumen mit schwacher Nachfrage eine höhere Nachfrage zu erzeugen und umgekehrt zu Zeiten von hoher Nachfrage diese zu verringern.

5.4.3 Kategorisierung der Szenarieneigenschaften

Die Einordnung nach dem in der Literaturanalyse aufgestellten Raster zur Erfassung der Cloud Computing Eigenschaften des Szenarios erfolgt in der folgenden Tabelle. Dabei wird den Daten aufgrund der in Kapitel 3.2.2 genannten Gründe eine legislative Sensibilität zugewiesen. Auch eine wirtschafts- und politische Relevanz kann dabei aus ähnlichen Gründen vorhanden sein. (Doleski, 2017)

Die Bereitstellung der Funktionalität der Cloud hat öffentlichen Charakter. Dies lässt sich durch den Zugriff der Kundinnen und Kunden von außen auf unterschiedliche Schnittstellen (z.B. über Web-Applikationen), welche den Zugriff auf die Datenbasis ermöglichen, begründen. (Rountree & Castrillo, 2014)

Dabei werden unterschiedliche Softwarefunktionen für Nutzerinnen und Nutzer bereitgestellt. Ein direkter Zugriff auf die Infrastruktur (z.B. die Datenbank) ist im ursprünglichen Kontext dieses Szenarios nicht vorhergesehen.

Daten	Legislativ sensibel		Wirtschaftlich sensibel	Nicht sensibel
Bereitstellung der Cloud	Öffentlich	Privat	Gemeinschaftlich	Hybrid
Angebot der Dienstleistung	SaaS		PaaS	IaaS

Tabelle 13: Kategorisierung – Szenario Energie-Cloud
In Anlehnung an: (Bowen, 2011)

5.4.4 Datengrundlage

Die Datengrundlage für die zugrundeliegende Proof-of-Concept-Implementierung stammt von der Datenbank der University of California und wurde von zwei französischen Wissenschaftlern zur Verfügung gestellt. (Hebrail & Berard, 2012)

Es handelt sich dabei um einen Datensatz mit neun Attributen und ungefähr zwei Millionen Messungen, wobei in dieser Arbeit nur die ersten 100.000 Datensätze benützt werden. Diese Messungen wurden von Dezember 2006 bis November 2010 durchgeführt. (Hebrail & Berard, 2012)

Folgende Attribute des Datensatzes werden vom Szenario genützt:

Bezeichnung	**Beschreibung**	**Beispielwert**
Date	Tag der Datenerfassung (Tag/Monat/Jahr)	13/1/2007
Time	Zeitpunkt der Datenerfassung (Stunden:Minuten:Sekunden)	22:17:00
voltage	Spannung (in Volt als Durchschnitt pro Minute)	228.710
sub_metering_1	Energie Teilmessung 1 (in Wattstunden) Küche (Geschirrspüler, Backofen, Mikrowelle)	25,000
sub_metering_2	Energie Teilmessung 2 (in Wattstunden) Wäscheraum (Waschmaschine, Trockner, Kühlschrank, Lampe)	1,000
sub_metering_3	Energie Teilmessung 3 (in Wattstunden) Klimaanlage und elektrische Wasserheizung	16,000

Tabelle 14: Datengrundlage – Szenario Energie-Cloud

In Anlehnung an: (Hebrail & Berard, 2012)

5.4.5 Hypothesen

Es gilt auch hier die wesentlichen Merkmale der vertraulichen Berechnung festzustellen. Hierfür werden noch weitere Hypothesen abgeprüft.

H3.1: Auf die Qualität des Resultates der Berechnungen hat die homomorphe Verschlüsselung im Kontext des Szenarios „Energie-Cloud“ keinen Einfluss.

H3.2: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Energie-Cloud“ einen höheren Speicherverbrauch.

H3.3: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Energie-Cloud“ einen höheren Leistungsverbrauch.

Weitere interessante Hypothesen sollen im Zusammenhang mit der benutzten CRT-Technik aufgestellt werden. Auf diese Methode zur effizienteren Verarbeitung von Daten in Vektorstruktur wird in den Kapiteln 6.4.1. und 6.4.2 vertieft eingegangen.

H3.4: Die Anwendung der CRT-Technik im Kontext der Berechnungen des Szenarios „Energie-Cloud“ hat keine Auswirkung auf die Qualität des Ergebnisses.

H3.5: Die Anwendung der CRT-Technik im Kontext der Berechnungen des Szenarios „Energie-Cloud“ erzeugt weniger Leistungsverbrauch, als die vertrauliche Berechnung ohne Anwendung der CRT-Technik.

H3.6: Die Anwendung der CRT-Technik im Kontext der Berechnungen des Szenarios „Energie-Cloud“ erzeugt weniger Speicherverbrauch, als die vertrauliche Berechnung ohne Anwendung der CRT-Technik.

6. IMPLEMENTIERUNG DER UNTERSUCHUNGSOBJEKTE

Dieses Kapitel soll die Vorgehensweise der Umsetzung der, für an die wirtschaftlichen Rahmenbedingungen angelehnten Szenarien mittels Technologien zur vertraulichen Berechnung, näher vorstellen. Für die unterschiedlichen Szenarien wurde eine geeignete Methodik gewählt. Diese hat keineswegs den Anspruch unter allen Umständen die geeignetste zu sein. Doch – wie auch in der Praxis üblich – sind Faktoren wie die Komplexität der Umsetzbarkeit und die Testbarkeit in Zusammenhang mit der Nachvollziehbarkeit in die Entscheidung eingeflossen. (Varia et al., 2015)

Da die Praktikabilität und die realitätsnahe Anwendung nicht durch ein objektives Testverfahren festgestellt werden können, sollen Erkenntnisse, welche aufgrund von rein technologischer Analysen getätigt werden, nicht direkt auf praxisnahe Problemstellungen übertragen werden. Diese Einschränkung wird auch aus wissenschaftlicher Sicht unterstützt, da bei der Übersetzung von vorhandenen Lösungen von Sachverhalten nicht auf dieselbe Vorgehensweise gesetzt werden kann. (Varia et al., 2015)

Dies liegt unter anderem daran, dass in Realanwendungen eine Form von bedingter Logik involviert ist. Für die Umsetzung dieser bedingten Rechenschritte gibt es im Wesentlichen keine direkte Unterstützung einer homomorphen Verschlüsselungsbibliothek. Eine reale Anwendung muss im ersten Schritt auf die Anforderungen von homomorpher Verschlüsselung – bzw. von entsprechenden Implementierungen homomorpher Verschlüsselungstechniken – abgeändert werden. Diese Modifikationen können selbst schon einen großen Einfluss auf die Leistung des Programms haben und sind meist auch nur in komplexer Form möglich. (Varia et al., 2015)

Das führt dazu, dass die Ergebnisse von Versuchen, welche unterschiedliche Bibliotheken gegeneinander testeten, nicht direkt auf die reale Anwendbarkeit übertragbar sind. Es benötigt Szenarien, welche beispielhafte – aus der Praxis entlehnte – Problemstellungen untersuchen und diese auf ihre Machbarkeit und Leistungsdaten evaluieren. In eingeschränkter Form können dann Aussagen über die Praktikabilität unter den entsprechenden Rahmenbedingungen getroffen werden. (Varia et al., 2015)

6.1 Implementierungen auf Basis von SEAL

Die im Kapitel 4.2.3 schon kurz vorgestellte Implementierung SEAL von Microsoft Research ist unter ständiger Weiterentwicklung. Ihre Aktualität im Vergleich zu anderen Bibliotheken sowie ihre breite Anwendung in wissenschaftlichen Forschungsprojekten legen die Nutzung von SEAL als Basis für Anwendungen, welche eine Bibliothek zur vollständigen homomorphen Verschlüsselung voraussetzen, nahe. Sie basiert auf den Erkenntnissen von Brakerski-Fan-Vercauteren (kurz B/FV) und berücksichtigt Standardisierungen des akademischen Konsortiums für fortgeschrittene vertrauliche Berechnungen – dem sogenannten „HomomorphicEncryption.org“-Konsortium. (Microsoft Research, 2018)

Die ursprüngliche Implementierung in C++ bietet aufgrund der angebotenen Schnittstellen zu C# auch einer weiteren, breiten Nutzerbasis die Möglichkeit, Applikationen umzusetzen. Dies ist vor allem interessant, da mehr als 60 Prozent der Entwickler C# als eine ihrer favorisierten Programmiersprachen sehen. (Stack Overflow, 2018)

In den folgenden Unterkapiteln sollen in knapper Form die Hauptbestandteile der Bibliothek und ihrer Komponenten sowie deren Einschränkungen dargestellt werden. Hierzu sei angemerkt, dass sich die aktuelle Version (SEAL 2.3.1) in Bezeichnungen, Optimierungen und auch teilweise in grundlegenden Funktionen von nicht viel älteren Versionen unterscheidet. (z.B. SEAL v2.2. von 2017) (Chen, Laine, & Player, 2017)

Die Nennung der englischen Bezeichnungen der einzelnen Funktionskomponenten in den folgenden Unterkapiteln soll einen Querverweis auf den Quellentext der Implementierung erleichtern.

6.1.1 Kodierung (Encoding)

Vor einer Verschlüsselung der zugrundeliegenden Zahlen müssen diese speziell kodiert (engl. encoding) werden. Klartexte werden beim angewendeten B/FV-Schema als Polynome eines Ringes interpretiert. Um Nutzern viele manuelle Umrechnungen zu ersparen, werden die Klartexte zuvor kodiert. Dafür bietet SEAL mehrere Möglichkeiten und unterschiedliche Parameter. Die Kodierung bietet sehr viel Potential in Hinblick auf die Verbesserung der Praxisfähigkeit für reale Anwendungsfälle. (Chen et al., 2018; Microsoft Research, 2018)

Neben der einfachen Skalarkodierung, welche aber aufgrund ihrer Ineffizienz nicht in SEAL implementiert ist, gibt es noch weitere Möglichkeiten zur Kodierung, welche abhängig von den Rahmenbedingungen der Problemstellungen gewählt werden sollen. (Microsoft Research, 2018)

Bei der Kodierung für ganzzahlige Werte liefert SEAL nicht nur eine effiziente Variante für die Kodierung von einzelnen Zahlen, sondern auch eine Variante für die Kodierung von mehreren ganzzahligen Werten im Sinne einer Stapelverarbeitung (engl. batching). Diese Prozedur wird CRT-Batching genannt. Die dahinterliegende Methodik wird im Kapitel 6.4.2 als besondere Anmerkung zur Implementierung näher vorgestellt. (Chen et al., 2018)

Bei Anwendungen, welche die Berechnung von rationalen Zahlen erfordern wird mittels eines eigenen Kodierers für Bruchzahlen (engl. fractional encoder) eine komplexere Variante des ganzzahligen Kodierers benützt. (Chen et al., 2018; Microsoft Research, 2018)

6.1.2 Verschlüsselung

Die unterschiedlichen Parameter zur effektiven und effizienten Verschlüsselung bei der Verwendung von SEAL werden zu Beginn an eine Instanz eines Objekts für Verschlüsselungsparameter übergeben. Hierbei muss ein Polynom mit der Struktur $x^n + 1$ übergeben werden, wobei „n“ ein Potenzwert einer beliebigen natürlichen Zahl zur Basis Zwei sein muss. (Chen et al., 2018; Microsoft Research, 2018)

Für den darauffolgenden Parameter – den Koeffizienten-Modulus – wird der Wert durch eine Methode an die Variable „n“ des zuvor beschriebenen Polynoms angepasst. In weiterer Folge ist noch der Klartext-Modulus anzugeben. Dieser Wert kann jeden ganzzahligen Wert, welcher durch eine Bitanzahl zwischen zwei und 60 repräsentierbar ist, annehmen und sollte bei Anwendungen mit hohen Leistungsansprüchen möglichst niedrig gesetzt werden. Die restlichen Parameter können automatisiert von SEAL vergeben werden, wovon in den Implementierungen für diese Masterarbeit auch Gebrauch gemacht wurde. Die Bedeutungen und die zulässigen Bereiche dieser Parameter werden in der zitierten Publikation näher erläutert. (Chen et al., 2018; Microsoft Research, 2018)

In weiterer Folge werden ein Schlüsselpaar und ein weiterer Schlüssel erzeugt. Das Schlüsselpaar – ein öffentlicher (engl. public) und ein privater Schlüssel (engl. private) – dienen zur Ver- und Entschlüsselung der kodierten Klartexte, der zusätzliche Schlüssel – Evaluierungsschlüssel (engl. evaluation key) - hingegen zur Relinearisierung. (Microsoft Research, 2018)

Die Relinearisierung mittels des Evaluierungsschlüssels wird vor allem bei der wiederholten Multiplikation von Schlüsseltexten benötigt um deren Größe zu kontrollieren und somit eine effiziente Parameterwahl zu ermöglichen. (Microsoft Research, 2018)

6.1.3 Klasse für Matrizenmanipulationen

Zur besseren Anwendung wurde für die Szenarien eine eigene Klasse für die vereinfachte Umsetzung von Matrizenmanipulationen geschaffen. Die Ideen zur dieser Implementierung basieren auf der Publikation von Du, Gustafson, Huang und Peterson vom MIT. Diese setzten ähnliche Anforderungen in C++ für eine ältere Version von SEAL um. Durch die in Teilbereichen sehr weitreichenden Neuerungen von SEAL, ist eine Vielzahl von Anpassungen dieser Methoden notwendig, dabei wurde außerdem aus Gründen der Einheit die Logik anstatt in C++ in C# implementiert. (Du et al., 2017)

Diese Klasse beinhaltet die Multiplikation und Addition von zwei verschlüsselten sowie von unverschlüsselten Matrizen. Außerdem ist die Multiplikation einer verschlüsselten oder unverschlüsselten Matrix mit einer Konstanten möglich. Auch die Berechnung der Determinante einer verschlüsselten oder unverschlüsselten Matrix wurde implementiert. In weiterer Folge kann auch die komplementäre Matrix einer verschlüsselten oder unverschlüsselten Matrix berechnet werden. (Du et al., 2017)

Diese Methoden waren für die unten angeführten Proof-of-Concept-Implementierungen notwendig, können jedoch auch für andere Zwecke benützt werden. In der folgenden Tabelle werden die Hauptfunktionen der Matrizenklasse kurz vereinfacht dargestellt. (Du et al., 2017)

Funktion	**Methodenbezeichnung**	**Anmerkung**
Addition	AddMatrix	
Multiplikation / *Multiplikation mit einer Konstanten*	MultiplyMatrix MultiplyConstant	Relinearisierung mittels Evaluierungsschlüssel notwendig
Leere Initialisierung	InitializeEmpty	Erstellt leere Schlüsseltexte
Transponierung der Matrix	GetTransposeMatrix	
Berechnung der Komplementären Matrix	GetAdjugateMatrix	
Berechnung der Determinante	GetDeterminantOfMatrix/ GetDeterminantMatrixRe-cursive	(auch als rekursive Methode für Matrizen n,m > 3)
Berechnung der Adjunkten Matrix	GetAdjugateMatrix/ GetAdjugateMatrixRecur-sive	(auch als rekursive Methode für Matrizen n,m > 3)

Tabelle 15: Funktionen zur Matrizenmanipulation
In Anlehnung an: (Du et al., 2017)

Im Gegensatz zu den anderen Methoden können für die Addition, Multiplikation, Initialisierung und Transponierung von Matrizen, aufgrund der vorhandenen Methoden von SEAL, relativ simple Prozeduren gewählt werden. Die Additionsfunktion übernimmt eine weitere Matrix und addiert alle Werte in denselben Zeilen- und Spaltennummern. Angewandt kann dies nur für Matrizen desselben Typs – also derselben Zeilen- und Spaltenanzahl – werden. (Du et al., 2017)

Die Multiplikation zweier Matrizen setzt eine Gleichheit zwischen der Spaltenanzahl der ausführenden Matrix und der Zeilenanzahl der übergebenen Matrix voraus. In unseren Szenarien werden auch wiederholte Multiplikationen von Matrizen benötigt. Dadurch war es – wie im vorigen Kapitel angemerkt – notwendig, dass man bei der entsprechenden Hilfsmethode einen Evaluierungsschlüssel zur Relinearisierung übergibt. Diese Relinearisierung findet nach der Berechnung statt und verhindert, dass eine zu lange Schlüssellänge zu einem falschen Ergebnis führt. Auch eine Matrizenmultiplikation mit einer Konstanten wurde ermöglicht. (Du et al., 2017)

Bei der Initialisierungsfunktion wird eine durch Spalten- und Zeilenanzahl definierte Menge von leeren Schlüsseltexten erstellt und in einer verschachtelten Listenstruktur retourniert. Die Transponierungsfunktion retourniert eine Matrix mit verdrehter Spalten- und Zeilendimension. (Du et al., 2017)

Die Berechnung der Determinante einer homomorph verschlüsselten Matrix ist jedoch etwas komplexer, doch folgt sie auch denselben Prinzipien wie die Berechnung von Matrizen in Klartext. Für Matrizen der Größe $0 < n, m \leq 3$ wurde eine effizientere direkte Berechnung angewandt. Für größere Matrizen wurde jedoch auf eine rekursive Prozedur, den Laplace-Entwicklungssatz, zurückgegriffen. Hierbei könnte durch andere Verfahren auch eine effizientere Berechnung erfolgen. (Du et al., 2017)

Für die Berechnungen der adjunkten Matrix – auch als komplementäre Matrix bezeichnet – wird ähnlich wie bei der Berechnung der Determinanten, eine effizientere Methodik für Matrizen mit der Eigenschaft $0 < n, m \leq 3$ angewandt. Größere Matrizen werden mittels einer rekursiven Prozedur berechnet. (Du et al., 2017)

6.1.4 Besondere Einschränkungen

6.1.4.1 Kompilieren

Beim Kompilieren des Quellcodes mittels Visual Studio müssen einige Voraussetzungen getroffen werden. Da die hinterlegte Referenz zur SEAL Bibliothek für 64-Bit kompiliert wurde, gilt dies zuerst auch für das jeweilige Projekt zwingend einzustellen.

