



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Sicheres Cloud Computing in der Praxis

Identifikation relevanter Kriterien zur Evaluierung der Praxistauglichkeit von Technologieansätzen im Cloud Computing Umfeld mit dem Fokus auf Datenschutz und Datensicherheit.

Dissertation
zur Erlangung des akademischen Grades

Dr. rer. nat.

Herr M.Sc. Paul Reinhold
geboren am 24. Februar 1987 in Zschopau

Fakultät für Informatik
an der Technischen Universität Chemnitz

Gutachter:

1. Prof. Dr. Wolfgang Benn
2. Prof. Dr. Wolfram Hardt
3. Prof. Dr. Dirk Labudde

Tag der Verteidigung:

02. Februar 2017

Online-Veröffentlichung:

<http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-222280>

Abstract

In dieser Dissertation werden verschiedene Anforderungen an sicheres Cloud Computing untersucht. Insbesondere geht es dabei um die Analyse bestehender Forschungs- und Lösungsansätze zum Schutz von Daten und Prozessen in Cloud-Umgebungen und um die Bewertung ihrer Praxistauglichkeit. Die Basis für die Vergleichbarkeit stellen spezifizierte Kriterien dar, nach denen die untersuchten Technologien bewertet werden.

Hauptziel dieser Arbeit ist zu zeigen, auf welche Weise technische Forschungsansätze verglichen werden können, um auf dieser Grundlage eine Bewertung ihrer Eignung in der Praxis zu ermöglichen. Hierzu werden zunächst relevante Teilbereiche der Cloud Computing Sicherheit aufgezeigt, deren Lösungsstrategien im Kontext der Arbeit diskutiert und State-of-the-Art Methoden evaluiert. Die Aussage zur Praxistauglichkeit ergibt sich dabei aus dem Verhältnis des potenziellen Nutzens zu den damit verbundene erwartenden Kosten. Der potenzielle Nutzen ist dabei als Zusammenführung der gebotenen Leistungsfähigkeit, Sicherheit und Funktionalität der untersuchten Technologie definiert. Zur objektiven Bewertung setzten sich diese drei Größen aus spezifizierten Kriterien zusammen, deren Informationen direkt aus den untersuchten Forschungsarbeiten stammen. Die zu erwartenden Kosten ergeben sich aus Kostenschlüsseln für Technologie, Betrieb und Entwicklung. In dieser Arbeit sollen die zugleich spezifizierten Evaluierungskriterien sowie die Konstellation der obig eingeführten Begriffe ausführlich erläutert und bewertet werden.

Für die bessere Abschätzung der Eignung in der Praxis wird in der Arbeit eine angepasste SWOT-Analyse für die identifizierten relevanten Teilbereiche durchgeführt. Neben der Definition der Praktikabilitätsaussage, stellt dies die zweite Innovation dieser Arbeit dar. Das konkrete Ziel dieser Analyse ist es, die Vergleichbarkeit zwischen den Teilbereichen zu erhöhen und so die Strategieplanung zur Entwicklung sicherer Cloud Computing Lösungen zu verbessern.

Inhaltsverzeichnis

Teil I	Einleitung und Grundlagen	1
1.	Einleitung	5
1.1.	Thesen der Arbeit	6
1.2.	Struktur der Arbeit	7
2.	Grundlagen	9
2.1.	Cloud Computing	9
2.2.	Cloud Computing und Sicherheit	14
2.3.	Taxonomie der Cloud-Sicherheit	17
2.4.	Cloud Computing und IT-Compliance	19
2.4.1.	Cloud Computing und Datenschutz	19
2.4.2.	Cloud Computing und Service Level Agreements	24
2.4.3.	Cloud Computing und Zertifizierung	26
Teil II	Theoretisch-technische Analyse und Evaluierung	27
3.	Cloud-Sicherheitsmanagement	29
3.1.	Zugangskontrolle	30
3.2.	Zugriffskontrolle	32
3.3.	Identitätsmanagement	33
3.4.	Schlüsselmanagement	36
3.5.	Logging und Auditierung	38
3.6.	IT Compliance	39
3.7.	Zusammenfassung	40
4.	Cloud-Applikationssicherheit	41
4.1.	Replikation und Verteilung	42
4.2.	Applikationslogik	43
4.2.1.	Verschleierung	43
4.2.2.	Beweisbasierte und verifizierte Berechnung	44
4.2.3.	Sichere Funktionsevaluierung	45
4.2.4.	Funktionale Verschlüsselung	52
4.2.5.	Zusammenfassung	54
4.3.	Datenbanken	55
4.4.	Dateisysteme	59
4.5.	Anonymisierung	62

4.6. Datenzugriff	66
4.7. Zusammenfassung	68
5. Trusted Cloud Computing	69
5.1. Trusted Plattform Module	70
5.1.1. Virtuelle Trusted Plattform Module	72
5.2. Sichere Ausführungsumgebungen	74
5.3. Hardware-Sicherheitsmodule	76
5.4. Zusammenfassung	78
6. Cloud-Virtualisierungssicherheit	79
6.1. Virtualisierung im Zusammenhang mit Cloud Computing	80
6.2. Sichere Virtualisierungslösungen	82
6.2.1. Trusted Virtualization	83
6.2.2. Geschachtelte Virtualisierung	84
6.3. Zusammenfassung	86
Teil III Technologie-Evaluierung nach ausgewählten Kriterien	87
7. Technologie Evaluierung	89
7.1. Evaluierungskriterien	90
7.1.1. Potenzieller Nutzen $\mathfrak{N}$	90
7.1.2. Kosten K	92
7.1.3. Praktikabilitätsaussage $\mathfrak{P}$	94
7.1.4. Leistungsfähigkeit L	95
7.1.5. Sicherheit S	97
7.1.6. Funktionalität F	99
7.1.7. Kategorisierung und Realisierung	100
7.2. Evaluierungsergebnisse	102
7.3. Auswertung und Diskussion	109
7.3.1. Cloud-Sicherheitsmanagement-Lösungen	112
7.3.2. Cloud-Applikationssicherheit-Lösungen	114
7.3.3. Lösungen für Trusted Cloud Computing	116
7.3.4. Cloud-Virtualisierungssicherheit-Lösungen	118
7.3.5. Cloud-Applikation und Cloud-Umgebung	120
8. Zusammenfassung	123
8.1. Fazit	124
Teil IV Anhang	129
A. Auflistung der Evaluierungsergebnisse	131
Literaturverzeichnis	246

Abbildungsverzeichnis

2.1. Cloud Service Modelle	10
2.2. Cloud-Deployment-Modelle	11
2.3. Systemsicherheitsbegriff	14
2.4. CIA-Prinzip der Datensicherheit	15
2.5. Schematische Darstellung von Verfahren der sicheren Verarbeitung	16
2.6. Taxonomie der Cloud-Sicherheit	18
2.7. Erweitertes CIA Prinzip der Datensicherheit durch den Datenschutz	21
3.1. Bestandteile des Cloud-Sicherheitsmanagements im Überblick	29
3.2. Aufbau und Teilnehmer eines föderierten Identitätsmanagementsystems	33
3.3. Überblick eines sicheren Log-Services	38
4.1. Übersicht über die Cloud-Applikationssicherheit	41
4.2. Szenarien der Applikationsreplikation und -verteilung auf zwei Cloud-Anbieter	42
4.3. Strukturierung Abschnitt Applikationslogik	43
4.4. Repräsentation verschiedener auswertbarer Funktionen	45
4.5. Aufbau und Funktion von Garbled Circuits	46
4.6. Erzeugung eines sicheren Ordered Binary Decision Diagram	47
4.7. Prinzip der homomorphen Verschlüsselung	48
4.8. Bootstrapping und Neuverschlüsselung in voll homomorphen Verschlüsselungsverfahren	49
4.9. Suchen innerhalb von verschlüsselten Daten	52
4.10. Darstellung von attribut-basierter Verschlüsselung	53
4.11. Die Architektur der CryptDB	55
4.12. Architektur der Relational Cloud	56
4.13. Unterscheidung Anonymisierung und Pseudonymisierung	62
4.14. Das Prinzip der Reidentifizierung	63
4.15. ORAM Datenorganisation	66
5.1. Bestandteile des Trusted Cloud Computing im Überblick	69
5.2. TMP-Chip Darstellung und interner Aufbau	70
5.3. Darstellung von vTMP Architekturen	72
5.4. Darstellung der TrustVisor Architektur	74
5.5. Datenschutzkonzept nach Maniatis	75
5.6. Die Abbildung eines IBM 4758	76
6.1. Bestandteile der Cloud-Virtualisierungssicherheit	79
6.2. Grundprinzip der Virtualisierung	80

6.3. Ansicht der NOVA Microhypervisor Architektur	82
6.4. SEC2 Architektur nach Hao et.al. [HLMS10]	83
6.5. Gegenüberstellung von sicheren Virtualisierungslösungen	84
6.6. CloudVisor und Xen-Blanket Architektur	85
7.1. Gegenüberstellung von Nutzen und Kosten	89
7.2. Gleichung zur Berechnung der Leistungsfähigkeit	96
7.3. Gleichung zur Berechnung der Sicherheit	98
7.4. Gleichung zur Berechnung der Funktionalität	100
7.5. Darstellung Kosten-Realisierungseffizienz Diagramm	110
7.6. Praktikabilität im Cloud-Sicherheitsmanagement	112
7.7. Praktikabilität in der Cloud-Applikationssicherheit	114
7.8. Praktikabilität im Trusted Cloud Computing	116
7.9. Verteilung in der Cloud-Virtualisierungssicherheit	118
7.10. Verteilung für Cloud-Applikation und Cloud-Umgebung	120

Tabellenverzeichnis

2.1. Zusammenfassung ausgewählter Cloud-Zertifizierungen	26
3.1. Zugriffskontrollmodelle für die Cloud	32
4.1. Zusammenfassung sicherer Datenbankspeicherlösungen	58
4.2. Zusammenfassung sicherer Datenspeichersysteme	61
4.3. k-Anonymisierung eines Datensatzes	64
7.1. Zusammensetzung des Evaluierungskriteriums Kosten K	92
7.2. Bewertungsmaßstabs für den Schlüssel K_{Tech}	92
7.3. Bewertungsmaßstabs für den Schlüssel K_{Pers}	92
7.4. Bewertungsmaßstabs für den Schlüssel K_{Entw}	92
7.5. Praktikabilitätskategorien	94
7.6. Evaluierungsfelder <i>Leistungsfähigkeit</i> und <i>Prototyp</i>	95
7.7. Bewertungsmaßstabs für den Mehraufwand	95
7.8. Evaluierungsfeld <i>Sicherheit</i>	97
7.9. Evaluierungsfeld <i>Funktionalität</i>	99
7.10. Evaluierungsfelder <i>Klassifikation</i> und <i>Akademische Informationen</i>	100
7.11. Zusammensetzung des Evaluierungsfeld <i>Realisierung</i>	101
7.12. Ergebnisse Cloud-Sicherheitsmanagement Teil 1	103
7.13. Ergebnisse Cloud-Sicherheitsmanagement Teil 2	104
7.14. Ergebnisse Cloud-Applikationssicherheit Teil 1	105
7.15. Ergebnisse Cloud-Applikationssicherheit Teil 2	106
7.16. Ergebnisse Trusted Cloud Computing und Virtualisierungssicherheit Teil 1	107
7.17. Ergebnisse Trusted Cloud Computing und Virtualisierungssicherheit Teil 2	108
7.18. Zusammenfassung der Evaluierungsergebnisse	109
7.19. Realisierungseffizienzzenarien	111
7.20. Zusammenfassung Cloud-Sicherheitsmanagement	113
7.21. Zusammenfassung Cloud-Applikationssicherheit	115
7.22. Zusammenfassung Trusted Cloud Computing	117
7.23. Zusammenfassung Cloud-Virtualisierungssicherheit	119

Teil I

Einleitung und Grundlagen

If you think technology can solve you security problems, then you don't understand the problems and you don't understand the technology. B.Schneier [Sch11b].

Einleitung

Cloud Computing bietet durch eine fortgeschrittene Art der Bereitstellung von IT Ressourcen in Form von Rechenleistung und Speicher neue Möglichkeiten der Auslagerung von Daten und Prozessen, wobei es der flexible und dynamische Bezug jederzeit erlaubt, die benötigte IT Infrastruktur dem aktuellen Bedarf anzupassen und damit die Kosten für diese Ressourcen zu optimieren. Neben den zahlreichen Vorteilen, die sich durch eine derartige Ressourcenbereitstellung ergeben, ist der Einsatz von Cloud Technologien stets mit Fragestellungen bezüglich der Datensicherheit und des Datenschutzes verbunden. Im Gegensatz zur klassischen Auslagerung von IT Infrastruktur liegen selten individuelle vertragliche Vereinbarungen und häufig keine exklusive Nutzung der IT-Ressourcen vor. Dies führt dazu, wie die Studien der BITKOM [BK13] und von Crisp Research [Cri14] bestätigen, dass die Akzeptanz von öffentlichen Cloud Services, in Europa und insbesondere Deutschland, für den geschäftsmäßigen Einsatz noch sehr gering ist.

Die Studie der Crisp Research AG hebt hervor, dass 40% der Studienteilnehmer Cloud-Lösungen ablehnen, da sie selbst oder ihre Kunden Bedenken bezüglich der Sicherheit der Unternehmensdaten haben. Als weitere Argumente gegen eine Cloud-Lösung werden vor allem erstens der hohe Aufwand für den Betrieb eines Cloud-Services und zweitens die hohen Kosten für die Neuentwicklung eines solchen Services hervorgehoben. Weitere genannte Argumente gegen eine Cloud-Lösung sind: Erstens, der hohe Aufwand des Betrieb eines Cloud-Services und zweitens die hohen Kosten für die Neuentwicklung einer Cloud-Services. Neben den geäußerten Sicherheitsbedenken können somit auch fehlende Ressourcen für Entwicklung, Betrieb und Support und fehlendes Know-How gegen Cloud-Lösungen sprechen.

Um diesen Bedenken entgegenzuwirken, beschäftigten sich in den letzten Jahren zahlreiche Forschungsprojekte mit der Sicherheitsproblematik. Projekte waren auf Ebene der EU *TClouds*¹ und *Trusted Cloud*² auf Bundesebene. Im Rahmen eines Teilprojekts der Trusted Cloud Initiative entstanden seitens des Autors die folgenden Publikationen:

- P. Reinhold, W. Benn, B. Krause, F. Goetz, and D. Labudde, *Hybrid cloud architecture for software-as-a-service provider to achieve higher privacy and decrease security concerns about cloud computing*, in CLOUD COMPUTING 2014, The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization, 2014, pp. 94–99
- P. Reinhold, W. Benn, B. Krause, F. Goetz, and D. Labudde, *Introducing a scalable encryption layer to address privacy and security issues in hybrid cloud environments*, in INTERNATIONAL JOURNAL ON ADVANCES IN SOFTWARE, 2014, Vol. 7, No. 3 and 4, pp. 727–739

Die genannten Arbeiten bieten einen Lösungsansatz, für das sichere Auslagern von Daten, als einen Teilaspekt der Cloud Computing Sicherheit und fügen sich damit folgerichtig in die Untersuchung dieser Arbeit ein, in welcher jedoch alle relevanten Teilbereiche des sicheren Cloud Computings untersucht und bewertet werden.

Das Ziel dieser Arbeit ist die Untersuchung einer breiten Basis von verschiedenartigen Technologieansätzen nach gemeinsamen Kriterien, um eine Vergleichbarkeit zu ermöglichen. Auf dieser Grundlage erfolgt im Anschluss die Bewertung der Praxistauglichkeit.

¹<http://www.tclouds-project.eu> letzter Zugriff 06.07.2015

²<http://www.trusted-cloud.de> letzter Zugriff 06.07.2015

1.1. Thesen der Arbeit

Die Thesen der vorliegenden Arbeit ergeben sich aus folgender, grundlegender Fragestellung, die im Rahmen dieser Arbeit beantwortet wird:

Ist es effizienter nutzerseitige Maßnahmen zum Schutz von Daten und Prozessen zu etablieren oder sollte dies durch den Anbieter des Cloud Services erfolgen?

Aus dieser Frage leiten sich folgende drei Fakten ab. Erstens: Es ist eine Unterscheidung zwischen dem Nutzer und dem Anbieter der Cloud notwendig. Zweitens: Zur Beantwortung dieser Frage muss idealerweise eine möglichst große Anzahl an Maßnahmen zum Schutz von Daten und Prozessen vorliegen. Drittens: Zur Bewertung der Effizienz sind technisches Verständnis sowie eine Vergleichsmöglichkeit der Maßnahmen erforderlich. Unter Annahme dieser drei Fakten ist diese Arbeit entstanden. Ferner leiten sich aus dieser Fragestellung folgende Thesen ab.

These 1. *Die Charakteristiken des Cloud Computings stehen im Widerspruch zu bestehenden Datensicherheits- und Datenschutzanforderungen.*

These 2. *Der notwendige Mehraufwand durch die Schutzmaßnahme ist vom Einfluss auf das gesamte Cloud-System abhängig.*

In der Arbeit wird ferner das *Prinzip der Delegation mit begrenztem Wissen* postuliert, welches sich aus der Fragestellung und den zwei aufgestellten Thesen ableitet.

Prinzip der Delegation mit begrenztem Wissen

In der Einleitung wurde verdeutlicht, dass Cloud Computing eine Form der Auslagerung darstellt und das Ziel des Nutzers eines Cloud-Services somit darin besteht, Daten, Prozesse oder Berechnungen in eine Cloud-Umgebung auszulagern. Mögliche Gründe können dafür sein, dass Daten zu groß sind, um diese lokal zu speichern oder die Berechnung zu aufwändig ist, um diese lokal auszuführen. Dabei ist der Nutzer nicht bereit, Daten oder Prozesse ungeschützt in die Cloud-Umgebung zu überführen, da er dem Cloud-Anbieter nicht oder nur begrenzt vertraut. Deshalb ergreift der Nutzer wirksame Maßnahmen zum Schutz von Daten und Prozessen, die das Wissen des Cloud-Anbieters über diese Daten begrenzen. Dieser Prozess unterliegt dabei dem *Prinzip der Delegation mit begrenztem Wissen*:

Je mehr Arbeit delegiert werden soll, aber je weniger Wissen darüber, desto mehr Arbeit hat der Delegierende. Diese Mehrarbeit kann durch Informationspreisgabe reduziert werden.

Wie das Prinzip verdeutlicht, führt die Auslagerung von Arbeit bei gleichzeitiger Wissensbegrenzung zum Anstieg der Arbeitslast des Auslagernden, um die Aufgabe vollständig zu erledigen. So ist es einem Anbieter, der kein Wissen oder Informationen über gespeicherte Daten besitzt, nicht möglich, diese zu durchsuchen. Möchte der Nutzer einen Teil der Daten, der einem bestimmten Suchkriterium entspricht, muss er alle Daten vom Cloud-Anbieter beziehen und die Suche selbst durchführen. Ein häufiges Durchsuchen der Daten würde somit einer Auslagerung der Daten widersprechen. Diesem Extremfall kann mit geeigneten Maßnahmen wie der Strukturierung und Indizierung entgegengewirkt werden. Gleichzeitig werden jedoch infolgedessen Informationen an den Anbieter übertragen. Beispiele sind: Informationen über Zugriffszeiten und -muster, Datengrößen und -struktur. Diese Art der Informationen werden auch als Metadaten bezeichnet.

1.2. Struktur der Arbeit

Die Arbeit ist in vier Teile untergliedert. Der erste Teil beschreibt die grundlegenden Konzepte und Begrifflichkeiten die in der Arbeit verwendet werden. Dabei geht es insbesondere um die in Abschnitt 2.3 eingeführte Taxonomie der Cloud-Sicherheit, denn Aufbau und Verständnis der weiteren Arbeit basieren maßgeblich auf dieser Struktur.

Der zweite Teil der Arbeit beschäftigt sich mit der Analyse und Diskussion von State-of-the-Art-Methoden und verdeutlicht die Ansätze der innerhalb der Taxonomie identifizierten Teilbereiche der Cloud Computing Sicherheit.

Der dritte Teil stellt die spezifizierten Kriterien vor, nach denen die im zweiten Teil der Arbeit beschriebenen Technologien evaluiert werden. Weiterhin beinhaltet dieser Teil Auswertung der Ergebnisse der Evaluierung sowie die Definitionen der Begriffe des potenziellen Nutzens, der Praktikabilitätsaussage und der Realisierungseffizienz.

Abschließend fasst Kapitel 8 die Ergebnisse der Arbeit bezüglich der aufgestellten Thesen zusammen und formalisiert das Prinzip der Delegation mit begrenztem Wissen.

2.1. Cloud Computing

Der Cloud Computing Begriff wird im Rahmen dieser Arbeit nach NIST [MG11] definiert. Diese häufig verwendete Definition, u.A. von [BIT10, BSI12, MPB⁺12, CSA11, Win10, CK10, Bas12, BGJ⁺13] fasst die notwendigen und charakteristischen Merkmale von Cloud Computing zusammen.

Begriff 1 (Cloud Computing). *ist ein Modell, dass es nutzergesteuert erlaubt, bei Bedarf jederzeit und überall über ein Netz auf einen geteilten Pool von konfigurierbaren Rechnerressourcen zuzugreifen, die schnell und mit minimalem Managementaufwand oder geringer Serviceprovider-Interaktion zur Verfügung gestellt werden können.*

Die Diskussion über diesen Begriff als neue Bezeichnung für eine bekannte und etablierte Technologie ist insofern unzutreffend, da per Definition nicht die technischen Details entscheidend sind, sondern die bedarfsgerechte, flexible und vom Nutzer gesteuerte Darbietung von Ressourcen oder Services. Um dies zu ermöglichen, finden sich neue und bereits etablierte Technologien zusammen. Cloud Computing als Marketingbegriff zu bezeichnen, ist damit ebenso unzutreffend, wie zu behaupten Cloud Computing sei etwas völlig Neues. Die hohe Bekanntheit des Cloud Computing Begriffs führt jedoch dazu, dass dieser oftmals für Marketingzwecke missbraucht wird. Dieser Versuch, das Marketing um ihre Produkte oder Dienstleistungen mit dem Begriff Cloud zu verstärken, wird in der Fachwelt als Cloud Washing bezeichnet.¹ Zudem sammeln sich um diesen Begriff zahlreiche weitere, so dass häufig pauschal alles als Cloud oder Cloud Computing bezeichnet wird. Die folgenden Abschnitte klären Begriff und deren Einordnung im Cloud Computing Umfeld und bilden damit zugleich die Grundlage für die nachfolgenden Kapitel dieser Arbeit. Dazu sind innerhalb der Arbeit folgende Rollen im Cloud-Umfeld definiert:

Begriff 2 (Cloud Service Anbieter, Cloud Service Provider, CSP). *bezeichnet einen Akteur, der einen Cloud Service (gegen Gebühr) anbietet und damit als Dienstleister agiert.*

Begriff 3 (Cloud User, Cloud Nutzer). *bezeichnet einen Akteur, der die Dienste eines Cloud Providers als Endverbraucher nutzt.*

Begriff 4 (Cloud Vermittler, Cloud Prosumer). *bezeichnet einen Akteur, der einerseits die Dienste eines Cloud Providers nutzt und andererseits ebenfalls als Cloud Provider, Dienste zur Verfügung stellt.*

Ein feinere Unterteilung bzgl. der Involvierten im Cloud Computing unternehmen Lui et al. [LTM⁺11, p. 19]. Die vorliegende Arbeit ist gleichermaßen für alle Akteure im Cloud Computing interessant, wobei Leser mit Vorkenntnissen im Cloud-Umfeld direkt auf den Abschnitt Sicherheit 2.2 verwiesen seien.

¹vgl. <http://clouduser.de/kommentar/top-cloud-computing-washer-diese-unternehmen-sagen-die-unwahrheit-uber-ihre-produkte-21125>

Cloud-Service-Modelle

Die Servicemodelle beschreiben die Art und Weise wie Dienste angeboten bzw. genutzt werden können. Im Folgenden werden drei grundlegenden Servicemodelle vorgestellt, wobei Cloud Provider auch mehrere dieser Modelle anbieten können. Die Abbildung 2.1 zeigt eine Übersicht der nachfolgend vorgestellten Modelle.

Präsentation Tier			SaaS
Business Tier		PaaS	
Daten Tier			
Betriebssystem/ VM-Images	IaaS	kein Zugriff	kein Zugriff
Konfiguration (CPU/Speicher)	umfangreiche Auswahl	eingeschränkte Auswahl	
Virtualisierung	kein Zugriff	kein Zugriff	
Hardware			

Abbildung 2.1.: Darstellung der drei grundlegenden Cloud-Servicemodelle: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) und Software-as-a-Service (SaaS). Die Servicemodelle beschreiben die Art und Weise wie Cloud-Dienste genutzt werden können und in welchem Bereich des Abstraktionsmodells der Cloud-Nutzer Zugriff oder Auswahlmöglichkeiten erlangt.

Begriff 5 (Infrastructure-as-a-Service (IaaS)). Dem Cloud-Nutzer werden grundlegende, virtuelle Hardwareressourcen (Prozessoren, Hauptspeicher, Plattenspeicher) angeboten. Mittels Managementkonsole lassen sich Ressourcen je nach Bedarf erstellen, starten, beenden bzw. löschen. Der Nutzer des Services kann in selbst erstellten, virtuellen Instanzen beliebige Software installieren. Der Nutzer hat keinen Zugriff auf die Virtualisierungsschicht oder die Hardware.

Begriff 6 (Platform-as-a-Service(PaaS)). Dem Cloud-Nutzer wird eine Entwicklungsplattform bereitgestellt. Der Provider bietet dabei eine Auswahl an Programmiersprachen, Frameworks, Bibliotheken und Werkzeugen an. Der Nutzer kann grundlegende Einstellungen zur Infrastruktur tätigen, beispielsweise eine Auswahl an Instanzen auf der entwickelte Applikationen ausgeführt wird. Einstellungen an Instanzen, welche die Applikationsbestandteile ausführen, sind nur eingeschränkt möglich. PaaS-Anbieter übernehmen Compiler- und Deployment-Aufgaben und entlasten somit die Softwareentwickler.

Begriff 7 (Software-as-a-Service (SaaS)). Dem Cloud-Nutzer wird eine Applikation bereitgestellt, die ihm der CSP auf Grundlage einer Cloud Architektur anbietet. Der Nutzer kann nur geringe Anpassungen in der Applikation vornehmen und hat bzgl. der unteren Schichten weder Zugriff noch Einstellungsmöglichkeiten.

Es existiert heute eine Vielzahl an CSP. Dabei bieten manche Provider, wie ProfitBricks², ausschließlich ein Servicemodell an, während hingegen Anbieter wie Amazon Web Service³ mehrere Modelle bereitstellen. Im letzteren Fall sind die Services der PaaS- und SaaS-Ebene zumeist kostenlos nutzbar und es werden nur die Kosten frakturiert, die auf der IaaS-Ebene anfallen.

²<https://www.profitbricks.de> letzter Zugriff 07.08.2015

³<http://aws.amazon.com/de> letzter Zugriff 07.08.2015

Cloud-Deployment-Modelle

Die Deployment-Modelle beschreiben die Art und Weise in der sich die IT-Infrastruktur der angebotenen Services befindet, wobei in drei Grundmodelle unterschieden wird. Die BITKOM [BIT10, p. 16ff] und die Cloud Security Alliance⁴ unterscheiden hingegen weitere Mischtypen, die sich jedoch durch Kenntnis dieser drei Grundtypen herleiten lassen. Die Abbildung 2.2 zeigt eine einfache Übersicht der drei Grundmodelle.

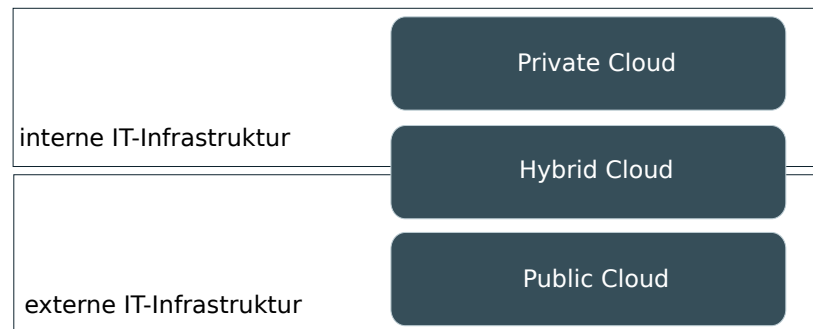


Abbildung 2.2.: Darstellung der drei grundlegenden Cloud-Deployment-Modelle: Private Cloud, Hybrid Cloud und Public Cloud. Das erste Modell befindet sich ausschließlich auf interner IT-Infrastruktur, das letztere dagegen vollständig auf externer Infrastruktur, die nicht unter dem Einfluss des Cloud-Nutzers steht. Das hybride Modell stellt eine Mischform aus interner und externer IT-Infrastruktur dar.

Begriff 8 (Private Cloud). Die Cloud-Infrastruktur (Hardware) befindet sich im Privatbesitz des Nutzers, der als Eigentümer und exklusiver Benutzer die Infrastruktur verwaltet, administriert und gepflegt. Dies stellt keine Form des Outsourcings dar.

Begriff 9 (Public Cloud). Die Cloud-Infrastruktur wird der Öffentlichkeit und damit externen Kunden angeboten. Obwohl es keine exklusive Nutzung gibt, wird durch Virtualisierung und nutzerbasierte Ressourcenverwaltung jedoch ein solcher Eindruck erzeugt. Die Public Cloud-Infrastruktur wird vom Cloud Provider verwaltet, administriert und gewartet. Es ist eine Form des vollständigen Outsourcings.

Begriff 10 (Hybrid Cloud). Diese Cloud-Infrastruktur stellt eine Mischform aus Private und Public Cloud dar und ist als Kombination aus exklusiven und öffentlich angebotenen Ressourcen zu verstehen. Damit ist dies als eine Form eines teilweisen Outsourcings zu bezeichnen.