Außerdem muss beachtet werden, dass SEAL in der Version von 2.3.1 verwendet wurde. Da neue Versionen oft in kurzen Abständen erscheinen, gilt es entweder etwaige Kompatibilitäten zuvor abzuprüfen oder die ursprüngliche Version 2.3.1 zu verwenden.

6.1.4.2 Datenübertragungen

Im Sinne der Szenarien werden oft Datenübertragungen von mehreren Parteien an eine zentrale Cloud-Infrastruktur notwendig sein. Da diese schwer in diesem Umfang prüfbar sind und eine realistische Darstellung des Sachverhalts nur mit großem Aufwand möglich ist, wurden die Daten innerhalb der Implementierungen bewusst nur einmalig aus einem statischen File importiert und von da an in einer Programmumgebung verarbeitet. Auch die Annahme von vorhandenen und zuvor bestimmten Protokollen für den sicheren Schlüsselaustausch und Datenübertragungen ermöglichte eine bessere Konzentration auf die Analyse der einzelnen Komponenten der Berechnungen der vertraulichen Daten.

6.2 Implementierung Szenario 1 – Emissions-Cloud

Bei der Implementierung dieses Szenarios wurde vor allem Wert auf den Einsatz einer variablen Datensatzanzahl in Zusammenhang mit unterschiedlichen Parametern gelegt. Hierbei stehen bei der Analyse die unterschiedlichen Ansprüche der Berechnungen im Vordergrund.

Da die mathematischen Operationen an sich keine hohe Komplexität aufweisen, ist besonders im Bereich der Berechnungen ein klarer Nachteil für die homomorphe Methodik zu erwarten. Doch besonders im Kontext des Systemaufbaus sind die unterschiedlichen Lastverteilungen innerhalb der Operationen von Interesse. (Armknecht et al., 2015)

Um die Analyse der Ergebnisse zu erleichtern wurden insgesamt drei unterschiedliche Varianten der Implementierung vorgenommen, welche auch bei der Ausführung des Programms ausgewählt werden können:

1. Variante mit vertraulicher und nicht-vertraulicher Berechnung
2. Variante mit vertraulicher Berechnung
3. Variante mit nicht-vertraulicher Berechnung

Die unterschiedliche Komplexität der Varianten führt dazu, dass Variante 2 und 3 weniger Schritte besitzen. Die unter 6.2.1. vorgestellte Ablaufbeschreibung gilt der ersten Variante.

6.2.1 Ablaufbeschreibung

Das Programm zur Durchführung dieses Szenarios beinhaltet grundsätzlich acht Schritte, bei welchen unterschiedliche Operationen durchgeführt werden:

1. Importieren der Datensätze aus der Datenquelle
2. Parametrisierung des SEAL-Kontextes und der notwendigen Schlüssel
3. Kodierung und Verschlüsselung der Daten
4. Aggregation der Emissionsdaten und Berechnung der Durchschnittswerte
5. Kalkulation der Emissionskennzahl für jeden Datensatz und dessen gewichteten Gesamtdurchschnitts
6. Entschlüsselung und Dekodierung der Resultate
7. Durchführung von Schritt 4 bis 7 mit unverschlüsselten Datensätzen
8. Vergleich der Resultate und etwaiger Differenzen

Beim Import der Datensätze wurde auf die im Kapitel 5.2.4 näher beschriebene Datengrundlage zurückgegriffen. Diese werden von einer lokal abgelegten Datenquelle in einer vom Anwender ausgewählten Anzahl verarbeitet.

Bei der Initialisierung der Verschlüsselungsparameter von SEAL wurde ein niedriger Polynom-Modulus („1x^2048 +1“) gewählt, da keine starke multiplikative Tiefe durch die Operationen erreicht wurde. Auch der Koeffizienten-Modulus konnte niedrig gewählt werden. Hierbei wurde auch in einigen Testdurchgängen die Berechnung mit höherem Polynom-Modulus durchgeführt, um im speziellen die potentiellen Einbußen für die Möglichkeit von nachträglichen komplexeren Datenverarbeitungsvorgänge zu evaluieren. Dies ist vor allem deshalb von Interesse, da diese Verschlüsselungsparameter nicht nachträglich – also nach der ersten Kodierung – geändert werden können. Der Vergleich mit unterschiedlichen Polynom-Modulus wird in den Auswertungen im Kapitel 7.1.1 näher dargestellt.

Im Gegensatz dazu, besitzt vor allem der Parameter mit dem Namen „PlainModulus“, also der Klartext-Modulus, eine hohe Relevanz für die Korrektheit der Ergebnisse. Diese Relevanz zeigt sich insbesondere im Zusammenhang mit einem Anstieg bei der Anzahl der zu verarbeiteten Datensätze. Die jeweilige Schwelle für ein korrektes Ergebnis und der jeweilige höhere Rechenaufwand werden in den Auswertungen im Kapitel 7.1.1 mit Analysedaten detailliert dargestellt. Als ausreichend groß dimensionierter Wert für 10.000 Datensätze hat sich 8.388.608 (2^23) herausgestellt.

Es wurden mittels eines Schlüsselgenerators ein öffentlicher Schlüssel und ein privater Schlüssel erzeugt. Der öffentliche Schlüssel wird bei Verschlüsselungen benützt, der private bei der Entschlüsselungen.

In diesem Szenario wurde ein Kodierer für gebrochene Zahlen benützt (siehe Kapitel 6.1.1 – „FractionalEncoder“). Dieser Kodierer wird parametriert mit dem zuvor bestimmten Klartext-Modulus und dem Polynom-Modulus. Zusätzlich werden noch weitere Parameter für die Definition der Klartextpolynome übergeben. Weitere Details werden in den Kommentaren zum Quellcode erläutert.

Die Erzeugung der Objekte mit den einzelnen Schlüsseltexten welche einen Wert darstellen, wird jeweils für die Stickoxid- und Kohlendioxidwerte durchgeführt. Dieser Schritt ist sehr laufzeitintensiv, unter anderem deshalb wird seine Ausführungszeit mit einer Diagnostikfunktion mitgestoppt und ausgegeben.

Die Durchführung der ersten mathematischen Operationen mit den kodierten und verschlüsselten Werten erfolgt im Schritt 4. Die Aggregation der Schlüsseltexte geschieht über das zuvor instanziierte „Evaluator“-Objekt. Bei der Berechnung der Mittelwerte gelangt man schon erstmals an die Grenzen der SEAL-Bibliothek. Das Dividieren durch einen anderen Schlüsseltext ist mittels Funktionen von SEAL nicht möglich und muss hierbei durch eine Multiplikation eines Klartext-Objekts umgangen werden. (Chen et al., 2018; Microsoft Research, 2018)

Dazu wird die Anzahl der vorhandenen Schlüsseltexte ermittelt und ein Quotient mit der Anzahl als Divisor und einer Eins als Dividenden berechnet. Der Wert des Quotienten wird dann kodiert und mit einer Funktion des „Evaluator“-Objekts mit der aggregierten Summe multipliziert.

Im nächsten Schritt (5) wird die gewichtete Emissionskennzahl für jeden einzelnen Emissionswert berechnet. Die Emissionskennzahl wurde auf Basis der Test- und Bewertungskriterien des „EcoTest“ des deutschen Automobilklubs ADAC ermittelt. (ADAC Fahrzeugtechnik, 2016) Er soll eine mögliche gewichtete Umweltkennzahl für etwaige Großstudien im Bereich der Analyse der Emissionsdaten darstellen. Dazu werden die verschlüsselten Emissionsdaten anhand einer linearen Skalierung in Punkte übergerechnet. Diese lineare Skalierung verläuft nach einem Muster, welches in den Bewertungskriterien des „EcoTest“ detailliert erklärt wird. Dabei werden eigene Punktwerte für zwei Emissionswerte des jeweiligen Emissionstyps definiert. Hierbei wurde für Stickoxidwerte nicht der Wertebereich von Testfahrten am stationären Prüfstand herangezogen, sondern dieser für den ADAC Autobahn-Test, welcher besser geeignet ist das Szenario nachzubilden. (ADAC Fahrzeugtechnik, 2016)

NOx (ADAC Autobahn)		**CO2**	
50 Punkte bei [g/km]	0 Punkte bei [g/km]	50 Punkte bei [g/km]	0 Punkte bei [g/km]
0,06	0,26	100	250

Tabelle 16: Wertebereiche für die lineare Skalierung nach dem ADAC EcoTest
In Anlehnung an (ADAC Fahrzeugtechnik, 2016).

Diese Vorgaben wurden mathematisch in die folgenden Funktionen umgeformt, welche darauf auf die einzelnen Werte angewendet wurden. Die Werte für Stickoxide wurden dabei in Milligramm pro Kilometer umgerechnet.

$$Punkte(NOx) = -0.25\,NOx\left[\frac{mg}{km}\right] + 65 \;\; bzw. \; Punkte\,(CO2) = -\frac{1}{3}\,CO^2\,[\frac{g}{km}] + 250/3$$

Für jedes einzelne erfasste Fahrzeug wurde anschließend ein gemischter gewichteter Punktewert berechnet. Dazu wurden die beiden zuvor berechneten Punkte mit dem selbstgewählten Faktor (Gewichtung) als Klartextpolynom multipliziert und addiert. Daraufhin erfolgt eine Aggregation und Mittelwertbestimmung analog zum Schritt 4. Schlussendlich wurde auch für diesen Schritt die Laufzeit mit einer Diagnostikfunktion aufgezeichnet und dem Anwender der Konsolenanwendung angezeigt.

Die darauffolgende Entschlüsselung und Dekodierung der Ergebnisse wird mit dem zuvor generierten privaten Schlüssel und dem Kodierer durchgeführt. Diese Ergebnisse werden daraufhin dem Anwender angezeigt. In den nächsten Schritten folgt eine wiederholte

Durchführung der Berechnungen, nur auf die unverschlüsselten Rohdaten. Daraufhin werden die Ergebnisse angezeigt und die Differenzen angegeben.

6.2.2 Besondere Anmerkungen

In diesem Beispiel wurde nicht von dem speziellen Stapelverfahren Gebrauch gemacht (siehe Szenario 3), welches eine Verarbeitung von Datensätzen in Vektoren effizienter machen würde. (Microsoft Research, 2018)

Diese Entscheidung lässt sich aber auch auf den Aufbau des Szenarios zurückführen. Da die einzelnen Datensätze von unterschiedlichen Fahrzeugen stammen und noch vor einer Datenübertragung verschlüsselt werden sollen, kann eine gleichzeitige Verschlüsselung von einem größeren Datenvektor nicht im Sinne einer Bestätigung des Konzepts des Szenarios sein.

Eine Relinearisierung war aufgrund der fehlenden Multiplikation von zwei unterschiedlichen Schlüsseltexten nicht notwendig. Dies hat positive Auswirkungen auf die Laufzeit des Programms. (Microsoft Research, 2018)

6.3 Implementierung Szenario 2 - Pharma-Cloud

Dieses Kapitel beschreibt den Programmablauf der Implementierung zur Realisierung des Szenarios Pharma-Cloud. Dabei werden auch zusätzliche Anmerkungen, welche spezielle Überlegungen im Quelltext kommentieren, dargebracht.

In diesem Szenario steht, im Gegensatz zu den bisherigen Szenarien, nicht die genaue Analyse der einzelnen Funktions- und Parameterauswirkungen im Vordergrund, sondern die Implementierung der, im Vergleich zu den bisherigen Szenarien, komplexeren mathematischen Operationen.

Hierbei wurden zwei unterschiedliche Varianten programmiert, um die Ergebnisse der Implementierungen überprüfen zu können:

1. Variante mit vertraulicher Berechnung
2. Variante mit nicht-vertraulicher Berechnung

Dabei wurde darauf geachtet, dass die zweite Variante denselben mathematischen Ablauf einhält wie die vertrauliche Variante. Es muss somit angemerkt werden, dass dafür nicht im Fokus stand das effizienteste Verfahren zur Lösung der linearen Regression für unverschlüsselte Daten auszuwählen.

6.3.1 Ablaufbeschreibung

Zusammengefasst sollen im Rahmen der Programmausführung folgende Schritte vollzogen werden:

1. Importieren der Datensätze aus der Datenquelle
2. Parametrisierung des SEAL-Kontextes und der notwendigen Schlüssel
3. Kodierung und Verschlüsselung der Daten
4. Initialisierung der Datenstruktur als Matrix-Objekt
5. Durchführung der Berechnungen zur linearen Regression
6. Entschlüsselung und Dekodierung der Ergebnismatrix

Im ersten Schritt werden die Datensätze in der vom Benutzer der Konsolenanwendung gewünschten Anzahl importiert. Die Anforderungen und deren Grundlage wurden im Kapitel 5.3.4 näher vorgestellt.

Dann wird ein SEAL-Kontext mit den notwendigen Schlüsseln erzeugt. In diesem Schritt ist es wichtig den Polynom-Modulus-Wert ausreichend groß zu wählen. Es wurden für unterschiedliche Datenmengen leistungseffiziente Parameter gewählt. Die Auswirkungen dieser Wahl, welche in

den Auswertungen im Kapitel 7.2.1 noch näher betrachtet werden, bestimmten maßgeblich die Qualität des Ergebnisses und die Laufzeit dieser Applikation. Standardmäßig wurde das Polynom $1\,x^{8192} + 1$ übergeben, welches für eine Datenmenge von bis zu 100 Datensätze ausreichend für annähernde Genauigkeit dimensioniert ist.

Auch hier wurden mittels eines Schlüsselgenerator-Objekts ein öffentlicher und ein privater Schlüssel zur Ver- und Entschlüsselung der Klartextobjekte erzeugt.

Abweichend zu den bisherigen Beispielen ist zudem der notwendige Einsatz eines Evaluierungsschlüssels zur Relinearisierung. Die theoretische Bedeutung und die Notwendigkeit einer Relinearisierung wurde in den Kapiteln 6.1.2 und 6.1.3 bereits näher vorgestellt.

Der Klartext-Modulus wurde auch in diesem Beispiel wieder als kritischer Faktor identifiziert. Eine ausreichend große Dimensionierung ist maßgeblich für die Korrektheit der Endergebnisse. Dieser wurde wieder in Zweierpotenzen angegeben und wurde mit einem Wert zwischen 2^20 bis 2^30 bestimmt.

Im dritten Schritt folgt nun die Initialisierung von Schlüsseltext-Objekten mittels der zuvor bestimmten Parameter. Die zuvor eingelesenen Daten werden kodiert und verschlüsselt. Dieser Vorgang kann sehr leistungsintensiv sein, deshalb wird dessen verbrauchte Zeit in Millisekunden von einer Diagnostikfunktion erfasst.

Im den folgenden Schritten wird auf die Methoden aus der Klasse Matrix zu gegriffen, welche zur Bereitstellung der Matrizenmanipulation und anderer Methoden implementiert wurde. Die Struktur und der Aufbau dieser Klasse wurde aufgrund ihrer übergeordneten Nutzbarkeit im Kapitel 6.1.3 schon näher erklärt. Diese Ablaufbeschreibung befasst sich nur noch mit der Verwendung der jeweiligen Methoden im Kontext der linearen Regression.

Die lineare Regression wird nach der sogenannten Methode der kleinsten Quadrate in Verbindung mit einer Implikation durch die Cramersche Regel durchgeführt. Weitere Anmerkungen und die Beweggründe zur Auswahl des Verfahrens und die mathematische Aufschlüsselung der Vorgehensweise werden im Folgekapitel 6.3.2 festgehalten.

Folgende Operationen sind für die hierbei angewandte Methode zur linearen Regression notwendig:

1. Transponieren der Koeffizientenmatrix
2. Multiplizieren von Matrizen
3. Berechnung der Adjunkten Matrix
4. Berechnung der Determinante von Matrizen
5. Berechnung der finalen Ergebnismatrix

Um eine lineare Regression mit homomorph verschlüsselten Daten umzusetzen gilt es die zuvor in Matrix-Objekte umgewandelten Eingangsdaten zu verarbeiten. Hierbei wird ein Matrix-Objekt für die Koeffizientenmatrix (X), sowie für den Vektor der abhängigen Werte (y) erzeugt.

Die unabhängigen Werte der einzelnen Parameter werden im ersten Teilschritt transponiert. Diese Funktion ist aufgrund ihrer Einfachheit nicht sehr laufzeitintensiv und wird deshalb nicht näher untersucht.

Danach müssen mehrere Multiplikationen von Matrizen unternommen werden. Zuerst gilt es, die zuvor transponierte Matrix mit dem Vektor y zu multiplizieren (A), danach wird dieselbe Matrix mit der ursprünglichen Matrix X multipliziert (B).

Im darauffolgenden Schritt wird die adjunkte Matrix der neu berechneten Matrix B ermittelt.

Zuletzt werden die zwei Ausgabewerte d und M berechnet. Die Berechnung der Determinante von B ergibt den Wert von d und die Matrix M wird durch die Multiplikation von A mit der adjunkten Matrix von B ermittelt.

Diese Ergebnisse werden im sechsten Schritt des Programmcodes entschlüsselt und dekodiert. Schlussendlich werden diese noch der Benutzerin oder dem Benutzer der Konsolenanwendung angezeigt.

6.3.2 Besondere Anmerkungen

Es gibt einige Rahmenbedingungen bei der Auswahl der möglichen Verfahren, welche man bei der Anwendung von homomorpher Verschlüsselung betrachten muss. Dies lässt sich unter anderem auf die Einschränkung der fehlenden Vergleichbarkeit von Schlüsseltexten zurückverfolgen. (Du et al., 2017)

Eine lineare Regression kann grundsätzlich mit der Methode der kleinsten Quadrate gelöst werden. Die Berechnung nach der folgenden Formel, welche das Minimierungsproblem der kleinsten Quadrate löst, stellt also das Ziel dieser Implementierung dar. Hierbei sind X die Koeffizientenmatrix und der Vektor y die vorhandenen Ergebnisse (im englischsprachigen Raum nennt man diese „labels“). (Björck, 1996)

$$\alpha = (X^T * X)^{-1} X^T y$$

Die Einschränkungen für die Durchführung einer linearen Regression mit verschlüsselten Daten haben viele häufig genutzte Varianten zur Lösung der Formel ausgeschlossen. Grundsätzlich sind unter anderem folgende Methoden trotzdem bei vollständiger homomorpher Verschlüsselung möglich: (Du et al., 2017)

1. Cholesky Zerlegung
2. Divisionsfreie Matrixinversion
3. Gradientenverfahren
4. Cramersche Regel

Die Cholesky Zerlegung kann aufgrund der aktuell fehlenden Fähigkeit zur Division in SEAL noch nicht implementiert werden. Außerdem wäre eine Implementierung der Cholesky Zerlegung nur anwendbar auf symmetrische Matrizen. Die divisionsfreie Matrixinversion wurde ebenfalls als ungeeignet erkannt, da sie eine gewisse Voraussicht über den zukünftigen Eigenwert der Matrix voraussetzt. (Lu, Kawasaki, & Sakuma, 2017) Diese Einschränkung ist aber bei praxisorientierten Szenarien nur stark eingeschränkt umsetzbar. (Lu et al., 2017) Das Gradientenverfahren kann sich vor allem bei einer dünnbesetzten Matrix als besonders geeignet herausstellen. Doch das Gradientenverfahren liefert keine eindeutige Lösung, sondern nur eine Lösung mit einem gewissen Fehlergrad. (Du et al., 2017)

Die Cramersche Regel ist eine exakte Methode um ein lineares Gleichungssystem zu lösen und bietet für niedrige Dimensionen direkte Methoden und für größere Dimensionen rekursive Methoden zur Berechnung der linearen Regression. Da die Berechnung der Inversen einer Matrix A, als Folgerung der Cramerschen Regel mit der folgenden Formel funktioniert, ermöglicht dies die Lösung der inversen Matrix innerhalb der Methode der kleinsten Quadrate: (Du et al., 2017)

$$A^{-1} = \frac{1}{\det(A)} * adj(A)$$

Der Verzicht auf Vergleichsoperationen und Wurzelberechnungen ermöglicht die Einhaltung der Einschränkungen der SEAL-Bibliothek. Wie auch bei der wissenschaftlichen Arbeit „Implementing ML Algorithms for HE-Encryption“ von Du et al. wurde deshalb die Entscheidung für den Einsatz der Cramerschen Regel getroffen. (Du et al., 2017)

Für die Implementierung wurde nun in die Formel zur Methode der kleinsten Quadrate die Berechnung der inversen Matrix nach der Cramerschen Regel eingesetzt. Dadurch ergibt sich folgende Formel, bei der das Produkt der Koeffizientenmatrix mit der transponierten Koeffizientenmatrix invertiert wird: (Du et al., 2017)

$$\alpha = \frac{1}{\det(X^T * X)} * adj(X^T * X)\ X^T\ y$$

Da schlussendlich die Division nicht möglich ist, werden – wie bei der dafür zugrundeliegenden Veröffentlichung von Du et.al. – zwei Werte ausgegeben, welche eine einfache und effiziente Nutzung der Resultate ermöglichen. (Du et al., 2017)

$$d = \det(X^T * X) \text{ und } M = adj(X^T * X) ** X^T * y$$

Um nun damit ein lineares Gleichungssystem zu lösen, müsste ein neuer (unverschlüsselter) Vektor v mit Koeffizienten in die folgende Formel eingefügt werden. Dabei werden die Ausgabewerte d und M noch entschlüsselt und mit dem neuen Koeffizienten multipliziert. (Aono, Hayashi, Phong, & Wang, 2015; Du et al., 2017)

$$y_v = dec(\tfrac{1}{d}) * dec(M) * v$$

6.4 Implementierung Szenario 3 – Energie-Cloud

In diesem Kapitel wird die Umsetzung des dritten Szenarios mittels der Verschlüsselungsbibliothek SEAL näher beschrieben. Es wurde innerhalb dieses Szenarios besonders Wert auf die leistungsarme Durchführung von simplen mathematischen Operationen, welche gleichzeitig auf viele Datensätze angewendet werden, gelegt. Dabei wird eine Methodik zur effizienten Stapelverarbeitung von SEAL 2.3.1. angewandt. Die Eingrenzungen aus dem Kapitel 6.1.4 sind dabei zu berücksichtigen.

Um die Durchführung der vertraulichen Berechnung mit den verschlüsselten Daten vergleichen zu können, werden folgende Varianten der Implementierung erstellt:

1. Variante mit vertraulicher Berechnung (mit CRT-Batching)
2. Variante mit vertraulicher Berechnung (ohne CRT-Batching)
3. Variante mit nicht-vertraulicher Berechnung

6.4.1 Ablaufbeschreibung

Die Abhandlung der dem Szenario nachempfundenen Problemstellung in der ersten Variante erfolgt in den folgenden Schritten:

1. Importieren der Datensätze aus der Datenquelle
2. Parametrisierung des SEAL-Kontextes und der notwendigen Schlüssel
3. Kodierung und Verschlüsselung der Daten mittels CRT-Batching
4. Durchführung von mathematischen Basisoperationen von verschiedenen verschlüsselten und unverschlüsselten Vektoren
5. Entschlüsselung und Dekodierung des Resultatvektors
6. Export des Ergebnisvektors

Innerhalb des ersten Schritts erfolgt das Einlesen einer von der Anwenderin oder des Anwenders gewünschten Anzahl von Datensätzen. Dabei ist zu bedenken, dass die Anzahl der Datensätze maßgeblich für die Auswahl der Parameter von SEAL ist, da es sonst zu fehlerhaften Resultaten kommt. Der Datensatz wurde im Kapitel Datengrundlage 5.4.4 näher erläutert.

Der zweite Schritt betrachtet die zuvor angesprochene Parameterauswahl. Dabei gilt es bei der Anwendung des Stapelverfahrens zur effizienten Datenverarbeitung einiges anzumerken. Das Kontext-Objekt von SEAL und die gewählten Parameter können auch geprüft werden, ob die Stapelverarbeitungstechnik eingesetzt werden kann. Nähere Informationen über die Voraussetzungen zur Durchführung werden im Kapitel 6.4.2. erläutert. Es wurden bei der Entwicklung starke Unterschiede in der Laufzeit festgestellt, welche im Kapitel 7.3.1 bei den Auswertungen detaillierter analysiert werden.

Die Kodierung und Verschlüsselung der einzelnen Datenvektoren – in diesem Fall die Wattstunden für jede einzelne Minute der drei unterschiedlichen Stromzähler – erfolgt mittels

„PolyCRTBuilder“-Objekts und nicht wie in den anderen Szenarien mittels eines Kodierers für gebrochene Zahlen. Zusätzlich wird auch ein Vektor mit stundengenauen Preisen eingelesen, welcher aber nicht verschlüsselt, sondern nur kodiert wird.

Die Addition jedes einzelnen Verbrauchswerts einer Minute erfolgt für die gesamten Vektoren in einzelnen Schritten. Daraufhin wird eine Multiplikation mit einem Klartextvektor durchgeführt. Dies erscheint im Sinne des Szenarios, da das Energieunternehmen die Preise selbst kodiert und diese mit den für sie nicht lesbaren Schlüsseltextvektoren multipliziert um damit einen stundengenauen Preis für den Verbrauch einzelner Stromzähler, oder auch für einen gesamten Haushalt, zu errechnen. Eine Relinearisierung der Ergebnisse mittels Evaluierungsvektor ist hierbei nach einigen Kontrolldurchläufen nicht notwendig gewesen. (Chen et al., 2018)

Schlussendlich wird der Vektor mit den Resultaten der Berechnungen entschlüsselt und dekodiert. Die Ergebnisse werden zu Analysezwecken in einer Datei abgespeichert. Im Rahmen des Szenarios würde dieser verschlüsselte Ergebnisvektor nur mittels des privaten Schlüssels der Kundin oder des Kunden über einen eigenen Client (z.B. eine mobile Applikation) entschlüsselt werden.

6.4.2 Besondere Anmerkungen

Die zuvor schon angesprochene Technik der Stapelverarbeitung von Daten, welche in vektorähnlichen Strukturen kodiert werden können, wurde in den letzten Versionen von SEAL etwas verändert. Grundsätzlich bietet sie die Möglichkeit einfache arithmetische Operationen auf Vektoren mit viel höherer Effizienz durchzuführen, als dies auf einzelne Schlüsseltexte möglich wäre. Dabei gilt es zu berücksichtigen, dass schon bei der ersten Kodierung und Verschlüsselung der Daten eines Vektors ein spezielles Kodierungsverfahren eingesetzt werden muss. (Chen et al., 2018)

Vereinfacht gesagt, wird somit aus einer gewissen Anzahl von Zahlen, anstatt individueller kodierter Klartextpolynome, ein gemeinsames Klartextobjekt. Ein Oberbegriff für eine solche Methodik ist SIMD (engl. Single Instruction, Multiple Data). (Chen et al., 2018)

Die Möglichkeit, diese Technik zu nutzen, ist bei SEAL aktiv, wenn der Klartext-Modulus eine Primzahl ist und dem Wert des zehnfachen Polynom-Modulus modulo 1 entspricht. Ansonsten würde das Kodieren der Vektoren mittels PolyCRTBuilder-Objekts fehlschlagen. (Chen et al., 2018)

Diese Kodierung funktioniert aufgrund der Theorie des sogenannten Chinesischen Restsatzes (engl. Chinese Remainder Theorem – CRT), welche Aussagen über die simultane Kongruenz von Zahlen über mehrere Ringsysteme trifft. (Chen et al., 2018)

Im Übrigen ermöglicht die SEAL-Bibliothek auch andere Operationen, wie etwa Rotationen, auf den Vektor anzuwenden. Dabei ist, aber die vorherige Erstellung eines Galois-Schlüssels notwendig. Dieser ähnelt – vereinfacht ausgedrückt – von der Funktionsweise einem Evaluierungsschlüssel, welcher bei Multiplikationen von Schlüsseltexten vonnöten ist. (Chen et al., 2018)

7. AUSWERTUNG DER IMPLEMENTIERUNGEN

Dieses Kapitel behandelt die Auswertungen der einzelnen Implementierungen. Dabei steht die Beantwortung der Forschungsfrage im Mittelpunkt dieser Analysen. Die Hypothesenüberprüfung stellt hierbei auf formalen Weg eine konkrete und eindeutig richtungsweisende Erkenntnisgewinnung dar. Doch zusätzlich zu den klar formulierten Hypothesen, benötigt es noch weitere Erkenntnisse, welche sich oft nur schwer auf eindeutigem Weg erschließen lassen. Es ist also eine besondere Herausforderung diese Analysen richtig zu interpretieren und im Fazit im Kontext ihrer Rahmenbedingungen auf eine allgemeine Praktikabilität zu übertragen. Auch in anderen Forschungsprojekten sind sich Wissenschaftler über den Spagat zwischen theoretischer und praktischer Anwendbarkeit bewusst. (Varia et al., 2015)

Grundsätzlich gibt es mehrere Faktoren, welche einen starken Einfluss auf die folgenden Ergebnisse haben. Einerseits ist die Parameterwahl, wie sich während der Entwicklung schon gezeigt hat, ein nicht zu unterschätzender Faktor, welcher bei jeder Aussage berücksichtigt werden muss. Andererseits spielt die zu verarbeitende Datenmenge hierbei auch eine große Rolle. Besonders interessant dabei ist die unterschiedliche Auswirkung dieser zwei Freiheitsgrade auf die jeweiligen Szenarien. Im Übrigen kann die Anzahl der durchzuführenden Berechnungen und die mathematische Komplexität der Szenarien eine starke Ursache für unterschiedliche Auswirkungen auf die zu untersuchenden Merkmale darstellen. (Chen et al., 2018)

Dabei wurde bei der Erstellung der Szenarien schon auf eine große Bandbreite von Unterschieden hinsichtlich der Art der Berechnung, der Datengrundlage und den angewendeten Hilfstechniken Wert gelegt. Das im Kapitel 5.1.2 gezeigte Diagramm hat den jeweiligen Fokus auf die unterschiedlichen Ausprägungen besser dargestellt und die Abgrenzung, sowie die Gemeinsamkeiten bei den Analyseschwerpunkten veranschaulicht. Dieser Vorsatz bei der Erstellung der Szenarien wird sich auch auf die Auswertung dieser auswirken.

Somit wird in jedem der folgenden Unterkapitel auf spezielle Veränderungen der besagten variablen Inputgrößen geachtet, um schlussendlich ein gesamtheitliches Bild über die Praxisfähigkeit der Berechnungsvorgänge im Rahmen der Anwendungsgebiete machen zu können.

Das Testgerät stellte hierbei ein HP Elitebook 840 G2 mit 8 GB Arbeitspeicher und einem Intel Core i5-5200U Prozessor mit 2.20 GHz dar und wurde mit Windows 10 Pro betrieben. Alle Tests wurden mit diesem Gerät durchgeführt. Es wurde jeweils darauf geachtet, dass es zu keiner übermäßigen Arbeitsspeicherauslastung kommt, um eine Vergleichbarkeit über die Datenmengen zu gewährleisten. Zu den Aussagen im Zusammenhang mit der erfassten Laufzeit in Millisekunden muss angemerkt werden, dass diese oft bei mehreren Durchläufen voneinander abweichen. Es wurden bei nicht plausiblen Messwerten mehrere wiederholte Messungen vorgenommen, um ein stabiles, aussagekräftiges Ergebnis zu erhalten.

Im Rahmen der statistischen Auswertung wurde einerseits auf Diagnostikfunktionen der Entwicklungsumgebung Visual Studio 2017, auf aufrufbare Funktionen aus den

Diagnostikbibliotheken von .NET und auf eigene manuelle Analysen der ausgegebenen Ergebnisse der Ausführungen zurückgegriffen. (Visual Studio 2017, 2018)

Insgesamt haben die Ausführungen der Versuche mehrere Stunden an reiner Laufzeit benötigt. Um die Vielzahl an Ergebnissen aus den Auswertungen besser verständlich zu machen, werden die Ergebnisse jedes einzelnen Szenarios schlussendlich noch kurz zusammengefasst und in Bezug auf ihre Praxisfähigkeit evaluiert.

7.1 Auswertung Szenario Emissions-Cloud

Im Zuge dieses Teilkapitels werden die Ergebnisse der Diagnostik über die Laufzeit, Speicherverbrauch und andere Merkmale im Zusammenhang mit der Ausführung der Implementierung für das Szenario Emissions-Cloud aufgezeigt und analysiert. Neben der reinen Analyse zur Beantwortung der zuvor aufgestellten Hypothesen, werden Evaluierungen durchgeführt um weitere Aspekte der Forschungsfrage zu beantworten.

7.1.1 Überprüfung der Hypothesen

Die zuvor aufgestellten Hypothesen werden nun überprüft. Dies geschieht durch unterschiedliche Erkenntnisse, welche durch Ergebnisse oder Statistiken belegt werden.

> *H1.1: Auf die Qualität des Resultates der Berechnungen im Kontext des Szenarios „Emissions-Cloud“ hat die homomorphe Verschlüsselung keinen Einfluss.*

Hierbei muss man die Parameterwahl genauer betrachten. Ein annähernd gleiches Ergebnis kann aber bei der Verwendung von unterschiedlichen Parametern sichergestellt werden. Dazu wurde der Klartext-Modulus-Parameter von 2.048 (2^11) sukzessive in 2er-Potenzschritten auf 262.144 (2^18) erhöht. Exemplarisch wurden in den folgenden Tabellen nur die Ergebnisse unter Berücksichtigung eines Parameterwertes von 2.048 respektive von 262.144 dargestellt. In den elektronisch beigefügten Materialien zu dieser Masterarbeit finden sich noch weitere Ergebnisse für Werte, welche zwischen den zwei vorgestellten Werten für den Klartext-Modulus liegen.

In der folgenden Tabelle werden die entsprechenden Ergebnisse der drei Endresultate dargestellt, welche jeweils mit homomorph verschlüsselten Datensätzen (mit HE in der Tabelle bezeichnet), sowie mit nicht verschlüsselten Datensätzen (mit NHE bezeichnet) berechnet wurden. Hierbei wurde eine aufsteigende Anzahl von Datensätzen benutzt, welche den Verlauf der Unschärfe und der Abweichungsausprägung anzeigen soll. Es gilt anzumerken, dass die Datensatzanzahl in keinem gleichmäßigen Verhältnis ansteigt.