Die Service- und Deployment-Modelle stellen eine Strukturierungsgrundlage für Cloud-Services dar und sind beliebig miteinander kombinierbar. Als verdeutlichendes Beispiel sei hier eine SaaS-Lösung skizziert: Der Cloud-Nutzer nutzt eine SaaS-Anwendung, welche durch einen Cloud-Vermittler mit Hilfe einer hybriden PaaS-Lösung entwickelt und betrieben wird. Bei normalen Lastverhältnissen werden die Applikationsserver des Cloud-Vermittlers sowie ausgelagerte Datenbankserver genutzt. Zu Spitzenlastzeiten werden kurzfristig weitere Applikations- und Datenbankserver vom CSP des Cloud-Vermittlers beansprucht. Der Vorteil für den Cloud-Nutzer besteht in einer hohen Verfügbarkeit und Leistungsfähigkeit des Systems in allen Lastzeiten. Ein Vorteil für den Cloud-Vermittler liegt in guten Entwicklungs- und Wartungsmöglichkeiten durch Redundanz sowie eine verbesserte Sicherung der Datenhaltung durch die Auslagerung. Bei Normalauslastung decken eigene Hardwareressourcen den Bedarf und durch die Kopplung an den CSP werden Sonderfälle wie Wartungs oder Spitzenlastzeiten abgedeckt. Wollte der Cloud-Vermittler derartige Fälle selbst abdecken, wäre ein Vielfaches an Personal und Hardware erforderlich.

⁴<https://cloudsecurityalliance.org>

Vorteile von Cloud Computing

Ziel des Cloud Computing ist es, die Entwicklung von bedarfsorientierten und kostenoptimalen Systemen zu ermöglichen, die vom Benutzer gesteuert werden. Die vom National Institute of Standards and Technology [MG11] und Buest [Bue10] beschriebenen Charakteristika entsprechen diesen Anforderungen und verdeutlichen damit die nachfolgend dargestellten Vorteile, die sich durch die Nutzung von Cloud Computing ergeben können:

1. **Kosteneinsparung:** Die Investitionskosten für Hardware werden geringer.
2. **Kostenoptimierung:** Es erfolgt eine Umwandlung von Investitionskosten zu Betriebskosten. Der Return on Invest (ROI) ist kürzer.
3. **On Demand:** Die Ressourcen werden zum benötigten Zeitpunkt bezogen und im Anschluss nach Bedarf wieder abgegeben.
4. **Pay-as-you-go:** Es werden nur die Ressourcen bezahlt, die tatsächlich genutzt wurden. Dabei wird abhängig vom Service- und Deployment-Modell pro Benutzer, pro Gigabyte oder pro Zeiteinheit abgerechnet.
5. **Keine feste Grundgebühr oder feste Bindung:** Bei einem idealen Cloud Computing Angebot ist keine monatliche/ jährliche Grundgebühr zu bezahlen und es bestehen keine zwingend langfristigen Bindungen.
6. **Hohe Verfügbarkeit/Zuverlässigkeit:** Die benötigten Ressourcen stehen zum erforderlichen Zeitpunkt verlässlich zur Verfügung.
7. **Hohe Skalierbarkeit:** Die Ressourcen lassen sich automatisiert neuen Bedürfnissen anpassen.
8. **Blackbox-Prinzip:** Die Umsetzung des Cloud Angebots ist nicht entscheidend. Der Service wird über eine offene, gut dokumentierte Schnittstelle genutzt. Ziel dieses Prinzips ist die Konzentration auf das Kerngeschäft.
9. **Automatisierung:** Nach einer Grundeinrichtung bzgl. der Bedürfnisse, sind keine weiteren manuellen Eingriffe während der Nutzung des Angebots notwendig.
10. **Zugriff über das Internet:** Weltweiter Zugriff auf den Service über das Internet. Diese Zentralisierung vereinfacht Software-Updates und ermöglicht Mobilität sowie Plattform- und Geräteunabhängigkeit.
11. **Sicherheit:** CSP bieten einen IT-Grundschutz, den sich vor allem KMUs selten leisten können und damit ein vergleichsweise hohes Datenschutzniveau.

Nachteile von Cloud Computing

Die Erreichung solcher Ziele wie Flexibilität, Bedarfsorientierung und Kostenoptimierung ist aber auch mit zahlreichen Problemen verbunden. In diesem Zusammenhang verweist Buest [Bue13] vor allem darauf, dass neben dem häufig ausschlaggebenden Einsparungen der Infrastrukturkosten, neu entstehende Kosten durch Personal mit notwendigen Kenntnissen und die Kosten für die Entwicklung von skalierbaren und ausfallsicheren Applikationen⁵ nicht vernachlässigt werden dürfen. Neben den Vorteilen der Cloud Nutzung können sich mit der Nutzung auch die nachfolgend dargestellten Nachteile ergeben:

1. **Zugriff über das Internet:** Anwendungen in der Cloud setzen eine Internetverbindung voraus. Zahlreiche Ausfallszenarien im Internet können zu einem Teil- oder Gesamtausfall der Cloud-Applikation führen.
2. **Datenschutz und Datensicherheit:** Cloud-Anwendungen stellen neue Herausforderungen an die Datensicherheit und den Datenschutz.
3. **Economic-Denial-of-Service-Attacks:** Diese Form des Angriffs kann Schaden in Form von Kosten für den Cloud Vermittler erzeugen.
4. **Lock in-Effekte:** Dieser Effekt tritt generell auf, wenn durch CSP angebotene Services genutzt, und in eigene Systeme integriert werden. Das Wechseln des Providers ist zumeist mit hohen Kosten verbunden. Fehlende Standards im Cloud Computing unterstützen diesen Effekt.
5. **Kostenexplosion bei Fehlern:** Durch automatisch, skalierende Cloud-Services können durch Softwarefehler oder falsche Konfigurationen ungewollt hohe Kosten entstehen.
6. **Transparenz/Verifikation:** Die Kehrseite des Blackbox-Prinzips sind fehlende Transparenz und Verifikationsmöglichkeiten.
7. **Compliance, Gouvernance und allgemeiner Kontrollverlust:** Durch das Blackbox-Prinzip leidet die Nachvollziehbarkeit und Durchsetzung unternehmensinterner Richtlinien.
8. **Rechtsgrundlage:** Durch international agierende CSP mit anhängenden Subunternehmen, internationalen Datentransfer und fehlenden Richtlinien ergeben sich unklare Rechtsgrundlagen.
9. **Insolvenz des Providers:** Fehlenden Verträge und langfristige Vereinbarungen verhindern eine Absicherungen gegen die Insolvenz des CSP.⁶
10. **Verzicht auf eigene IT-Kompetenz:** Durch den Verzicht auf IT im Unternehmen wird diesbezüglich keine Know-How aufgebaut, so dass sich eine Abhängigkeit zu Cloud-Dienstleistern ergibt. Zudem verschieben sich die Kenntnisse des IT-Personals in Richtung IT-Management und Services.

Dabei ergeben sich einige Nachteile durch die geforderten Ziele und Vorteile. So sind die Nachteile fehlender Transparenz und Verifizierbarkeit durch den Blackbox-Charakter der Cloud bedingt. Diese Widersprüche führen zur Notwendigkeit einer Kompromissbereitschaft: Entweder es wird auf einen Teil der Transparenz verzichtet oder diese ist nur durch Einführung eines dementsprechenden Kostenfaktors zu erreichen.

⁵Näher Informationen zum diesem Thema bietet Wilder [Wil12].

⁶Die Verantwortung über die Daten von Dritten bleibt beim Cloud-Nutzer [SB12].

2.2. Cloud Computing und Sicherheit

Der Begriff der Sicherheit innerhalb der IT-Branche ist breit gefächert und komplex. Als Grundlage für die Verwendung in dieser Arbeit dienen die nachfolgend eingeführten Begriffe. Der Sicherheitsbegriff wird nach der Vorlage von Schneier [Sch11b] verwendet, nach dessen Meinung sich die Sicherheit eines Systems in drei entscheidende Faktoren aufteilt. Fehlt einer dieser Faktoren, ist das System nicht mehr als sicher anzusehen. Die Abbildung 2.3 stellt den Sicherheitsbegriff von Schneier schematisch dar.

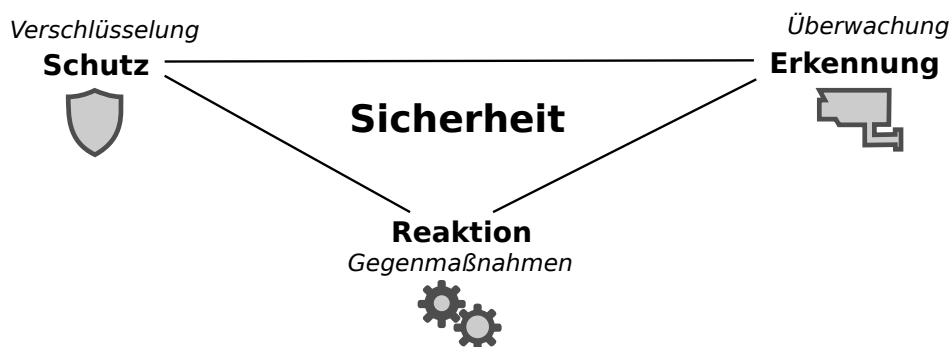


Abbildung 2.3.: Das Prinzip der Systemsicherheit nach Schneier [Sch11b] teilt sich in die drei Faktoren Schutz, Erkennung und Reaktion auf. Nur bei Präsenz aller drei Faktoren kann ein System als sicher angesehen werden.

Schneier verweist insbesondere drauf, dass die Erkennung und Reaktion essentiell sind. Kein noch so sicheres System kann seinen Schutz und damit seine Sicherheit aufrechterhalten, wenn es Verstöße oder Angreifer nicht erkennt und keine entsprechende Gegenmaßnahmen ergreift. Diese Gegenmaßnahmen müssen nicht zwingend gegen Angriffe gerichtet sein, sondern können im einfachsten Fall bedeuten, das System stets auf dem aktuellsten Stand zu halten. Der Fokus dieser Arbeit liegt auf den Maßnahmen zum Schutz. Bedrohungsszenarien in Form von Maleware, Firewall- und Penetrationstests werden hingegen nicht vertiefend behandelt.⁷ Damit grenzt sich diese Arbeit vom Thema der Sicherheit von Netzwerken, Intrusion Detection und Begriffen wie Security Information and Event Management (SIEM) ab.

Datensicherheit

Der Datensicherheitsbegriff dieser Arbeit orientiert sich an den von Ertel [Ert03] definierten Hauptzielen zum Schutz von Informationen der modernen Kryptographie. Neben diesen Zielen kommt der Verfügbarkeit der Daten im Cloud-Umfeld eine besondere Bedeutung zu, die durch Ertel nicht berücksichtigt wird. Die Definitionen orientieren sich zudem am Terminus des National Institute of Standards and Technology [GR95].

Begriff 11 (Vertraulichkeit, Confidentiality). *Nur dazu berechnigte Individuen sollen in der Lage sein, die Daten oder die Nachricht zu lesen oder Informationen über ihren Inhalt zu erlangen*

Begriff 12 (Integrität, Integrity). *Die Daten müssen nachweislich vollständig und unverändert sein. Veränderungen dürfen nur nach festgelegter und autorisierter Art und Weise erfolgen.*

Begriff 13 (Verfügbarkeit, Availability). *Die Daten müssen erreichbar und damit abrufbar sein.*

⁷Eine Übersicht möglicher Angriffsszenarien bietet [Win10].

Im Zusammenhang mit diesen drei Begrifflichkeiten wird in der Fachwelt zumeist auf das vom National Institute of Standards and Technology [NIS02] definierte CIA-Prinzip verwiesen. Das in Abbildung 2.4 illustrierte Prinzip wird ebenso von Winkler [Win11, p. 14], Krutz [KD10, p. 125], Mather et al. [MKL09, p. 130] und Boamping [BA12] verwendet.

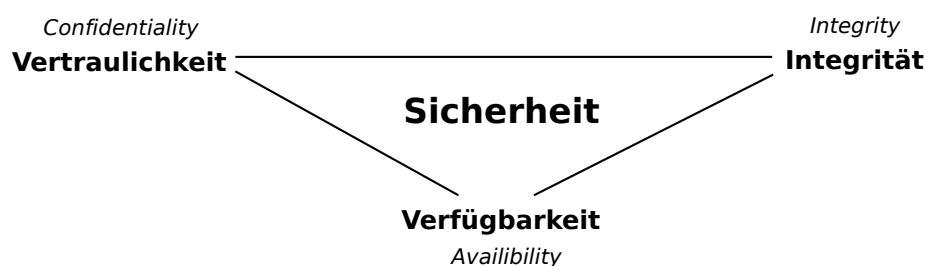


Abbildung 2.4.: Das CIA-Prinzip der Datensicherheit ist das Analogon zum Sicherheitsbegriff für Systeme nach Schneier. Nur durch Präsenz aller Faktoren: Vertraulichkeit, Integrität und Verfügbarkeit ist die Sicherheit der Daten gewährleistet.

Begriff 14 (CIA-Prinzip). *Das CIA-Prinzip beschreibt ein Sicherheitsprinzip von gespeicherten Daten. Nur durch gleichzeitige Präsenz aller drei Merkmale (Confidentiality, Integrity, Availability), kann die Datensicherheit gewährleistet sein. Die Merkmale sind von voneinander abhängig. Das CIA-Prinzip entspricht einer Verknüpfung von Kryptographie und modernen IT Anforderungen, wie Hochverfügbarkeit und Skalierung.*

Dieses Prinzip eignet sich sehr gut für die Bewertung von sicheren Cloud-Speicher-Services. Bugiel et al. [BSSS11] beschreiben zudem einen Ansatz für die Erweiterung dieses Prinzips auf Services innerhalb von Cloud-Umgebungen. Die Abhängigkeit der Merkmale sei durch folgendes Beispiel verdeutlicht. Um die Verfügbarkeit zu erhöhen, werden Daten trivialerweise mehrfach abgelegt. Mehrfach abgelegte Daten bieten potenziellen Angreifern jedoch eine größere Angriffsfläche, so dass die Vertraulichkeit mit steigender Verfügbarkeit sinkt. Wie Winkler [Win11] hervorhebt, erhöhen kryptographische Verfahren nicht notwendigerweise alle hier genannten Merkmale. Die Verfügbarkeit ist zudem kein Merkmal der Kryptographie und kann darüber hinaus weitere Anforderungen wie Hochverfügbarkeit, Back-Ups und Archivierung (Langzeitspeicherung) stellen. Die kryptographischen Merkmale stehen der Verfügbarkeit zudem häufig entgegen und erhöhen damit die Komplexität der Lösungsansätze.

Sichere Verarbeitung und Berechnung von Daten

Da Cloud-Umgebungen nicht nur als Datenspeicher dienen, sondern auch Prozesse, Geschäftslogik und vollständige Applikationen abbilden, besteht die Anforderung auch diese sicher abzubilden. Dieses Vorhaben ist jedoch weit komplexer als ausschließlich die Datenhaltungsschicht abzusichern. Trotz intensiver Forschung und vielversprechender Erfolge existieren bisher keine allgemeingültigen Lösungsansätze um beliebige Informationen sicher zu verarbeiten. Dennoch ist dieser Forschungsbereich in den letzten Jahren vor allem durch Gentry [Gen09] weiter vorangeschritten. Diese Arbeit bietet im Abschnitt 4.2 mit Ausführungen zur sicheren Applikationslogik einen Überblick dieses Verfahrens und beleuchtet weitere praktisch relevante Forschungsergebnisse.

Die Abbildung 2.5 stellt einen schematischen Überblick über Verfahren dar, die es erlauben, Daten zu verschlüsseln und gleichzeitig jedoch weiterhin eine Verarbeitung dieser Daten zu ermöglichen. Zum Vergleich werden ebenfalls probabilistische Verfahren, wie AES, und unverschlüsselter Klartext ebenfalls aufgetragen, um eine bessere Einordnung zu ermöglichen. Die einzelnen Verfahren werden in den folgenden Kapiteln der Arbeit näher beleuchtet.

Sicherheit

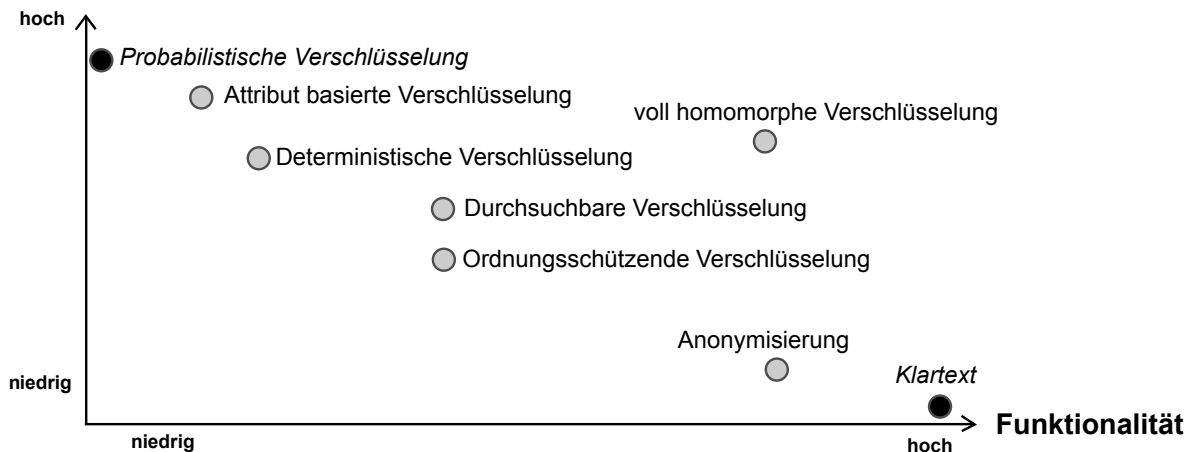


Abbildung 2.5.: Vergleich der Sicherheit und Funktionalität von Verfahren der sicheren Verarbeitung. Zur besseren Orientierung sind probabilistische Verfahren sowie unverschlüsselter Klartext ebenfalls aufgetragen. Zur erkennen ist, dass mit zunehmender Funktionalität die Sicherheit abnimmt.

Die Abbildung 2.5 dient der Verdeutlichung, dass mit zunehmender Funktionalität die Sicherheit des Verfahrens abnimmt, da Metainformationen zum Klartext einem Angreifer verfügbar werden. Das Prinzip der Delegation mit begrenztem Wissen wird bereits deutlich. Je mehr Funktionalität gefordert wird, desto mehr Informationen müssen preisgegeben werden. Ist der Nutzer dazu nicht bereit, kann dieser die Funktionalität nicht auslagern und muss diese auf Nutzerseite selbst abbilden.

2.3. Taxonomie der Cloud-Sicherheit

Die allgemein anerkannten Klassifizierungen im Cloud Computing wurden im Abschnitt 2.1 vorgestellt und diskutiert. Darüber hinaus gibt es in der Fachwelt keine weiteren Standards oder einheitlichen Einteilungen. Es existieren jedoch Vorschläge seitens der Cloud Security Alliance [CSA11], der Europäische Agentur für Netz- und Informationssicherheit [ENI12] und des National Institute of Standards and Technology [LTM⁺11]. Einen weiteren Vorschlag unterbreiten der Autor Gonzalez et al. [GMR⁺12], welcher zudem auf die starke Kontextabhängigkeit des Sicherheitsbegriffs in Cloud-Lösungen hinweist. Gonzanlez et al. erzeugen ihre Taxonomie in Bezug auf Risiken und Schwachstellen in der Cloud erzeugt und unterteilen daran das Cloud- Sicherheitsthema. Eine weitere Einteilung nehmen Fernandes et al. [FSG⁺14] vor, deren umfangreiche Taxonomie alle wesentlichen Punkte enthält und sich mit der im Folgenden vorgestellten Taxonomie überschneiden.

Das Ziel der hier entwickelten Taxonomie ist eine Einteilung bisheriger Lösungsansätze und deren praktische Anwendung. Der Fokus ist somit praktisch orientiert und soll in erster Linie die Entwicklung sicherer Cloud-Systeme unterstützen. Aus diesem Grund orientiert sich die hier diskutierte Darstellung an jener von Fehling et al. [FLR⁺14] zum Thema Cloud Computing Patterns. Ferner orientiert sich die entwickelte Taxonomie inhaltlich an den Arbeiten von [CGJ⁺09, Ker12, BGJ⁺13, BBI⁺13]. Die Abbildung 2.6 illustriert die für diese Arbeit gültige Taxonomie der Cloud Computing Sicherheit.

Analog zu Fehling et al. [FLR⁺14] stehen die Cloud-Applikation und die Cloud-Umgebung im Mittelpunkt, denn diese stellen die Aufteilung in Bereiche dar, die durch den Nutzer bzw. Provider zu beeinflussen sind. Somit bezieht sich die Taxonomie unmittelbar auf die Fragestellung und Thesen aus Abschnitt 1.1 sowie auf das in diesen Abschnitt ebenfalls erläuterte Prinzip der Delegation mit begrenztem Wissen.

Eine Cloud-Applikation besteht aus einer Präsentationsschicht, welche die Schnittstelle zum Cloud-Nutzer darstellt, einer Schicht, welche die Geschäftslogik abbildet und einer Datenschicht, welche die Bereitstellung und Speicherung der Daten übernimmt. Ob diese 3-Tier-Architektur im praktischen System derartig Anwendung findet, ist für die Betrachtung innerhalb der vorliegenden Arbeit unerheblich und ist primär der Strukturierung förderlich. Die Cloud-Applikation befindet sich innerhalb einer Cloud-Umgebung, welche Ressourcen für die Applikation bereitstellt. Neben Hardwareressourcen bietet die Cloud-Umgebung zudem weiterhin eine Virtualisierungsumgebung (IaaS) oder Entwicklungsumgebung (PaaS) an. Auf Letzteres wird im Rahmen der Arbeit nicht tiefer eingegangen, da PaaS-Umgebungen in Bezug auf sicherheitstechnische Anpassungen durch Vorgaben des CSP sehr stark eingeschränkt sind.

Erkennbar ist, dass sowohl die Cloud-Applikation als auch die Umgebung eine Form der Verwaltung benötigen, die mit der Box *Cloud Security Management* verbunden ist und welche die wesentlichen Faktoren im Sicherheitsmanagement beinhalten. Analog ist dies auch für die Lösungsansätze zur Datenspeicherung zu verstehen, da diese ebenfalls innerhalb der Cloud-Umgebung eingesetzt werden können, um Nutzern sichere Speicherlösungen als Service anzubieten. Die Boxen dienen zudem der kompakte Darstellung der Kapitel des zweiten Teils der Arbeit. Zur besseren Orientierung wird zu Beginn des jeweiligen Kapitels die entsprechende Box mit detaillierteren Informationen erneut dargestellt.

Im Ergebnis visualisiert die Abbildung 2.6 die Struktur der Arbeitsinhalte. Darüber hinaus wird diese Taxonomie sowie die Einteilung in Cloud Applikation und Cloud Umgebung, auf Grund des Thesenbezugs aus Kapitel 1.1 regelmäßig referenziert.

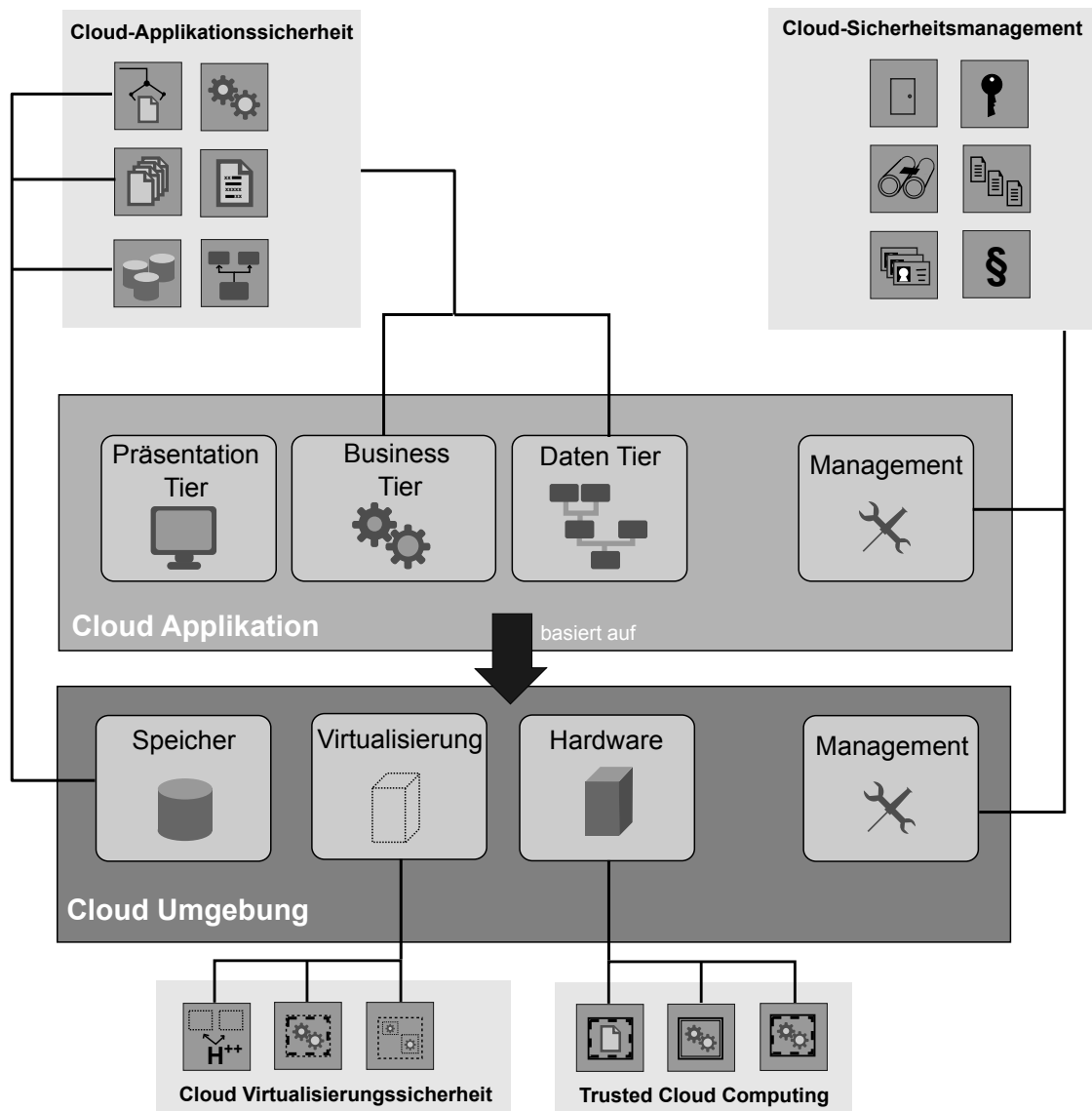


Abbildung 2.6.: Die Taxonomie der Cloud-Sicherheit verdeutlicht den Zusammenhang zwischen Aspekten der IT-Sicherheit und des Cloud Computing. Erkennbar ist die Aufteilung in Cloud-Applikation und Cloud-Umgebung. Erkennbar ist eine 3-Tier-Architektur der Cloud-Applikation, die zeigt welche Sicherheitsmaßnahmen ergriffen werden können. In der ebenfalls aufgeteilten Cloud-Umgebung werden analog verknüpfte Sicherheitsmaßnahmen dargestellt. Damit visualisiert die Abbildung die Struktur der Inhalte und des Aufbaus der vorliegenden Arbeit.

2.4. Cloud Computing und IT-Compliance

Im Kapitel wird das Spannungsfeld zwischen Cloud Computing und der Einhaltung von gesetzlichen oder vertraglichen Regelungen beschrieben. Es verdeutlicht datenschutzrechtliche Probleme die bei der Verarbeitung von personenbezogenen Daten in Cloud-Umgebungen entstehen können. Der Fokus der Diskussion liegt auf der deutschen und europäischen Rechtsgrundlage, die mit internationalen Regelungen verglichen werden. Zudem befasst sich der Abschnitt mit den Grundlagen von Service Level Agreements und diskutiert abschließend verschiedene Cloud-Zertifikate, mit denen sich CSP auszeichnen können.

2.4.1. Cloud Computing und Datenschutz

Die folgend beschriebenen Aussagen dienen der Referenzierung und Klärung eines gemeinsamen datenschutzrechtlichen Verständnisses im Bereich des Cloud Computings. Das *Unabhängigen Landeszentrum für Datenschutz Schleswig-Holstein* (ULD) veröffentlichte bisher zahlreiche, empfehlenswerte Zusammenfassungen und Beiträge wie [Wei10, Han12, BM12, Pau11].

Nationale Regelung - Bundesdatenschutzgesetz Gemäß §2 Abs. 4 BDSG müssen nicht öffentliche Stellen beim Einsatz von Cloud Computing die Vorschriften des BDSG beachten. Im Artikel 2 Abs. 1 Grundgesetz (GG) in Verbindung mit Art. 1 Abs. 1 GG ist der Datenschutz des Einzelnen konkret festgeschrieben und als *Recht auf informationelle Selbstbestimmung* definiert. Vor diesem rechtlichen Hintergrund sind Datenerhebungen nur mit Einwilligung des Betroffenen oder aus überwiegendem Allgemeininteresse zulässig. Dies gilt ebenfalls für die Verarbeitung von personenbezogenen Daten (§4a Abs. 1 BDSG). Schmidt [SB12] bezeichnet die Einwilligung gemäß §4a BDSG hingegen als praxisuntauglich, da im Cloud-Umfeld meist eine Vielzahl von Unterauftragsnehmern beteiligt sind, und es dem Servicenutzer möglicherweise unbekannt ist, auf welchen Servern Daten gespeichert werden.

Dennoch kann die Datenverarbeitung infolge der Beauftragung durch den Nutzer einer Cloud Services i.S.d. §11 BDSG datenschutzrechtlich möglich werden, so dass dieser Regelung im nationalen Kontext gegenüber der im §28 BDSG Vorzug zu geben ist. So kann Cloud Computing als Auftragsverhältnis i.S.v. §11 BDSG zwischen dem CSP als Auftragsnehmer und dem Nutzer als Auftraggeber angesehen werden. Als Konsequenz muss sich der Auftragsnehmer an die Weisungen des Auftraggebers halten, der wiederum für alle Datenschutzvorschriften verantwortlich ist. Würde der Auftragsnehmer von diesen Weisungen abweichen, fällt dies unter §4 Abs. 1 BDSG und verlangt nach der Einwilligung des Betroffenen. Eine Mustervereinbarung zum Datenschutz in Auftragsverhältnissen nach § 11 BDSG findet sich im Internet.⁸

Da der Auftraggeber für alle Datenschutzvorschriften verantwortlich ist, fordert §11 Abs. 2 S. 1 BDSG, den Auftragsnehmer unter Prüfung der seinerseits getroffenen technischen und organisatorischen Maßnahmen mit Sorgfalt auszuwählen. Dabei wird der Auftragsnehmer im Cloud-Umfeld diesen Forderungen kaum nachkommen können (bzw. wollen), so dass entsprechend Möglichkeiten der Zertifizierungen und Auditierung gemäß §9a BDSG geschaffen wurden. Wie Schmidt [SB12] feststellt, wird im Falle der ISO-27001 nur die Einhaltung der Datensicherheit und nicht das Einhalten des Datenschutzes zertifiziert. Ferner ist für die privilegierte Auftragsdatenverarbeitung nach §11 BDSG eine kontinuierliche Vorlage von Prüfberichten und Datenschutzaudits nötig.

Der §28 BDSG bietet weitere Möglichkeit der datenschutzrechtlich zulässigen Übermittlung von Daten in die Cloud. Hierbei ist die Anwendung von §28 Abs. 1 S. 1 Nr. 2 BDSG theoretisch als datenschutzrechtlich zulässig einzustufen. Laut Schmidt muss danach berechtigtes Interesse seitens des

⁸Siehe https://www.datenschutz.hessen.de/mustervereinbarung_auftrag.htm, letzter Zugriff 18.09.2014

CSP erforderlich sein und kein Grund zur Annahme bestehen, dass das schutzwürdige Interessen des Nutzer durch die Verarbeitung oder Nutzung der personenbezogenen Daten, die Interessen des CSP überwiegen. Berechtigtes Interesse ist in diesem Zusammenhang als entstehende wirtschaftlicher Vorteil zu verstehen. Schutzwürdiges Interesse heißt, dass es keinerlei Beeinträchtigungen des Persönlichkeitsrechts seitens des Betroffenen gibt. Insofern der CSP ein verlässlicher Anbieter ist und die Daten nicht in die Kategorie besondere personenbezogene Daten i.S.v. §3 Abs. 9 BDSG fallen, kann die Interessenabwägung zu Gunsten der Verwendung von Cloud Computing fallen. Dennoch weist Schmidt [SB12] darauf hin, in der Praxis eine genaue Überprüfung des Einzelfalls vorzunehmen. Dies sind ebenfalls die Empfehlungen des BSI [BSI12].

Internationale Regelungen Internationale Datentransfers sind nach §4b Abs. 2 S. 2 BDSG generell verboten, insofern der Betroffene ein schutzwürdiges Interesse am Ausschluss der Übermittlung hat. Dabei wird bei der Übermittlung von Daten innerhalb der EU/EWR und in unsichere und sichere Drittstaaten unterschieden. Letztere sind die von der EU-Kommission festgelegte folgende Staaten: Schweiz, Kanada, Israel, Argentinien, Guernsey und die Insel Man. Für die Mitgliedstaaten des EWR gelten die Vorschriften der EU- Datenschutzrichtlinien 95/46/EG zum Schutz natürlicher Personen bei der Verarbeitung von personenbezogenen Daten und zum freien Datenverkehr.⁹ Der Datentransfer im EWR ist nach §4b Abs. 2 S. 1 BDSG für personenbezogene Daten erlaubt. Die Übermittlung besonderer persönlicher Daten i.S.v. §3 Abs. 9 BDSG ist gemäß §28 Abs. 6 BDSG jedoch grundsätzlich verboten.

Eine Sonderrolle nehmen die von der EU als unsicherer Drittstaat eingeschätzten USA ein. Für die USA wurde von der EU-Kommission die *Safe Harbor Zertifizierung* beschlossen.¹⁰ Diese soll in den USA ein angemessenes Datenschutzniveau, i.S.v. § 4b Abs. 2 Satz 2 BDSG, anerkennen, wenn sich das datenimportierende US-Unternehmen an die *Safe Harbor Principles* hält. Jedoch handelt sich um eine freiwillige Selbstzertifizierung und einer Registrierung in die *Safe Harbour List* des US-Handelsministeriums. Aus Grund unzureichender Kontrollen seitens der US-Behörden gibt es zahlreiche Kritiken an der Safe Harbour Zertifizierung, wie Marnau [MS11] und Schmidt[SB12, S.50-57] aufzeigen. Daher wird nach Beschluss des *Düsseldorfer Kreises* vom 28./29. April 2010 [Kre10] das datenexportierende Unternehmen verpflichtet, die Safe Harbor Zertifizierung des US-Unternehmens zu überprüfen. Kann zunächst ein angemessenes Schutzniveau angenommen werden, ist in einem zweiten Schritt ist zu prüfen, ob der Nutzung des Cloud-Dienstes nicht schutzwürdige Interessen des Betroffenen nach § 4b Abs. 2 S. 2 entgegenstehen. Generell dürfen, wie im nationalen Fall, die CSP kaum bereit sein, jedem Kunden ihre Datenschutzkonformität nachzuweisen, so dass Zertifizierungen und Auditierungen die einzige Möglichkeit darstellen, den Behörden ein angemessenes Datenschutzniveau nachzuweisen. Trotz zahlreicher Bemühungen nationaler und europaweiter Projekte¹¹ hat sich bisher kein internationaler Standard etabliert. Im Gegenteil: Es gehen Cloud Provider dazu über, Rechenzentren in Deutschland zu eröffnen. So wurde am 23.Oktober 2014 ein Amazon AWS Rechenzentrum in Betrieb genommen¹². Dies verdeutlicht die schwierige Sachlage für deutsche Unternehmen, datenschutzkonform Daten in die internationale Cloud auszulagern.

⁹Vgl. <http://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:31995L0046>, letzter Zugriff 27.10.2014

¹⁰Für Details siehe Schmidt [SB12] S. 47/48

¹¹Vgl. <http://www.tclouds-project.eu>, Marnau et al.[MSS⁺11] und <http://www.optimis-project.eu>, letzter Zugriff 15.10.2014

¹²<http://aws.amazon.com/de/region-frankfurt/>, letzter Zugriff 25.10.2014

Datenschutzziele

Zur Berücksichtigung der Vorgaben des Datenschutzes in der Evaluierung der Lösungsansätze, werden in diesem Abschnitt die Schutzziele im Datenschutz nach Bock und Meissner [BM12] eingeführt. Das CIA Prinzip der Datensicherheit kann, wie Abbildung 2.7 darstellt, leicht um die Datenschutzziele erweitert werden.

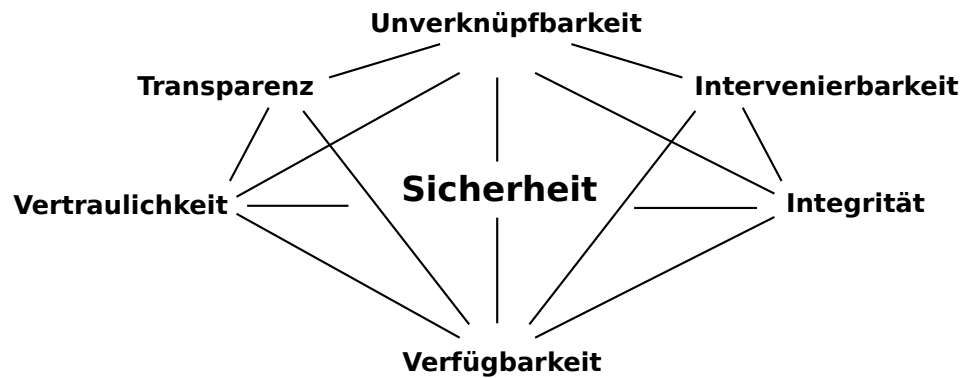


Abbildung 2.7.: Das CIA-Prinzip der Datenschutzziele ist die Erweiterung des CIA-Prinzips der Datensicherheit. Die Abbildung soll verdeutlichen, dass Datensicherheit und Datenschutz nur durch Präsenz aller sechs Faktoren gewährleistet ist. Alle dargestellten Faktoren werden als Kriterien für die Evaluierung von Technologien und Lösungsansätzen im Teil 3 dieser Arbeit genutzt.

Grundanliegen der Einführung von Schutzzielen ist es, diese als Kriterien für die Evaluierung von Technologien und Lösungsansätzen zu nutzen. Eine Auflistung entsprechender Kriterien erfolgt in Abschnitt 7.1 im Teil III der Arbeit. Ferner sei darauf hingewiesen, dass die ersten drei aufgeführten Schutzziele meist in IT-Systemen im Zusammenhang mit der Datensicherheit adressiert werden. Aus diesem Grund wurden die entsprechenden Begrifflichkeiten bereits im Abschnitt 2.2 eingeführt. Die nachfolgend beschriebenen Schutzziele werden mit Verweis auf die Vorschriften des Bundesdatenschutzgesetzes (BDSG) und der EU-Datenschutzrichtlinie (DSRL) in den Zusammenhang mit Cloud Computing gebracht.

Vertraulichkeit Der Begriff wurde bereits in Definition 11 eingeführt. Im Gesetzestext wird diese in Anlage Nr. 1-5 zu § 9 BDSG explizit hervorgehoben. Besonders S. 2 Anlage zu § 9 BDSG drückt dies aus: *Dabei sind insbesondere Maßnahmen zu treffen, die je nach Art der zu schützenden personenbezogenen Daten oder Datenkategorien geeignet sind.* Ebenso verankert ist die Vertraulichkeit in Artikel 16 DSRL und Artikel 17 S. 1 DSRL auf EU-Ebene. Die Sicherung der Vertraulichkeit ist der erste und häufigste Ansatz bei der Etablierung von sicheren Cloud-Umgebungen. Dazu gehören neben dem Gebäudeschutz des Datenzentrums (ISO 27001), Authentifizierungsverfahren und Identitätsmanagementsysteme. Diese und weitere Maßnahmen werden in Abschnitt 3 ausführlich erörtert. Weiterhin bieten sichere Cloud Dateisysteme Datenvertraulichkeit und integrieren zudem meist ein Rechtezugriffssystem. Detaillierte Ausführungen dazu enthält Abschnitt 4.4.

Integrität Der Begriff wurde in Definition 12 eingeführt. Das Integritätskriterium wird explizit in Anlage Nr. 3-5 zu § 9 Abs. 1 BDSG hervorgehoben, und betrifft die Zugriffskontrolle, Weitergabekontrolle und Eingabekontrolle. Auf EU Ebene wird in Artikel 6(d) DSRL darauf Bezug genommen. Die Integrität stellt ein weiteres typisches Merkmal der Datensicherheit in Cloud Umgebungen dar. Im Vergleich zu Vertraulichkeit ist die Datenintegrität jedoch selten explizit Forschungsgegenstand. Eine Ausnahme hierbei ist HAIL von Bowers et al.[BJO09]. Die Details sind im Abschnitt 4.4 zur Thematik der *sicheren Cloud-Dateisysteme* enthalten.

Verfügbarkeit Der Begriff wurde bereits mit der Definition 13 eingeführt. zur Verfügbarkeit ist explizit in Anlage Nr. 7 zu § 9 Abs. 1 BDSG hervorgehoben, dass *zu gewährleisten [ist], dass personenbezogene Daten gegen zufällige Zerstörung oder Verlust geschützt sind (Verfügbarkeitskontrolle).* Ebenso wird auf EU-Ebene in Artikel 17 S. 1 DSRL Bezug genommen. Das Verfügbarkeitskriterium betrifft im Cloud-Umfeld Themen wie Backup, Ausfallsicherheit, Wiederherstellung, Redundanz und Verteilung von Daten und Prozessen. Weitere Information sind im Abschnitt 3 zum *Cloud-Sicherheitsmanagement* enthalten.

Begriff 15 (Transparenz, Transparency). *bezeichnet die objektive Nachvollziehbarkeit, Überprüfbarkeit und Bewertbarkeit von Prozessen, Datenverarbeitungen sowie die Auskunftserteilung bzgl. der Datenspeicherung.*

Das Schutzziel ist nur indirekt im BDSG verankert. So entsprechen die Unterrichtungspflicht nach § 4 Abs. 3 BDSG, die Benachrichtigungspflicht gemäß §§ 19 a Abs. 1 bzw. §33 Abs. 1 BDSG sowie der Auskunftsanspruch nach § 34 Abs. 1 S. 1 BDSG der Umsetzung dieses Schutzziels. Auf EU-Ebene sind dazu analog im Artikel 10 f. und Artikel 12(a)n der DSRL Festlegungen getroffen. Das Transparenzkriterium umfasst im Cloud Computing Themen wie Auditierung, Logging, und Monitoring, sowohl auf Cloud-Applikationsebene als auch auf Umgebungsebene. Die Details sind im Abschnitt 3 aufgeführt.

Begriff 16 (Nicht-Verkettbarkeit, Unlikability). *ist die (ausschließlich) zweckgebundene Benutzung von Informationen in der ursprünglich beabsichtigten Art und Weise.*

Dieses Schutzziel ist in den §§ 4 Abs. 3 S.1 Nr. 2, 14 Abs 2-5, 28 Abs. 2,3,5 und 8, 31 und 39 fixiert. Die entsprechende EU-Richtlinie enthält Artikel 6(b) DSRL. Weiterhin ist das Schutzziel im Telemediengesetz (TMG) § 15 Abs.3 TMG sowie in §3 a BDSG unter der Verwendung von Pseudonymen verankert. Als Kriterium für die Evaluierung spielt die Nicht-Verkettbarkeit insbesondere bei der Zugangskontrolle (Abschnitt 3.1) und Identitätsmanagement Systemen (Abschnitt 3.3) eine Rolle. Die Anforderungen an eine Anonymisierung, Pseudonymisierung werden in Abschnitt 4.5 vertiefend behandelt.

Begriff 17 (Intervenierbarkeit, Intervenability). *beschreibt die Möglichkeit der wirksamen Ausübung der Rechte eines Betroffenen.*

Diese Schutzziel ist in den §§ 20 und 35 sowie Artikel 12 und 14 DSRL erwähnt und regelt die Rechte auf Berichtigung, Sperrung und Widerspruch (Entfernen/Löschen) von persönlichen Daten. Die Intervenierbarkeit stellt im Cloud-Umfeld zumeist ein Problem dar. Ein solches Problem besteht darin, Daten jederzeit vollständig aus der Cloud Umgebung zu erhalten und sicher und dauerhaft zu entfernen. oftmals werden derartige Szenarien anbieterseitig nicht vorgesehen. Das Verfügbarkeitschutzziel, mit der Forderung nach Redundanz und Wiederherstellungsmöglichkeiten, steht hier scheinbar im Widerspruch dazu. Selbst wenn der Cloud Anbieter vorsieht, alle Daten vollständig zu löschen, bleibt die Frage was mit Backups die (unter anderem) diese Daten enthalten geschehen soll. Die Beantwortung Letzterem fällt in das Forschungsgebiet des selektiven Löschens. Interessanterweise besteht diese Problematik nicht nur für explizite Backups oder Langzeitarchivierungen, sondern auch für VM-Laufzeitbackups (Snapshots), die für eine stabile Cloud Umgebung unerlässlich sind und vielmehr im Verfügbarkeitschutzziel gefordert werden. Diese automatisierten Sicherungen entziehen sich dabei häufig der expliziten Kontrolle der Cloud Anbieter und werden im Hintergrund verwaltet. Längerfristig vorgehalten, unterliegen diese ebenfalls dem Schutzziel der Intervenierbarkeit.

2.4.2. Cloud Computing und Service Level Agreements

Die Autoren Jaatun et al. [JBU12] fassen Service Level Agreements (SLA) in drei fundamentale Fragen zusammen: Was wird geliefert? Wohin wird es geliefert? Wann wird es geliefert? SLA werden nach Ansicht der Autoren verwendet, um die Qualität des Services zu spezifizieren. Die messbaren QoS Parameter sind im SLA dargelegt und umfassen laut Jaatun in der Regel die Leistungsfähigkeit, Verfügbarkeit und Ausfallsicherheit, wobei der Fokus auf der Verfügbarkeit des Services läge¹³ und Sicherheitsaspekte dagegen dagegen nicht angesprochen. Jaatun et al. schlagen eine Etablierung von *Sicherheits-SLAs* vor, welche folgende Punkte beinhalten sollten:

- Die Beschreibung des angebotenen Services.
- Die Sicherheitsanforderungen, an die sich der Anbieter bereit ist, zu binden
- Den Prozess, der die Sicherheit überwacht (Monitoring), inkl. der Beweisführung, -sicherung und -verantwortlichkeiten.
- Den Prozess, wie Zwischenfälle gemeldet werden, inkl. Ansprechpartner und Zeitraum zum Lösen des Zwischenfalls.
- Die Konsequenzen beim Verstoß gegen SLA-Regularien für Anbieter und Nutzer inkl. Haftungsausschlüsse.
- Die rechtlichen und regulatorischen Festlegungen, für welchem Bereich die SLA-Gültigkeit besitzt und für welchem sie keine Gültigkeit besitzt.

Als mögliche Beispiele geben die Autoren an: *Alle Nutzerinformation müssen verschlüsselt gespeichert werden. Alle Textnachrichten müssen digital signiert sein. Alle Standortdaten müssen minimal 48 Stunden und maximal 168 Stunden geloggt werden.*

Trotz dieser Bemühungen existieren selbst bei einflussreichen Cloud- Anbietern bisher keine praktischen Umsetzungen. Ein Problem ist die Absolutität: es gibt im Gegensatz zur Verfügbarkeit keine Gewährleistung einer 90%igen Sicherheit. Zudem sind Sicherheitsanforderungen häufig so formuliert, dass Aspekte beschrieben werden, die nicht eintreten sollen, was es schwierig gestaltet, diese präventiven Maßnahmen erfolgreich zu verifizieren. Für eine bessere Kontrolle und Handhabung der SLA in automatisierten Umgebungen, wie in der Cloud, schlagen die Autoren Meland et al. [MBJ⁺12] eine maschinenlesbare Formulierung der SLAs vor.

Laut Gangadharan et al. [GP11] bezeichnen Cloud-SLA als spezifizierte Erwartungen und Verpflichtungen eines Anbieters und Nutzers bzgl. der Service Charakteristiken in geschäftsmäßiger Form, so dass diese Charakteristiken gemessen, überwacht und verwaltet werden können. Ihrer Meinung nach beschreibt Cloud-SLA ein rechtlich durchsetzbares Dokument, welches die minimalen Leistungsmerkmale beschreibt, welchem der Anbieter im Serviceangebot zugesagt hat. Die Vertragsbedingungen welche zwischen Nutzer und Anbieter vereinbart werden sollen, sind in der Arbeit von Gangadharan et al. [GP11] formuliert. Dabei bezeichnen die Autoren die Transparenz als goldene Regel für eine erfolgreiches Datenschutzniveau, konstatieren in diesem Zusammenhang aber zugleich, dass die Charakteristiken des Cloud Computings dies sehr schwierig gestalteten. Diese Feststellung bekräftigt die Annahme der These 1 dieser Arbeit aus Abschnitt 1.1. Die Autoren führen grundlegende Technologien wie Identitätsmanagement, Verschlüsselungstechnologien und Architekturmuster als wichtige Unterstützung gegen Datenschutzbedenken auf.

¹³Der Stand im August 2014 von Amazon Web Services bestätigen dies, denn in den SLA ist ausschließlich die Verfügbarkeit thematisiert. <http://aws.amazon.com/de/ec2/sla/>, <http://aws.amazon.com/de/s3/sla/>, <http://aws.amazon.com/de/rds/sla/>, letzter Zugriff 09.11.2014

Gemäß des Cloud Computing Grundsatzes, das alles als Service zu bewerten sei, versucht die Arbeit von Allison et al.[AC11] den Datenschutz als Service (Privacy as a Service) anzubieten. Die Autoren schlagen dazu eine Unterteilung von privaten Informationen, bzw. Information die dem Datenschutz unterliegen vor. Eine Trusted Third Party soll zudem die erweiterten SLA in Form dieses Privacy-as-a-Service überwachen. Der Autor Baset [Bas12] vergleicht in seiner Arbeit SLAs von verschiedenen Cloud-Anbieter, darunter Amazon¹⁴, Microsoft¹⁵ und Rackspace¹⁶ und kommt dabei zu zwei wesentlichen Feststellungen. Erstens treffen die Anbieter in den SLAs nur Vereinbarungen über die Verfügbarkeit, keiner der Anbieter macht Leistungsgarantieren bzgl. seines Services. Zweitens übernimmt keiner der Anbieter bei Verletzungen gegen seine Verfügbarkeitsgarantien automatisch Vergütungen. In jeden Fall obliegt dies dem Servicenutzer die SLA Verletzung fristgemäß anzuzeigen und vielmehr noch eindeutig zu belegen. Amazon Web Services verweist in diesem Zusammenhang auf vier Anforderungen:¹⁷

1. Die Betreffzeile muss die Worte *SLA Credit Request* enthalten.
2. Datum und Uhrzeit des Zwischenfalls sind anzugeben.
3. IDs der betroffenen Instanzen sind zu benennen.
4. Entsprechende Logdateien sind bereitzustellen, die den Fehler des Zwischenfalls beweisen.

Die Europäische Kommission¹⁸ [Com15] bietet eine Übersicht für die Cloud- SLAs sowie eine Zusammenfassung bisheriger Forschungsergebnisse in diesem Bereich.

¹⁴<http://aws.amazon.com>, letzter Zugriff 07.11.2014

¹⁵<http://azure.microsoft.com/>, letzter Zugriff 07.11.2014

¹⁶<http://www.rackspace.com>, letzter Zugriff 07.11.2014

¹⁷Vgl. <http://aws.amazon.com/de/ec2/sla/>, letzter Zugriff 09.11.2014

¹⁸European Commission Directorate General Communications Networks, Contents and Technology Unit E2 - Software and Services, Cloud

2.4.3. Cloud Computing und Zertifizierung

Im Cloud-Umfeld sind Zertifizierungen eine wichtige Form, das Vertrauen des Nutzers in einem angebotenen Service gegenüber zu erhöhen und oftmals, wie in Abschnitt 2.4.1 verdeutlicht wurde, die einzige Möglichkeit datenschutzgerecht Cloud Computing zu betreiben. Dazu wird der Service gegen mess- und vergleichbare Kriterien geprüft, wodurch dem Cloud-Nutzer ein transparenteres und objektiveres Angebot zur Verfügung steht. Bei der Zertifizierung kann zwischen zwei Arten unterschieden werden, der internen und externen Zertifizierung. Kritisch betrachtet entspricht nur letzteres einer tatsächlichen Zertifizierung. Im internen Fall werden gegen vorgegebene Standards und Kriterien, wie dem BSI-Grundschutz, Selbstkontrollen durchgeführt. Bei der externen Zertifizierung erfolgt die Kontrolle hingegen durch eine unabhängige Kommission. Grundsätzlich bieten Zertifizierungen nach Schneider et al. [SLS13] folgende Vorteile:

- Qualitätssicherung und Erhöhung der Vertrauenswürdigkeit
- Verbesserung des innerbetrieblichen Verständnisses für Sicherheit und Qualität
- Kostenreduktion bei der Geschäftsanbahnung

Die Tabelle 2.1 bietet eine Übersicht ausgewählter Zertifizierungsmöglichkeiten für Cloud-Services und Cloud-Umgebungen.

Tabelle 2.1.: Die Zusammenfassung der Cloud-Zertifizierungen vergleicht ausgewählte Zertifizierungsmöglichkeiten für CSP. Dabei werden die Kosten und Umfang der Zertifizierung sowie deren Nutzen in Deutschland bzw. der EU und der Nutzen weltweit verglichen. Abschließend wird die Art der Zertifizierung benannt.

Zertifikat	Zertifizierer	Kosten	Umfang	Nutzen in D/EU	Nutzen weltweit	Art
TÜV Zertifikat [TR14]	TÜV Rheinland	hoch	sehr hoch	sehr hoch	mittel	extern
CSA STAR [CSA14]	Cloud Security Alliance	gering - mittel	mittel - hoch	gering - mittel	mittel-hoch	intern + extern
EuroCloud Star Audit [GW11]	EuroCloud	mittel	hoch	hoch	gering	extern
EuroPriSe [Una08]	ULD	mittel	hoch	hoch	gering	extern
FedRAMP [Fed15]	US Regierung	mittel	mittel	gering	gering, USA hoch	extern
Tust in Cloud [iC15]	SaaS Eco System	gering	gering	gering	gering	intern

Zu erkennen ist, dass das TÜV Rheinland Zertifikat das umfangreichste, gleichzeitig jedoch mit den höchsten Kosten verbunden ist. Diese Zertifizierung ist daher vor allem für größere Unternehmen bedeutsam. Inwiefern die anderen Zertifikate für KMUs geeignet sind, kann an dieser Stelle eben so wenig beantwortet werden wie die Frage nach der Außenwirkung hier vorgestellter Zertifikate zum potenziellen internationalen Cloud-Nutzer. Grundsätzlich ermöglichen alle Arten der Zertifizierung eine höhere Form der Transparenz, eines der geforderten Schutzziele im Abschnitt 2.4.1.

Teil II

Theoretisch-technische Analyse und Evaluierung

Cloud-Sicherheitsmanagement

Das Kapitel Cloud-Sicherheitsmanagement diskutiert Problemstellungen, die mit der Verwaltung von Cloud-Applikationen und Cloud-Umgebungen verbunden sind, wobei der Fokus auf sicherheitsrelevanten Fragestellungen und Maßnahmen zum Schutz liegt, die im Rahmen der Evaluierung an den in Abschnitt 7.1 definierten Kriterien bewertet werden. Dabei werden diese Maßnahmen nicht explizit aufgelistet, sondern fügen sich logisch in die Beschreibung der cloudspezifischen Probleme und Besonderheiten ein. Die Abbildung 3.1 verdeutlicht die einzelnen Teilgebiete, die in diesem Kapitel erörtert werden.



Abbildung 3.1.: Die Bestandteile des Cloud-Sicherheitsmanagements im Überblick. Die Abbildung visualisiert den Aufbau dieses Kapitels und ist die detaillierte Darstellung der Cloud-Sicherheitsmanagement-Box aus dem Kapitel Taxonomie der Cloud-Sicherheit.

Die Zugangs- und Zugriffskontrolle spielen in jedem IT-System eine Rolle, bei dem mehr als ein Nutzer involviert ist. Dazu muss sich ein Nutzer authentifizieren und bekommt ggf. Rechte zugewiesen, die er innerhalb des Systems wahrnehmen kann. Zur Verwaltung zahlreicher solcher Nutzer werden Identitätsmanagementsysteme genutzt. Für diese gibt es in Cloud-Anwendungen spezielle Anforderungen auf die im Kapitel vertiefend eingegangen wird. Das Thema Schlüsselmanagement spielt vor allem dann eine Rolle, wenn Daten in verschlüsselter Form vorliegen und von verschiedenen Nutzern benötigt werden. Hierbei sind Schlüsselerzeugung und -verteilung in dynamischen und komplexen Cloud-Umgebungen nicht immer trivial. Auditierungs- und Logging Prozesse bilden das Rückgrat der IT-Compliance und sind damit ebenfalls Bestandteil dieses Kapitels.

3.1. Zugangskontrolle

Die Zugangskontrolle stellt sicher, dass nur berechtigte Nutzer Zugang zum System erlangen. Dieser Abschnitt beschäftigt sich zunächst mit allgemeinen Prinzipien und verdeutlicht im weiteren Verlauf spezifische Probleme, die im Cloud Computing-Umfeld auftreten können. Dabei werden im Rahmen der Diskussion nur wissensbasierte Authentifikationsverfahren erläutert. Für weiterführende Informationen sei auf Standardliteratur wie Eckert [Eck13] und Poguntke [Pog07] verwiesen.

Im Allgemeinen authentifiziert sich ein Nutzer bei Services mit Hilfe von Credentials. Dies können Informationen wie Benutzername, Passwort, PIN oder hardwaregenerierte Signaturen sein. Bei vielen Cloud-Services ist dies die einzige Form der Authentifikation, die zudem weit verbreitet ist und eine hohe Akzeptanz der Nutzer besitzt. Entscheidend für die Sicherheit ist hierbei die Wahl eines sicheren Passwortes. Eine einfache Möglichkeit die Sicherheit weiter zu erhöhen ist die Verwendung einer Mehrfaktor-Authentifizierung, in Form von Einmalpasswörtern, TANs und PINs. Challenge-Response (CR) Verfahren stellen eine Verallgemeinerung der Passwort Authentifizierung dar. Das Prinzip der CR-Verfahren besteht ebenso wie bei Verwendung von Einmalpasswörtern darin, diese nicht mehrfach zu übertragen, sondern bei jedem Login neue Credentials zu erzeugen. Zum Teil finden auch mehrere Server Round-trips in Form von Frage-/Antwortspielen statt. CR-Verfahren werden in der Praxis insbesondere im Mobilfunk- oder Drahtlosnetzwerkbereich eingesetzt.¹

Zero-Knowledge-Verfahren(ZK) stellen eine Spezialfall von CR- Verfahren dar und bieten die Möglichkeit, einen Teilnehmer vom Protokoll vom Besitz eines Geheimnisses des anderen Teilnehmers des Protokolls zu überzeugen, ohne jedoch Teile des Geheimnisses preiszugeben. Die praktische Realisierbarkeit wurde von Feige et al.[FFS88] nachgewiesen. Anwendung findet diese Form der Authentifizierung z.B. in PayTV-Systemen², auch im Lösungsansatz von Bertino et al. [BPFS09]. Weitere Anwendung finden ZK-Verfahren in den anonymen Credential-Systemen U-Prove [MR14], Idemix [IRZ14] und AnonymousCloud [KH12]. Der große Vorteil dieser Verfahren besteht darin, dass Angreifer nicht an private Informationen gelangen können, da diese den Nutzer niemals verlassen. Die Nachteile sind ein höherer Berechnungs- und Kommunikationsaufwand [TPPG11].