PlainModulus 2^11 Datensätze - HE		NOx Durchschnitt [mg/km]	CO2 Durchschnitt [g/km]	Emissions-Punkte Durchschnitt [Punkte]
10	HE	115,308	145,291	35,665
	NHE	115,308	145,291	35,665
100	HE	122,837999999997	146,292099999997	34,4020199999902
	NHE	122,838	146,2921	34,40202
250	HE	117,409959999994	148,654999999993	34,0653130457467
	NHE	117,40996	148,655	34,9011726666667
750	HE	119,708586666677	143,32287589698	4,01922644861897
	NHE	119,708586666667	149,565826666666	34,4349351111111
1.000	HE	121,70602000001	327,597369019886	72,067605626583
	NHE	121,70602	150,60971	33,9961356666667
5.000	HE	16,731644587344	-25,0177744659912	10,5550866605691
	NHE	116,264444	143,396966	35,7740712666667
10.000	HE	-8,10252108961499	22,1760516625545	9,49797255964525
	NHE	116,448347	138,20904	36,4382092833333

Tabelle 17: Resultate – PlainModulus 2^11 – Szenario 1
In Anlehnung an: (Visual Studio 2017, 2018)

Es gilt nun zu untersuchen, ob und wie stark diese Abweichungen sind, um eine Aussage auf den Qualitätseinfluss des Ergebnisses tätigen zu können. Gut zu erkennen ist dabei, dass grundsätzlich vier Kategorien von unterschiedlichen Abweichungen existieren:

1. Keine Abweichungen (siehe Tabelle 13 und 14: 10 Datensätze)
2. Geringfügige Abweichungen (siehe Tabelle 13: 100 und 250 Datensätze; siehe Tabelle 14: 750, 1.000, 5.000, 10.000 Datensätze)
3. Starke Abweichungen (siehe Tabelle 13: ab 750 Datensätzen)

Die erste Phase, bei der keine Abweichungen vorkommen, ist unter den Rahmenbedingungen nur bei sehr geringen Datenmengen aufgetreten. Bei richtiger Parameterwahl lässt sich zumindest die zweite Kategorie – also nur geringfügige Abweichungen – realisieren. Dies kann durchaus Auswirkungen auf die Praktikabilität besitzen.

Starke Abweichungen deuten auf andere Ursachen hin. Hierbei galt es in den Versuchen vor allem die Parameter des SEAL-Kontextes abzuändern. Wenn man den Verlauf solch starker

Abweichungen näher betrachtet, dann erkennt man bei Berechnungen mit niedriger Komplexität – wie in diesem Szenario – die problematischen Funktionen, welche die vorher definierten Wertebereiche eventuell überschreiten lassen. In diesem Fall dürfte dies die Aggregationsfunktion des SEAL-„Evaluator“-Objekts sein, welche zuerst die Berechnung der Emissionspunkte, dann der Kohlendioxidwerte und zuletzt der Stickoxidwerte beeinflusst.

Ergebnisse bei einem Klartext-Modulus-Parameter von 262.144 (2^18):

PlainModulus 2^18 *Datensätze - HE*		NOx Durchschnitt [mg/km]	CO2 Durchschnitt [g/km]	Emissions-Punkte Durchschnitt [Punkte]
10	HE	115,308	145,291	35,665
	NHE	115,308	145,291	35,665
750	HE	119,708586666677	149,56582666668	34,4349351111151
	NHE	119,708586666667	149,565826666666	34,4349351111111
1.000	HE	121,70602000001	150,609710000013	33,9961356666703
	NHE	121,70602	150,60971	33,9961356666667
5.000	HE	116,264443999884	143,396965999858	35,774071266653
	NHE	116,264444	143,396966	35,7740712666667
10.000	HE	116,448346999884	138,209039999863	36,4382092832812
	NHE	116,448347	138,20904	36,4382092833333

Tabelle 18: Resultate – PlainModulus 2^18 – Szenario 1
In Anlehnung an: (Visual Studio 2017, 2018)

Als endgültige Antwort auf die Hypothese kann man nur eine eingeschränkte Bestätigung der aufgestellten Hypothese eindeutig verifizieren. Besonders bei niedriger Datensatzanzahl kann eine vollständige Gleichheit erreicht werden, doch selbst bei Versuchsvorgängen mit hohem Parameterwert für den Klartext-Modulus gibt es eine minimale Abweichung. Diese Unschärfe kann je nach Kontext keine, oder auch eine sehr starke Auswirkung auf die tatsächliche Qualität der Berechnung haben. (Armknecht et al., 2015)

> *H1.2: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Emissions-Cloud“ einen höheren Speicherverbrauch.*

Zur Beantwortung der Frage nach den Speicherverbrauch wurde das vollständige Szenario mit bis zu 10.000 Datensätzen und einem gleichbleibendem Klartext-Modulus von 2^30 (Polynom-

Modulus 2^11) berechnet und dessen Speicherverbrauch aufgezeichnet. Hierbei kann sich die Hypothese in erster Linie klar bestätigen lassen. Insbesondere von Interesse ist aber auch das Verhalten des Speicherverbrauchs bei unterschiedlicher Parameterwahl und der Anzahl der Datensätze.

Anzumerken gilt es hierbei, dass darauf geachtet wurde, dass der Arbeitsspeicher des Testgerätes nicht vollständig ausgenutzt wurde, um zu verhindern, dass diese Limitierung einen Einfluss auf die Erkenntnisse hat. In der folgenden Grafik wird die unterschiedliche Entwicklung des Speicherverbrauchs bei homomorpher Verschlüsselung, im Vergleich zur Entwicklung ohne Verschlüsselung, dargestellt. Dabei ist zu beachten, dass die Y-Achse die Wertebereiche in MB („Megabyte“) pro zu berechnenden Datensatz anzeigt (also MB/Datensatz).

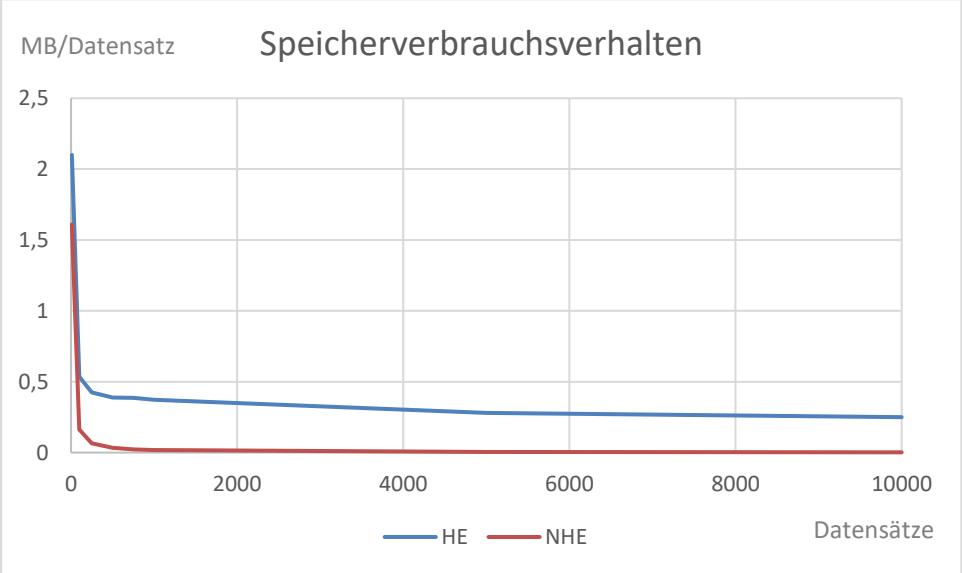

Abbildung 8: Speicherverbrauchsverhalten – Szenario 1
In Anlehnung an: (Visual Studio 2017, 2018)

Hierbei ist interessant, dass die homomorphe Verschlüsselung sich grundsätzlich ähnlich wie die Variante ohne vertrauliche Berechnung entwickelt. Auch im Zusammenhang mit der grundsätzlich höheren relativen Speicherbelegung bei vertraulicher Berechnung, lassen die Verlaufslinien auf ein ähnliches Verhalten bei vertraulicher Berechnung hindeuten.

> *H1.3: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Emissions-Cloud“ einen höheren Leistungsverbrauch.*

Die dritte Hypothese zu diesem Szenario behandelt den Leistungsverbrauch. Hierbei lässt sich die Hypothese unter den Rahmenbedingungen dieser Versuchsumgebung bestätigen. Neben dem absoluten Mehrverbrauch ist auch die Relation zu den Parametern und der Datensatzanzahl für eine Aussage zur endgültigen Anwendbarkeit in der Praxis von Relevanz.

Der Leistungsverbrauch wurde auf mehreren Arten gemessen. Unter anderem wurde, um eine eindeutige Zuweisung der einzelnen Phasen (siehe Implementierung des Szenarios in Kapitel

6.2.1) zu gewährleisten, eine interne Diagnostikfunktion zum Aufzeichnen der verbrauchten Millisekunden benutzt. In der folgenden Tabelle wird die Berechnung unter Einsatz von homomorpher Verschlüsselung mit der Berechnung ohne homomorphe Verschlüsselung gegenübergestellt. Hierbei lässt sich gut erkennen, dass die homomorphe Verschlüsselung einen sehr viel höheren Leistungsverbrauch besitzt. Der Versuch wurde hierbei mit den Werten 2.048 für PolyModulus und 2^30 für den PlainModulus durchgeführt.

Datensätze	HE [ms]	NHE [ms]
10	1.743	7
100	6.004	9
500	27.759	14
1.000	53.523	16
5.000	270.837	19
10.000	533.629	39

Tabelle 19: Leistungsverbrauch Gesamtvergleich – Szenario 1
In Anlehnung an: (Visual Studio 2017, 2018)

Wenn man den Parameter „PolyModulus“ näher betrachtet, welcher auch durchaus eine wichtige Rolle für die Beantwortung der Praxisfähigkeit der Berechnung unter homomorpher Verschlüsselung spielt, dann zeigt sich ein eindeutiges Bild. Eine gut überlegte Wahl des Parameters wirkt sich anscheinend sehr stark auf das Laufzeitverhalten aus. Die folgende Tabelle zeigt exemplarisch zwei aufsteigende Wertebereiche für den Polynom-Modulus (und dem Koeffzienten-Modulus analog dazu, siehe dazu Kapitel 6.1.2). Hierbei ist besonders interessant, dass sich die falsche Parameterwahl auch im Verhältnis zum Anstieg der Datenmenge überproportional zur Laufzeit auswirkt.

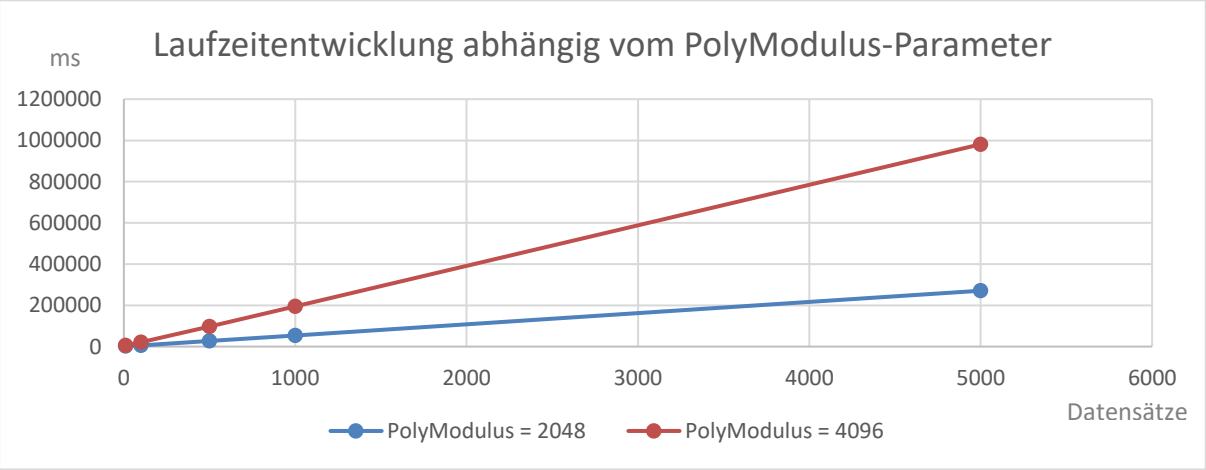

Abbildung 9: Laufzeitentwicklung abhängig vom PolyModulus-Parameter – Szenario 1
In Anlehnung an: (Visual Studio 2017, 2018)

Um mehr Details über den tatsächlichen Leistungsverbrauch zu erhalten, kann man die Auswirkungen der einzelnen Methodenaufrufe des Programmcodes näher betrachten. Dies wurde mithilfe einer Diagnostikfunktion von Visual Studio 2017 bewerkstelligt und gibt Einblicke in die Unterschiede des Anteils am absoluten Leistungsverhalten. (Visual Studio 2017, 2018)

In der nachfolgenden Grafik lässt sich gut erkennen, dass einige wenige Methoden einen hohen Anteil an der absoluten CPU-Auslastung besitzen. Im Vordergrund ist die Verschlüsselungsmethode der SEAL-Bibliothek, gefolgt von einer Hilfsmethode zur Multiplikation von Schlüsseltexten mit Klartexten und der Entschlüsselungsfunktion. Besonders der hohe Anteil der Verschlüsselungsmethode an der absoluten Laufzeit kann eine sinnvolle Erkenntnis für die praxisnahe Umsetzung dieses Szenarios sein. Im Vergleich dazu sieht man auch die sehr geringe Beanspruchung der Recheneinheit durch Methoden wie „Add“ der Klasse „Collection“. Bei der folgenden Grafik ist besonders auf die logarithmische Darstellung der Y-Achse zu achten. Diese führt zu einem viel schwächer wirkenden Unterschied zwischen höchster und niedrigster CPU-Auslastung, aber ermöglicht den besseren Vergleich innerhalb der niedrigen und hohen Wertebereiche. (Varia et al., 2015)

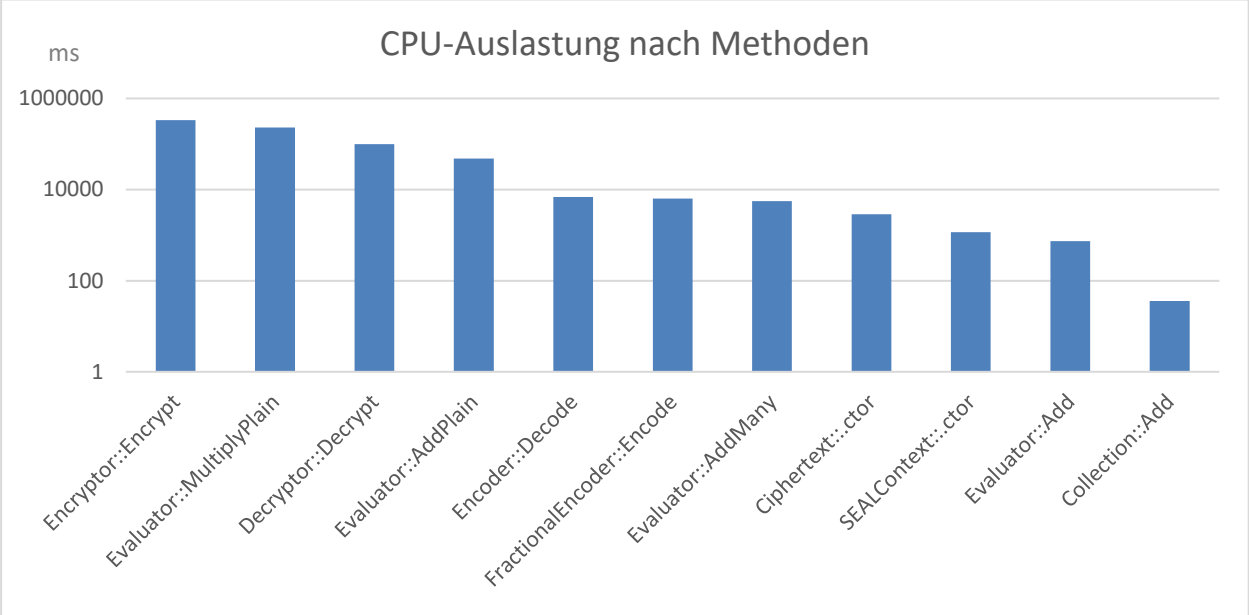

Abbildung 10: CPU Auslastung Methodenaufteilung – Szenario 1
In Anlehnung an: (Visual Studio 2017, 2018)

Unabhängig von den exakten Methoden wurde die Laufzeitbetrachtung für eine unterschiedliche Anzahl von Datensätzen auch für einzelne Berechnungsabschnitte (siehe Kapitel 6.2.1 Implementierung) durchgeführt. In der folgenden Grafik werden die Datensätze auf der X-Achse und die Laufzeit auf der Y-Achse dargestellt. Hierbei lässt sich gut erkennen, dass Abschnitt 3 (Kodierung und Verschlüsselung der Daten) und Abschnitt 5 (Berechnung der Emissionskennzahlen und Berechnung des gewichteten Durchschnitts) ähnlich hohe absolute Laufzeitanteile besitzen. Die Laufzeit des Abschnitts 4 (Aggregation der Datensätze und Berechnung der durchschnittlichen Stickoxid- und Kohlendioxidwerte) hingegen verhält sich im

Verlauf des Anstiegs der Datensatzanzahl eher wie Abschnitt 7 (Durchführung aller Berechnungsschritte an unverschlüsselten Daten). Dies deutet darauf hin, dass die Methoden in Abschnitt 5 und 3 eindeutig zu vermeiden sind, wogegen sich die Methoden des Abschnittes 4 auch für hohe Datenmengen als praktikabel erweisen. Diese Auffälligkeit ist besonders interessant, da man vor allem einen Unterschied in den multiplikativ aufwendigen Abschnitten erkennt. In Abschnitt 5 werden zum Beispiel die Multiplikationen für die lineare Skalierung auf jeden einzelnen Messwert durchgeführt.