Für das Cloud Computing sind vor allem Ansätze zur Authentifizierung in verteilten Systemen von Bedeutung. Wie Ludescher et al. [LFB12] in diesem Zusammenhang hervorheben, ist *die dynamische Bereitstellung von virtuellen Ressourcen insofern eine Herausforderung, [als] dass auch die Authentizität der dynamisch hinzugefügten Ressourcen gewährleistet werden muss und keinen potenziellen Angreifer darstellt.*

Eine OpenSource-Implementierung eines solchen verteilten Ansatzes ist Shibboleth³. Vor allem im akademischen Umfeld ist diese Form der Authentifizierung verbreitet. Der Nutzer identifiziert sich mit einem digitalen Ausweis, der von einem Identifikationsserver der Institution bereitgestellt wird, dem der Nutzer angehört. Shibboleth bietet durch die Identifikationsserver eine Möglichkeit den Datenschutz des Nutzers zu verbessern, da dieser die Credentials des Nutzers bereitstellt. Dennoch bietet Shibboleth laut [XYM⁺13] keinen Schutz vor der Erstellung von digitalen Spuren, d.h. der im vorherigen Kapitel Schutzziele 2.4.1 angesprochenen Unverknüpfbarkeit.

Ein in Telekommunikationssystemen verwendetes Protokoll ist DIAMETER [Ven01], das beispielsweise bei Authentifizierungen von Mobilfunkgeräten, z.B. bei der Nutzung von LTE verwendet

¹Vergleiche <http://tools.ietf.org/html/rfc2195>, letzter Zugriff 22.04.2014.

²Vergleiche Patent: System for controlling access to broadcast transmissions <http://patentimages.storage.googleapis.com/pdfs/US5481609.pdf>, letzter Zugriff 23.04.2014

³Vergleiche <http://shibboleth.net>, letzter Zugriff 23.04.2014

wird. Dieses Protokoll stellt eine Umsetzung einer AAA-Architektur dar, welche die Möglichkeit der Authentifizierung, Autorisierung und des Accountings bietet. Für das Accounting bietet DIAMETER Möglichkeiten der Protokollierung von Verbindungsaufbau, -dauer und Datenvolumen als Grundlage für die Erstellung von Rechnungen. Das Protokoll setzt dabei auf sicheren Kommunikationsprotokollen, wie IPSec [DH03] oder TLS⁴ auf.

Der bekannte Authentifizierungsservice Kerberos [NT94], basierend auf dem Needham-Schroeder-Protokoll von Needham und Schroeder [NS78], bietet die Möglichkeit einer beidseitigen Authentifizierung⁵. Der vergleichsweise aufwändige Aufbau des Systems, mit Ticket System und getrennten Authentifizierungs- und Key-Storage-Servern bietet in der aktuellen Version eine hohe Sicherheit für Nutzung und Service-Provider beiderseits. Kerberos unterstützt dabei einen Single-Sign-on, ist aber wie angedeutet auf eine vertrauenswürdige Drittpartei angewiesen. Es existieren verschiedene Implementierungen.⁶ Die Vorschläge der Autoren [BLS⁺09, JSGI09, TJA10] der Integration von Kerberos in Cloud-Umgebungen zur Erhöhung der Sicherheit werden in folgenden beiden Arbeiten aufgegriffen.⁷ So erweitert der Ansatz von Pecarina et al. [PPL12] den Kerberos um die Möglichkeit mit anonymisierten Clientzugriffen umgehen zu können. Die Autoren Ludescher et al. [LFB12] beschreiben in ihrem Ansatz die Integration von Kerberos in einer hybriden Cloud-Umgebung und gehen dabei auf die Problemstellung der Authentifizierung von dynamisch hinzugefügten VMs ein. Ein besonderer Ansatz der Zugangskontrolle von Gun et al. [GWP⁺11] ist Logical Attestation, der auf der Verwendung von sicherer Hardware⁸ basiert. Dazu führen die Autoren Bezeichner ein, die einen fälschungssicheren, maschinen-lesbaren Ausdruck repräsentieren. Der Bitstrom der diesen Bezeichner repräsentiert, stellt hierbei ein Credential dar. Ferner haben die Autoren zur sicheren Ausführung von Applikationen das Betriebssystem Nexus entwickelt, das die beschriebene Authentifizierungsmethode integriert.

⁴Vgl. <http://datatracker.ietf.org/wg/tls/charter/>, letzter Zugriff 08.04.2014

⁵Sowohl Client als auch Server müssen ihre Authentizität wechselseitig belegen.

⁶Vgl. <http://web.mit.edu/kerberos/www/>, <http://www.h51.org/>, <http://josefsson.org/shishi/>, letzter Zugriff 10.04.2014

⁷Vgl. <https://www.oasis-open.org/committees/download.php/38245/Kerberos-Cloud-use-cases-11june2010.pdf>, letzter Zugriff 10.04.2014

⁸Für eine detaillierte Beschreibung dieser Technologie siehe Abschnitt 5.1: Trusted Platform Module.

3.2. Zugriffskontrolle

Nach der Authentifizierung von Objekten und Subjekten werden diesen zumeist spezielle Rechte eingeräumt. Dieser Vorgang wird als Autorisierung bezeichnet und hier nicht vertiefend behandelt. Die Verwaltung und Überwachung dieser Rechte und Zugriffsbeschränkungen basiert dabei auf einer Sicherheitsstrategie. Auf Informations-fluss-Strategien wird an dieser Stelle jedoch nicht eingegangen, dagegen werden die in der Praxis verwendeten Zugriffskontrollstrategien beschrieben. Die Zugriffskontrolle verwaltet den Zugriff auf Ressourcen und ist eines der zentralen Themen in der IT-Sicherheit. Dabei stellt sich die Frage, wie der Zugriff auf Objekte so zu begrenzen ist, dass das need-to-know Prinzip eingehalten wird. Zur Beantwortung der Fragestellung wird im Folgenden die Relevanz bekannter Zugriffskontrollmodelle für Cloud-Applikationen erörtert. Für die detaillierte Beschreibung einzelner Modelle sei auf Standardliteratur wie [Eck13, S. 261 ff.] verwiesen.

Im Discretionary Access Control (DAC) Modell hat jedes Objekt einen Besitzer, der entscheidet, wer welche Zugriffsrechte auf dieses Objekt hat. Es werden keine global gültigen Rechte festgelegt. Durch die Nutzersteuerung eignet sich das Modell vor allem für kleine Personengruppen. Da keine einheitlichen Systemstandards festgelegt sind, gestaltet sich die Wartung und Skalierung über Personen und Personengruppen als schwierig.

Das rollenbasierte Zugriffsmodell(Role Based Access Control - RBAC) nach Ferraiolo und Kuhn [FK09] setzt auf systemweit festgelegte Rollen, die einem Subjekt zugeordnet werden können. Dabei entscheidet die Rolle über Objektzugriffe. Das Modell setzt den Fokus auf die Rollen und nicht auf Subjekte oder Objekte. Bacon et al.[BMY02] stellen in ihrer Publikation ein RBAC für verteilte Systeme vor. Auch die Cloud-Speicherlösung von Zhou et al.[ZVH12] wird durch ein RBAC-System geschützt.⁹ Durch systemweite Standards wird die Wartbarkeit und Skalierbarkeit verbessert. Zudem lassen sich RBAC-Modelle laut Rawat [RS12] gut mit anderen Modellen kombinieren.

Im Mandatory Access Control (MAC) Modell dominieren systembestimmte globale Festlegungen, nutzerspezifische Regeln. Subjekte und Objekte werden global gekennzeichnet. Ein Subjekt kann ein Objekt einsehen, wenn es mindestens die gleiche globale Kennzeichnung (z.B. Sicherheitseinstufung) hat. Durch Fokussierung systemweiter Standards wird die Wartbarkeit und Skalierbarkeit in besonderem Maße unterstützt.

Neben diesen Grundmodellen der Zugriffskontrolle existieren weitere Modelle. Ein hybrides Modell zwischen DAC und RBAC wird von Yu et al. [YWRL10] vorgeschlagen. Die Autoren verbinden mit jeder Datei eine Menge von Attributen und teilen jedem Nutzer eine Zugriffsstruktur zu, welche über diesen Attributen definiert ist. Die technische Umsetzung erfolgt mittels kryptografischer Primitive wie attributbasierender Verschlüsselung (ABE). Die im Abschnitt 4.2.4 beschriebene ABE, ist der Ansatz die Zugriffskontrolle und Datenverschlüsselung zu vereinen und aus diesem Grund für praxisrelevante Einsatzszenarien in der Cloud bedeutsam. Die Tabelle 3.1 fasst die Grundmodelle mit ihrer Eignung für Cloud-Anwendungen noch einmal zusammen.

Tabelle 3.1.: Zusammenfassung und Bewertung ausgewählter Zugriffsmodelle für Cloud-Umgebungen.

Modell	Nutzer-- steuerung	zentrale Wartung	Skalier- barkeit	Subjekt Klas- sifizierung	Objekt Klas- sifizierung	Besonderheiten
DAC	sehr hoch	schlecht	schlecht	✗	✓	für kleine Gruppen mit wenigen Objekten
RBAC	mittel	gut	mittel	✓	✓	für mittlere Gruppengröße, Verwaltung der Rollen nötig
MAC	gering	gut	gut	✓	✓	für große Gruppen mit vielen Dateien, Administration nötig

⁹Für Details siehe Abschnitt 4.4: Dateisysteme.

3.3. Identitätsmanagement

Laue und Stiernerling [LS10] bieten eine Übersicht über Identitäts- und Zugriffsmanagement für Cloud-Anwendungen und verweisen zugleich auf folgende technisch-organisatorische Probleme und rechtliche Risiken:

- Problem 1: Veränderung der Mitarbeiterstruktur (Versetzung/Entlassung)
- Problem 2: Kaum Möglichkeiten einer zentralen Verwaltung (Credentials/Rollen)
- Problem 3: Selten ausreichende Logging-, bzw. Auditierungsmöglichkeiten
- Problem 4: Credential-Inflation

Die Einführung von Identitätsmanagementsystem (IMS) soll die Lösung obiger Probleme unterstützen. Nach Hussain [Hus10] ermöglicht ein IMS die effektive Erzeugung, Speicherung und Nutzung von Identitätsinformationen zur Authentifizierung und Autorisierung dazugehöriger Individuen in Organisationen. Ein weiteres Ziel ist es, wie Bernito et al. [BPFS09] und Ranchal et al. [RBO⁺10] hervorheben, die Risiken von Identitätsdiebstahl und -missbrauch zu minimieren. Es existieren zwei allgemeine Ansätze von IMS, die nachfolgend im Zusammenhang mit Cloud Computing erläutert werden. In einem nutzerzentrierten Identitätsmanagement *Centric Identity Management* liegt der Fokus auf der Nutzersicht. Bestehende Lösungen wie OpenID¹⁰ erweisen sich im Cloud Computing-Umfeld als ungenügend. Die Autoren Anging et al. [ABR⁺10] weisen darauf hin, das OpenID sehr anfällig für Phishing-Attacken sei. Förderierte Identitätsmanagementsysteme *Federated Identity-Management Systems (FIMS)* erlauben laut Hussain [Hus10], dass sich Nutzer einmalig bei Identitätsprovidern authentifizieren. Danach ist es möglich eine Vielzahl von Anbietern zu nutzen, ohne sich zu re-authentifizieren. Nach Hussain existieren in einem FIMS folgende drei Hauptkomponenten: Nutzer, Identitätsprovider (IdP) und CSP. Bernito et al. [BPFS09] fügen einen weiteren Teilnehmer, die Registrierungsstelle, hinzu. Die Abbildung 3.2 verdeutlicht diesen Aufbau schematisch.

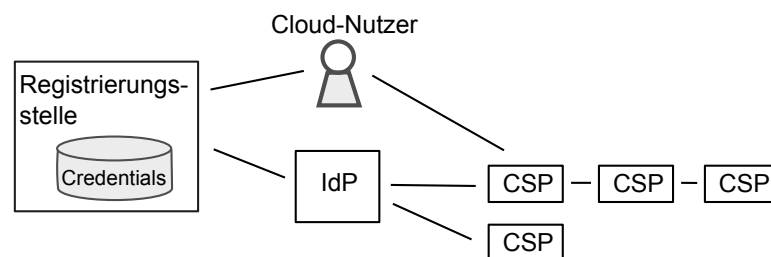


Abbildung 3.2.: Aufbau und Teilnehmer eines föderierten Identitätsmanagementsystems. CSP bieten Zugang zu Daten und Services. Identitätsprovider (IdP) zertifizieren die Identitätsattribute der Nutzer und steuern die Verteilung dieser Informationen. Die Registrierungsstellen speichern und verwalten die Authentifizierungsinformationen des Cloud-Nutzers.

Die CSP bieten Zugang zu Daten und Services über das Internet, während die IdP die Identitätsattribute der Nutzer zertifizieren und die Verteilung dieser Informationen steuern. Die Registrierungsstellen speichern und verwalten Informationen (Credentials) die im Zusammenhang mit der Identifizierung stehen und beispielsweise bei der Mehrfaktor-Authentifizierung verwendet werden.

¹⁰<http://openid.net> letzter Zugriff 12.06.2014. OpenID ist ein dezentrales Authentifikationsprotokoll, welches dem Nutzer hilft deren verschiedenen digitalen Identitäten zu verwalten und damit eine bessere Kontrolle über preisgegebene Informationen zu erhalten.

Wie Chow et al.[CHHY12] in ihrer Arbeit zusammenfassen, muss ein sicheres ISM für den Cloud-Einsatz die folgenden Sicherheitsaspekte und Funktionalitäten berücksichtigen:

- Unverknüpfbarkeit - Diese Eigenschaft schützt den Nutzer mehrerer Services davor, dass einzelne Transaktionen in einem Service, auch bei Zusammenarbeit mehrerer Serviceanbieter, nicht auf einen Nutzer zurückgeführt werden können. Das Anlegen von Nutzerprofilen wird damit erschwert.
- Delegierbare Authentifikation - Ein Serviceanbieter soll in der Lage sein eine Authentifizierung an einem seinerseits genutzten Service weiterzureichen, ohne den Nutzer zu zwingen sich erneut zu authentifizieren.
- Anonymität - Der CSP soll den Nutzer authentifizieren können ohne dessen wahre Identität oder Credentials zu kennen, bzw. zu erfahren.
- Verbindlichkeit, Haftbarkeit - Der Nutzer darf die Anonymisierung nicht ausnutzen oder Handlungen abstreiten können. CSP können im Zweifelsfall bei einer vertrauenswürdigen Partei die Identität erfragen, um bei böartigen oder rechtswidrigen Handlungen eingreifen zu können.
- Benutzerdefinierte Zugangskontrolle - Der Nutzer kann festlegen, zu welchen Information der Provider Zugriff erlangen soll.
- Einmalige Registrierung - Der Nutzer muss sich nur einmal registrieren um seine Credentials zu erhalten.

Die Forschungsergebnisse von Chow et al. berücksichtigen diese Anforderung durch die Verwendung zufälliger Gruppensignaturen. Dies sind datenschutz-orientierte Signaturen, in denen ein Gruppenmanager Schlüssel zur Signierung ausstellen und an Gruppenmitglieder verteilen kann. Mit Hilfe dieser Signaturen kann zwar gezeigt werden, dass eine Person zweifelsfrei zur Gruppe gehört, jedoch nicht deren genaue Identität. Die Methode bietet neben den ZK-Verfahren eine weitere Möglichkeit der anonymen Authentifizierung. Für die Realisierung obiger Anforderungen sind häufig mehrere, teils vertrauenswürdige Parteien, notwendig. Ferner erhöhen sichere Authentifizierungsprotokolle den Aufwand für die Kommunikation teils erheblich.

Der Ansatz von Bertino et al. [BPFS09] bezieht ebenfalls vertrauenswürdige Drittparteien mit ein und folgt dem in Abbildung 3.2 dargestellten Prinzip. Der Fokus ihrer Arbeit ist die mehrfaktorielle, anonyme Authentifizierung auf der Grundlage von ZK-Verfahren. Die Autoren betrachten zudem die Problematik semantischer Vergleiche der Formularfelder¹¹ bei der Registrierung für Services um den Grad der Automatisierung zu erhöhen. Keine Berücksichtigung findet dagegen die von Chow et al. geforderte Unverknüpfbarkeit.

Neuere Forschungsergebnisse wie von Xiong et al. [XYM⁺13] zeigen weitere Verbesserungen. So berücksichtigt dieser Ansatz die SLAs des CSP, um die Transparenz des Gesamtsystems zu erhöhen. Ein weiterer Vorteil ihres Lösungsvorschlages ist die hohe Skalierbarkeit des ISM.

Die zahlreichen Verbesserungen äußern sich jedoch in einer stark angestiegenen Komplexität. So wurde die Anzahl der Teilnehmer bei der Registrierung auf fünf erweitert und der Registrierungsvorgang in vier Phasen aufgeteilt. Laut Schneier [Sch11b] stellt die Komplexität eines Systems einen starken Gegenpol zur Systemsicherheit dar. Somit ist eine Akzeptanz des Ansatzes von Xiong et al. im praktischen Umfeld fragwürdig.

Im Gegensatz schlagen die Lösungsansätze von Angin et al.[ABR⁺10] und Ranchal et al.[RBO⁺10]

¹¹Z.B. der Vergleich von *credit card*, *credit-card* oder *creditcard information*.

die Verwendung von *active Bundles* vor, um nicht auf vertrauenswürdige Drittparteien angewiesen zu sein. Die von Ben et al. [BOL09] entwickelten *active Bundles* bieten eine Möglichkeit private, sensible Daten zu schützen und beinhalten neben den Daten eine virtuelle Maschine, die vom Zielrechner ausgeführt werden muss. Dieser Ansatz ist dabei in der Lage die Vertrauenswürdigkeit seiner Umgebung einzuschätzen und verfügt über einen Schutzmechanismus, der beim Verletzen der Integrität des *active Bundles* sensible Daten rückstandslos löscht. Entscheidender Nachteil des Lösungsvorschlags ist die Übertragung des *active Bundles* zum CSP. Leider gehen die Autoren in ihrer Arbeit weder auf die Größe, welche durch eine integrierte virtuelle Maschine erheblich sein sollte, noch auf die Möglichkeit einer effektiven Übertragung ein. Dieser Aspekt und weitere Fragen, wie den Kostenträger für die benötigten Ressourcen des *active Bundles* beim CSP, stellen die Praktikabilität dieses Ansatzes ebenfalls in Frage. Der Abschnitt verdeutlicht das Zusammenspiel von Datenschutz, Anonymität und Schlüsselmanagement bei der Verwaltung von Identitäten im Cloud-Umfeld. Es zeigt sich, dass mit zunehmenden Sicherheitsanforderungen sowohl die Komplexität als auch der Aufwand für die Entwicklung und den Betrieb eines solchen Systems steigen. Das postulierte Prinzip der Delegation mit begrenztem Wissen wird an dieser Stelle deutlich: Das Begrenzen des Wissens seitens des CSP bzgl. der Identität des Nutzers oder dessen Nutzungsprofil führt zu einer größeren Arbeitslast beim Cloud-Nutzer selbst. Diese kann einerseits entweder erneut an dritte Parteien ausgelagert, deren Wissen jedoch wiederum begrenzt wird und so zu einem starken Komplexitätsanstieg führt. Oder wird andererseits in eine komplexe Datenstruktur überführt, die unpraktikablen Mehraufwand verursacht und die Kosten für die Verarbeitung durch den CSP stark erhöht. Somit bleibt die Frage offen, wer trotz stetiger Verbesserungen der Sicherheit, die Kosten für Entwicklung, Verbreitung und Betrieb eines sicheren Identitätsmanagementsystems trägt.

3.4. Schlüsselmanagement

Auf die Standardaufgaben des Schlüsselmanagements, wie die Erzeugung, Speicherung und Vernichtung, wird an dieser Stelle nicht eingegangen. Hierzu sei auf die Arbeit von Eckert [Eck13] verwiesen, die diesen Vorgang ausführlich beschreibt. Die Ausführungen in dieser Arbeit, betreffen die Probleme im Schlüsselmanagement, die im Zusammenhang mit dynamischen, hierarchischen Nutzergruppen oder verteilten Ressourcen, wie sie im Cloud Computing Umfeld typisch sind, entstehen.

Von zentraler Rolle für die Sicherheit und Akzeptanz von Cloud- Lösungen ist die Frage der an welcher Stelle, d.h. in der Cloud- Applikation oder der Cloud-Umgebung, und durch wen, d.h. durch den Cloud-Nutzer oder den CSP, die Schlüsselverwaltung durchgeführt wird. Analog zu Identitätsmanagementsystemen ist grundsätzlich in nutzerzentrierte und föderierte Schlüsselmanagement-Systeme zu unterscheiden, die nachfolgend kurz erläutert werden.

Bei nutzerzentrierten Systemen behält der Nutzer die Kontrolle über die Schlüssel. Aus Effizienzgründen erfolgt dies häufig im Form eines Master-Keys, der als Wurzel des Schlüsselmanagement-Systems vorstellbar ist und von dem weitere Schlüssel abgeleitet werden. Ohne diesen Master-Key ist kein Zugriff auf die Daten möglich. Damit bietet dieser Ansatz den Vorteil, dass der CSP ohne die explizite Einwilligung des Nutzers nicht auf dessen Daten zugreifen kann. Gleichzeitig besteht jedoch den Nachteil, dass bei Verlust des Schlüssels die Daten unwiederbringlich verloren sind. Der Nutzer hat dem Prinzip der Delegation ohne Wissen folgend, zusätzliche Arbeit durch die Wissensbegrenzung.

Bei föderierten Systemen übernimmt der CSP die Schlüsselverwaltung. Dies bietet den Vorteil, dass der Nutzer keine Verantwortung über die Schlüsselverwaltung übernehmen muss und die Daten trotzdem gegenüber Dritten geschützt sind. Nachteil dieser Lösung ist jedoch die sich ergebende Möglichkeit für den CSP, dass dieser die Daten auch ohne explizites Einverständnis des Nutzers entschlüsseln kann. Ferner existieren in AGBs oder SLAs ggf. Klauseln zur Herausgabe der Schlüssel bei verdächtigem Verhalten an staatliche Behörden. Auch an dieser Stelle wird das Prinzip der Delegation mit begrenztem Wissen deutlich: die Informationspreisgabe senkt den zusätzlichen Aufwand auf der Seite des Cloud-Nutzers deutlich.

Es sei zudem angemerkt, dass das Problem des Schlüsselmanagements nicht nur im Falle der Verschlüsselung von persistenten Daten relevant ist, sondern auch für Data-in-Use-Verschlüsselungen, die im Abschnitt 4.2 erörtert werden. Die Autoren Danezis et al. [DLB11] weisen in diesem Zusammenhang insbesondere auf vollständig homomorphen Verfahren hin. Wie in Kapitel 4.2.3 erläutert wird, sind diese hochgradig ineffizient. Doch selbst unter der Annahme sie wären effizient, ist die Verwaltung der Schlüssel dennoch ungeklärt. Erschwerend kommt die von Dijk und Juels [DJ10] bewiesene Tatsache hinzu, dass eine Gruppierung der Schlüssel bei diesen Verfahren unmöglich ist. Dies bedeutet, dass ein Teilen von Berechnungen analog zum Teilen einer Datei zwischen verschiedenen Nutzern ausgeschlossen ist.

Die Autoren Zarandioon et al. [ZYG12] beschreiben zwei generelle Ansätze im Schlüsselmanagement im Dateiumfeld. Zum einen die klassische *Access Control List*, bei der für jede Datei eine Liste zu verwaltender Schlüssel erforderlich ist¹². Dieser Ansatz bietet feingranulare Zugriffe, ist jedoch kaum skalierbar. Zum anderen der von Kallahalla et al. [KRS⁺03] eingeführte Ansatz Dateien in Form einer Gruppe mit dem selben Schlüssel zu sichern. Zur Vermeidung unnötigen Mehraufwands beim Entzug von Rechten, führen die Autoren das Prinzip der *Lazy Revocation* und der *Key Regression* ein.

Wird das Zugriffsrecht auf ein Objekt entzogen, so ist es nötig, dieses mit einem neuen Schlüssel zu versehen, um zu verhindern, dass auf das entsprechende Objekt weiter zugegriffen werden kann.

¹²Z.B. genutzt von Golimund et al.[GMSW06]

Das Prinzip der Lazy Revocation erlaubt es, diese Neuverschlüsselung auf den nächsten Schreibzugriff¹³ des Objekts zu verschieben. Dies beeinträchtigt die Sicherheit insofern, dass ein Angreifer die Datei weiterhin lesen kann, indem er sich den Schlüssel kopiert. Ohne das Prinzip der Lazy Revocation, wäre er gezwungen das gesamte Objekt zu behalten. Valide Schlüssel die frühere Zugreifende nach wie vor kennen - welche auch weiterhin zugreifen sollen - werden als *dirty* bezeichnet. Das Bereinigen der dieser Schlüssel, geschieht durch Ersetzung des alten Schlüssels und Neuverschlüsselung aller mit dem alten Schlüssel verschlüsselten Objekte mit einem neuen Schlüssel. Um diese Neuverschlüsselung effizienter zu gestalten führen Kallahalla et al. [KRS⁺03] eine neue Strategie im Key Updating Schema ein.

Dieser Ansatz ist hoch skalierbar, bietet jedoch keine feingranularen Zugriffsberechtigung. Daher bezeichnen Zarandioon et al. [ZYG12] beide Ansätze als ungeeignet für das Cloud-Umfeld. Stattdessen schlagen die Autoren einen Ansatz auf Basis einer attributbasierenden Verschlüsselung mit anonymen Zugriff vor. Für den anonymen Zugriff verwenden die Autoren attributbasierende Signaturen, um die Credentials des Nutzers zu schützen. Die Autoren Santos et al. [SRGS12] sind zudem der Meinung, dass attributbasierende Verschlüsselungsverfahren den entstehenden Mehraufwand des Schlüsselmanagement reduzieren können. Der entwickelte Service von Xu und Sandghu [XS07] bietet einen weiteren Ansatz, der durch eine 3-Faktor- Authentifizierung geschützt ist und die kryptographischen Schlüsseln in Form von Netzwerken aufteilt. Zudem bietet der Service eine Kompromittierungserkennung und dem Nutzer die Möglichkeit unmittelbare Sperrung von Zugangsdaten vorzunehmen.

Da Verschlüsselung der Quasi-Standard für die Sicherheit in der Cloud ist, sollte das Schlüsselmanagement bereits beim Entwurf von cloudbasierenden Applikationen und Systemen berücksichtigt werden. Hervorgehoben wurde, dass die Verwaltung der Schlüssel der entscheidende Punkt für die Sicherheit der Cloud-Applikation und Cloud-Umgebung bzw. deren Daten. Das Prinzip der Delegation mit begrenztem Wissen ist bei der Verwaltung der Schlüssel anschaulich darstellbar. Soll der CSP keine Kenntnis der Schlüssel erlangen, muss das Schlüsselmanagement als zusätzliche Aufgabe seitens des Cloud-Nutzers erfüllt werden. Ist der Nutzer bereit diese Schlüssel preiszugeben, kann die zusätzliche Arbeit vollständig abgegeben werden. Neben diesen beiden Extrema existieren Möglichkeiten nur einen Teil des Wissens preiszugeben und damit nur einen Teil der Arbeit zu übernehmen.

Eine umfangreiche Auflistung von kommerzielle Lösungen bietet die Übersicht [Wik15].

¹³Im Gegensatz zum Prinzip der *Lazy Re-Encryption*, welche beim nächsten Lese- oder Schreibzugriff neu verschlüsselt, verbessert dies die Leistungsfähigkeit auf laut Zarandioon [ZYG12] erheblich.

3.5. Logging und Auditierung

Die Autoren Schneier und Kesley [SK99] schlagen als erste Forscher einen praktischen Ansatz für einen sicheren Logging-Service in nicht vertrauenswürdigen Umgebungen wie in der Cloud vor. Darauf aufbauend beschreiben Smiraglia und Ramunno im Rahmen eines technischen Berichts des TClouds-Projekts [SR15b, SR15a] einen sicheren Logging-Service. Die Abbildung 3.3 stellt eine Übersicht der integrierten Bestandteile des Services dar und soll zur Verdeutlichung des Kerngedankens von sicherem Logging dienen.

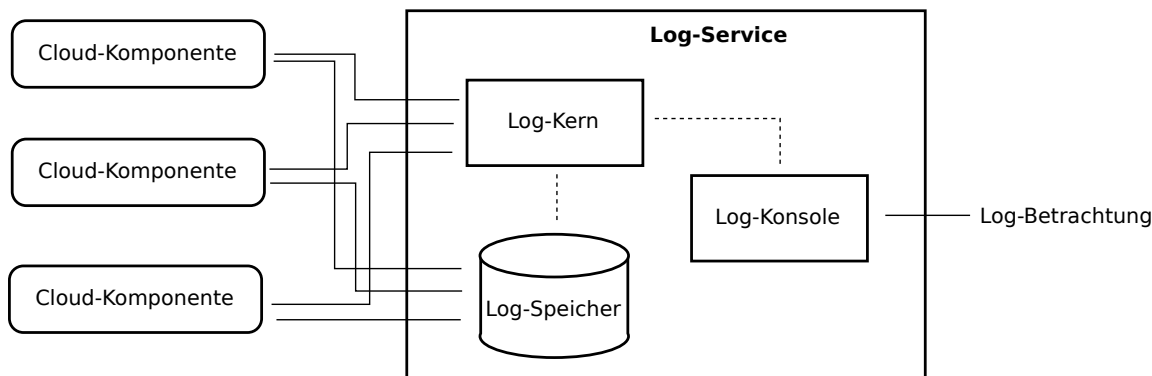


Abbildung 3.3.: Überblick eines Log-Services nach Smiraglia & Ramunno [SR15a]. Der Log-Kern, der das Initialisieren neuer Logging-Sessions und das Verifizieren bereits vorhandener Sessions steuert, stellt die Hauptkomponente dar. Der Log-Speicher übernimmt die Speicherung der Log-Einträge, die durch die Cloud-Komponenten erzeugt wurden. Die Log-Konsole stellt die Schnittstelle zur Serviceverwaltung dar und bietet Verifikationsmöglichkeiten.