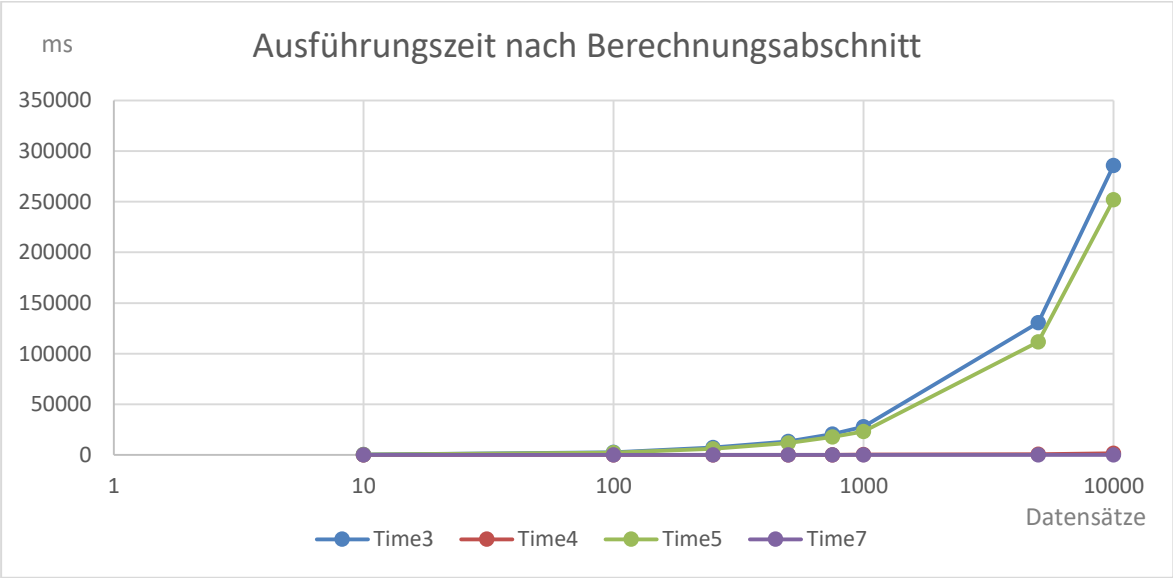

Abbildung 11: Laufzeit nach Berechnungsabschnitt – Szenario 1
In Anlehnung an: (Visual Studio 2017, 2018)

7.1.2 Evaluierung der Praxisfähigkeit

In diesem Unterkapitel werden die wichtigsten Ergebnisse der Analysen für das Fazit zusammengefasst. Ein gleiches Ergebnis im Kontext der Rahmenbedingungen des Szenarios lässt sich nur mit korrekten Parametern erreichen. Es hat sich dabei eindeutig feststellen lassen, dass die Variante mit homomorpher Verschlüsselung einen höheren Leistungsverbrauch hat, als die Rechenoperationen an unverschlüsselten Daten – selbst bei annähernd optimaler Parameterwahl.

Die Qualität des Ergebnisses hängt bei den Datenverarbeitungen des Szenarios hauptsächlich vom verwendeten Klartext-Modulus ab, welcher in Verhältnis der notwendigen Datensatzanzahl gewählt werden muss. Einen Einfluss auf die Qualität des Ergebnisses durch den Polynom-Modulus konnte man trotz Erhöhung des Parameterwertes (ausgehend von 2.048) nicht feststellen.

In Bezug auf die Veränderung der Laufzeit bei höherer Datensatzanzahl wurde festgestellt, dass diese wesentlich von der Wahl des Polynom-Modulus abhängt. Es gilt also für Anwendungen in der Praxis, welche laufzeiteffizient durchgeführt werden sollen, diesen Parameter so niedrig wie möglich zu wählen.

Eine interessante Erkenntnis ist das ähnliche Verhalten von vertraulicher und offener Berechnung in Bezug auf ihre Speicherauslastung in Verhältnis zur Veränderung der Datensatzanzahl. Dies lässt implizieren, dass bei ähnlichen Anwendungsgebieten eine Erhöhung der Speicherressourcen im annähernd gleichen Verhältnis wie unter normaler Berechnung erfolgen muss.

Im Rahmen der Untersuchung wurde für den konkreten Vorgang der Verschlüsselung, die höchste absolute Laufzeit aufgewendet. Dies lässt für die Durchführung des Szenarios in der Praxis den Schluss ziehen, dass die integrierten Recheneinheiten der einzelnen Fahrzeuge ausreichend gut ausgestattet sein müssen.

Bei der Implementierung von ähnlichen Szenarien in der Praxis muss außerdem auch sehr auf die Auswahl der Methoden geachtet werden, da die Ergebnisse der Analyse den Schluss zu lassen, dass sich ähnliche Berechnungen durch Umstrukturierung auf effizientere Weise lösen lassen. Besonders häufige Multiplikationen sollten vermieden werden (siehe Auswertungen Hypothese H1.3).

7.2 Auswertung Szenario Pharma-Cloud

Dieses Kapitel befasst sich mit der Auswertung der Ergebnisse rund um die Tests der Implementierung für das Szenario zur Pharma-Cloud. Der zu analysierende Unterschied der Erkenntnisse bezieht sich in technologischer Hinsicht auf die, im Vergleich zu den anderen Szenarien, höhere mathematische Komplexität. Besonders die multiplikative Tiefe gibt Rahmenbedingungen für die Parameterzustände vor, welche die Leistung beeinflussen sollten. (Chen et al., 2018; Du et al., 2017)

7.2.1 Überprüfung der Hypothesen

Wie in den Auswertungen zum Szenario der Emissions-Cloud werden auch in diesem Kapitel hauptsächlich drei Hypothesen beantwortet. Unterschiede in der Beantwortung gibt es aber einige, vor allem aufgrund der technischen Unterschiede.

> *H2.1: Auf die Qualität des Resultates der Berechnungen im Kontext des Szenarios „Pharma-Cloud“ hat die homomorphe Verschlüsselung keinen Einfluss.*

Die erste Hypothese behandelt die Ergebnisqualität, also die Korrektheit, im Zusammenhang mit homomorpher Verschlüsselung. Die theoretische Frage, ob es unter gewissen Rahmenbedingungen zu annähernd korrekten Ergebnissen kommen kann, kann bejaht werden. Auf die allgemeine Frage zur Übertragung der Erkenntnisse auf die Praxis gibt es keine eindeutige Antwort. Aber eine Tendenz aufgrund des Verhaltens mit unterschiedlichen Rahmenbedingungen kann gemessen werden. Dabei muss angemerkt werden, dass der

gesamte Datensatz, mit den zu Verfügung gestandenen Mitteln, nicht in innerhalb von weniger als zwei Stunden mit vermeintlich ausreichend großen Parameterset durchgeführt werden konnte. Ein annähernd korrektes Ergebnis für 100 Datensätze des Szenarios konnte bei folgenden Parametern im Kontext der Untersuchungsumgebung ermittelt werden:

- Klartext-Modulus: 2^30 (1.073.741.824)
- Polynom-Modulus: 8192
- Evaluierungsschlüssel (Anzahl in Bit): 32
- Ergebnisvariable „d“ (siehe Kapitel 6.3.2) unter Verschlüsselung: (ohne Verschlüsselung) 142.821.443.686.128,00 (142.821.443.686.048,00)
- Differenz: 80

Zur Überprüfung der Hypothesen wird in diesem Rahmen nicht mehr so detailliert auf die Unterschiede bei unterschiedlichen Klartext-Modulus eingegangen, da die Untersuchungen gezeigt haben, dass keine weiteren Erkenntnisse, zusätzlich zu denen aus der Auswertung des Szenarios Emissions-Cloud, getroffen werden können.

Im Gegensatz dazu kann hierbei aber zusätzlich auch ein starker Einfluss des Polynom-Modulus auf die Ergebnisqualität festgestellt werden. Dieser trat in diesem Umfang bei der Analyse des Emissions-Cloud Szenarios nicht auf. Dort konnte der Parameter mit 2.048 niedrig gehalten werden. Bei der Ausführung der Implementierung wurden testweise auch niedrigere Werte wie 2.048 evaluiert, doch führten diese in keinem der getesteten (ab 5 Datensätze) Versuche zu einem annähernd korrektem Ergebnis. Beispielhaft dafür ist die Berechnung mit dem nächstniedrigeren Wert (4.096) dargestellt. In der folgenden Abbildung ist das Verhalten der Ergebnisse (in diesem Fall der Variable d, siehe Kapitel 6.3.1) mit unterschiedlichen Eingangsvariablen veranschaulicht.

Ergebnisdifferenz *für „d“ (HE-NHE)*

Datensätze	`PolyModulus="1x^4096+1"`	`PolyModulus="1x^8192+1"`	`PolyModulus="1x^16384+1";`
10	-9,94E+17	-0,02	-0,02
25	-9,11E+18	0,09	0,09
50	6,21E+18	10,50	10,5
75	7,41E+17	24,60	24,60
100	8,31E+18	80	80

Tabelle 20: Ergebnisqualität abhängig von PolyModulus und Datenmenge – Szenario 2
In Anlehnung an: (Visual Studio 2017, 2018)

Ein ausreichend hoher Wert des Polynom-Modulus ist also eine Bedingung für ein annähernd korrektes Ergebnis. Anzumerken sei hierbei auch die fehlende Stabilität der Ergebnisse bei einem niedrigeren Wert (siehe Tabelle 20, Spalte für 4.096) des Polynom-Modulus, die zu unterschiedlichen Ergebnissen für dieselben Eingangswerte bei mehrmaliger Durchführung

führte. Ein höherer Wert führte aber nicht zu einem besseren Ergebnis im Kontext des Untersuchungsaufbaus.

H2.2: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Pharma-Cloud“ einen höheren Speicherverbrauch.

Der Speicherverbrauch der Implementierung mit homomorpher Verschlüsselung ist unabhängig von der Parametrierung höher als bei der Ausführung ohne vertrauliche Verarbeitung der Daten. Dies wird in der folgenden Grafik mit unterschiedlicher Datensatzanzahl näher veranschaulicht. Dabei wurde der Polynom-Modulus mit 8.192 und der Klartext-Modulus mit 2^30 festgelegt. Es ergibt sich ein eindeutiges Bild, welches trotz der geringen Datensatzanzahl (X-Achse) einen hohen Anstieg für die Variante mit vertraulicher Berechnung darstellt. Eine Verzehnfachung der Datensatzanzahl führte zu mehr als einer Verfünffachung der Speicherauslastung bei Anwendung von homomorpher Verschlüsselung. Im Gegensatz dazu verändert sich die Speicherauslastung bei der Variante ohne vertrauliche Berechnung nur äußerst geringfügig.

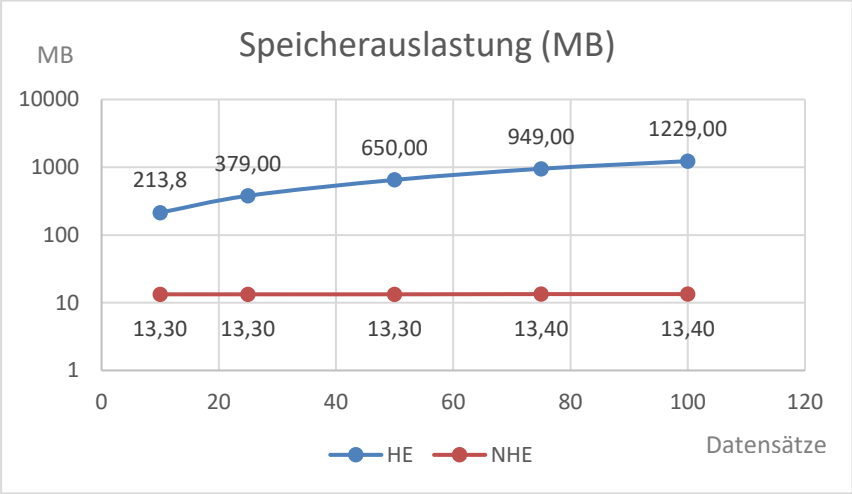

Abbildung 12: Speicherauslastung in MB Vergleich HE und keine HE – Szenario 2
In Anlehnung an: (Visual Studio 2017, 2018)

Von besonderem Interesse sind die zuvor bei der Beantwortung der Hypothese H2.1 angemerkten Qualitätssprünge im Zusammenhang mit der Laufzeit. Da dem Parameter Polynom-Modulus schon beim ersten Szenario ein großer Einfluss auf die Laufzeit nachgewiesen werden konnte, gilt es hierbei insbesondere das Verhältnis zwischen Ergebniskorrektheit und Laufzeiteffizienz zu untersuchen. Dies wird für die Hypothese H2.3 näher betrachtet.

H2.3: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Pharma-Cloud“ einen höheren Leistungsverbrauch.

Der Leistungsverbrauch ist, wie zuvor schon angekündigt, relativ stark abhängig von der Parameterwahl und das bezieht sich vor allem auf den Polynom-Modulus („PolyModulus“). In der folgenden Tabelle kann man die Entwicklung, anhand der steigenden Datensatzanzahl in Zusammenhang mit einem steigenden Parameterwert, gut erkennen.

Hierbei führt eine Verzehnfachung der Datensatzanzahl ungefähr zu einer Versiebenfachung bei niedrigeren bzw. zu ungefähr einer Verachtfachung bei höherem Wert für den Polynom-Modulus. Dies bedeutet, dass auch der relative Anstieg der Laufzeit bei höheren Parametwerwert höher ist.

Leistungsverbrauch und Speicherauslastung

Datensätze	`PolyModulus="1x^8192+1"`		`PolyModulus="1x^16384+1";`	
	CPU (ms)	Speicher (MB)	CPU (ms)	Speicher (MB)
10	238.021	213,8	1.007.185	696
25	466.036,00	379,00	2.291.001	1286
50	917.340,00	650,00	3.907.648	2601
75	1.311.214,00	949,00	6.234.079	3010
100	1.615.237,00	1229,00	8.338.921	4214

Tabelle 21: Leistungsverbrauch und Speicherauslastung – Szenario 2
In Anlehnung an: (Visual Studio 2017, 2018)

Um noch mehr Erkenntnisse über das Laufzeitverhalten der Implementierung zu gewinnen wurde eine genauere Betrachtung der Verteilung der Eigenmethoden und der externen Methoden durchgeführt. Dies wurde mit der erweiterten Diagnosefunktion von Visual Studio 2017 durchgeführt. (Visual Studio 2017, 2018)

Bei der Analyse der Eigenmethoden, steht die logische Struktur der Berechnungen (siehe Kapitel 6.3.1) im Mittelpunkt. Die Methode zum Multiplizieren verbraucht in absoluten Messwerten mehr als das 18-fache, als die Berechnung der Determinante. Dies zeigt die Bedeutung von effizienten Multiplikationen von Matrizen im Zusammenhang mit Regressionsmethoden auf. Bei der folgenden Grafik wird diese Aufteilung veranschaulicht. Zur besseren Sichtbarkeit der einzelnen Werte wurde, aufgrund der hohen Spanne des Wertebereichs, die Y-Achse zur Basis 10 logarithmiert.

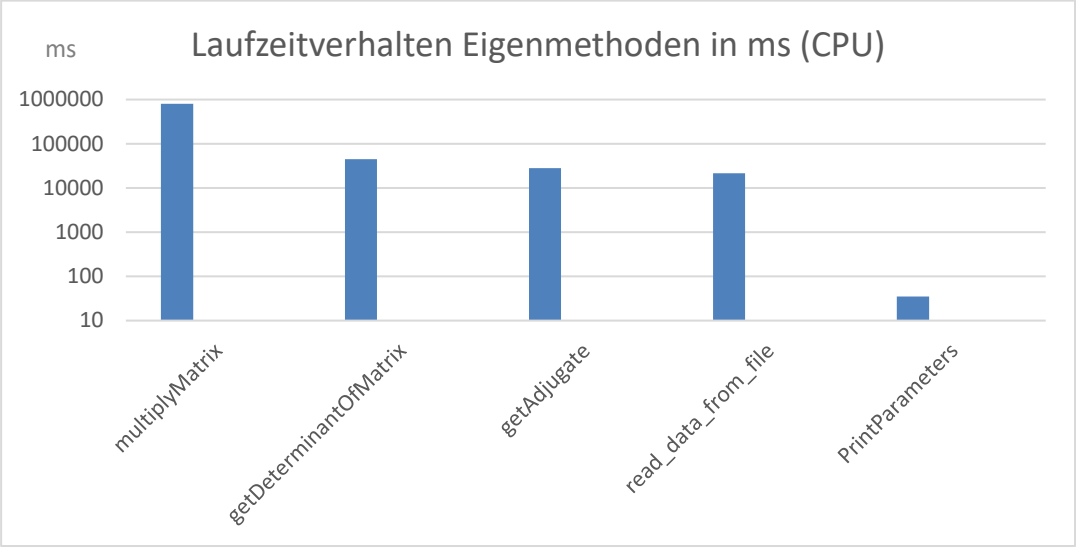

Abbildung 13: Laufzeitverhalten der Eigenmethoden – Szenario 2
In Anlehnung an: (Visual Studio 2017, 2018)

Die Analyse der externen Methoden aus der Bibliothek SEAL wird in der folgenden Grafik näher dargestellt. Dabei fällt wieder die hohe Bedeutung der Multiplikation auf. In Zusammenhang mit einer Multiplikation wurde jeweils eine Relinearisierung durchgeführt, welche im Kapitel 6.1 näher erklärt wird. Im Vergleich dazu beanspruchte die Addition nur verhältnismäßig wenig Laufzeit, in absoluten Zahlen, für sich.

Insgesamt passt das dargebrachte Bild zu den Erkenntnissen der vorherigen Analyse der Eigenmethoden. Es lässt sich insgesamt ein hoher Fokus auf multiplikative Vorgänge erkennen. Auch bei der folgenden Grafik wurde die Y-Achse – aus denselben Gründen wie bei der vorherigen Grafik – zur Basis 10 logarithmiert.