Der Log-Kern stellt die Hauptkomponente dar, welche in einer vertrauenswürdigen Umgebung ausgeführt wird. Die wesentlichen Aufgaben dieser Komponente sind das Initialisieren neuer Logging-Sessions und das Verifizieren bereits vorhandener Sessions. Der Log-Speicher übernimmt die Speicherung der Log-Einträge. Die Log-Konsole stellt die Nutzerschnittstelle zum Management des Services dar und bietet Möglichkeiten zur Verifikation. Die Cloud-Komponente integriert ein Logging-Service-Modul welches die Log-Einträge erstellt und die Kommunikationsfähigkeit mit dem Log-Service sicherstellt. Diese befindet sich außerhalb der vertrauenswürdigen Umgebung und bildet den Datenaustausch mittels REST, HTTPS und JSON ab. Im technischen Bericht beschreiben die Autoren zudem die Integration in OpenStack. Kryptographisch wird der Prozess nach den HashChain-Verfahren von Schneier und Kesley [SK99] umgesetzt. Die Vertraulichkeit der Daten wird durch eine symmetrische Verschlüsselung gewährleistet, so dass das System die Integrität aller Log-Einträge bis zu einem Angriffszeitpunkt X garantieren kann. Der Angreifer kann die Log-Einträge vor Zeitpunkt X nicht modifizieren, ohne dass dies bemerkt würde. Dies wird als *forward integrity* bezeichnet.

In einem technischen Bericht beschreiben Pulls et al. [PWVG12] den Bereich des verteilten, sicheren Loggings mit Rücksicht auf den Datenschutz. Aime [Aim10] schlägt einen Ansatz eines verteilten vertrauenswürdigen Logging-Service auf Basis von sicherer Hardware vor.¹⁴ Zudem sei an dieser Stelle auf zwei alternative Ansätze verwiesen, welche für die Vertrauenswürdigkeit eine Trusted Third Party integrieren: Wang et al. [WWRL10] und Wang et al. [WLL12]. Für einen tiefgreifenderen Einblick in diesen Bereich sei auf die Thesis von Pulls [Pul12] verwiesen.

¹⁴Für eine detaillierte Beschreibung der Technologie siehe Abschnitt 5.1: Trusted Platform Module.

3.6. IT Compliance

Dieser Abschnitt erörtert Lösungsansätze, mit denen die Einhaltung gesetzlich vorgeschriebener bzw. in SLAs getroffener Reglementierungen gesichert werden. Thematisch fügt sich dieser Abschnitt demnach an das Kapitel Datenschutz 2.4.1 und SLAs 2.4.2 an. Insbesondere werden technische Maßnahmen bzw. Forschungsansätze beschrieben, welche die IT-Compliance im Cloud Computing Umfeld unterstützen.

Der Ansatz von Zellag et al. [ZK12] beschreibt Möglichkeiten einer quantitativen Aussage zur Dateninkonsistenz von verteilten Applikationen in der Cloud. Der Cloud-Nutzer kann damit erkennen, ob der CSP die Daten wie gefordert redundant speichert oder ob, um Kosten zu reduzieren, darauf verzichtet wird.

Zur Wahrung der Compliance stellen Brandic et al. [BDA⁺10] eine Lösung namens Compliant Cloud Computing vor, die Cloud- Anwendungen von ausgewählten CSP nach Regeln eines *Compliance Level Agreements* (CLA) ausführt. Diese CLA werden durch eine von den Autoren entwickelte Domain-Specific-Language spezifiziert. Damit ist es möglich spezifische Anforderungen der Daten, wie beispielsweise die Speicherung innerhalb eines Landes oder der EU, anzugeben. Diese Lösung setzt ein grundsätzliches Vertrauen in den CSP voraus, da keinerlei Vertraulichkeit bzgl. der Daten geboten wird. Abhängig vom Umfang der Applikation kann die Formulierung der CLA zudem sehr aufwändig sein.

Ein anderes Problemfeld beschreiben Bleikertz et al. [BKNS12] am Beispiel der sicheren Wartung von Clouds. Der vorgestellte Lösungsansatz beschreibt die Absicherung von virtuellen Maschinen bei der Wartung gegenüber den Administratoren. Dazu führen die Autoren verschiedene Privilege-Level ein (z.B. read-only-Level) um die Integrität und Vertraulichkeit der VMs zu verbessern. Diese Lösung richtet sich an CSP oder Betreiber von privaten Cloud-Umgebungen.

Ein weiteres Feld im Bereich der IT-Compliance adressieren Watson et al. [WL12], indem sie untersuchen in welcher Region die Daten gespeichert werden. Häufig existieren datenschutzrechtliche, verbindliche Richtlinien, welche Daten wo gespeichert werden müssen. Die Autoren führen dazu einen *Proof of Location* ein, welcher vom Cloud-Nutzer verwendet werden kann, um sich von der Lokation seiner Daten zu überzeugen. Wie die Autoren prototypisch nachweisen konnten, eignet sich dieses Konzept dazu CSP zu identifizieren, die ihren Profit erhöhen und dafür günstigen Speicher im Ausland zukaufen.

3.7. Zusammenfassung

Das Kapitel beschreibt Problemstellungen, die sich mit der Verwaltung von Cloud-Applikationen und Cloud-Umgebungen ergeben können und erläutert mögliche Lösungen. Der Fokus liegt dabei auf Untersuchung bereits vorhandener Ansätze und der Bewertung welche Auswirkung diese auf die Verwaltung von Cloud-Systemen haben. Die untersuchten Cloud-Lösungsansätze wurden nach den in Kapitel 7.1 spezifizierten Evaluierungskriterien bewertet, die Auflistung der Ergebnisse findet sich im Kapitel 7.2.

Generell ist festzustellen, dass es häufig eine Unterscheidung in nutzergesteuerter und föderierter Verwaltung vorgenommen wird. Bei der nutzergesteuerten Verwaltung nimmt der Cloud-Nutzer alle Verwaltungsaufgaben selbst wahr und ist damit für die Sicherheit selbst verantwortlich. Der Umfang dieser Aufgaben ist von der Größe des Cloud-Systems, Anzahl der Nutzer und der Nutzergruppen abhängig. Dabei wirken diese Zusatzaufgaben dem ursprünglichem Auslagerungsgedanken, der mit dem Cloud Computing verbunden ist, entgegen. Die Alternative besteht darin, CSP oder Dritten zu vertrauen und die neu angefallenen Verwaltungsaufgaben ebenfalls auszulagern. Diese Situation entspricht exakt der Aussage des in Abschnitt 1.1 erläuterten Prinzip der Delegation mit begrenztem Wissen. Ferner wird deutlich, dass die Erkenntnisse aus diesem Kapitel die Annahme der These 1 unterstützen. Klassische Sicherheitsmaßnahmen wie Authentifizierung, und Identitäts- oder Schlüsselmanagement bieten häufig zu starre Konzepte, um der hohen Dynamik und starken Vernetzung von CSP in Cloud-Umgebungen die gleiche Sicherheit zu bieten wie in klassischen Client-Server-Anwendungen. Hier ergeben sich zwei Alternativen: Entweder wird die Flexibilität und Dynamik der Cloud-Umgebung reduziert, um bisherige Maßnahmen weiterhin einsetzen zu können, oder die bestehenden Maßnahmen werden erweitert. Letzteres führt jedoch häufig zu einem sprunghaften Anstieg der Komplexität und ist aus diesem Grund sicherheitstechnisch als kritisch zu bewerten.

Cloud-Applikationssicherheit

Dieses Kapitel beschäftigt sich mit Sicherheitslösungen für Cloud- Applikationen. Somit sind die hier vorgestellten Lösungen in der Regel vom Cloud-Nutzer anwendbar, ohne einen Zugriff auf die Cloud-Umgebung zu benötigen. Die Struktur des Kapitel orientiert sich an der im Abschnitt 2.3 eingeführten Taxonomie der Cloud-Sicherheit. Im Abschnitt der Applikationslogik geht vor allem die Möglichkeiten zur sicheren Berechnungen in der Cloud zu verdeutlichen. Für die Sicherung der Datenschicht werden verschiedene Maßnahmen zum sicheren Datenmanagement diskutiert. In Abbildung 4.1 werden die konkreten Teilbereiche visualisiert, auf die das Kapitel näher eingeht.

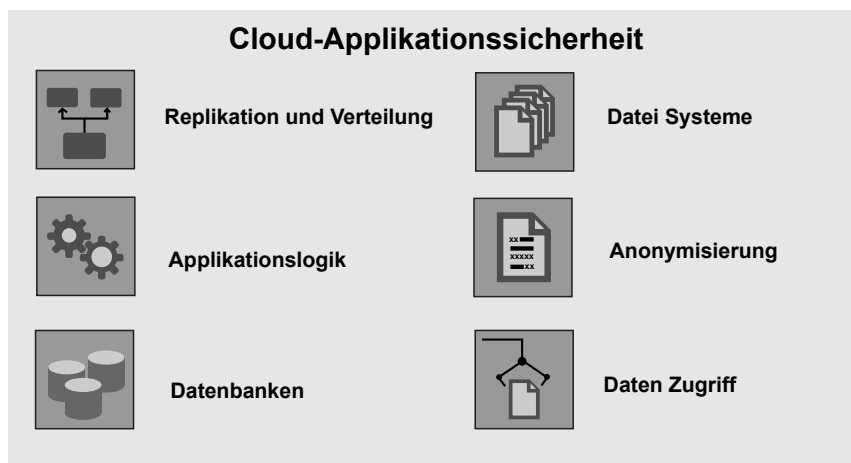


Abbildung 4.1.: Die Bestandteile des Cloud-Applikationssicherheit im Überblick. Die Abbildung visualisiert den Aufbau dieses Kapitels und ist die detaillierte Darstellung der Cloud-Applikationssicherheit-Box aus dem Kapitel Taxonomie der Cloud-Sicherheit.

Im Abschnitt zur *Replikation und Verteilung* werden Möglichkeiten der Verteilung von Cloud-Applikationen über verschiedenen Cloud- Umgebungen hinweg erläutert. Die Maßnahmen zur *Applikationslogik*, als zweiter Punkt, stellen den umfangreichsten Teil dieses Kapitels dar und beleuchten, wie angedeutet, Lösungs- und Forschungsansätze zur sicheren Ausführung von Applikationen. Die Abschnitte zu *Datenbanken* und *Dateisysteme* untersuchen Lösungen zur Speicherung von Daten in verteilten Cloud- Umgebungen. Maßnahmen zur *Anonymisierung* von Daten werden im darauf folgenden Abschnitt erörtert, dabei wird auf Besonderheiten, die bei der Anonymisierung von Daten in öffentlichen Umgebungen eine Rolle spielen, eingegangen. Der letzte Abschnitt bzgl. der *Datenzugriffe* untersucht Methoden, mit denen Informationen zu schützen sind, die beim Datenzugriff preisgegeben werden.

4.1. Replikation und Verteilung

Ein trivialer Ansatz der Replikation und Verteilung, den auch von Bohli et al.[BGJ⁺13] vertreten, ist der Grundgedanke die vollständige Applikation bei zwei nicht kooperierenden Cloud-Providern zu betreiben. Die obere Darstellung 1) in Abbildung 4.2 verdeutlicht dieses Szenario.

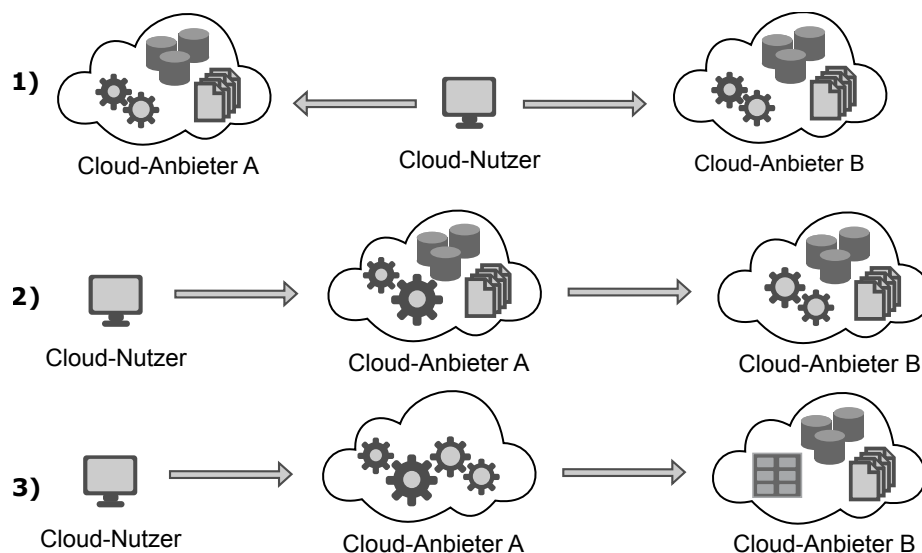


Abbildung 4.2.: Szenarien der Applikationsreplikation und -verteilung auf zwei Cloud-Anbieter. Fall 1) stellt die vollständige Replikation einer Cloud-Applikation auf zwei CSP dar. Fall 2) unternimmt eine Aufteilung in zwei Rechenschritte vor. Fall 3) verdeutlicht eine Aufteilung nach Applikationsschichten in Geschäftslogik und Datenschicht.

Dies bietet den Vorteil einer höheren Ausfallsicherheit und die Möglichkeit, Ergebnisse durch einen Vergleich verifizierbar zu machen. Die Replikation der gesamten Applikation ist jedoch in der praktischen Anwendung fraglich. Wie Bohli et al. hinweisen, müssen die Daten entweder einzeln in beide unabhängige Systeme eingegeben werden oder es existiert ein Service um beide Services miteinander zu verknüpfen. Zudem vervielfachen sich die Kosten entsprechend der Anzahl der Replikationen.

Eine weitere Möglichkeit, in Abbildung 4.2 2) dargestellt, ist eine Aufteilung von n Rechenschritten auf n nicht kooperierende CSP. Dies setzt wiederum unabhängige und umfangreiche Rechenschritte voraus. Sollten diese Rechenschritte vorhanden sein, stellt diese Verteilung im Sinne der Ausfallsicherheit und geringen Umfang auch des Datenschutzes und der Datensicherheit eine praktikable Lösung dar. Durch die Verteilung werden die Cloud-Provider effektiv daran gehindert, Kenntnisse über schutzwürdige Daten und Prozesszusammenhänge des Gesamtsystems zu erlangen.

Der letzte Ansatz ist die Verteilung von Applikationsschichten auf verschiedene, nicht kooperierende, CSP. Im einfachsten Fall wie im unteren Szenario 3) in Abbildung 4.2 wird eine Unterteilung von Business und Data Layer vorgenommen. Die Sicherheit basiert in diesem Fall analog zum vorherig beschriebenen Ansatz darauf, dass kein Cloud-Provider eine vollständige Systemansicht hat. Wobei Anbieter A vollständigen Einblick in die Geschäftslogik und Anbieter B die vollständige Ansicht der Daten hat.

In den folgenden Abschnitten 4.2 *Applikationslogik*, 4.4 *Dateisysteme* und 4.3 *Datenbanken* wird auf Lösungsansätze eingegangen, die es den CSP zu erschweren, Einsicht auf Daten und Prozesse zu erlangen.

4.2. Applikationslogik

Im Abschnitt werden Maßnahmen und technologische Ansätze erläutert, die es ermöglichen, die Ausführung von Programmlogik zu schützen eingegangen. Die theoretischen Grundlagen dieses Gebiets liefert das Feld der sicheren Funktionsevaluierung (SFE), das erstmals durch Yao [Yao82] vorgestellt wurde. Dieser Ansatz erlaubt es, geschützt Funktionen auf bestimmten Daten mehrerer Parteien auszuführen. Grundsätzlich war dieser Ansatz jedoch durch seine ineffiziente Implementierung für viele Applikationen ungeeignet. Laut Troncoso et al. [TPPG11] wurden jedoch trotz dieser Hindernisse in den vergangenen Jahren zahlreiche effiziente Techniken für spezielle Anwendungsfälle entwickelt. nachfolgend beschäftigt sich der Abschnitt mit Techniken zur Verschleierung, beweisbasierte Berechnung, SFE, homomorphe und funktionale Verschlüsselung, wobei auf Grund des großen Potenzials die letzten drei Ansätze umfangreich vorgestellt werden. Dabei werden die vorgestellten und diskutierten Maßnahmen zum Schutz der Cloud-Applikationslogik nach den in Abschnitt 7.1 definierten Evaluierungskriterien bewertet. Das Bild 4.3 stellt zur besseren Übersichtlichkeit die Strukturierung des Abschnitts dar.

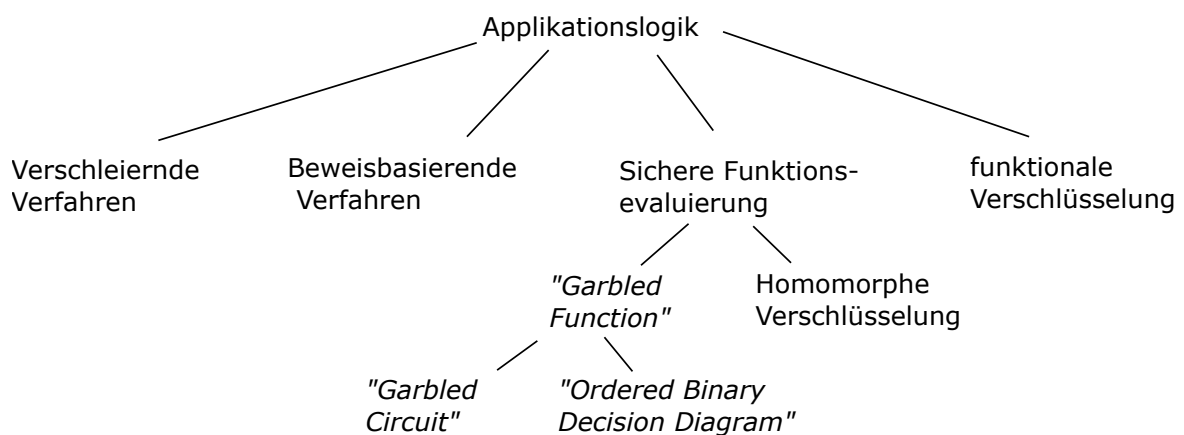


Abbildung 4.3.: Die folgenden Teile des Abschnitts sind entsprechend der Abbildung untergliedert.

4.2.1. Verschleierung

Die Verschleierung stellt nach Troncoso et al. [TPPG11] einen clientseitigen Ansatz zum Datenschutz dar. In die realen Eingabedaten des Cloud-Services wird ein so genanntes *Noise* (Störgeräusch) in Form von echt wirkenden Eingabedaten injiziert, um die realen Daten zu verstecken. Dabei kann das Noise mit verschiedene Strategien erzeugt werden. Die auf der historischen Wahrscheinlichkeit basierende Strategie (HPNGS) ist eine effiziente Möglichkeit, Noise ausgehend von der Wahrscheinlichkeitshistorie, zu generieren. Generell zielen solche Verschleierungsverfahren primär darauf ab, die Auftrittswahrscheinlichkeit zu verbergen. Das Ziel ist es die Varianz aller Auftrittswahrscheinlichkeiten so gering wie möglich zu halten.

Wie Zhang et al. [ZZY⁺12] hervorheben, ist der Datenschutz jedoch häufig von großer Varianz. In der Cloud liegt eine Vielzahl verschiedener sensibler Daten vor, welche datenschutzkritische Rückschlüsse erlauben. Es gilt entsprechend nicht nur, echte Service-Anfragen innerhalb von Verschleierte zu erkennen, sondern auch Assoziationsregeln über echte Anfragen aufstellen können. Wenn zwei dieser Anfragen durch eine Assoziationsregel mit einander verbunden werden, ist die Wahrscheinlichkeit hoch, dass auf eine bestimmte weitere Anfrage zur Folge hat. Dies kann die Charakteristik eines Verhaltensmusters des Nutzers sein. Diese Muster können dessen Identität verraten.

ten und somit ein potenzielles Datenschutzrisiko darstellen. Die zwei Schlüsselkriterien der Noise-Generierung sind nach Zang et al., die Noise- Erzeugungswahrscheinlichkeit und die Noise- Injektionsintensität. Diese schlagen eine Strategie der Generierung von Noise auf Grundlage der Wahrscheinlichkeit eine Assoziationsregel aufstellen zu können vor. Der beschriebene Ansatz erzeugt ihrer Meinung nach mehr und gezielter Noise als die HPNGS-Ansätze und verweisen darauf, dass dem Nutzer mit einen effektiveren Datenschutz höhere Kosten entstehen. In welcher Weise das Verhältnis Datenschutz - Kosten festgelegt wird, sei dessen Entscheidung. Obwohl eingehende Daten in Cloud-Umgebungen bei einigen Anbietern keine Kosten erzeugen, sorgt die Vielzahl an Noise-Daten für erhöhten Speicherbedarf in Cloud-Applikationen. Es ist aus diesem Grund fraglich, ob Verschleierungstechniken effektiv zu einer Erhöhung der Sicherheit von Cloud-Applikationen beitragen können. Obwohl Sicherheitsmaßnahmen immer für erhöhte Kosten sorgen, ist dieser Ansatz als ineffizienter zu bewerten als beispielsweise klassische Verschlüsselungen.

4.2.2. Beweisbasierte und verifizierte Berechnung

Als weiteres Gebiet beschreiben die Autoren Setty et al. [SVP⁺12] das grundlegende Problem von beweisbasierter und verifizierter Berechnung (Proof based verified Computation (PbvC)) wie folgt: Ein Computer V , bekannt als Verifizierer, hat eine Berechnung B mit einer Eingabe x , die er von einem Computer P , bekannt als Beweiser (Prover), berechnen lassen möchte. P gibt y als Ergebnis der Berechnung dem Computer V zurück. V und P können eine effiziente Interaktion führen, wobei diese Interaktion günstiger für V sein sollte, als die lokale Berechnung von $B(x)$. Zudem soll V in der Lage sein sich vom korrekten Ergebnis y von P zu überzeugen oder anderenfalls ein falsches Ergebnis mit hoher Wahrscheinlichkeit ablehnen zu können. Im Allgemeinen ist der PbvC Ansatz nicht effizient. Wie Setty et al. [SVP⁺12] hinweisen, ist die praktische Anwendbarkeit vor allem durch massive Parallelisierung in naher Zukunft, jedoch abzusehen. Das umgekehrte Vorgehen, dass der Client seine Berechnungen mittels Zero-Knowledge-Beweise gegenüber dem Cloud-Anbieter verifizieren muss, ist effizient, praktisch einsetzbar und wird beispielsweise von Danezis et al. [DLB11] an praxisrelevanten Beispielen diskutiert.

4.2.3. Sichere Funktionsevaluierung

Der Begriff der sicheren Funktionsevaluierung (SFE) wurde bereits 1982 durch Yao [Yao82] eingeführt. Die Autoren Kolesnikov et al. bieten mit ihrer Arbeit [KSS10] einen hervorragenden Einblick in das Thema. Grundlegendes Ziel von SFE ist es, zwei misstrauenden Parteien eine Möglichkeit zu bieten, beliebige Funktionen zu berechnen ohne dass dabei einer Partei Eingabedaten der anderen Partei bekannt werden. Mit Bezug auf die Cloud sind die Parteien Cloud-Nutzer und Cloud-Anbieter. SFE geht damit weit über die sichere Kommunikation zwischen Parteien hinaus. Bedeutsam ist vor allem Yao's Millionärsproblem, unter anderem von Ioannidis et al. [IG08] diskutiert. SFE basiert grundlegend auf zwei Ansätzen, der *Garbled Functions (GF)*¹ und *Homomorpher Verschlüsselung (Homomorphic Encryption – HE)*. In den folgenden Abschnitten wird deutlich, dass beide Vor- und Nachteile besitzen. Während GF es notwendig machen die Funktion zu übertragen und damit die Kommunikationskomplexität mindestens linear zur Größe der Funktion ist, ermöglichen es diese jedoch, alle aufwändigen Operationen vorzuberechnen und damit die Latenzzeiten gering zu halten. HE benötigen aufwändige Public-Key-Verfahren, die Kommunikationskomplexität ist jedoch insgesamt geringer. Beide Ansätze erfordern, wie in Abschnitt 3.4 hervorgehoben, ein Schlüsselmanagement.

Garbled Functions

GF stellen eine Verallgemeinerung der von Yao [Yao86] beschriebenen Garbled Circuits² dar. Dabei verschlüsselt eine Partei die Funktion, während die andere die geschützten Eingaben enthält und die GF ausführt, d.h. auf die Eingabedaten anwendet. Die erste Partei ist anschließend in der Lage des Ergebnis zu entschlüsseln. Seit der Einführung ist SFE ein intensives Forschungsthema und hat sich durch Automatisierung, Optimierungen und Effizienzsteigerung eine praktische Relevanz erlangt. Hervorzuheben im Zusammenhang mit praxisrelevanten GF sind die Autoren Schneider T. [Sch08, Sch10a, Sch10b, Sch11a], Kolesnikov V. [KK12, KSS10, KSL⁺08] und Sadeghi A. [SS10a, SSWH10]. Der entscheidende Aspekt für SFE ist die Repräsentation der GF. Denn obwohl die Partei, welche die unbekannte Funktionen ausführt, keine Informationen über Funktion, Eingaben und Ausgaben erhalten soll, muss die GF eine auswertbare Form besitzen. Kolesnikov et al. [KSS10] unterscheiden dazu drei Darstellungen, die in Abbildung 4.4 illustriert werden.

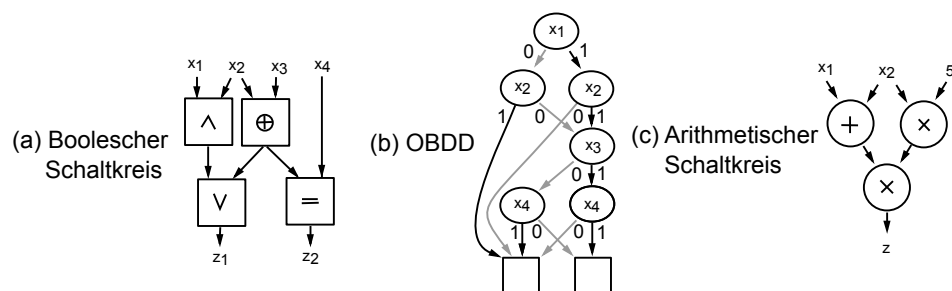


Abbildung 4.4.: Drei Darstellungen von auswertbaren Funktionen. Die Darstellung (a) zeigt einen Booleschen Schaltkreis mit x_n als binäre Eingabewerten, logischen Verknüpfungen und z_n als Ausgabewerten. Die mittlere Darstellung (b) illustriert ein Ordered Binary Decision Diagram (OBDD), welches traversiert wird und in Abhängigkeit der Eingabewerte x_n zwei Ausgabewerte liefert. Darstellung (c) zeigt einen arithmetischen Schaltkreis. Im Gegensatz zum Booleschen Schaltkreis werden die Eingabewerte nicht logisch, sondern arithmetisch verknüpft um die Ausgabewerte zu berechnen.

¹garbled engl. unkenntlich machen

²circuits engl. Schaltkreise

Die GF wird zur sicheren Berechnungen in eine Anzahl von Booleschen Schaltkreisen oder OBDD überführt. GF lassen sich zwar effizient ausführen, da schnelle symmetrische Verschlüsselungsmethoden verwendet werden, benötigen jedoch Hinweise, so dass die SFE zur Berechnung unter Verschlüsselung mit Hinweisen zählt. Dies ist einer der wesentlichen Unterschiede zu homomorphen Verfahren und gleichzeitig der Grund seiner Effizienz und praktischen Anwendbarkeit. Die auf Booleschen Schaltkreise basierenden Garbled Circuits (GC) wurden von Yao [Yao86] eingeführt. Dieser repräsentiert die Funktion, die von der Partei ausgeführt wird, welche keine Informationen über Funktion oder Ergebnis erhalten soll. Die Abbildung 4.5 illustriert dies.

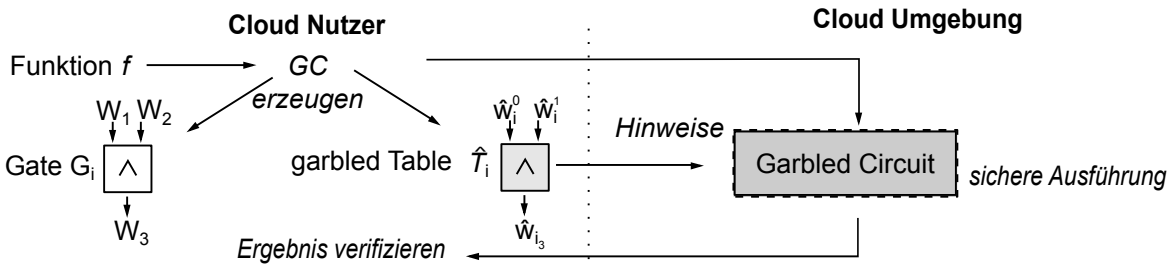


Abbildung 4.5.: Schematische Übersicht über die Funktionsweise und Erzeugung von Garbled Circuits GC. Die detaillierte Beschreibung befindet sich im Text unterhalb.

Ein GC besteht aus Gates G_i mit den Eingängen W_1 und W_2 . Die Anordnung der G_i in der Form, dass die Funktion korrekt ausgeführt wird, entspricht dem booleschen Schaltkreis der Funktion f . Für jedes W_i werden zwei zufällig aussehende Werte $\hat{w}_i^0, \hat{w}_i^1$ erzeugt, welche die Verschlüsselung von 0 und 1 von W_i repräsentieren. Für die Ausgabe W_3 von G_i wird bei der Ausführung des GC eine Hilfsinformation benötigt. Dazu dient die *garbled table* $\hat{T}_i$, die alle Kombinationen an Werte als Eingabe beinhaltet. Dieses Vorgehen erlaubt es, G_i unter Verschlüsselung zu berechnen, ohne weitere Informationen preiszugeben. Diese Methodik wird für den gesamten booleschen Schaltkreis angewandt, so dass dieser Gate für Gate ausgewertet werden kann. Zu beachten ist dabei das exponentielle Wachstum von $\hat{T}_i$ bei Vergrößern der Anzahl W_i . Obige Beschreibung stellt das grundlegende Prinzip dar, heute existieren jedoch zahlreiche effiziente Techniken zur Optimierung. Beispiele sind [EK11, HSE⁺11] mit der Verwendung von XOR-Gates für die Konstruktion von GCs oder die von Kolesnikov et al. [KK12] Optimierungen durch die Prinzipien der Informationstheorie. Die Autoren [GT08, IS10, JKSS10] nutzen dagegen Hardwareunterstützung zur effektiveren Konstruktion und Berechnung. Ferner seien Malka et al. [Mal11] mit der Optimierungen für Streaming-Szenarien aufgeführt. Eine frei verfügbare Implementierung in Java ist in Internet verfügbar.³ Die alternative Möglichkeit der sichere Berechnung mittels Ordered Binary Decision Diagram (OBDD) erfolgt, wie von Kruger et al. [KJGB06] beschreiben, analog zu GCs. Das Ziel ist die Erzeugung einer unkenntlich gemachten Version $\tilde{O}$ des OBDD O . Dazu wird das OBDD um Dummy-Knoten erweitert, so dass immer die gleiche Anzahl an variablen Evaluierungen pro Pfadtraversierung gewährleistet ist. Dieses Prinzip wird bei ORAM im Abschnitt 4.6 analog verwendet und dient der Vermeidung von Metainformationspreisgaben. Zudem werden die Knoten, bis auf den Startknoten K_1 , zufällig angeordnet. Jeder Knoten K_i wird mit einer booleschen Variable b_j gekennzeichnet und in einen unkenntlich gemachten Knoten $\tilde{K}_i$ umgewandelt. Für jeden Knoten K_i wird ein zufälliger Schlüssel S_i erzeugt, der die Informationen von K_i sowie die Nachfolgerknoten und deren Schlüssel verschlüsselt. Die Ausführung von $\tilde{O}$ ist Traversierung des OBDD entsprechend der Eingabe. Um zu verhindern, dass der Ausführende beide Folgeknoten durchläuft, wird die rechte bzw. linke Folgeknoteninformation

³<https://hcrypt.com/sfe/>

mit b_j verschlüsselt. Die Abbildung 4.6 stellt die Erzeugung eines unkenntlich gemachten OBDD $\tilde{O}$ schematisch dar.

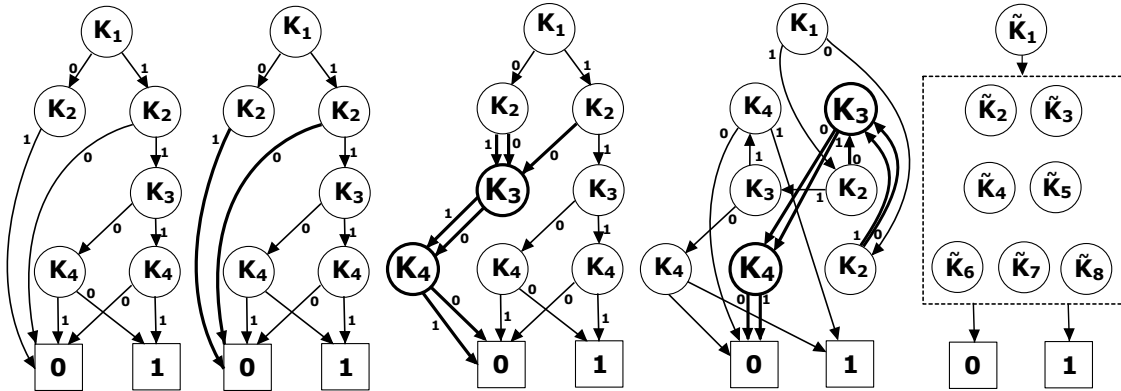


Abbildung 4.6.: Die Erzeugung einer unkenntlich gemachten Version eines OBDD zur sicheren Berechnung von Funktionen. Die stärker hervorgehobenen Teile sind im Zuge der Erzeugung von $\tilde{O}$ hinzugefügte Knoten und Pfade. Das Bild ist zudem von links nach rechts zu betrachten.

Homomorphe Verschlüsselung

Dieser Teilabschnitt erörtert die HE, wobei im weiteren Verlauf vor allem die Problematik der voll homomorphen Verschlüsselung (Fully Homomorphic Encryption – FHE) diskutiert wird, die lange Zeit als unmöglich galt.

Die homomorphe Verschlüsselung ist eine Form der Verschlüsselung, die es erlaubt, spezielle Berechnungen auf den Geheimtexten auszuführen und ein verschlüsseltes Ergebnis zu erzeugen. Entschlüsselt entspricht dies dem gleichen Ergebnis, wie der Berechnungen im Klartext. Geheimtext und Klartext verändern sich gleichermaßen, daher der Name *homomorph*. Die HE ist dabei eine Form der Berechnung unter Verschlüsselung ohne Hinweise. Im Gegensatz zu dem im letzten Abschnitt beschriebenen GCs. In der Fachliteratur werden zahlreiche teilweise homomorphen Verfahren wie RSA [RSA78], ElGamal [ElG85] oder Paillier [Pai99] beschrieben, wobei letztes der Einführung in das Themengebiet dienen soll.

Bei dem von Pascal Paillier 1999 entwickelten Kryptosystem handelt es sich um ein probabilistisch, asymmetrisches Verschlüsselungsverfahren. Zudem besitzt es eine additiv-homomorphe Eigenschaft, die laut Paillier [Pai99] zu den folgenden Eigenschaften führt:

$$\forall m_1, m_2 \in \mathbb{Z}_n \text{ und } k \in \mathbb{N}$$

$$D(E(m_1) \cdot E(m_2) \bmod n^2) = m_1 + m_2 \bmod n \quad (4.1)$$

$$D(E(m)^k \bmod n^2) = km \bmod n \quad (4.2)$$

Die Eigenschaft (4.1) bietet die Möglichkeit, zwei Klartexte zu addieren, in dem die Geheimtexte multipliziert werden. Die Eigenschaft (4.2) zeigt die Möglichkeit auf, das k -fache des Klartextes zu erhalten. Die Verwendung dieser Art von homomorphen Verfahren entspricht der Verwendung von arithmetischen Schaltkreisen, wie es in Abbildung 4.4 (b) dargestellt ist. Diese Eigenschaften können genutzt werden um sichere und gleichzeitig effektive Applikationen zu entwickeln. So ist es damit möglich sichere, elektronische Wahlen abzubilden, indem die Wahlentscheidung wird mit 0

bzw. 1 angegeben und verschlüsselt wird sowie eine anschließende Multiplikation aller verschlüsselten Entscheidungen die Anzahl der positiven Stimmen wiedergibt. Eine weitere interessante Eigenschaft ist die Möglichkeit des *Reshuffling*, bei dem der Geheimtext mit 0 multipliziert wird, welches effektiv eine Addition mit 0 auf m bedeutet. Da es sich um ein probabilistisches Verfahren handelt, führt dies zu einem neuen Geheimtext, jedoch mit gleichem Klartext m .

Mit Hilfe von teilweise homomorphen Verfahren lassen sich, wie beschrieben, spezielle Berechnungen sicher unter Verschlüsselung durchführen. Jedoch zeigen diese Grenzen bzgl. der Anwendung beliebiger Funktionen auf den Geheimtext auf, wie es durch die Erzeugung von GF möglich ist. Gentry [Gen10] beschreibt in seiner Publikation als erster eine Möglichkeit, beliebige Funktionen auf verschlüsselten Daten zu berechnen und führt dazu neben den üblichen Algorithmen eines Verschlüsselungsverfahrens Schlüsselerzeugung, Verschlüsselung und Entschlüsselung, ein vierter Algorithmus Evaluierung hinzu. Dieser Algorithmus ist mit einer Menge an erlaubten Funktionen F_E verbunden. Die nachfolgende Abbildung 4.7 veranschaulicht das beschriebene Funktionsprinzip. Mit Hilfe der Evaluierung soll die Funktion durch eine zweite Partei, die im Cloud Computing dem

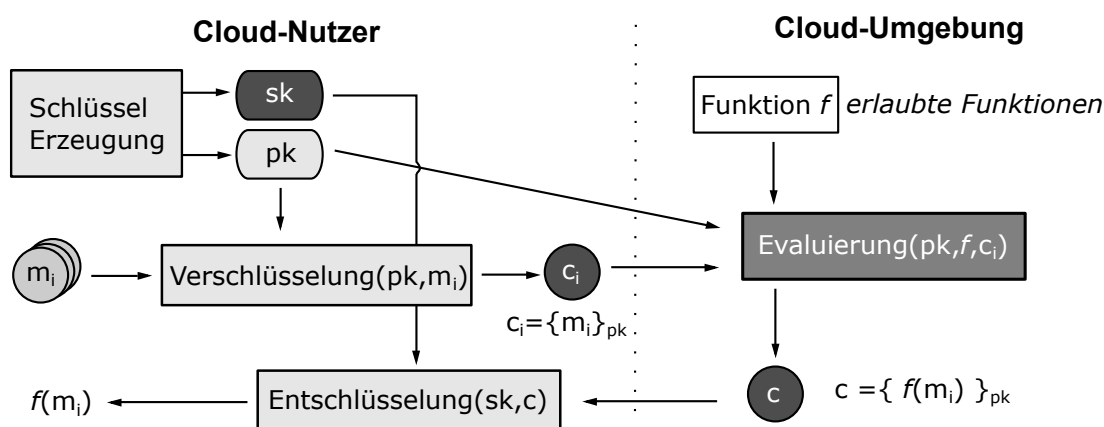


Abbildung 4.7.: Das Prinzip der homomorphen Verschlüsselung unterteilt sich auf den Cloud-Nutzer und die Cloud-Umgebung. Der Nutzer erzeugt einen privaten und einen öffentlichen Schlüssel (sk, pk). Die Nachrichten m_i werden die Geheimtexte c_i . Die Schreibweise $c_i = m_{i,pk}$ drückt aus, dass c_i die Verschlüsselung von m_i unter dem öffentlichen Schlüssel pk darstellt. In der Cloud-Umgebung erfolgt die Berechnung der Funktion f . Anschließend wird der resultierende Geheimtext c an den Nutzer übertragen. Mit seinem privaten Schlüssel sk ist dieser in der Lage den Funktionswert $f(m_i)$ zu erhalten.

CSP entspricht, berechnet werden. Um eine Aussage über die Komplexität von f , und damit über die effiziente Berechnung machen zu können führt [Gen10] jedoch boolesche Schaltkreise⁴ ein. Dies bietet den Vorteil die Berechnung von f in kleine, einfache Schritte aufzuteilen. Ein voll homomorphes Verfahren ist möglich, wenn mit dem Evaluierungsalgorithmus beliebig oft addiert, subtrahiert und multipliziert werden kann. Das vorgestellte Verfahren von Gentry basiert auf Abständen von Vielfachen großer Zahlen. Dabei ist entscheidend, dass dieser Abstand die gleiche Parität wie der Klartext m hat, d.h. das er gerade oder ungerade ist. Das Anwenden des Evaluierungsalgorithmus auf einen Geheimtext verändert diesen Abstand. Ist diese Änderung zu stark, so ist das Ergebnis der ausgeführten Funktion nicht länger korrekt. Wichtig zu verstehen ist, dass die hier vorgestellten homomorphen Eigenschaften auf Bit-Ebene ausgeführt werden. Nur aus diesen Grund ist nicht der Abstand von Bedeutung, sondern dessen Parität. Wird durch die Evaluierung dieser zu stark ma-

⁴Vgl. Abb.4.4 (a)

nipuliert, so ist die Wahrscheinlichkeit hoch, dass sich die Parität ändert. Daraus ergeben sich zwei Fakten: Erstens reicht das obig vorgestellte System offensichtlich nicht aus, um beliebig komplexe Funktionen sicher zu berechnen. Zweitens wird bereits an dieser Stelle der enorme Aufwand deutlich, der nötig wäre um die wie im Fall von Gleichung 4.1 zwei Klartexte m_1, m_2 zu addieren. Da dies Bitweise geschieht muss mit Hilfe der vorhandenen booleschen Schaltkreise ein Addierwerk konstruiert werden, dass der arithmetischen Berechnung entspricht.

Zur Lösung des Problems mit dem Veränderung der Abstände, schlägt Gentry [Gen10] das Prinzip des *Bootstrappings* vor.⁵ Dazu wird der Evaluierungsalgorithmus genutzt, um zum einerseits einen Teil der Funktion f zu berechnen und andererseits die Abstandsveränderung rückgängig zu machen. Da die einzige Möglichkeit diese Veränderung rückgängig zu machen im Entschlüsseln besteht, wird innerhalb des Evaluierungsalgorithmus eine Neuverschlüsselung vorgenommen. Die nachfolgende Abbildung 4.8 verdeutlicht den Aufbau eines addierenden Neuverschlüsselungsschaltkreises.

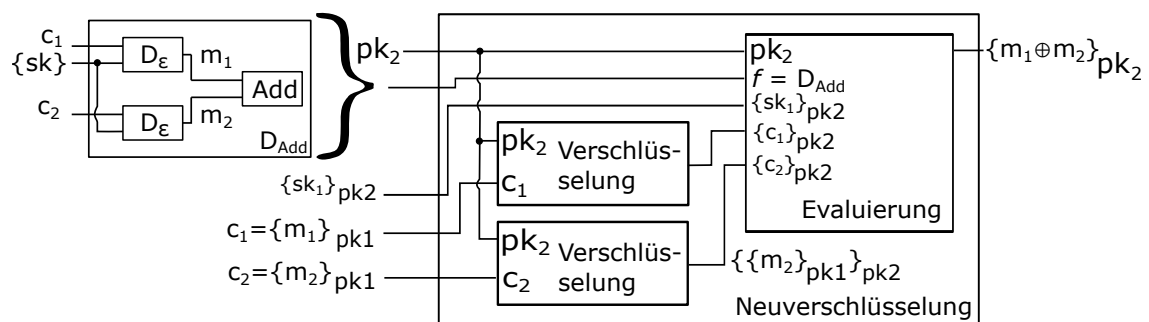


Abbildung 4.8.: Die Abbildung zeigt das Vorgehen der voll homomorphen Verschlüsselung durch die Nutzung von Bootstrapping in Form eines Neuverschlüsselungsschaltkreises. Die öffentlichen Schlüssel pk_i sind zur besseren Übersicht farblich hervorgehoben. Der Neuverschlüsselungsschaltkreis addiert die beiden Geheimtexte c_1, c_2 und entfernt das Noise des resultierenden Geheimtextes $m_1 + m_2$. Grund dafür ist das der resultierenden Geheimtext mit einem neuen öffentlichen Schlüssel pk_2 verschlüsselt ist. Die Funktion die auf die Eingabewerte angewandt wird, ist D_{Add} . Neben der Addition führt diese Funktion eine Entschlüsselungen mit dem gegebenen privaten Schlüssel durch. Dennoch liegen die Werte m_1, m_2 zu keinem Zeitpunkt unverschlüsselt vor, da diese vor der Evaluierung mit dem Schlüssel pk_2 neuverschlüsselt werden.

Der entscheidende Effekt des Vorgehens ist, dass die eingehenden Geheimtexte c_1 und c_2 mit dem öffentlichen Schlüssel pk_1 (blau) verschlüsselt übergeben werden, der resultierende Geheimtext jedoch mit dem öffentlichen Schlüssel pk_2 (rot) verschlüsselt ist. Der wichtigste Aspekt ist die Evaluierung der addierenden Entschlüsselungsfunktion D_{Add} . Diese entfernt die Abstandsveränderung unter pk_1 , fügt jedoch eine Veränderung des Abstands durch die Addition unter pk_2 hinzu. Interessant hierbei ist, dass die innere Verschlüsselung unter pk_1 entfernt werden kann, obwohl c_1 und c_2 weiterhin mit pk_2 verschlüsseln ist. Durch diesen Effekt liegen die Daten trotz Neuverschlüsselung nie im Klartext vor. Durch Rekursion dieses Vorgehensweise ist ein voll homomorphes Verschlüsselungsverfahren realisierbar.

Obwohl damit das Vorhandensein eines voll homomorphen Verschlüsselungsverfahrens bewiesen wurde, verdeutlicht die Beschreibung den enormen Aufwand nötig ist um beliebige Funktionen auf verschlüsselte Daten anzuwenden. Die Laufzeiten für eine bootstrapping Operation geben Gentry und Halevi [GH11] nach hochgradigen Optimierungen zwischen 30 Sekunden und 30 Minuten an,

⁵<http://www.google.com/patents/US8630422>

die allem von der Wahl der Sicherheitsparameter abhängen. Die letztere Zeitangabe mit Sicherheitsparametern zustande kamen, die laut der Autoren *möglicherweise so sicher sind wie RSA-1024* sind. Die Größe des öffentlichen Schlüssels pk beträgt bei dieser Parameterwahl bereits 2,25 GByte. Die Ergebnisse zeigen, dass Berechnungen beliebiger Funktionen unter Verschlüsselung möglich, jedoch hochgradig unpraktikabel sind. Die Implementierungen nach Gentry finden sich im Internet.⁶ Seit der Publizierung im Jahr 2009 gab es in der Fachwelt zahlreiche Vorschläge für die Optimierung. Beispiele sind Smart und Vercauteren [SV10] oder Stehle und Steinfeld [SS10b].

Die Autoren [GHS12, NT12, GPS13] bieten weiterhin optimierte Verfahren, welche durch das Packen von Klartexten und Schlüsseloptimierungen eine Stapelverarbeitung von Geheimtexten ermöglichen. Das generelle Anliegen der Optimierungen ist es, die Anzahl der Bootstrapping-Operationen und damit die aufwendigen Neuverschlüsselungsoperationen zu verringern. Obwohl FHE-Verfahren bei der Anwendung unpraktikabel sind, werden diese für spezielle Fälle in der Praxis eingesetzt. Im Jahr 2013 fand dazu in Okinawa der Workshop on Applied Homomorphic Cryptography statt. Die Beiträge von Kamara und Raykova [KR13] versuchten beispielsweise FHE per Map-Reduce-Verfahren zu parallelisieren, Kamara und Wei [KW13] dagegen untersuchten eine Verbindung zwischen GCs und FHE herzustellen. Der Beitrag von Tsujii et al. [TDF⁺13] nutzt das Paillier-Kryptosystem, um eine sichere Datenverarbeitung zu ermöglichen. Dieses vorgeschlagene Modell setzt jedoch zahlreiche teilweise vertrauenswürdige Teilnehmer voraus. Statistische sichere Berechnungen über 100 Datensätze erfolgenden dabei in rund 20 Minuten. Ein weiterer Ansatz von Brenner et al. [BWvS11] bietet Sicherheit in Form einer simulierten verschlüsselten CPU. Die Autoren beschreiben in ihrer Arbeit ausführlich die implementierten Komponenten. Mittels eines Assemblers können beliebige Programme erstellt werden, die auf der simulierten CPU sicher ausgeführt wird. Die Arbeit von Mani et al. [MSG13] beschäftigt sich mit der Verwendung von FHE bzw. homomorpher Verschlüsselung im Kontext von sicheren Datenbanken. Die Autoren kommen zu dem Ergebnis, dass bisherige praktische Lösungen durch die Verschlüsselung der Daten nur ein sehr geringen Anteil der Query-Verarbeitung innerhalb der Cloud realisiert werden kann und der Rest vom Client übernommen werden muss. Eine entscheidende Fragestellung der Autoren war, wie man eine Query in einen FHE Schaltkreis überführt. Dazu bildeten diese arithmetische und vergleichende Operatoren auf Grundlage bitweiser XOR und AND Operationen ab. Basierend auf diesen werden im Anschluss typische Datenbankoperatoren definiert. Leider enthält die Arbeit keine Ergebnisse bzgl. einer der praktischen Umsetzung.

Wie bereits Gentry [Gen10] hervorhebt, können Geschwindigkeitsverbesserungen im sublinearen Bereich durch Random-Access(RAM) nicht mit verschlüsselten Daten funktionieren und damit konsequenter Weise auch nicht mit FHE. Ohne Wissen über die Daten ist zumindest ein linearer Aufwand mit der Anzahl der Eingaben erforderlich. Anderenfalls würden zwangsweise Informationen preisgegeben. Diese Prinzipien werden ebenfalls in Abschnitt ORAM 4.6 erläutert.

Durch die feste Größe der Schaltkreise ist auch die Größe der Ausgabe fest. So muss eine Suchanfrage spezifizieren, wie groß die Menge der Ausgabe ist, die erwartet wird beispielsweise ein MByte oder n Zeilen einer Datenbank. Ist die Datenmenge geringer, muss entsprechend aufgefüllt, umgekehrt abgeschnitten werden. Dies ist laut Gentry unvermeidbar, solange die Cloud nichts über die Anfrage und die Daten weiß und keine Informationen preisgegeben werden sollen.

Von Dijk und Ari [DJ10] wurde zudem bewiesen, dass Berechnungen von FHE unter dem gleichen Schlüssel stattfinden müssen und daher FHE mit verschiedenen Schlüsseln nicht existieren können. Das bedeutet, dass gemeinsame Berechnungen, ohne das Teilnehmer Eingaben bzw. Ergebnisse anderer Teilnehmer erfahren, mittels FHE nicht möglich sind.

Neben diesen Grenzen von FHE stellten sich Neahrig et al. [NLV11] im Rahmen ihrer Forschung die Frage, ob HE überhaupt praktikabel sein kann und implementieren dafür ein teilweise homo-

⁶Vgl. <https://github.com/hcrypt-project/libScarab>

morphes Verfahren um damit Funktionen wie das arithmetische Mittel zu berechnen. Die Aussage der Autoren *it seems that all known fully homomorphic encryption schemes have a long way to go before they can be used in practice* stellt die heutige und zukünftige praktische Verwendbarkeit in Frage. Dennoch konnten die Autoren zeigen, dass spezielle Anwendungsfälle sehr wohl praktische Relevanz haben.⁷ Bemerkenswert ist die Argumentation von Danezis et al.[DLB11], die im Ergebnis ihrer Arbeiten behaupten, dass das Problem des sicheren Cloud Computings, auch mit praktikable FHE nicht gelöst ist. Dabei begründen sie dies damit, dass die Kosten anderer kryptographischer Operationen das Gesamtsystem trotzdem ineffizient macht. Zudem verweisen die Autoren drauf, dass dem Server das Ergebnis, ohne eine Entschlüsselung, nicht von Nutzen für weitere Berechnungen sein kann. Mittels zusätzlichen Roundtrip müsste der Nutzer dem Server eine Entscheidung anweisen, was ohne Zweifel die Performance der Applikation stark senkt. Ferner bietet FHE dem Nutzer keine Möglichkeiten einer Verifikation der Ergebnisse.

⁷Beispiele sind: Summe von 100 Zahlen benötigt 20ms, Berechnung der Summe und Summe der Quadrate für arithmetisches Mittel und Standardabweichung benötigt 6s

4.2.4. Funktionale Verschlüsselung

Wie Boneh et al. [BSW11] feststellen, existieren im allgemeinen Forschungsumfeld zwei grundlegende Annahmen zum Thema Verschlüsselung:

1. Verschlüsselung ist eine Methode, Nachrichten oder Daten an einzelne Personen zu senden, die einen privaten Schlüssel besitzen und mit Hilfe dieses Schlüssels die Nachrichten entschlüsseln können.
2. Der Zugriff auf verschlüsselte Nachrichten entspricht dem *Alles-oder-Nichts-Prinzip*, d.h. entweder kann entschlüsselt und der Klartext vollständig eingesehen werden, oder kann nicht entschlüsselt werden, womit keine Informationen über den Klartext bekannt werden.

Auch wenn dies für klassische symmetrische/asymmetrische Verfahren zutrifft, gibt es zahlreiche Verfahren, die über obige Annahmen hinaus gehen. Die *funktionale Verschlüsselung* wurde erstmals 2011 von Boneh et al. [BSW11] definiert und kategorisiert. Dabei handelt es sich bei funktionaler Verschlüsselung um Verfahren, die dem Besitzer des privaten Schlüssels die Möglichkeit bietet, eine spezifische Funktion auf verschlüsselten Daten auszuführen, jedoch nichts (weiter) über die Daten zu lernen. Zu diesen Verfahren zählen unter anderem *Identity Based Encryption* [Sha85, BF01, BPR⁺08], *Broadcast Encryption* [FN94, HS02, BGW05], *Durchsuchbare Verschlüsselung*, *Attributbasierende Verschlüsselung* und *Ordnungsschützende Verschlüsselung*. Auf die Letzteren wird nachfolgend eingegangen.

Durchsuchbare Verschlüsselung

Die durchsuchbare Verschlüsselung (Searchable Encryption) ist ein kryptographisches Primitiv, das es erlaubt, innerhalb verschlüsselter Daten ein gegebenes Muster zu suchen [TPPG11]. Kamara et al. [KL10] und Sun et al. [SLHL14] bieten in ihren Arbeiten einen umfassenden Einblick in diese Thematik. Die Abbildung 4.9 zeigt ein typisches Szenario der Suche auf verschlüsselten Daten.

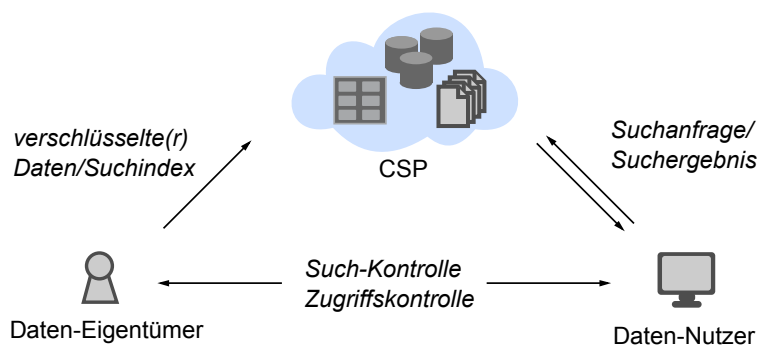


Abbildung 4.9.: Die Suche innerhalb von verschlüsselten Daten angelehnt an Cao et al. [CWL⁺11]. In diesem Szenario wird eine Unterscheidung zwischen Daten-Eigentümer und Daten-Nutzer vorgenommen und der Suchvorgang damit um einen Schritt erweitert. So muss der Datennutzer entsprechende Berechtigungen für die Daten und den Suchzugriff erhalten.

Diese zeigt die typischen Teilnehmer eines sicheren Suchsystems involvierter CSP, Datenbesitzer und Datennutzer. Weiterhin verdeutlicht die obige Abbildung den Ablauf der Suche. Der Datenbesitzer speichert die verschlüsselten Daten und verschlüsselten Indizes innerhalb der Cloud Server. Während die Daten mittels eines Standardverfahrens (z.B. AES) verschlüsselt werden können, ist für die Verschlüsselung der Indizes ein spezielles Verfahren notwendig. Möchte ein Nutzer auf die Daten

zugreifen, generiert er dafür eine verschlüsselte Version des Suchbegriffs mit dem gleichen speziellen Verfahren. Dadurch ist es möglich, Indizes zu durchsuchen und die entsprechenden Dateien zu finden ohne jedoch den Suchbegriff oder den Index offenzulegen.

Durchsuchbaren Verschlüsselung lassen sich ebenfalls in symmetrische und asymmetrische Suchverfahren unterteilen. Wesentliche Unterschiede sind neben der Leistungsfähigkeit, die Erzeugung der verschlüsselten Suchbegriffe und die Flexibilität der Verfahren. Beim symmetrischen Suchverfahren kann nur der Datenbesitzer die Begriffe erzeugen und der Nutzer würde diese beim Besitzer anfordern, vgl. Abbildung 4.9. Song et al. [SWP00] stellen einen praktikablen, symmetrisch verschlüsselnden Ansatz vor, der von Popa et al. [PRZB11] sowie Schaad et al. [SBK⁺14] genutzt wird, um in verschlüsselten SQL-Queries einen *like* Operator zu erhalten. Detailliertere Informationen dazu enthält der Abschnitt zum Thema Datenbanken 4.3.

Die gebotenen Sicherheitsgarantien der durchsuchbaren Verschlüsselung unterscheiden sich von denen der klassischen Verschlüsselungsverfahren. Es sei angemerkt, dass nach bestimmter Zeit (z.B. nach vielen Suchanfragen) eine bestimmte Menge von Dokumenten, die ein bestimmtes Wort enthalten, nützliche Informationen preisgeben können. Daher ist es möglich, dass der Server Annahmen über die Suchmuster des Nutzer machen kann, um Suchbegriffe zu erraten. Wichtig für das Verständnis der Problemstellung ist die Tatsache, dass der Suchprozess Informationen an den Provider weitergibt und dass das Weitergegebene ist exakt das, was der CSP beim Prozess des Datenbereitstellens lernen würde (z.B. das die gelieferte Dateien ein gemeinsames Wort enthalten). Mit anderen Worten: die Information die der CSP erhält, wird nicht durch das kryptographische Primitiv preisgegeben, sondern durch die Art wie der Service genutzt wird. Entscheidend ist die Erkenntnis, dass die Informationspreisgabe in gewissem Sinne einem effizienten und verlässlichen Cloud Speicher-Service innewohnt. Die einzige bekannte Alternative besteht darin, dem Provider eine falsch positive Menge an Informationen schicken zu lassen und das Ergebnis vom Client lokal zu filtern. Dies bedeutet Ineffizienz und Mehraufwas in Kommunikation und Berechnung. Das beschriebene Primitiv verdeutlicht hier stark das Prinzip der Delegation mit begrenztem Wissen.

Attributbasierte Verschlüsselung

Attributbasierte Verschlüsselung (Attribute based Encryption – ABE) ist ein kryptographisches Primitiv, das es erlaubt, Geheimtexten und/oder Nutzerschlüsseln eine Policy zuzuordnen, die den Zugriff auf verschlüsselten Informationen regelt [Mül11]. Der Begriff wurde erstmals 2005 von Sahai und Waters [SW05] im Ergebnis ihrer Forschungsarbeiten geprägt. Dabei werden zwei Arten von ABE unterschieden, wie Abbildung 4.10 dargestellt.