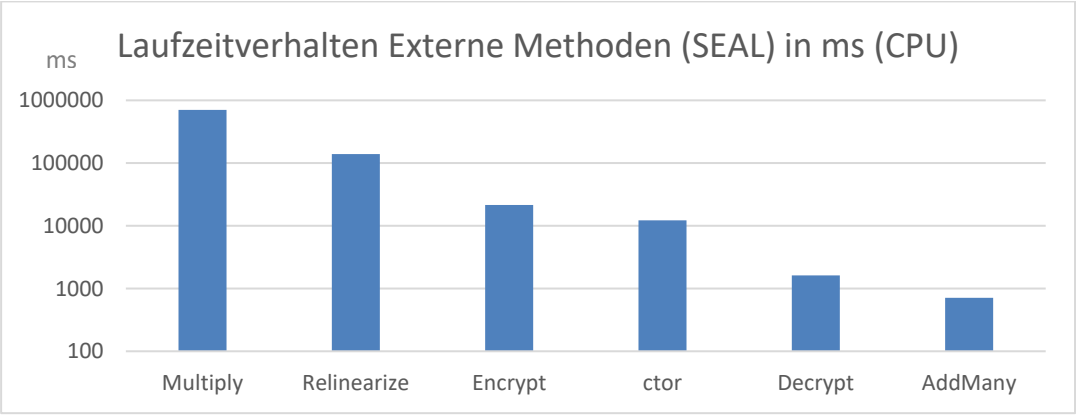

Abbildung 14: Laufzeitverhalten der externen Methoden (SEAL) – Szenario 2
In Anlehnung an: (Visual Studio 2017, 2018)

7.2.2 Evaluierung der Praxisfähigkeit

Die Auswertungen zur Analyse der Hypothese nach der Ergebnisqualität im Kontext mit der Umsetzung von homomorpher Verschlüsselung bei komplexeren mathematischen Vorgängen, insbesondere mit einer höheren multiplikativen Tiefe, können die erste Hypothese (H2.1) unter starken Einschränkungen bestätigen. Der Kompromiss zwischen der Korrektheit des Ergebnisses und der absoluten Laufzeit muss für jedes Anwendungsgebiet in der Realität abgewogen werden, um dessen tatsächliche Umsetzungsmöglichkeiten definieren zu können. (Armknecht et al., 2015)

Hierbei ist vor allem ein starker Einfluss der Wahl des richtigen Polynom-Modulus-Wertes aufgetreten. Dieser unterscheidet sich im Vergleich zum ersten Szenario besonders in einem grundsätzlich höheren Wertebereich. Eine kleinstmögliche Reduktion führt zu gravierend falschen Ergebnissen (siehe Tabelle 20), aber eine Erhöhung dieses Parameters führte in den Versuchen zu keiner Verbesserung.

Die Wahl, zwischen einer höheren Flexibilität in der zukünftigen Durchführung weiterer Berechnungen an den bereits verschlüsselten Datensätzen, oder einer effizienten Berechnung, erfordert eine weitreichende Entscheidung in der Konzeptionsphase des verteilten Systems. Ein Abtausch zwischen den Prioritäten lässt sich durch die Auswertungen bezüglich der zweiten Hypothese feststellen. Auch bei einer stark eingeschränkten Anzahl von Datensätzen konnten dabei durchaus sehr hohe Laufzeitdifferenzen bei unterschiedlicher Parameterwahl festgestellt werden. Das lässt sich vor allem durch die Ergebnisse der Tabelle 21 bestätigen.

Auch der Speicherverbrauch hängt stark von der Parameterwahl ab, doch ist hier der Abstand zwischen der unterschiedlichen Anzahl von Datensätzen nicht so stark, wie dies bei der Laufzeit der Fall ist.

Schlussendlich lässt sich zusammenfassen, dass ein annähernd korrektes Ergebnis mit ausreichend Ressourcen möglich ist. Doch bei einer Einschränkung der Ressourcen – wie für diese wissenschaftliche Arbeit in Kapitel 7 angemerkt – sind auch schon kleine Zuwächse bei der Datensatzanzahl nicht mehr in angemessener Zeit und mit den vorhandenen Arbeitsspeicherreserven zu berechnen.

Für eine Verbesserung dieser Ausgangssituation benötigt man – wie in den Analysen zur Laufzeit angemerkt – vor allem eine effizientere Methode zur Multiplikation von Schlüsseltexten. Dies würde die Umsetzung von ähnlichen Problemlösungen in der Praxis, im Rahmen einer realen Umsetzung, ermöglichen. Gleichzeitig sollten Algorithmen zur Lösung von mathematischen Problemen gefunden werden, die größtmöglich auf Multiplikation von Schlüsseltexten verzichten können.

7.3 Auswertung Szenario Energie-Cloud

Dieses Kapitel beschreibt die Auswertungen und Analysen rund um das letzte Szenario. Dieses wird in den Kapiteln 5.4 näher vorgestellt und die Implementierung wird im Kapitel 6.4 erläutert. Da in diesem Kapitel eine besondere Art der Datenverarbeitung angewandt wurde, wird besonderes Augenmerk auf die Praktikabilität dieser Methodik gelegt.

7.3.1 Überprüfung der Hypothesen

Als Hauptziel gilt in erster Linie die Beantwortung der zugrundliegenden Hypothesen. Diese wurden im Kapitel 5.4.5 zur Szenarienerstellung aufgestellt.

> *H3.1: Auf die Qualität des Resultates der Berechnungen hat die homomorphe Verschlüsselung im Kontext des Szenarios „Energie-Cloud“ keinen Einfluss.*

Hierbei wird die Variante mit Stapelverarbeitung der Variante ohne verschlüsselte Daten gegenübergestellt. Dazu wurden Resultate für bis zu 16.384 Datensätze auf einmal erzeugt und es wurden diese miteinander verglichen. Dies wird wiederholt durchgeführt und mit unterschiedlichen Parameterwerten für den Klartext-Modulus sowie Polynom-Modulus getestet. Alle Ergebnisse über 16.384 Werte sind vollständig ident bei folgender Parameterwahl.

- Stapelgröße: 16.384
- Polynom-Modulus: 16.384
- Klartext-Modulus: 163.841

Auch mit kleineren Stapelgrößen – wie 4.096 Datensätzen – kann ein identes Ergebnis erzeugt werden. Im Vergleich zu den bisherigen Szenarien muss angemerkt werden, dass es sich hierbei nur um ganzzahlige Ergebnisse handelt. Größere Stapelgrößen konnten, entweder aufgrund unzureichender Erfüllung der Bedingungen von SEAL in Bezug auf das CRT-Stapelverfahren (siehe dazu Kapitel 6.3.2), oder allgemein in Bezug auf den Parameter für den Polynom-Modulus (siehe dazu Kapitel 6.1.2), nicht fehlerfrei ausgeführt werden.

Die Ergebnisse für die Stapelgrößen von 4.096 und 16.384 wurden der elektronischen Beilage dieser Masterarbeit angefügt.

H3.2: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Energie-Cloud“ einen höheren Speicherverbrauch.

Auch diese Hypothese kann bestätigt werden. Insgesamt erfordert die vertrauliche Berechnung in jedem Testdurchlauf einen höheren Speichverbrauch, als die Berechnung von unverschlüsselten Daten. Die möglichen Vorteile des Stapelverfahrens im Vergleich zur bisherigen Methode werden durch die Überprüfung der Hypothese 3.5 analysiert. In der folgenden Tabelle sind exemplarisch Speicheranforderungen für zwei unterschiedlich parametrisierte Durchläufe dargestellt.

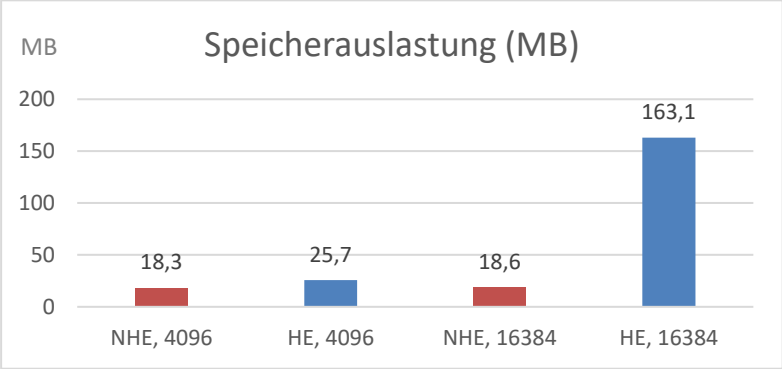

Abbildung 15: Vergleich der Speicherauslastung von vertraulicher Berechnung – Szenario 3
In Anlehnung an: (Visual Studio 2017, 2018)

Auch die CPU-Auslastung wurde im selben Kontext untersucht. Die Auswertungen zur Überprüfung dieser Hypothese sind sehr interessant für die Beantwortung der Forschungsfrage. Die Antwort, inwieweit eine sehr simple Berechnung für eine große Anzahl von Daten auf einmal in effektiver Weise durchgeführt werden kann, sollte mit einem Vergleich der Berechnung unter homomorpher Verschlüsselung und einer Durchführung derselben Berechnung auf unverschlüsselte Daten, gefunden werden. Hierbei wurde, aufgrund der Limitierungen von SEAL bezüglich des maximalen Polynom-Modulus von 16.384, eine maximale Datensatzanzahl von 16.384 gewählt, welche innerhalb einer Programmausführung verarbeitet wird. Als Gegenvergleich wurde dieselbe Prozedur auf 4.096 Datensätze angewandt.

H3.3: Die Berechnung mithilfe von homomorpher Verschlüsselung verursacht im Kontext des Szenarios „Energie-Cloud“ einen höheren Leistungsverbrauch.

Hierbei konnte die Hypothese bestätigt werden, welche der Variante mit vertraulicher Berechnung eine höhere Leistungsauslastung vorhergesagt hat. In der folgenden Tabelle sind die exemplarischen Messwerte zur Bestätigung der Vermutung dargestellt. Dabei war die Laufzeit der Verarbeitung von unverschlüsselten Daten ungefähr um den Faktor 67 für 4.096 Datensätze bzw. 730 für 16.384 Datensätze höher.

	Leistungsverbrauch in ms	
Datensätze	Vertrauliche Berechnung (mit CRT)	Berechnung mit unverschlüsselten Daten
4096	5.721	85
16348	71.532	98

Tabelle 22: Leistungsvergleich mit CRT – Szenario 3
In Anlehnung an: (Visual Studio 2017, 2018)

Zusätzlich dazu wurden folgende Hypothesen im Kontext des Stapelverfahrens evaluiert. Dabei ist von Interesse, ob der Einsatz solcher Techniken tatsächlich die Praktikabilität erhöhen kann. Dies wäre gegeben, wenn die Ergebnisse für Speicher- und Leistungsverbrauch einen eindeutigen Vorteil für die Durchführung mittels CRT-Batching aufzeigen würden.

Im Rahmen der Testdurchläufe sind die Effizienzvorteile in Bezug auf Speicherverbrauch und CPU-Leistung eindeutig erkennbar. Die Durchführung ohne Stapelverfahren wurde dabei mit einem effizientem Parameter für Polynom-Modulus von 2.048, sowie einem ausreichend großem Klartext-Modulus von 2^30, gestaltet.

H3.4: Die Anwendung der CRT-Technik im Kontext der Berechnungen des Szenarios „Energie-Cloud“ hat keine Auswirkung auf die Qualität des Ergebnisses.

Hierbei sehen wir uns den Unterschied bei den Ergebnissen von vertraulicher Berechnung mit und ohne Stapelverarbeitungstechnik nach CRT an. Bei 4.096 und 16.348 Datensätzen treten keine abweichenden Resultate auf. Auch diese Ergebnisse wurden der elektronischen Beilage zugefügt. Dieser Schluss passt mit dem Ergebnis der Überprüfung der Hypothese H3.1. zusammen.

H3.5: Die Anwendung der CRT-Technik im Kontext der Berechnungen des Szenarios „Energie-Cloud“ erzeugt weniger Leistungsverbrauch, als die vertrauliche Berechnung ohne Anwendung der CRT-Technik.

Dies ist ein sehr interessanter Aspekt für die Aussage über die Praktikabilität der CRT-Batching Methodik. Besonders der Leistungsverbrauch hat bisherige Szenarien mit hohen Datensätzen oft schwer testfähig gemacht, da ihre Ausführung bis zu mehrere Stunden am Testgerät (siehe Kapitel 7) verlangte.

In der folgenden Darstellung wurden die Laufzeiten der Durchläufe mit korrekten, übereinstimmenden Ergebnissen veranschaulicht. Dabei ist erkennbar, dass die Durchläufe mit

Stapeltechnik einen Vorteil gegenüber der individuellen Verarbeitung besitzen. Dies ist vor allem interessant, da für große Stapel von 16.384 und 4.096 Datensätzen ein höherer Polynom-Modulus gewählt werden muss. In den vorherigen Szenarien hat sich gezeigt, dass dieser Parameter einen hohen Einfluss auf die Laufzeit besitzt, aber im Kontext dieses Szenarios ist der Vorteil durch den Einsatz der CRT-Technik größer. Es zeigt sich, dass obwohl in der Ausführung ohne Einsatz der effizienten Hilfstechnik (in der Grafik als „Non CRT“ (NCRT) bezeichnet) ein Polynom-Modulus-Wert von nur 2.048 genutzt wurde, die Laufzeit bei 4.096 Datensätzen, in etwa um das 40-fache höher ausfällt, als bei Anwendung der Stapelverarbeitungstechnik. Bei der Untersuchung der gleichzeitigen Verarbeitung von 16.348 Datensätzen kann dabei die Laufzeit ungefähr auf ein Zehntel reduziert werden.

Dies zeigt auf, dass unter den Rahmenbedingungen der Experimentalumgebung eine Anwendung der Hilfstechnik einen großen Vorteil bringen kann. Gleichzeitig wird der Vorteil verhältnismäßig kleiner, je größer die Datenvektoren sein müssen, da der Nachteil gegenüber dem kleineren Polynom-Modulus-Wertes (siehe Auswertung Szenario 1) verhältnismäßig kleiner wird. In der folgenden Grafik sind die Messwerte für die eben aufgestellten Erkenntnisse veranschaulicht.

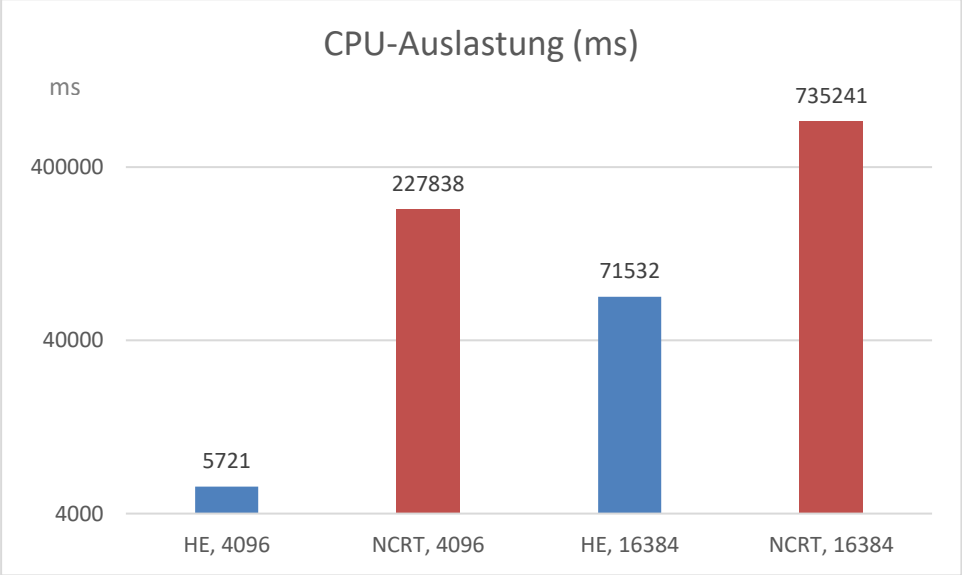

Abbildung 16: CPU-Auslastung der Stapelverarbeitungstechnik in Vergleich – Szenario 3
In Anlehnung an: (Visual Studio 2017, 2018)

> *H3.6: Die Anwendung der CRT-Technik im Kontext der Berechnungen des Szenarios „Energie-Cloud“ erzeugt weniger Speicherverbrauch, als die vertrauliche Berechnung ohne Anwendung der CRT-Technik.*

Aliquot zu Laufzeit wurde auch für den Speichverbrauch eine Hypothese aufgestellt. Im Kontext des Speicherverbrauchs ist die Entwicklung ähnlich wie die der Laufzeitmessung. In der Abbildung 17 wurde der unterschiedliche Speichverbrauch dargestellt. Hierbei kann festgestellt werden, dass die Speicherauslastung bei niedrigeren Blockgrößen (wie 4.096) und bei einem

Verzicht auf die Hilfstechnik, ungefähr dem 16-fachen der vergleichbaren Messwerte der Ausführung unter Nutzung der Stapelverarbeitungstechnik entspricht.