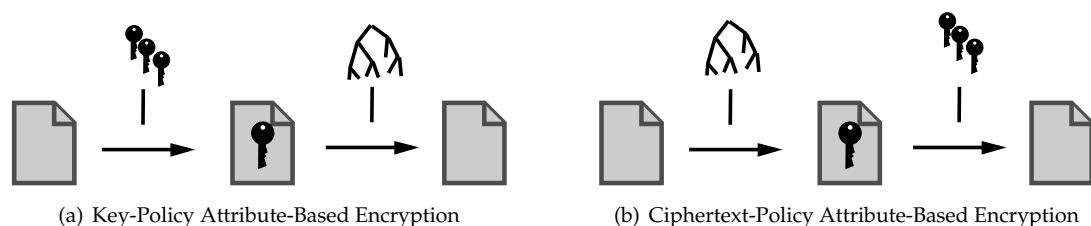


Abbildung 4.10.: Die schematische Darstellung zweier Arten von attribut-basierter Verschlüsselung analog zu Müller [Mül11]. Die links dargestellte Key-Policy Attribute-Based Encryption nutzt für die Verschlüsselung eine Anzahl von Schlüsselattributen, und für die Entschlüsselung eine Policy. Im zweiten, rechts dargestellten Fall, ist dies umgekehrt.

Wie Abbildung 4.10(a) veranschaulicht, wird der Geheimtext im Falle einer *Key-Policy Attribute-Based Encryption* (KP-ABE) mit einer Anzahl von Attributen verschlüsselt, wobei jeder (geheime) Nutzerschlüssel ist mit einer Policy verbunden, die festlegt welche Geheimtexte der Nutzer entschlüsseln darf. Eine solche Policy ist in der Regel ein boolescher Ausdruck. Die *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) nutzt die Policy bei Verschlüsselung der Geheimtexte. Wie die Abbildung 4.10(b) darstellt, kann jeder Nutzer mit den entsprechenden Attributen den Geheimtext entschlüsseln, vergleiche hierzu Abbildung . Die Attribute sind üblicherweise relevante Credentials, wie Abteilungs- oder Firmenzugehörigkeit, bestimmte Stati oder Gültigkeitszeiträume. In der Fachwelt gibt es eine Vielzahl an ABE-Verfahren mit verschiedenen Funktionalitäten und Policy-Sprachen, als Beispiele seien [Cha07, AI09, ZH10, LW11] aufgeführt. Nach Santos et al. [SRGS12] reduziert sich beim Einsatz von ABE der Mehraufwand des Schlüsselmanagements. Dies begründet sich mit der Verkettung von Zugriffsrechten und der sich daraus ergebenden Möglichkeit Daten zu entschlüsseln. Auf spezielle Gruppenschlüssel und deren Verwaltung, die häufig eine besondere Herausforderung im Schlüsselmanagement darstellen, kann verzichtet werden. Dennoch verschiebt sich der Aufwand dahingehend, dass eine Verwaltung der Policies bzw. der Attribute erforderlich wird.

Ordnungsschützende Verschlüsselung

Agrawal et al. [AKSX04] prägten 2004 in ihrer Arbeit erstmals das kryptographische Primitiv der ordnungsschützenden Verschlüsselung (Order Preserving Encryption – OPE). OPE ist ein deterministisches Verschlüsselungsschema, in dem die Verschlüsselungsoperation die numerische Reihenfolge des Klartextes beibehält [BCLO09]. Natürlich kann ein solches Schema nicht die Standard-Sicherheitsanforderungen wie IND-CPA⁸ erfüllen, da es einerseits deterministisch ist und andererseits die Reihenfolgebeziehungen der Klartexte offenlegt. OPE wird auf Grund seiner Eigenschaften von Propa et.al [PRZB11] und Schaad et al. [SBK⁺14] in der Datenbankverschlüsselung eingesetzt und beispielsweise größer-gleich Vergleiche auf verschlüsselten Daten. Schaad et al. [SBK⁺14] nutzen das Schema von Boldreva [BCLO09, BCO11], das OPE mit nachweislich optimaler Sicherheit und Unveränderlichkeit.

4.2.5. Zusammenfassung

Im Abschnitt werden Maßnahmen beschrieben, um die Applikationslogik einer Cloud-Applikation gegenüber dem CSP oder Dritten zu schützen. Dieser Bereich erlebte vor allem durch die FHE von Gentry ein breites Interesse in der Forschung. Durch Verfügbarkeit von kostengünstigen, leicht beziehbaren IT-Ressourcen aus der Cloud sind diese zunehmend von praktischer Relevanz. Doch auch mit neuen und effizienten Verfahren sind nicht alle Probleme gelöst und es entstehen neue Herausforderungen. Die diskutierten alternative Ansätze der funktionalen Verschlüsselung zeigen Möglichkeiten, die Applikationslogik auf nicht oder nur teilweise vertrauenswürdige Cloud-Server auszulagern. Regelmäßig müssen dafür Leistungs- oder Funktionalitätseinbußen hingenommen werden, wodurch deren praktische Anwendbarkeit in Frage gestellt wird. Das Prinzip der Delegation mit begrenztem Wissen wird in diesem Abschnitt besonders deutlich, Verfahren wie GF und HE, die nur wenig Informationen preisgeben, sind aufwändig und erzeugen zusätzliche Arbeit vom Nutzer. Die vorgestellten Verfahren der funktionalen Verschlüsselung geben vergleichsweise viele Informationen preis und bieten effizientere Berechnungen und weniger zusätzliche Arbeit auf der Seite des Cloud-Nutzers.

⁸Bei diesem Kriterium geht es um die Ununterscheidbarkeit von Geheimtexten. Diese ist für deterministische Verfahren trivialerweise nicht zu erreichen.

4.3. Datenbanken

In diesem Abschnitt werden Lösungen für die Speicherung von Daten, in Form von Datenbanken, in Cloud-Applikationen erläutert und diskutiert. Im Fokus der beschriebenen Ansätze liegt somit der Schutz des Daten-Layers, welcher innerhalb der Taxonomie aus Kapitel 2.3 aufgeführt wird. Dabei unterscheidet der Abschnitt Datenbankverschlüsselung nach Bajaj und Sion [BS11], welche einerseits Datenbanken, die gemeinsam mit sicherer Hardware agieren und andererseits Datenbanksysteme, die verschlüsselte Anfragen auf verschlüsselten Daten differenzieren. Erstere werden im Rahmen dieser Arbeit im Abschnitt Trusted Computing 5 erörtert und auf Letztere wird im nachfolgend eingegangen.

Sichere, geschützte Datenbankabfragen und Lösungen haben enge Beziehungen zu funktionalen Verschlüsselungsmethoden wie durchsuchbare oder ordnungsschützende Verschlüsselung, die im vorigen Abschnitt beschrieben wurden. Da auch in Datenbanken mit Indizes gearbeitet wird, ist der Zusammenhang unmittelbar erklärbar.

Hacigümüs et al. [HILM02] schlagen in ihrer Arbeit als eine der Ersten eine Methode vor, SQL-Queries über teilweise verschleierte und ausgelagerte Daten auszuführen. Die Autoren teilen die Daten in geheime Partitionen auf und schreiben die Queries über Originaldaten in der Form neu, dass diese einen Identifikator für die entsprechende Partition erhalten. Die Information, die dadurch an den Server gereicht wird, ist 1-aus-s, wobei s die Partitionsgröße ist. Mit dem Vorgehen wird die client- und serverseitige Verarbeitung als Funktion bzgl. der Datensegmentgröße ausgeglichen. In einem Extremfall ist der Datenschutz vollständig, durch die kleine Segmentgröße, komprimiert, die client-seitige Verarbeitung aber minimal. Das andere Extrem ist ein hochgradiger Schutz der Daten auf Kosten der clientseitigen Verarbeitung der Queries. Diese würde vollständig auf der Nutzerseite erfolgen, nachdem dieser die gesamte Datenmenge empfangen hat. Das Prinzip der Delegation mit begrenztem Wissen wird unmittelbar deutlich.

Die Autoren Hore et al. [HMT04] suchen nach der optimalen Größe der Partitionen für eine bestimmte Anzahl von Queries. Analog zum vorhergehenden Ansatz, wird eine Datenpartitionierung vorgenommen, welche einen *fast-private* Index von Attributen als sensitiv auffasst. Die Autoren diskutieren zudem einen *Sicherheits-Nutzen Trade-off* für ihren Index, das heißt: wie viele Informationen muss preisgegeben werden, damit sich ein Auslagern noch lohnt. Der Hauptnachteil dieser Lösungsansätze ist der lineare Sicherheits-Kosten Trade-off, die Kosten also gleich mit der zunehmenden Sicherheit steigen.

Ein moderner Ansatz ist die von Popa et al. [PRZB11, PZB11] entwickelte CryptDB. Diese Lösung steht für eine proxybasierte verschlüsselte Datenbank mit SQL-basierenden sicheren Abfragen. Die Abbildung 4.11 illustriert die Architektur dieses Ansatzes.

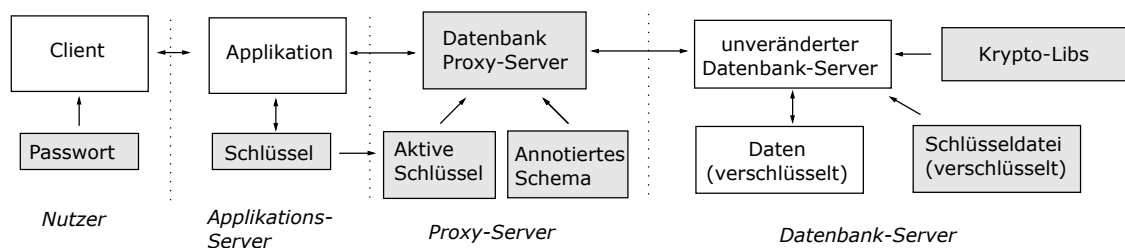


Abbildung 4.11.: Die Architektur der CryptDB nach Popa et al. [PRZB11] zeigt die Möglichkeiten einer proxy-server-basierten Datenbankverschlüsselung. Die Sicherheitskomponenten sind dunkler dargestellt.

Die Schlüsselideen von CryptDB sind die *SQL-aware Encryption* und die *Adjustable Query-based*

Encryption. Erste beschreibt die effektive Kombination von erwünschten SQL-Merkmalen und Verschlüsselungsverfahren. Beispielsweise eignen sich deterministische Verfahren, um in SQL-Queries *WHERE* Klauseln durch Vergleiche zu realisieren. Die Daten der Datenbank sind dabei geschachtelt und mit verschiedenen Verfahren verschlüsselt. Um alle Arten von Queries zu ermöglichen, ist die Verschlüsselungsebene der Daten anpassbar (Adjustable Query-based Encryption). Der CryptDB-Ansatz bietet nach Aussagen von Popa et al. Vertraulichkeit, aber garantiert keine Integrität, Aktualität oder Vollständigkeit der gelieferten Ergebnisse. Der Lösungsansatz ist mit geringen Anpassungen von jeder Applikation auf gewöhnliche DBMS einsetzbar, zudem ist der Quellcode für Forschungszwecke frei zugänglich⁹. Die von Reinhold et al. [RBK⁺14b] entwickelte Lösung *Encryption Layer* basiert unter anderem auf diesem Lösungsansatz.

Die Lösung Monomi, von Tu et al. [TKMZ13], kann als Erweiterung von CryptDB verstanden werden. Die Autoren beschreiben erweiterte Möglichkeiten um auch analytische Queries verschlüsselt auszuführen. Diese nutzt dabei, analog zu Hacigümüs et al., eine Aufteilung der Rechenanteile zwischen Nutzer und CSP. Für eine effektive Nutzung von Mononmie ist ein hohes Anteil an analytischen Queries erforderlich, damit sich der Mehraufwand lohnt. Im praktischen Umfeld ist dies durch Nutzung von Werkzeugen, wie beispielsweise Hibernate¹⁰, nicht immer der Fall. Für einfache Datenbankoperationen ist CryptDB Lösung zumeist ausreichend.

Die von Curino et al. [CJP⁺11] vorgestellte Relational Cloud ist ein Prototyp eines Database-as-a-Services. Dieser wurde entwickelt, um die CryptDB, die zur Gewährleistung der Vertraulichkeit genutzt wird, mit zusätzlicher Sicherheit in Form von Verfügbarkeit, Backup und Skalierung zu erweitern. Die Abbildung 4.12 illustriert die Architektur des Ansatzes.

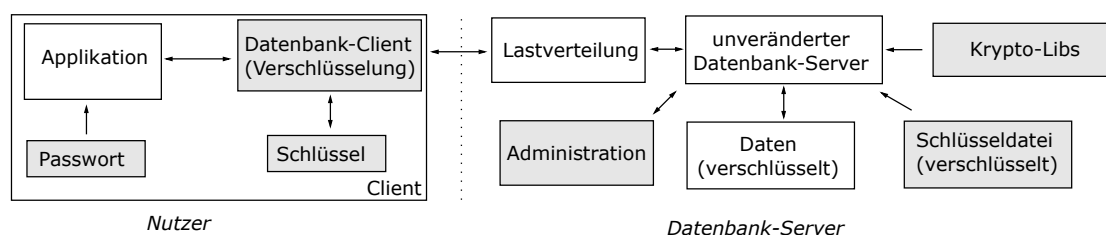


Abbildung 4.12.: Die Architektur der Relational Cloud nach Curino et al. [CJP⁺11] zeigt die Möglichkeit einer client basieren Datenbankverschlüsselung. Die Sicherheitskomponenten sind dunkler dargestellt.

Zu erkennen ist, dass, im Gegensatz zum CryptDB Ansatz, auf Einsatz eines Proxy Servers, der die Verschlüsselungsaufgaben wahrnimmt, verzichtet wurde und werden nutzerseitig im entwickelten Client vollzogen. Leider wurde im Gegensatz zur CryptDB-Lösung kein Quellcode veröffentlicht. Die Autoren Schaad et al. [SBK⁺14] analysieren, basierend auf dem Ansatz der CryptDB (Adjustable Query-based-Encryption) die Fragestellung, ob man optimale Sicherheits-Policys für eine ausgelagerte Datenbank findet, welchen den Sicherheits- und Funktionsanforderungen vollständig entspricht. Dabei stellen die Autoren einen Lösungsansatz vor und zeigen, dass die Fragestellung nicht trivial zu beantworten ist. Das Ziel der Autoren ist es, die praktische Eignung der Adjustable-Database-Encryption zu verdeutlichen und die dabei entstehenden Probleme der Datenklassifizierung zu lösen.

Ein weiterer Ansatz wird von Park et. al [PPL11] mit der Lösung PKIS beschrieben, die praktisch, anwendbare Möglichkeit der Schlüsselwort-Suche über verschlüsselte Daten in einer Datenbank vorschlägt. Als drei Parteien sind in diesem Ansatz Nutzer, Gruppenadministrator und Datenserver involviert. Der Administrator verwaltet dabei Gruppen- und Suchschlüssel aller Gruppen und ist mit

⁹ [hgit://g.csail.mit.edu/cryptdb](https://github.com/mit-cryptdb/cryptdb) letzter Zugriff:09.07.2015

¹⁰Für Details zu Hibernate siehe <http://hibernate.org> (letzter Zugriff 16.06.2015)

dem Proxyserver des CryptDB-Ansatzes vergleichbar. Die Umsetzung des Verfahrens erfolgte in einem MS SQL Server 2000, über welche die Autoren beschreiben jedoch keine Details veröffentlicht haben. Der praktische Anwendungsfall der PKIS-Lösung zielt vor allem auf das Dokumentenmanagement und das Teilen von Daten ab. Der Mehraufwand im Vergleich zu einem unverschlüsselten System wird ebenfalls nicht angegeben.

Neben der Sicherung von relationalen Datenbanken, sollen an dieser Stelle auch Ansätze für nicht relationale Datenbanken erörtert werden. Die Literaturrecherche ergab jedoch, dass in diesem Gebiet weit weniger stark geforscht wird, als in dem von sicheren relationalen Datenbanken. NoSQL-Datenbanken zeichnen sich vor allem durch ihre geringere Strukturiertheit (Schemalosigkeit) gegenüber relationalen Datenbanken ab. Aus diesem Grund liegt die Annahme nahe, dass eine verschlüsselte NoSQL-Datenbank einfacher zu realisieren sei als eine relationale Datenbank, dies konnte im Rahmen dieser Arbeit jedoch weder veri- noch falsifiziert werden. Ein Ansatz zur Absicherung von NoSQL-Datenbanken von Guo et al. [GZLL13] nutzten die teilweise homomorphe Verschlüsselungssystem von Elgamal und Paillier. Damit versuchen die Autoren die Daten innerhalb einer BerkeleyDB [Ora15a] zu schützen und sind dennoch in der Lage weitere Datenbankoperationen auszuführen. Ferner implementierten die Autoren eigene Algorithmen zur Indizierung und Datenmanipulation.

Analog zu Bocker et al.[BT10], sollen in diesem Abschnitt abschließend die Lösungen zweier öffentlicher CSP bzgl. der Sicherheit verglichen werden. Die Autoren stellen in ihrer Arbeit Microsoft Azure SQL und Amazon RDS gegenüber und stellen fest, dass diese Standardlösungen neben der praktischen Verwendbarkeit weitere Vorteile durch Verfügbarkeit und Redundanz bieten. Andererseits bietet keiner der Services standardmäßig einen Vertraulichkeitsschutz der Daten. Azure SQL unterstützt keinen Schutzmaßnahmen bis auf verschlüsselten Kommunikation.¹¹ Der Service Amazon RDS [AWS13] unterstützt Oracle Transparent Data Encryption [Ora15b] und diese ohne zusätzliche direkte Kosten.¹² Diese Unterstützung bietet Schutz für persistente Daten und ist mit einem providerseitiges Schlüsselmanagement kombinierbar. Die Daten werden bei Bedarf on-the-fly auf der Seite des Cloud-Providers ent- und verschlüsselt. Bei gleichzeitiger Nutzung des sicheren providerseitiges Schlüsselmanagement AWS CloudHSM¹³, ist zudem eine Vertraulichkeit der gespeicherten Daten gegenüber Dritten und dem CSP realisierbar. Die Kosten für diese Lösung steigen jedoch stark an, da die zusätzlich zu Verfügung gestellten Ressourcen frakturiert werden.

¹¹Stand August 2013

¹²Die Kosten können durch gestiegene IT-Ressourcen jedoch indirekt steigen.

¹³Details zu CloudHSM bietet der Abschnitt Trusted Cloud Computing 5.3

Zusammenfassung

Dieser Abschnitt beschreibt Datenbanksysteme, die verschlüsselte Anfragen auf verschlüsselten Daten auswerten, die laut Bajaj und Sion [BS11] auch als *Queries on encrypted data* Systeme bezeichnet werden. Die Tabelle 4.1 bietet eine Zusammenfassung der Informationen aus den Evaluierungsergebnissen im Teil III der Arbeit. Dabei werden die Bewertungen nach den in Abschnitt 2.4.1 diktierten Schutzzielen¹⁴ dargestellt. Eine ausführliche Erläuterung der Kriterien enthält der Abschnitt 7.1 der vorliegenden Arbeit.

Tabelle 4.1.: Die Zusammenfassung verschiedener sicherere Datenbanklösungsansätze nach den in der Arbeit eingeführten Schutzzielen.

Bezeichnung	CO	IG	AV	UL	TR	IN	Anmerkungen
Hacigumus et al. [HILM02]	✓	✗	✓	✗	✓	✗	keine
Hore et. al. [HMT04]	✓	✗	✓	✗	✓	✗	keine
CryptDB [PRZB11]	✓	✗	✓	✗	✓	✗	keine
MONOMI [TKMZ13]	✓	✗	✓	✗	✓	✗	CryptDB Erweiterung
Relational Cloud [CJP ⁺ 11]	✓	✓	✗	✗	✓	✗	CryptDB Erweiterung
Azure SQL[BT10]	✗	✗	✓	✗	✓	✗	Standard Servicenutzung
Amazon RDS [AWS13]	✗	✗	✓	✗	✓	✗	Standard Servicenutzung
Amazon RDS* [AWS13]	✓	✓	✓	✗	✓	✓	*in Kombination mit Oracle Transparent Data Encryption und AmazonHSM
PKIS [PPL11]	✓	✗	✓	✗	✓	✗	Dokumentenmanagement
Guo et al. [GZLL13]	✓	✗	✓	✗	✓	✗	NoSQL Lösung

Wie die Tabelle verdeutlicht sind die Lösungen darauf gerichtet, eine Vertraulichkeit der Daten zu erreichen. Nur ein Forschungsansatz, die Relational Cloud, bietet beispielsweise erweiterten Integritätsschutz an. Leider gab es in den letzten Jahren keine weiteren Veröffentlichungen zu diesem Ansatz. Grundsätzlich ist damit festzustellen, dass der Bereich der nicht relationalen Datenbanken in der Forschung bisher nur von geringem Interesse ist. Dies ist in sofern verwunderlich, da nicht relationale Datenbanken, nach Meinung des Autors weniger komplexe Anforderungen an die technische Realisierung stellen und quasi-konsistente Datenzustände einer leichteren Handhabung von Transaktionen und damit eine einfachere Verwaltung von Schlüsseln ermöglichen könnten.

Deutlich wird in diesem Abschnitt erneut der Bezug zum Prinzip der Delegation mit begrenztem Wissen, das im Abschnitt 1.1 eingeführt wurde. Wie zu Beginn dieses Abschnitts diskutiert, stellten auch die Autoren Hacigumus et al. und Hore et. al. Untersuchungen bzgl. der Verteilung von Arbeitslast zwischen Client (Cloud-Nutzer) und Server (Cloud-Anbieter) in Abhängigkeit von der Bereitwilligkeit der Informationspreisgabe, an. Sie kommen dabei zur gleichen Aussage, die auch durch das Prinzip der Delegation mit begrenztem Wissen formuliert wird.

¹⁴Diese sind als Vertraulichkeit (CO), Integrität (IG), Verfügbarkeit (AV), Unverknüpfbarkeit (UL), Transparenz (TR), Interventionsfähigkeit (IN) in der Tabelle 4.1 aufgeführt.

4.4. Dateisysteme

In diesem Abschnitt werden Lösungen für die Speicherung von Daten, vorzugsweise Dateien, in Cloud-Applikationen vorgestellt und diskutiert. Der Fokus der beschriebenen Ansätze liegt demnach auf dem Schutz des Daten-Layers, der sich innerhalb der eingeführten Taxonomie aus Kapitel 2.3 befindet.

Dabei orientiert sich der Aufbau dieses Abschnitts an der zeitlichen Entwicklung der Forschungsansätze, indem angefangen von verteilten, sicheren Dateisystemen in klassischen Client-Server-Szenarien, im weiteren Verlauf des moderne Ansätze für dynamische Cloud-Umgebungen vorgestellt werden. Der häufige Bezug der modernen Lösungen auf die klassischen Ansätze, ist der Grund der anfänglichen Erörterung.

Als klassische Ansätze werden im Folgenden SiRiUS von Goh et al. [GSMB03], Plutus von Kallahalla et al. [KRS⁺03] und SUNDR von Li et al. [LKMS04] bezeichnet. Ursache für die Charakterisierung ist, dass viele der modernen sicheren Cloud-Speicherlösungen auf den Erfahrungen dieser Lösungsansätze aufbauen und ihre Ansätze an diesen messen. SiRiUS ist ein sicheres Dateisystem, das entwickelt wurde, um unsichere Netzwerke und P2P File Systeme wie NFS¹⁵, CIFS¹⁶ und OceanStore¹⁷ zu schützen. Die Lösung nimmt an, dass der Netzwerkspeicher nicht vertrauenswürdig ist und bietet eigene kryptographische Zugangskontrollen für das Teilen von Daten. Das Schlüsselmanagement und Rechteentzug ist einfach und mit minimaler Kommunikation umgesetzt. Die Konsistenz des Dateisystems wird von SiRiUS durch Hash-Tree Konstruktionen unterstützt. Die von Kallahalla et al. [KRS⁺03] entwickelte Lösung Plutus ist ein kryptographisches Speicher-System welches erlaubt, sicher Daten zu speichern und zu teilen, ohne dabei Vertrauen in den File Server zu setzen. Plutus bietet ein hoch skalierbares Schlüsselmanagement, das individuellen Nutzern die direkte Kontrolle darüber erlaubt, wer Zugriff auf seine Daten hat. Die von Kallahalla et al. durchgeführten Messungen der Leistungsfähigkeit bestätigen, dass Plutus eine hohe Sicherheit bieten kann und dabei ein ähnlicher Mehraufwand wie beim Verschlüsseln des gesamten Netzwerkverkehrs entsteht. SUNDR ist ein Netzwerk-Dateisystem, das von Li et al. [LKMS04] für die integritätsgeschützte Speicherung auf nicht vertrauenswürdigen Servern entwickelt wurde. Die Lösung erlaubt dem Client alle Versuche nicht autorisierter Datenveränderungen von böswilligen Serverbetreibern oder Nutzern aufzuzeigen. Das Protokoll von SUNDR hat eine Eigenschaft namens *Fork Consistency*, welche Clients garantiert, dass alle Integritäts- oder Konsistenzfehler erkannt werden, so lange die Dateiänderungen offen liegen. Die beschriebene Implementierung ist vergleichbar mit NFS, jedoch wird nach Aussage der Autoren eine signifikant höhere Sicherheit geboten.

Die nachfolgende Untersuchung der Ansätze für sichere Speicherlösungen mit Fokus auf Cloud-Umgebungen zeigt das große Interesse der Forschergemeinde, praktikable Ansätze zu liefern. So wird der Ansatz Cryptree von Grolimund et al. [GMSW06] von einem CSP in der Praxis eingesetzt. Die Lösung ist eine kryptographische Datenstruktur bestehen aus Schlüsseln und kryptographischen Links. Der Cryptree kann als gerichteter Graph mit Schlüsseln als Knoten und kryptographischen Links als Kanten angesehen werden. Zudem haben, wie der Name nahelegt, Cryptrees häufig eine Baumstruktur. Durch diese Baumstruktur kann Cryptree benutzt werden, um effektiv Schlüssel von verschachtelten Ordnern in kryptographischen Dateisystemen zu verwalten. Es genügt, einen Schlüssel zu kennen, um rekursiv alle abgeleiteten Schlüssel (der Unterordner) zu beziehen. Dies ist eine mächtige Eigenschaft bei der Gestaltung von Zugangskontrollen eines kryptographischen Dateisystems. Anwendung findet dieser Ansatz beim Wuala Cloud-Speicher¹⁸. Die Lösung

¹⁵Network File System, für Details siehe: <https://tools.ietf.org/html/rfc3530>, letzter Zugriff: 15.08.2014

¹⁶Common Internet File System: <https://technet.microsoft.com/en-us/library/cc939973.aspx>, letzter Zugriff: 15.08.2014

¹⁷CFür Details siehe: <https://oceanstore.cs.berkeley.edu/info/overview.html>, letzter Zugriff: 15.08.2014

¹⁸Weiter Informationen unter <https://www.wuala.com/de/>, letzter Zugriff am 17.06.2015

CS2 von Kamara et al. [KBR11] ist ein kryptographisches Cloud-Speicher-System, das Vertraulichkeit, Integrität und Verifizierbarkeit bietet ohne die Nutzer zu beeinträchtigen. CS2 bietet Sicherheit gegenüber dem CSP, wobei die Nutzer weiterhin in der Lage effizient sind auf ihre Daten zuzugreifen, via Suchschnittstelle zu suchen und Daten sicher hinzuzufügen und zu löschen. Möglich wird dies durch ein symmetrisches Verfahren der durchsuchbaren Verschlüsselung, wie es im Abschnitt 4.2.4 vorgestellt wird.

CloudProof, die Lösung von Popa et al. [PLM⁺11] bietet als sicheres Speichersystem die Möglichkeit Integritätsverstöße und Konsistenzprobleme aufzudecken. Zudem bietet das System Beweise, dass diese Verstöße von einer dritten Partei (z.B. dem CSP) begangen wurden. Laut Meinung der Autoren ist dieses beweisbasierende System entscheidend, um einen Anspruch auf einen Ausgleich beim Fehlverhalten innerhalb eines Cloud-Services geltend machen können. Evaluierungen von Popa et al. zeigen, dass Sicherheitsmechanismen angemessene Kosten haben. Die Latenz wird bei Lese- und Schreibzugriffen um rund 15% verringert, der Durchsatz um etwa 10%. Eine Erweiterung dieses Ansatzes stellen Albeshri et al. [ABGN12] vor.

In der Lösung von Somorovsky et al. [SMT⁺12] wurde besonders auf die Integration von mobilen Endgeräten und Smartcards Rücksicht genommen. Die Umsetzung erfolgte auf der Basis von *XML Signatures* und *XML Verschlüsselung*, kombiniert mit *SAML*. Zudem konzipierten die Autoren ein mehrstufiges Schlüsselkonzept mit Nutzer-, Daten- und Gruppenschlüssel, um das Schlüsselmanagement zu vereinfachen.

Die Lösung von Zhou et al. [ZVH12] nutzt dagegen eine rollenbasierte Zugriffskontrolle. Die verwendete rollenbasierte Verschlüsselung erzwingt RBAC-Vorschriften als erweiterte Form der in Kapitel 4.2.4 vorgestellten attributbasierenden Verschlüsselung. Das Schema kann Daten für eine Rolle im RBAC-System verschlüsseln, so dass nur die Nutzer welche die Systemrolle besitzen, die Daten entschlüsseln können. Xiong et al. [XZY⁺12] stellen mit ihrem Ansatz, CloudSeal, eine End-to-End Verschlüsselungslösung zur Speicherung und Verteilung von großen Dateien (vorrangig Videos) vor. Die Lösung verwendet eine Kombination aus symmetrischer Verschlüsselung (AES) und Proxy-Reencryption, um die Vertraulichkeit der Daten zu gewährleisten, wobei die Autoren den Mehraufwand als akzeptabel bewerten. Gleichzeitig stellen diese fest, dass trotz der Nutzung eines Content Delivery Networks¹⁹ der entstehenden Mehraufwand, durch die kryptographischen Operationen gegenüber den Übertragungszeiten, der großen Dateien, kaum ins Gewicht fallen. Konkret heißt es dazu: *The cryptographic time is at least 40 times faster than the content delivery time.*

Bessani et al. [BCQ⁺13] bieten mit DEPSKY, welches Bestandteil des TCloud-Forschungsprojekts ist, eine sichere Möglichkeit, Dateien redundant und sicher bei verschiedenen CSP zu speichern. Trotz der Nutzung von 4 unabhängigen Cloud-Anbietern verdoppeln sich die Kosten lediglich.