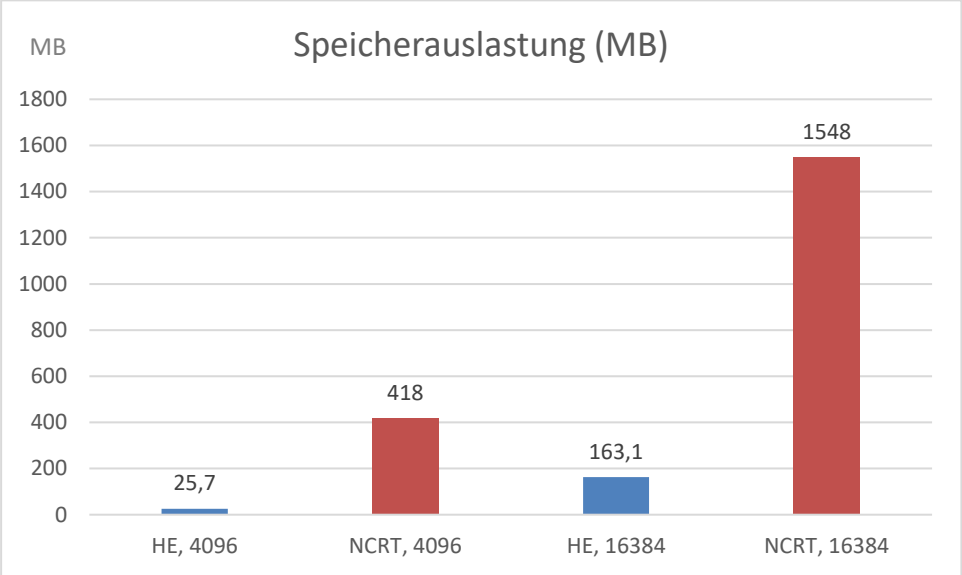

Abbildung 17: Speicherauslastung Vergleich für Stapelverarbeitungstechnik – Szenario 3
In Anlehnung an: (Visual Studio 2017, 2018)

7.3.2 Evaluierung der Praxisfähigkeit

In Bezug auf das Szenario der Energie-Cloud konnten einige wertvolle Erkenntnisse für weitere praktikable Anwendungsgebiete gewonnen werden. Einerseits hat die Technik zur Stapelverarbeitung einige Nachteile bei der Konstruktion von Berechnungen und auch Einschränkungen in Bezug auf die anwendbaren Operationen. Andererseits bietet sie eine, in Bezug auf Laufzeit und Speicherverbrauch, effiziente Alternative.

Wenn es gelingt die notwendigen mathematischen Operationen innerhalb der einschränkenden Rahmenbedingungen der Stapelverarbeitungstechnik zu konzeptionieren, dann kann man unter Berücksichtigung eines ähnlichen Kontextes wie im erstellten Szenario, von einer ungefähren Verringerung der notwendigen Laufzeit um den Faktor 3 bis 7 ausgehen. Dies kann die Praktikabilität maßgeblich beeinflussen.

Besonders für Operationen mit vorbestimmten und unveränderlichen Vektorgrößen kann diese Hilfstechnik einen Türöffner für den Einsatz von homomorpher Verschlüsselung in praxisnahen Anwendungen darstellen. Es ist auch auf effiziente Weise möglich vollkommen idente Ergebnisse zu erlangen, wobei berücksichtigt werden muss, dass hierbei nur ganzzahlige Ergebnisse verglichen wurden.

8. FAZIT UND AUSBLICK

Diese wissenschaftliche Arbeit wurde mit dem Leitgedanken geschrieben zwei Welten miteinander zu vereinen. Eine neue moderne Technologie ermöglicht für viele unterschiedliche Sichtweisen neue Wege zur Lösung von bestehenden Problemen. Durch die technische Komplexität und die hohen Vorstellungen im Zusammenhang mit der Umsetzung dieser Technologie entstanden große Divergenzen zwischen den Vorstellungen und den aktuellen Umsetzungsfähigkeiten. (siehe Kapitel 2) Dieses Spannungsfeld galt es innerhalb dieser Masterarbeit aufzuarbeiten, um einen kleinen Teil zur besseren Einschätzung der Anwendungsmöglichkeiten beitragen zu können. Die Forschungsfrage (siehe Kapitel 2) wird, aufgrund der gewonnen Erkenntnisse der Auswertungen, innerhalb dieses Fazits beantwortet. Dazu gilt es die Antwort in zwei Bereiche zu unterteilen. Im technologischen und wissenschaftlichen Fazit wird die Antwort grundsätzlich auf die Frage, was die aktuellen technologischen Möglichkeiten sind, bezogen. Der Abschnitt über das wirtschaftliche und gesellschaftliche Fazit soll dann diese Möglichkeiten im Kontext der Erwartungen und Bedürfnisse aus unterschiedlichen Bereichen der Wirtschaft, Politik oder anderen Organisationen betrachten.

8.1 Technologisches und wissenschaftliches Fazit

Im Rahmen des technologischen und wissenschaftlichen Fazits muss beachtet werden, dass die Aussagen, welche im Rahmen der Auswertungen getroffen worden sind (siehe Kapitel 7), insbesondere unter Berücksichtigung des Untersuchungskontextes, gesehen werden müssen.

Die ausgiebige Literaturanalyse und das Studieren einzelner Umsetzungen von Bibliotheken, welche den Einsatz von homomorphen Verschlüsselungstechniken erlauben, hat ein sehr heterogenes Bild ergeben. Eine zentrale Aussage kann in diesem Fall nur lauten, dass für den individuellen Einsatzzweck die optimale Art von homomorpher Verschlüsselung – wie etwa partielle oder vollständige homomorphe Verschlüsselungen – und weiters auch die geeignetste Implementierung, aufbauend auf der theoretischen Grundlage der Verschlüsselung, gewählt werden muss. (Armknecht et al., 2015; Furht & Escalante, 2010; Sadeghi & Schneider, 2010)

In Bezug auf den Einsatz einer vollständig homomorphen Verschlüsselung kann SEAL viele Kriterien als Grundlage von Umsetzungen der technologischen Anforderungen erfüllen. Auch wenn es noch Einschränkungen bei der praktischen Anwendung gibt. (Chen et al., 2017) Hierbei sei zum Beispiel die fehlende Division zweier Schlüsseltexte erwähnt (siehe Kapitel 6.1.1). (Chen et al., 2018; Du et al., 2017)

Die Erfassung von grundsätzlichen Problematiken bei der technischen Umsetzung von Anwendungsszenarien in konkreten Implementierungen war eines der Ziele dieser Masterarbeit. Dabei wurde auch vermehrt aufgezeigt, welche Grenzen dabei auftreten können. Vor allem die Auswertungen zum ersten Szenario („Emissions-Cloud“) bieten Einblick in die Möglichkeiten an Leistungs- und Speicherauslastung durch eine präzise Parameterwahl.

Beim Einsatz im Bereich der präskriptiven Statistik, also einem Themengebiet, welches in der Wissenschaft zurzeit große Aufmerksamkeit genießt, gibt es viele Ansätze homomorphe Verschlüsselung zu benützen (siehe Kapitel 4.3.4). (Aslett et al., 2015a) Auch innerhalb dieser Arbeit wurde eine lineare Regression (siehe Kapitel 6.3.2) implementiert, welche ein einfacher Grundstein für eine vertrauliche Berechnung solcher Problemstellungen sein könnte. Hierbei muss aber eindeutig festgestellt werden, dass die vollständige homomorphe Verschlüsselung mittels SEAL einen außerordentlich hohen Leistungsverbrauch und eine hohe Speicherauslastung bei der Umsetzung dieses Szenarios benötigte. Als Anregung für die weitere Forschung in diesem Gebiet gilt es ähnliche Szenarien mit anderen Bibliotheken oder Umgebungen, welche eventuell für statistische Verfahren optimiert worden sind, umzusetzen und mit diesen Ergebnissen zu vergleichen.

Der Mehraufwand im Bereich der Laufzeit zeigt sich allgemein sehr divergent. Keines der Szenarien konnte in annähernd gleicher Zeit auf verschlüsselte Daten durchgeführt werden, auch nicht, wenn nur eine kleine Datensatzanzahl (z.B. 10 Datensätze) als Vergleichsgrundlage diente. Im Rahmen des ersten Szenarios („Emissions-Cloud“) reichte der Faktor, um den die Laufzeit bei Verarbeitung der Daten in verschlüsselter Form Anstieg, von ungefähr 250 bis zu 13.700. Das zweite Szenario wies den höchsten Anstieg der Laufzeit aus. Hierbei erreichten die Werte für den Wachstumsfaktor der Laufzeit in Millisekunden von ungefähr 14.900 bis zu 198.500. Das dritte Szenario konnte, aufgrund einer innerhalb dieser Arbeit im Kapitel 6.4.2. vorgestellten Hilfsfunktion, bessere Werte für die Laufzeit im Verhältnis zur unverschlüsselten Variante erreichen. In diesem Fall konnte unter gewissen Rahmenbedingungen ein verhältnismäßig niedriger Faktor von ungefähr 70 erreicht werden. Ein ungünstigerer Kontext führte, aber auch bei dieser Variante zu einer Erhöhung der Laufzeit um den Faktor 14.306.

Die hohen Unterschiede lassen sich auf diverse Quellen zurückführen. Eine besondere Erkenntnis liefert dabei die Abarbeitung von Daten in zuvor definierten Vektorstrukturen im dritten Szenario. Diese können eventuell mittels des Stapelverfahrens von SEAL auf eine äußert effiziente Weise gelöst werden. Die Untersuchungen ergaben eine bis zu 40-fach niedrigere CPU-Laufzeit im Vergleich zur aliquoten Verarbeitung derselben verschlüsselten Daten, ohne Zuhilfenahme dieser Technik. Auch die Speicherauslastung profitiert vom Einsatz dieser Funktionserweiterung von SEAL. In Zukunft wäre es von hohem Wert, diese Funktion weiterzuentwickeln, um passgenaue Szenarien noch praktikabler zu machen.

8.2 Wirtschaftliches und gesellschaftliches Fazit

Die Erwartungen der Wirtschaft sind sehr groß, wenn es um die Sicherung der Privatsphäre beim Einsatz von Technologien des Cloud Computing geht. (Bitkom & KPMG, 2018a; Ghorbel et al., 2017) Hierbei wurden im dritten Kapitel einige Fakten über den derzeitigen Einsatz von Cloud Computing analysiert. Daraus ergibt sich eine Vielzahl an Anwendungsgebieten von vertraulichem Cloud Computing in der Geschäftswelt. Diese Entwicklung hat sich in den letzten Jahren durch den Fokus auf personenbezogenen Datenschutz in gewissen Bereichen stark verschärft. (Bitkom & KPMG, 2018a)

Doch auch die Politik kann von möglichen Varianten zur vertraulichen Berechnung im Rahmen von Cloud Computing profitieren. Als Beispiel ist dabei der Gesundheitssektor zu nennen, welcher sensible Daten in großem Umfang analysiert. (Griebel et al., 2015; Sadeghi & Schneider, 2010) Im Rahmen dieser Arbeit wurden auch andere konkrete Anwendungsgebiete, welche auch schon mittels homomorpher Verschlüsselung umgesetzt worden sind, im gesellschaftlichen Kontext näher betrachtet (siehe Kapitel 4.3).

Die aufgestellten Szenarien hatten unterschiedliche Einflüsse aus Politik und Wirtschaft und dienen damit einer realistischeren Einschätzung der Praktikabilität der verwendeten Technologie. Die Ergebnisse der Analysen über die Testimplementierungen geben grundsätzlich ein positives Bild über die Umsetzbarkeit solcher Szenarien. Doch es gilt besonders bei einer hohen Datensatzanzahl einen genauen Blick auf die Laufzeit- und Speicherauslastungsentwicklung im Verhältnis zu den Rahmenbedingungen zu werfen.

Bei der zukünftigen Ausarbeitung von potentiellen Einsatzgebieten von homomorpher Verschlüsselung können diverse Erkenntnisse dieser Masterarbeit als direkte Basis dienen. In erster Linie gilt es Berechnungsschritte so weit wie möglich zu vereinfachen. Dabei spielt auch die Vermeidung von Multiplikationen von verschlüsselten Werten eine große Rolle, was unter anderem aus den Ergebnissen der Auswertungen der Versuchsreihen des ersten Szenarios hervorgeht (siehe Kapitel 7.1.1). Die Aufteilung von Operationsgruppen, wie der Kodierung und Verschlüsselung, auf unterschiedliche Recheneinheiten eines Systems, erfordert im Kontext des Einsatzes von vollständig homomorpher Verschlüsselung eine zielgerichtete Ressourcenausstattung.

Viele einfache und vor allem starre Szenarien ließen sich aktuell, anhand der Erkenntnisse der Auswertungen, umsetzen. Die zusätzlich benötigte Rechenleistung liegt dabei zwar auf einem hohen Niveau, doch muss diese in Relation zum Wert des Schutzes der zugrundeliegenden Daten gesehen werden. Starre Szenarien, damit ist eine gleichbleibendes Set von Rechenoperationen an den verschlüsselten Daten gemeint, können unter gewissen Umständen ungemein von Hilfstechniken von SEAL profitieren. In diesem Bereich könnte man, unter den Rahmenbedingungen der Versuchsumgebung, die höchste Praktikabilität in naher Zukunft sehen.

Im Rahmen der Interpretation der Resultate der Auswertungen der Szenarien 1 und 2 lassen sich insgemein folgende Schlüsse ziehen: Eine hohe Relevanz spielt die Verteilung der Aufgaben von Kodierung, Verschlüsselung, Berechnung und Entschlüsselung, da diese in ihren Methoden oft eine sehr unterschiedlich hohe Leistungs- und Speicherauslastung besitzen. In praktischen Szenarien muss somit nicht nur ein Augenmerk auf die Ressourcen der berechnenden Cloud Computing Infrastruktur geworfen werden, sondern auch auf die verschlüsselnden Systeme. Diese können in gewissen Anwendungsszenarien durchaus einen Flaschenhals darstellen (siehe Kapitel 7.1.1).

Bei einer Vielzahl von unterschiedlichen Operationen auf einem gemeinsamen Datensatz, deren Ablauf sich eventuell auch noch während der Laufzeit ändert, hat sich in den Versuchen gezeigt, dass die Parameterwahl von außerordentlicher Bedeutung sein kann. Diese Einschränkung trifft Szenarien oft stark in der Einfachheit ihrer Umsetzung, da die optimale Parametrisierung weitgehende Auswirkungen auf die Qualität der Ergebnisse besitzt (siehe vor allem Kapitel 7.1.1).

Im Kontext der Ergebnisse der Auswertungen kann der Schluss gezogen werden, dass die Anwendungsgebiete, welche mathematisch simple Operationen auf einen gleichbleibend großen Datensatzblock beinhalten, eine verhältnismäßig hohe Möglichkeit zur praktikablen Umsetzung besitzen.

Eine weitere Einschränkung liefert die teilweise abweichende Ergebnisqualität, trotz annähernd optimaler Parameterwahl (siehe Kapitel 7.1 und 7.2). Dabei gilt es zu unterscheiden, ob eine geringfügige Abweichung noch tragbar für den praktischen Sinn des Szenarios ist. Als Erkenntnis dieser wissenschaftlichen Arbeit kann der Schluss gezogen werden, dass Anwendungen mit zwingend exakten Berechnungen von rationalen Zahlen vermehrt Problematiken entgegenstehen. Eine Anwendung ohne Fehlertoleranz, wie es zum Beispiel in der Finanzbranche vorkommt, kann entweder nur in stark eingeschränkter Form umgesetzt werden, oder muss die aufkommenden Engpässe schon in der Konzeption des Anwendungssystems mitberücksichtigen. (Armknecht et al., 2015) Dies könnte zum Beispiel durch die Mitberücksichtigung von schrittweisen Berechnungen mit Kommunikationsschritten geschehen.

Eine weiterführende Implementierung der Szenarien mit anderen Bibliotheken zur homomorphen Verschlüsselung wäre eine interessante Fortsetzung dieser Masterarbeit. Die Fortschritte der homomorphen Verschlüsselung könnten die laufende Überprüfung der Ergebnisse dieser Forschungsarbeit mit dem zukünftigen Stand der Technologie anregen.

ABKÜRZUNGSVERZEICHNIS

ADAC	Allgemeiner Deutscher Automobil Club
B/FV	Brakerski / Fan-Vercauteren
CC	Creative Commons
CO2	Kohlenstoffdioxid
CPU	Central Processing Unit
CRT	Chinese Remainder Theorem
CUDA	Compute Unified Device Architecture
CWI	Centrum Wiskunde & Informatica Amsterdam
DIN	Deutsche Industrie Norm
FHE	Fully homomorphic encryption
FTP	File Transfer Protocol
GB	Gigabyte
GHz	Gigahertz
GSW	(Scheme of) Gentry, Sahai, and Waters
HE	Homomorphic Encryption
ISO	International Standardization Organization
LWE	Learning with errors
MB	Megabyte
NCRT	Non CRT-Batching
NHE	Non Homomorphic Encryption
NOx	Stickoxide
RDE	Real Driving Emissions
RSA	(Verfahren von) Rivest, Shamir und Adleman
SEAL	Simple Encryption Arithmetic Library

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

LITERATURVERZEICHNIS

Achtnicht, M., Kesternich, M., & Sturm, B. (2018). *Die „Diesel-Debatte“: Ökonomische Handlungsempfehlungen an die Politik* (ZEW policy brief No. 3/2018). Mannheim: Zentrum für Europäische Wirtschaftsforschung (ZEW). Abgerufen von http://hdl.handle.net/10419/176696

ADAC Fahrzeugtechnik. (2016). EcoTest - Test- und Bewertungskriterien (ab 09/2016). Abgerufen 18. August 2018, von https://www.adac.de/_mmm/pdf/28843_292234.pdf

Adida, B. (2008). Helios: Web-based Open-Audit Voting. In *USENIX security symposium* (Bd. 17, S. 335–348).

Adida, B. (2018). helios-server: Helios server. Abgerufen 7. Juli 2018, von https://github.com/benadida/helios-server

Aono, Y., Hayashi, T., Phong, L. T., & Wang, L. (2015). Fast and Secure Linear Regression and Biometric Authentication with Security Update. *IACR Cryptology ePrint Archive*, *2015*, 692.

Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C. A., & Strand, M. (2015). A Guide to Fully Homomorphic Encryption. *IACR Cryptology ePrint Archive*, *2015*, 1192.

Aslett, L. J. M., Esperança, P. M., & Holmes, C. C. (2015a). A review of homomorphic encryption and software tools for encrypted statistical machine learning. *ArXiv*, *1508*(06574). Abgerufen von http://arxiv.org/abs/1508.06574

Aslett, L. J. M., Esperança, P. M., & Holmes, C. C. (2015b). Encrypted statistical machine learning: new privacy preserving methods. *ArXiv*, *1508*(06845). Abgerufen von http://arxiv.org/abs/1508.06845

Barton, T. (2014). *E-Business mit Cloud Computing: Grundlagen, praktische Anwendungen, verständliche Lösungsansätze*. Wiesbaden: Springer Vieweg.

Bedner, M. (2013). *Cloud Computing: Technik, Sicherheit und rechtliche Gestaltung* (Bd. 14). Kassel: Kassel University Press.

Bellare, M., & Rogaway, P. (2005). Introduction to modern cryptography. *Ucsd Cse*, 207.

Bitkom, & KPMG. (2016). Cloud Computing in deutschen Unternehmen nach Branche 2016 | Umfrage. Abgerufen 8. Juli 2018, von https://de.statista.com/statistik/daten/studie/305570/umfrage/einsatz-von-cloud-computing-in-deutschen-unternehmen-nach-branche/

Bitkom, & KPMG. (2018a). Cloud Computing - Kriterien bei der Auswahl eines Cloud-Providers 2017 | Umfrage. Abgerufen 30. Juni 2018, von

https://de.statista.com/statistik/daten/studie/545924/umfrage/kriterien-bei-der-auswahl-eines-cloud-providers-in-deutschen-unternehmen/

Bitkom, & KPMG. (2018b). Public Cloud - Art der gespeicherten Daten in deutschen Unternehmen nach Größe 2017 | Umfrage. Abgerufen 4. Juli 2018, von https://de.statista.com/statistik/daten/studie/714300/umfrage/art-der-gespeicherten-daten-in-der-public-cloud-in-unternehmen-nach-groesse/

Björck, Å. (1996). *Numerical methods for least squares problems*. Philadelphia: Society for Industrial and Applied Mathematics.

Bos, J. W., Castryck, W., Iliashenko, and I., & Vercauteren, F. (2016). *Privacy-friendly Forecasting for the Smart Grid using Homomorphic Encryption and the Group Method of Data Handling* (No. 1117). Abgerufen von https://eprint.iacr.org/2016/1117

Bowen, J. A. (2011). Cloud Computing: Issues in Data Privacy/Security and Commercial Considerations. *Computer & Internet Lawyer*, *28*(8), 1–8.

Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2011). *Fully Homomorphic Encryption without Bootstrapping* (No. 277). Cryptology ePrint Archive. Abgerufen von https://eprint.iacr.org/2011/277

Brenner, M. (2012). *Rechnen mit verschlüsselten Programmen und Daten* (Dissertation). Leibniz Universität Hannover, Hannover.

Brenner, M. (2016). Microsofts entrauscht homomorphe Krypto-Library SEAL [Fachzeitschrift]. Abgerufen 29. Juni 2018, von https://www.heise.de/security/meldung/Microsofts-entrauscht-homomorphe-Krypto-Library-SEAL-3243299.html

Brenner, M., Perl, H., & Smith, M. (2012). Practical Applications of Homomorphic Encryption (S. 10). Gehalten auf der SECRYPT.

Brocato, M. (2018). Mockaroo. Abgerufen 7. Juli 2018, von https://www.mockaroo.com/help/about

Bundesministerium für Wirtschaft und Energie. (2016). IT-Sicherheit für die Industrie 4.0. Abgerufen 31. Juli 2018, von https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/it-sicherheit-fuer-industrie-4-0.html

Carpov, S., Nguyen, T. H., Sirdey, R., Constantino, G., & Martinelli, F. (2016). Practical Privacy-Preserving Medical Diagnosis Using Homomorphic Encryption (S. 593–599). IEEE. https://doi.org/10.1109/CLOUD.2016.0084

Çetin, G. S., Chen, H., Laine, K., Lauter, K., Rindal, P., & Xia, Y. (2017). Private queries on encrypted genomic data. *BMC Medical Genomics*, *10*(2), 45. https://doi.org/10.1186/s12920-017-0276-z

Chatterjee, A., & Sengupta, I. (2015). Searching and Sorting of Fully Homomorphic Encrypted Data on Cloud. *IACR Cryptology ePrint Archive*, *2015*, 981.

Chen, H., Han, K., Huang, Z., Jalali, A., & Laine, K. (2018). Simple Encrypted Arithmetic Library v2.3.0, 35.

Chen, H., Laine, K., & Player, R. (2017). Simple Encrypted Arithmetic Library - SEAL v2.2. In M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. A. Ryan, V. Teague, … M. Jakobsson (Hrsg.), *Financial Cryptography and Data Security* (Bd. 10323, S. 3–18). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-70278-0_1

Chillotti, I., Gama, N., Georgieva, M., & Izabachène, M. (2016). Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds. In J. H. Cheon & T. Takagi (Hrsg.), *Advances in Cryptology – ASIACRYPT 2016* (Bd. 10031, S. 3–33). Heidelberg: Springer. https://doi.org/10.1007/978-3-662-53887-6_1

Chillotti, I., Gama, N., Georgieva, M., & Izabachène, M. (2017). Faster Packed Homomorphic Operations and Efficient Circuit Bootstrapping for TFHE. In T. Takagi & T. Peyrin (Hrsg.), *Advances in Cryptology – ASIACRYPT 2017* (Bd. 10624, S. 377–408). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-70694-8_14

Chirgwin, R. (2018). IBM's homomorphic encryption accelerated to run 75 times faster. Abgerufen 29. Juni 2018, von https://www.theregister.co.uk/2018/03/08/ibm_faster_homomorphic_encryption/

CSIRO Data 61. (2018a). *Python Paillier Documentation* (S. 46). Abgerufen von https://python-paillier.readthedocs.io/en/stable/

CSIRO Data 61. (2018b). Who we are - Data61. Abgerufen 5. Juli 2018, von https://www.data61.csiro.au/en/Who-we-are

Dai, W. (2018). cuFHE: CUDA-accelerated Fully Homomorphic Encryption Library. Abgerufen 21. Mai 2018, von https://github.com/WeiDaiWD/cuFHE

Dai, W., & Sunar, B. (2015). cuHE: A Homomorphic Encryption Accelerator Library. In *Cryptography and Information Security in the Balkans* (S. 169–186). Springer, Cham. https://doi.org/10.1007/978-3-319-29172-7_11

DinosoftLabs. (2018). *Lock, secure, signal, wifi icon* [Security Double Colour Blue & Black vol 4]. Abgerufen von https://www.iconfinder.com/icons/2537320/lock_secure_signal_wifi_icon

Doleski, O. D. (Hrsg.). (2017). *Herausforderung Utility 4.0: wie sich die Energiewirtschaft im Zeitalter der Digitalisierung verändert*. Wiesbaden: Springer Vieweg.

Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2017). Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE*, *105*(3), 552–567.

Du, Y., Gustafson, L., Huang, D., & Peterson, K. (2017). Implementing ML Algorithms with HE, 14.

Ducas, L., & Micciancio, D. (2014). *FHEW: Bootstrapping Homomorphic Encryption in less than a second* (No. 816). Abgerufen von https://eprint.iacr.org/2014/816

Dyer, J., Dyer, M., & Xu, J. (2017). Practical Homomorphic Encryption Over the Integers. *arXiv*, *1702*(07588). Abgerufen von http://arxiv.org/abs/1702.07588

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, *32*(2), 407–499.

Feiler, L., & Horn, B. (2018). *Umsetzung der DSGVO in der Praxis: Fragen, Antworten, Muster*. Wien: Verlag Österreich.

Flügge, B. (Hrsg.). (2016). *Smart Mobility: Trends, Konzepte, Best Practices für die intelligente Mobilität*. Wiesbaden: Springer Vieweg.

Fontaine, C., & Galand, F. (2007). A Survey of Homomorphic Encryption for Nonspecialists. *EURASIP Journal on Information Security*, *2007*, 1–10. https://doi.org/10.1155/2007/13801

Furht, B., & Escalante, A. (Hrsg.). (2010). *Handbook of cloud computing*. New York: Springer.

Gentry, C. (2009). *A fully homomorphic encryption scheme* (PhD Thesis). Stanford University.

Ghorbel, A., Ghorbel, M., & Jmaiel, M. (2017). Privacy in cloud computing environments: a survey and research challenges. *Journal of Supercomputing*, *73*(6), 2763–2800. https://doi.org/10.1007/s11227-016-1953-y

Greenberg, A. (2009). IBM's Blindfolded Calculator. Abgerufen 4. Juli 2018, von forbes/2009/0713/breakthroughs-privacy-super-secret-encryption

Griebel, L., Prokosch, H.-U., Köpcke, F., Toddenroth, D., Christoph, J., Leb, I., … Sedlmayr, M. (2015). A scoping review of cloud computing in healthcare. *BMC Medical Informatics and Decision Making*, *15*(1). https://doi.org/10.1186/s12911-015-0145-7

Guldner, M., Spieldenner, T., & Schubotz, R. (2018). NEXUS: Using Geo-fencing Services without revealing your Location. In *Proceedings of the 2018 Global Internet of Things Summit*. Bilbao: IEEE Communications Society. Abgerufen von https://www.dfki.de/web/forschung/publikationen/renameFileForDownload?filename=NEXUS_accepted.pdf&file_id=uploads_3529

Halevi, S. (2018). shaih/HElib: An Implementation of homomorphic encryption. Abgerufen 5. Juli 2018, von https://github.com/shaih/HElib

Halevi, S., & Shoup, V. (2014). Algorithms in HElib. In *Advances in Cryptology – CRYPTO 2014* (S. 554–571). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44371-2_31

Halevi, S., & Shoup, V. (2018). Faster Homomorphic Linear Transformations in HElib. In H. Shacham & A. Boldyreva (Hrsg.), *Advances in Cryptology – CRYPTO 2018* (Bd. 10991, S. 93–120). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-96884-1_4

Hansen, M. (2012). Datenschutz im Cloud Computing. In *Daten- und Identitätsschutz in Cloud Computing, E-Government und E-Commerce* (S. 79–95). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30102-5_4

Hebrail, G., & Berard, A. (2012). Individual household electric power consumption Data Set. Abgerufen 3. August 2018, von https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption

Helios Voting. (2018). Helios Voting. Abgerufen 7. Juli 2018, von https://vote.heliosvoting.org/

Hilpert, H., Thoroe, L., & Schumann, M. (2011). Real-Time Data Collection for Product Carbon Footprints in Transportation Processes Based on OBD2 and Smartphones. In *2011 44th Hawaii International Conference on System Sciences* (S. 1–10). Kauai, HI: IEEE. https://doi.org/10.1109/HICSS.2011.356

Hoffstein, J., Pipher, J. C., & Silverman, J. H. (2014). *An introduction to mathematical cryptography* (2. Aufl.). New York: Springer.

Ibarrondo, A. (2018). Pyfhel: PYthon For HELib, implements basic functionalities of HElib as a Homomorphic Encryption library such as sum, mult, or scalar product in Python. Includes Afhel, a C++ abstraction of HElib. Abgerufen 5. Juli 2018, von https://github.com/ibarrond/Pyfhel

Kaiser, T. (2017). Digitale Transformation, aber wie? – Von der Spielwiese zur Umsetzungsplanung. In *Herausforderung Utility 4.0* (S. 69–87). Springer Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-15737-1_4

Kim, M., & Lauter, K. (2015). Private Genome Analysis Through Homomorphic Encryption. In *BMC medical informatics and decision making* (Bd. 15). BioMed Central Ltd. Abgerufen von https://www.microsoft.com/en-us/research/publication/private-genome-analysis-through-homomorphic-encryption/

Kuo, A. M.-H. (2011). Opportunities and challenges of cloud computing to improve health care services. *Journal of Medical Internet Research*, *13*(3).

Lab41. (2018). PySEAL. Abgerufen 1. Juli 2018, von https://github.com/Lab41/PySEAL (Original work published 19. Oktober 2017)

Lauter, K., López-Alt, A., & Naehrig, M. (2014). Private computation on encrypted genomic data. In *International Conference on Cryptology and Information Security in Latin America* (S. 3–27). Springer.

Lu, W., Kawasaki, S., & Sakuma, J. (2017). Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data. Internet Society. https://doi.org/10.14722/ndss.2017.23119

Microsoft Research. (2018). Simple Encrypted Arithmetic Library (SEAL). Abgerufen 2. Juli 2018, von https://www.microsoft.com/en-us/research/project/simple-encrypted-arithmetic-library/

Münzl, G., Pauly, M., & Reti, M. (2015). *Cloud Computing als neue Herausforderung für Management und IT*. Berlin: Springer Vieweg.

Nigam, V. K., & Bhatia, S. (2016). Impact of Cloud Computing on Health Care. *Int Res J Eng Technol*, *3*(5), 2804–10.

Ning, Z., Wubulihairen, M., & Yang, F. (2012). PM, NOx and butane emissions from on-road vehicle fleets in Hong Kong and their implications on emission control policy. *Atmospheric Environment*, *61*, 265–274. https://doi.org/10.1016/j.atmosenv.2012.07.047

Paillier, P. (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In J. Stern (Hrsg.), *Advances in Cryptology — EUROCRYPT '99* (S. 223–238). Heidelberg: Springer.

PySyft: A Library for Private, Secure, Multi-Owner Deep Learning - Currently Pre Alpha. (2018). Python, OpenMined. Abgerufen von https://github.com/OpenMined/PySyft (Original work published 2017)

Rothblum, R. (2011). Homomorphic Encryption: From Private-Key to Public-Key. In Y. Ishai (Hrsg.), *Theory of Cryptography* (Bd. 6597, S. 219–234). Heidelberg: Springer. https://doi.org/10.1007/978-3-642-19571-6_14

Rountree, D., & Castrillo, I. (2014). *The Basics of Cloud Computing : Understanding the Fundamentals of Cloud Computing in Theory and Practice*. Amsterdam: Syngress.

Sadeghi, A.-R., & Schneider, T. (2010). Verschlüsselt Rechnen: Sichere Verarbeitung verschlüsselter medizinischer Daten am Beispiel der Klassifikation von EKG-Daten. In *Workshop Innovative und*

sichere Informationstechnologie für das Gesundheitswesen von morgen (perspeGKtive'10) (Bd. P-174, S. 11–25). Bonner Köllen Verlag. Abgerufen von http://thomaschneider.de/papers/SS10.pdf

Sarunski, M. (2016). Big Data – Ende der Anonymität?: Fragen aus Sicht der Datenschutzaufsichtsbehörde Mecklenburg-Vorpommern. *Datenschutz und Datensicherheit - DuD*, *40*(7), 424–427. https://doi.org/10.1007/s11623-016-0630-x

Smart, N. P., & Vercauteren, F. (2010). Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In P. Q. Nguyen & D. Pointcheval (Hrsg.), *Public Key Cryptography – PKC 2010* (Bd. 6056, S. 420–443). Heidelberg: Springer. https://doi.org/10.1007/978-3-642-13013-7_25

Spiecker, I. (2017). Smart Home, Smart Grid, Smart Meter – digitale Konzepte und das Recht an Daten. In *Herausforderung Utility 4.0* (S. 285–300). Springer Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-15737-1_15

Stack Overflow. (2018). Stack Overflow Developer Survey. Abgerufen 10. August 2018, von https://stackoverflow.com/insights/survey/2018/?utm_source=so-owned&utm_medium=social&utm_campaign=dev-survey-2018&utm_content=social-share

Varia, M., Yakoubov, S., & Yang, Y. (2015). HEtest: A Homomorphic Encryption Testing Framework. In *International Conference on Financial Cryptography and Data Security* (S. 213–230). Heidelberg: Springer.

Vernamlab. (2018). cuHE: CUDA Homomorphic Encryption Library. Abgerufen 5. Juni 2018, von https://github.com/vernamlab/cuHE (Original work published 18. Jänner 2016)

Visual Studio 2017. (2018). Visual Studio 2017 - Debuggen, Profilerstellung und Diagnose. Abgerufen 20. September 2018, von https://visualstudio.microsoft.com/de/vs/features/debugging-and-diagnostics/

Wang, S., Zhang, Y., Dai, W., Lauter, K., Kim, M., Tang, Y., … Jiang, X. (2016). HEALER: homomorphic computation of ExAct Logistic rEgRession for secure rare disease variants analysis in GWAS. *Bioinformatics*, *32*(2), 211–218. https://doi.org/10.1093/bioinformatics/btv563

Winkelhake, U. (2017). *Die digitale Transformation der Automobilindustrie*. Heidelberg: Springer. https://doi.org/10.1007/978-3-662-54935-3

Xu, C., Chen, J., Wu, W., & Feng, Y. (2016). Homomorphically Encrypted Arithmetic Operations Over the Integer Ring. In F. Bao, L. Chen, R. H. Deng, & G. Wang (Hrsg.), *Information Security Practice and Experience* (Bd. 10060, S. 167–181). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-49151-6_12

Yasuda, M., Shimoyama, T., & Kogure, J. (2014). Secret computation of purchase history data using somewhat homomorphic encryption. *Pacific Journal of Mathematics for Industry*, *6*(1). https://doi.org/10.1186/s40736-014-0005-x

Zalakeviciute, R., Rybarczyk, Y., López-Villada, J., & Diaz Suarez, M. V. (2018). Quantifying decade-long effects of fuel and traffic regulations on urban ambient PM2.5 pollution in a mid-size South American city. *Atmospheric Pollution Research*, *9*(1), 66–75.