Jäger et al. [JMR⁺14] bieten einen anderen Ansatz, der Schutz gegen Insider Angriffe bietet, indem der Daten-Lebenszyklus durch eine spezielle Ausführungsumgebung in Kombination mit Hardware geschützt wird. Das heißt, die Verarbeitung von sensiblen Daten kann nur erfolgen, wenn alle software- und hardwareseitigen Voraussetzungen erfüllt sind. Das Prinzip kann als Verallgemeinerung von sicherer Hardware²⁰ angesehen werden, welche speziell geschützt ist und im Falle eines erkannten Angriffs alle sensiblen Daten innerhalb der Ausführungsumgebung löscht. Ob die gebotene Sicherheit des Lösungsansatzes von Jäger et al. mit der von Hardwaremodulen gleich einzuschätzen ist, ist jedoch fraglich und kann in dieser Arbeit nicht abschließend bewertet werden. Die hohe Komplexität dieser Ansätze widerspricht jedoch nach Schneier [Sch11b] grundsätzlich einer hohen Sicherheit. Ferner erzeugt der Ansatz einen starken Vendor Lock-In, da auf exklusive proprietäre Ausrüstung im Rechenzentrum zurückgegriffen werden muss. Weitere Arbeiten, die im Rahmen der Evaluierung untersucht wurden, an dieser Stelle jedoch nicht ausführlicher beschrie-

¹⁹Das Ziel ist eine bessere Cloud-Anbindung mit kürzeren Latenzzeiten.

²⁰Vergleiche dazu Kapitel 5.

ben werden, sind: Hourglass Schemes von Dijk et al. [DJO⁺12] CloudFilter von Papagiannis et al. [PP12], HAIL von Bowers et al. [BJO09], Venus von Shrear et al. [SCC⁺10] sowie die Arbeiten von [WJ10, WLL12].

Zusammenfassung

Im Abschnitt werden sichere Datenspeicherlösungen in Cloud- Umgebungen erörtert. Die Tabelle 4.2 bietet eine Zusammenfassung der Informationen aus den Evaluierungsergebnissen im Teil III der Arbeit. Es werden die Bewertungen nach den in Abschnitt 2.4.1 diskutierten Schutzziele²¹ dargestellt. Eine ausführliche Erläuterung der Kriterien enthält der Abschnitt 7.1. Im Rahmen der untersuchten Lösungen ist DEPSKY von Bessani et al. [BCQ⁺13] hervorzuheben. Diese bietet neben dem Schutz der Vertraulichkeit, die nahezu alle Lösungen ermöglichen, zusätzliche Sicherheit durch die redundante Speicherung bei verschiedenen, nicht kooperierenden, Cloud Anbietern. Dabei werden die Daten aufgeteilt, um die Kosten nicht zu stark steigen zu lassen. Damit unterstreicht diese Lösung nach Meinung des Autors die Idee des Cloud Computings, durch das Nutzen verschiedener CSP über standardisierte Schnittstellen die Sicherheit bei akzeptablen Entwicklungsaufwand zu erhöhen. Die Standards ermöglichen eine leichte Integration weiterer CSP und verringern somit den Vendor Lock-in-Effekt. Negativ fallen die erhöhten Kosten auf, die jedoch bei der redundanten Speicherung unvermeidbar sind.

Eine weitere wissenschaftlich interessante Arbeit wird von Bowers et al. [BDJ⁺11] vorgestellt und geht auf die Möglichkeit ein, zu Prüfen, ob die Daten beim CSP tatsächlich redundant gespeichert werden. Dabei findet laut Aussage der Autoren eine außergewöhnliche Kombination von *coding theory, cryptography, and hardware profiling* statt. Der Lösungsansatz hat nach Autorenaussage nur geringe Anforderungen und kann parallel zu anderen Schutzmaßnahmen angewandt werden.

Tabelle 4.2.: Die Zusammenfassung untersuchter sicherer Datenspeichersysteme mit deren Bewertung der ausgewählten Evaluierungskriterien.

Bezeichnung	CO	IG	AV	UL	TR	IN	Anmerkungen
SiRiUS [GSMB03]	✓	✓	✗	✗	✗	✓	klassischer Ansatz
Plutus [KRS ⁺ 03]	✓	✓	✗	✗	✓	✓	klassischer Ansatz
SUNDR [LKMS04]	✗	✓	✓	✗	✗	✓	klassischer Ansatz
Wang et al. [WLL12]	✗	✓	✓	✗	✓	✗	Cloud Auditierung
Bowers et al. [BDJ ⁺ 11]	✓	✗	✓	✗	✓	✗	Redundanzüberprüfung
Cryptree [GMSW06]	✓	✗	✗	✗	✓	✓	durch CSP produktiv eingesetzt
Sec2 [SMT ⁺ 12]	✓	✓	✗	✗	✓	✓	mobile Endgeräte
Kamara et al. [KL10]	✓	✓	✓	✗	✓	✓	Konzept
CloudProof [PLM ⁺ 11]	✓	✓	✗	✗	✓	✓	keine
HAIL [BJO09]	✗	✓	✓	✗	✗	✓	Integritäts- und Verfügbarkeitsfokus
CS2 [KBR11]	✓	✓	✗	✗	✗	✓	keine
Zhou et al. [ZVH12]	✓	✗	✗	✗	✓	✓	keine
Venus [SCC ⁺ 10]	✗	✓	✗	✗	✓	✗	mind. 2 Clients online
CloudSeal [XZY ⁺ 12]	✓	✗	✓	✗	✗	✓	große Dateien im Fokus
DEPSKY [BCQ ⁺ 13]	✓	✓	✓	✗	✓	✓	Bestandteil TClouds Projekt
Sealed Cloud [JMR ⁺ 14]	✓	✗	✓	✗	✓	✗	Trusted Clouds Teilprojekt
Hourglass [DJO ⁺ 12]	✓	✗	✗	✗	✓	✓	keine
Cloud Filter [PP12]	✓	✗	✗	✗	✓	✓	keine
SPORC [FZFF10]	✓	✓	✗	✗	✓	✓	keine

²¹Dies sind als Vertraulichkeit (CO), Integrität (IG), Verfügbarkeit (AV), Unverknüpfbarkeit (UL), Transparenz (TR), Interventionierbarkeit (IN) in der Tabelle 4.1 aufgeführt.

4.5. Anonymisierung

Anonymisierung und Pseudonymisierung sind Maßnahmen zur Gewährung des Datenschutzes, in dem sensible oder personenbezogene Informationen (sog. Identifizierer) gelöscht oder ersetzt werden. Streng genommen unterscheiden sich Anonymisierung und Pseudonymisierung darin, dass bei der Anonymisierung schützenswerte Informationen unwiderruflich gelöscht oder ersetzt werden und eine spätere Zuordnung unmöglich ist. Bei der Pseudonymisierung wird die Information in der Art ersetzt, dass eine spätere Zuordnung weiterhin möglich bleibt, wobei die schützenswerte Information an anderer Stelle gesichert werden muss. Die Unterscheidung wird anhand von Abbildung 4.13 in Form einer anonymisierten 4.13(a) und einer pseudonymisierten Nachricht 4.13(a) verdeutlicht.

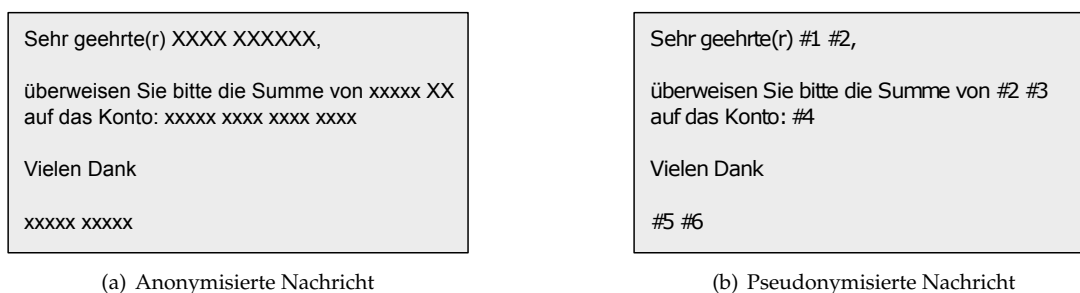


Abbildung 4.13.: Die Unterscheidung von Anonymisierung und Pseudonymisierung.

Dabei entscheidet im Fall der Pseudonymisierung der Ort der Speicherung, ob dem Datenschutzrecht entsprochen wird. Im §30 BDSG heißt es dazu: *Werden personenbezogene Daten geschäftsmäßig erhoben und gespeichert, [...], sind die Merkmale gesondert zu speichern mit denen Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können.* Da der Anwendungsfall der strengeren Definition von Anonymisierung stark begrenzt ist, wird folgend, bei der Verwendung des Begriffs Anonymisierung, der Fall Zwei der Pseudonymisierung gemeint.

Die Verschlüsselung bietet zwar Datenschutz, jedoch ist die Verarbeitungseffizienz, wie in Kapitel 4.2 erläutert, häufig vergleichsweise gering. Verfahren zur Anonymisierung verstecken laut Zhang et al.[ZLS11] die Verbindung von sensiblen Datenkombination um den Datenschutz zu gewährleisten, obwohl die Daten im Klartext vorliegen. Die gewährleistete Sicherheit wird demnach für die verbesserte Verarbeitungseffizienz verringert. Zu beachten ist jedoch die Gefahr einer möglichen Reidentifizierung von bestehenden Merkmalskombinationen durch externe, öffentlich verfügbare Informationen. Diese Problemstellung wird im Folgenden unter dem Begriff k-Anonymität erörtert.

k-Anonymität

In einem ersten Ansatz zur Anonymisierung von Datensätzen geht es darum, diese nach zuvor definierten Attributen zu durchsuchen, um alle Felder unter diesen Attributen aus den Daten zu entfernen oder deren Inhalt zu ersetzen. Das Ziel ist dabei, einen Datensatz zu erzeugen, der frei von identifizierenden Attributen ist. Dieser Prozess wird als *Deidentifikation* eines Datensatzes bezeichnet. Die Deidentifikation eines Datenbestandes bedeutet jedoch nicht gleichzeitig, dass dieser Datenbestand auch anonym ist. Unter Zuhilfenahme externer Daten ist es möglich, in einem deidentifizierten Datensatz individuelle Personen dennoch zu identifizieren. Beispiele sind Daten durch staatliche Behörden (Meldeämter, Steuerbehörden, KFZ-Zulassungsstellen) oder andere öffentlich verfügbare

Daten wie Telefonbücher oder Vereinslisten. Wie Sweeney [Swe02] belegt, ist es mit Hilfe weniger Identifikationsmerkmale, wie beispielsweise Postleitzahl, Geburtsdatum und Geschlecht, möglich, 63% bis 87% der Bevölkerung zu identifizieren. Unter dieser Voraussetzung können z.B. Melderegisterdaten verwendet werden, um Personen in deidentifizierten und veröffentlichten Krankenhausdaten zu identifizieren. Dies ist in der Abbildung 4.14 übersichtlich dargestellt.

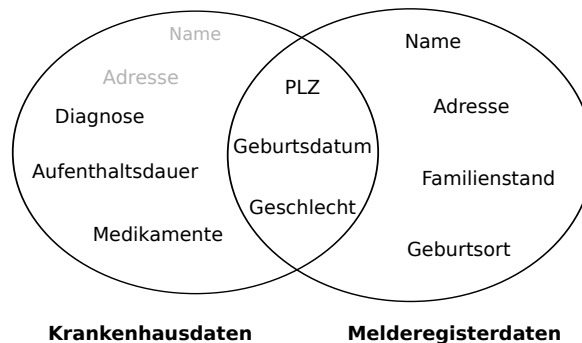


Abbildung 4.14.: Das Prinzip der Reidentifizierung von anonymisierten Datensätzen nach Sweeney[Swe02]. Durch das Zusammenführen können die anonymisierten Krankenhausdaten mit Hilfe von Melderegisterdaten anhand der Kombination dreier gemeinsamer Merkmale wieder konkreten Personen zugeordnet werden.

Stimmen die drei Attribute Postleitzahl, Geburtsdatum und Geschlecht in je einem Datensatz aus beiden Datenbeständen überein, kann der Eintrag aus den Krankenhausdaten erfolgreich dem aus den Melderegisterdaten zugeordnet werden. Da in den Melderegisterdaten die identifizierenden Attribute Name und Adresse vorhanden sind, ist somit auch der Eintrag in den Krankenhausdaten identifizierbar geworden. Dieses Vorgehen wird als Reidentifizierung bezeichnet.

Im Folgenden wird ein privater Datenbestand betrachtet, der mit Hilfe eines k-Anonymitätsverfahren anonymisiert und daraufhin veröffentlicht werden soll. Die Attribute des Datensatzes können in drei disjunkte Teilmengen eingeteilt werden. Dabei stellen die explizit identifizierenden Attribute die erste Teilmenge dar. Sie werden, im Sinne der Deidentifikation, vor der eigentlichen Ausführung des k-Anonymitätsverfahren festgelegt und entfernt. Die zweite Teilmenge bilden die Attribute, welche die zu veröffentlichende Information beinhalten. Diese dürfen durch das k-Anonymitätsverfahren nicht verändert werden, da sie der Grund für die Datenherausgabe sind. Die verbliebenen Attribute stellen die dritte und letzte Teilmenge dar. Diese können für eine Wiederidentifizierung individueller Personen verwendet werden. Deshalb gilt es, diese durch das k-Anonymitätsverfahren zu schützen. Nach LeFevre [LeF07] und El Emam et al. [EEJS⁺06] unterteilt sich die Teilmenge wie folgt:

1. Identifizierer: Die Menge der explizit identifizierenden Attribute, wie z. B. Name, Adresse, Sozialversicherungsnummer.
2. Quasi-Identifizierer: Die Menge der Attribute, die vor der Veröffentlichung anonymisiert werden müssen. Beispiele sind Postleitzahl, Geburtsdatum, Geburtsort, Geschlecht, Familienstand, Religion und Staatsangehörigkeit.
3. Sensitive Attribute: Die Menge der Attribute, die bei der Veröffentlichung nicht anonymisiert werden dürfen, jedoch sensitive Informationen über individuelle Personen enthalten, so dass sie bei der Veröffentlichung einzelnen Personen nicht mehr zuzuordnen sein dürfen. Beispiele sind die ärztliche Diagnose einer Krankheit oder die verabreichten Medikamente.

Das Vorgehen eines k-Anonymitätsverfahrens sei durch die folgenden Tabelle 4.3 veranschaulicht. Der Ablauf des Verfahrens ist von oben nach unten nachvollziehbar. Oben befindet sich der vollständige, in der Mitte der deidentifizierte und unten der k-anonyme Datensatz mit $k = 2$.

Tabelle 4.3.: k-Anonymisierung eines Datensatzes: oben der vollständige Datensatz, mittig der deidentifizierte und unter der k-anonyme Datensatz ($k=2$). Die Informationen im Datensatz sind frei gewählt und haben keinen Bezug zu real existierenden Personen.

Name	Adresse	PLZ	Geb.Datum	Geschlecht	Diagnose
E.Weber	Bachstr. 15	16404	30.04.1982	feminin	Krebs
R.Meyer	Dorfstr. 11	16405	27.10.1982	feminin	HIV
A.Mueller	Hauptstr. 1	16406	01.07.1982	maskulin	HIV
D.Schmid	Waldstr. 6	16422	16.03.1982	maskulin	Krebs
G.Lange	Hauptstr. 5	16423	12.11.1984	feminin	Rheuma
A.Wagner	Hauptstr. 4	16404	06.01.1984	feminin	Krebs
		PLZ	Geb.Datum	Geschlecht	Diagnose
		16404	30.04.1982	feminin	Krebs
		16405	27.10.1982	feminin	HIV
		16406	01.07.1982	maskulin	HIV
		16422	16.03.1982	maskulin	Krebs
		16423	12.11.1984	feminin	Rheuma
		16404	06.01.1984	feminin	Krebs
		PLZ	Geb.Datum	Geschlecht	Diagnose
		164xx	xx.xx.1982	feminin	Krebs
		164xx	xx.xx.1982	feminin	HIV
		164xx	xx.xx.1982	maskulin	HIV
		164xx	xx.xx.1982	maskulin	Krebs
		164xx	xx.xx.1984	feminin	Rheuma
		164xx	xx.xx.1984	feminin	Krebs

Die k-Anonymität eines Datenbestandes wird bei der Ausführung des k-Anonymitätsverfahren durch zwei Strategien erzeugt. Diese beiden Techniken sind die Generalisierung und die Unterdrückung. Für detaillierte Informationen wird auf die folgenden Arbeiten verwiesen: [SS98, ABGP05, Del06, AY07, CdVFS07, SK09, Seb10, Zie11].

Das Ziel des Abschnitts ist es, zum einen zu zeigen, dass insbesondere bei der Veröffentlichung von anonymisierten Daten eine Reidentifikation einzelner Datensätze leicht möglich sein kann. Verfahren zur k-Anonymität können unterstützen, da jeder Datensatz mindestens k mal vorgehalten und dabei die maximale Anzahl an Informationen bereitstellt wird. Andererseits zeigt die k-Anonymität, dass eine Klassifizierung der Daten Voraussetzung ist. Insbesondere das automatisierte Klassifizieren von Daten ist ein offenes und stark erforschtes Gebiet der modernen Informationstechnologie, in dem Begriffe wie IT-Forensik und Big Data häufig aufgeführt werden. In Cloud-Umfeld finden zudem Datenkategorisierung statt, um verschiedene Sicherheitslevel zu etablieren. Dies ist insbesondere bei hybriden und Multi-Cloud-Umgebungen der Fall.

Die Ansätze von Bugiel et al. [BSSS11], Papagiannis et al. [PP12] und Zhang et al. [ZZCW11] gehen von vertrauenswürdigen Umgebungen aus, in denen sicherheitskritische Berechnungen stattfinden. Alle anderen Verarbeitungen werden zu einem nicht vertrauenswürdigen Cloud-Anbieter ausgelagert. Das hybride Modell von Zhang et al. [ZZCW11] nutzt eine Private Cloud, um sicherheitskritischen Berechnungen auszuführen. Dies bietet den Vorteil keinen Cloud-Provider vertrauen zu müssen, jedoch den Nachteil der Verwaltung der Private Cloud. Für die Klassifizierung, die festlegt

welche Berechnungen in der vertrauenswürdigen Cloud ausgeführt werden müssen, beschreiben die Autoren Ansätze, diesen Vorgang zu automatisieren. Auch der Cloud-Filter-Ansatz von Papa-
giannis et al. [PP12] beschreibt einen Klassifizierungsansatz, jedoch auf der Basis von Data Loss
Prevention.

Neben der Anonymisierung von Daten existieren zudem Ansätze einer anonymen Identifizierung.
Dieser scheinbare Widerspruch wird bereits in Kapitel 3.1 angedeutet. Neben den dort genannten
Lösungen bieten Khan et al. [KH12] mit Hilfe eines anonymen Credential-Systems und basierend
auf dem Tor-Netzwerk die Möglichkeit einer anonymen Berechnung in der Cloud. Eine hervorra-
genden Einblick in diese Thematik der Anonymisierung bietet die Thesis von Hussain [Hus10].

4.6. Datenzugriff

Die Verschlüsselung von Daten bietet nicht immer die gewünschte oder benötigte Sicherheit der Daten. Zugriffsmuster können ebenfalls eine kritische Menge an Informationen preisgeben. Folgt auf einer Abfolge von Datenzugriffen immer die gleiche Aktion, wie beispielsweise ein Bezahlvorgang, eine Autorisierung oder eine andere sicherheitskritische Operation kann der Server an sensible Informationen gelangen. Und das, wie Pinkas und Reinman [PR10] hinweisen, trotz der sicheren Verschlüsselung von Daten.

Oblivious RAM (ORAM) ist ein von Goldreich und Ostrovsky [Gol87, Ost90, GO96], eingeführtes kryptographisches Primitiv zum Verstecken von Speicher-Zugriffsmustern. In seiner ursprünglichen Bedeutung sollte es Software vor Reverse Engineering schützen, indem es den Speicherzugriff (RAM) verschleiern. Wie Berkeley et al. [SSS12] aufzeigen, ist mit dem Trend zum Cloud Computing, bei der Auslagerung von Datenspeichern ORAM erneut Bedeutung.

Der klassische ORAM-Ansatz von Goldreich [GO96], hat eine amortisierte Kommunikations- und Berechnungskomplexität von $O(\log^3 n)$ bei einer Datenbankgröße von n . Die vom Server gespeicherte Datenbank ist eine Menge von n semantisch-sicher, symmetrisch- verschlüsselten Blöcken. Unterstützte Operationen sind $\text{read}(\text{id})$ und $\text{write}(\text{id}, \text{newvalue})$. Die Daten sind in $\log_2(n)$ Level einer Pyramide organisiert. Dabei besteht Level i aus bis zu 2^i Buckets, jeder der Blöcke ist einem der 2^i Buckets auf diesem Level durch eine Hash Funktion zugeordnet. Durch Hash-Kollisionen kann jeder Bucket 0 bis $O(\log n)$ Blöcke enthalten. Die Abbildung 4.15 verdeutlicht die Speicherstruktur.

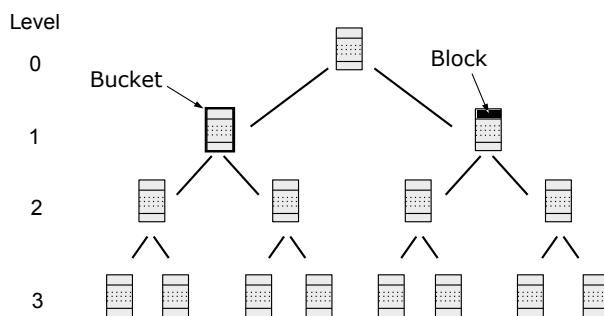


Abbildung 4.15.: Die Datenorganisation in ORAM. Dabei besteht Level i aus bis zu 2^i Buckets, jeder Block ist einem der 2^i Buckets auf diesem Level durch eine Hash Funktion zugeordnet. Durch Hash-Kollisionen kann jeder Bucket 0 bis $O(\log n)$ Blöcke enthalten.

Der moderne Ansatz hat das klassische hierarchische Schema mit Verfahren wie Kuckucks-Hashing [PR04] und Bloom Filter [Blo70] optimiert. Als in der Praxis problematisch erweist sich beim klassischen Ansatz vor allem die hohe Anzahl an Client-Server-Roundtrips, die sich in einer hohen, praktisch unvereinbaren, online Latenzzeit äußern. Dieser Herausforderung nehmen sich Willimans et al. [WS12] an, indem die Autoren einen Single-Round Ansatz vorstellen. Dieser ist der erste Single-Roundtrip ORAM mit poly-logarithmischem Zeitbedarf und benötigten Client Speicher von logarithmischer Größe. Durch geeignete Vorverarbeitung des Clients kann der Server den interaktiven Prozess des Traversierens des ORAM-Speicherbaums bearbeiten, ohne zusätzliche Informationen zu erlernen. Möglich machen dies die Autoren durch Bloomfilter und einer geeigneten Entschlüsselungsmethode auf Serverseite. Die Voraussetzungen sind ausreichender flüchtiger Client Speicher (logarithmisch abhängig von der ausgelagerte Datenmenge) und persistenter Speicher für das Schlüsselmanagement. Daraus resultieren die online Kommunikations- und Berechnungskosten von $O(\log n)$ und die Offline pro Anfrage Kosten von $O(\log^2 n \log \log n)$. Dies ergibt bei einer Daten-

bankgröße von 1TB und 150 ms Kosten für den Roundtrip, pro Query ca. 4 Sekunden Reaktionszeit [WS12].

Der Ansatz Shroud bietet einen cloud basierten ORAM-Ansatz für große Datenmengen. Dazu führen Lorch et al. [LPM⁺13] neue Techniken wie *oblivious aggregation*, eine Möglichkeit der sicheren²² Datenzusammenfassung mit Hilfe von sicherer Hardware, an. Diese arbeiten als Client Proxy Server beim Cloud Provider. Durch diese Maßnahme bleiben die kritischen Roundtrips CSP-intern. Dies stellt einen kosteneffizienten Lösungsansatz der Latenzproblematik dar. Ferner zielt der Ansatz darauf ab, die Latenz durch massive Parallelisierung zu verbessern. Die Autoren untersuchen dabei auf Einfluss von Block-Größe, Anzahl von sicheren Coprozessoren und Parallelisierung von Bucket-Zugriffen. Das Ergebnis ist die signifikante Kostenreduzierung, um den Faktor 1444 [LPM⁺13]. Jedoch sind die Zugriffszeiten immer noch wesentlich höher als für den ungesicherten Zugriff. Auch die Lösung *Oblivstore* von Stefanov et al. [SS13] stellt einen weiteren schritt der praktischen Verwendbarkeit von ORAM-Speichersystemen dar.

Trotz aller Forschungsbestrebungen gelten ORAM-Lösungen bis heute als zu ineffizient für den Einsatz in praktischen Systemen. Ursache dafür ist die erwähnte Notwendigkeit stets alle Buckets zu durchlaufen, selbst wenn die Daten bereits im ersten Bucket enthalten waren. Dies ist alternativlos, da anderenfalls Informationen an den Server preisgegeben würden. Hinzu kommen die zusätzlichen Aufgaben des Nutzers durch Vorverarbeitung, Schlüsselmanagement und benötigten lokalen Speicher.

Das Prinzip der Delegation mit begrenztem Wissen, erfährt in diesem Abschnitt eine Bestätigung in extremer Form. Es wird deutlich dargelegt, welche Auswirkungen es hat, die preisgegeben Informationen auf ein Minimum zu reduzieren. Die Verarbeitung seitens des Cloud-Anbieters wird ineffizient und die zusätzlichen Aufgaben, zum Erhalt der Funktionalität steigen auf der Seite des Cloud Nutzers stark an.

²²Sicher im Sinne, dass keine Informationen, auch keine Metainformationen, über die Daten an den Server gelangen.

4.7. Zusammenfassung

Das Kapitel diskutiert die in der jüngsten Forschung dargestellten Möglichkeiten für den Cloud-Nutzer eines IaaS (selten auch PaaS), um die Sicherheit einer Cloud-Applikation in Bezug auf Datenschutz, Datensicherheit oder beidem zu erhöhen. Der Abschnitt unterteilt sich dabei gemäß der in Abschnitt 2.3 eingeführten Taxonomie.

Die beschriebenen Replikations- und Verteilungsszenarien innerhalb des Abschnitts 4.1 zeigen allgemeine Möglichkeiten, Cloud-Applikationen über verschiedene Cloud-Anbieter zu verteilen. Die Ansätze können dabei kombiniert werden und sind auch für interne oder hybride Szenarien von grundsätzlichem Interesse.

Der Abschnitt 4.2 stellt Maßnahmen vor, die getroffen werden können um Applikationen oder Berechnungen sicher auszuführen. Insbesondere wird deutlich, dass nach derzeitigem Forschungsstand keine allgemeinen und effizienten Ansätze existieren, Applikationslogik vollständig zu schützen. Dazu wird auf verschiedene bekannte Forschungsbereiche eingegangen und sowohl Grundlagen zu Technologien als auch deren Grenzen erläutert. Die in der Forschung und Populärwissenschaft breit diskutierte voll homomorphe Verschlüsselung wird dabei in aller Ausführlichkeit erörtert, denn das Ziel der Arbeit ist es die Funktionsweise dieses Verfahren soweit zu verdeutlichen, dass trotz der herausragenden Forschungsergebnisse, die Probleme des Verfahrens deutlich werden. Der in vielen wissenschaftlichen Publikationen vertretene Standpunkt, das Problem des sicheren Cloud Computings sei mit effizienten FHE Verfahren gelöst, wurde eindeutig widerlegt.

Die Abschnitte zu Datenbanken und Dateisystemen beschreiben Lösungen zur Datenspeicherung in verteilten Cloud Umgebungen. Die Ergebnisse der Untersuchungen bestätigen, dass in der jüngeren Forschung zahlreiche effiziente Lösungen existieren. Die entsprechenden Abschnitte fassen daher die untersuchten Ansätze tabellarisch zusammen. Problemstellungen, wie die Reidentifizierung von Datensätzen, werden im Zusammenhang mit der Anonymisierung von Daten im Abschnitt 4.5 ausführlich diskutiert, indem das Verfahren der k -Anonymisierung eingeführt und an Beispielen erörtert wird. Der letzte Abschnitt untersucht Methoden, zum Schutz Informationen, die beim Datenzugriff preisgegeben werden. Leider existieren nach gegenwärtigem Forschungsstand keine Verfahren, die in praktischen Umgebungen eingesetzt werden könnten.

Das Kapitel verdeutlicht, dass der Nutzer zahlreiche Möglichkeiten hat, die Sicherheit innerhalb der Cloud Applikation zu erhöhen. Bis auf Datenspeicherlösungen haben die untersuchten Ansätze jedoch selten praktische Reife erreicht oder sind bereits potenziell für eine produktiv einsetzbare Implementierung geeignet. Eine ausführliche Einschätzung enthält der Evaluierungsteil III dieser Arbeit. Zahlreiche untersuchte Ansätze eignen sich zudem für die Erhöhung der Sicherheit innerhalb der Cloud-Umgebung. Insbesondere sichere Dateisysteme und Datenbanklösungen können auch von CSP effektiv umgesetzt werden.

Dieses Kapitel unterstützt somit die Annahme der These 1. Klassische Sicherheitsmaßnahmen, wie besonders bei den Datenspeichersystemen deutlich wurde, werden in aktuellen Forschungssätzen für Cloud-Umgebungen erweitert. Auch der Abschnitt der sicheren Applikationslogik macht deutlich, dass neue Sicherheitsmaßnahmen, wie funktionale Verschlüsselungen, zumeist erst durch Cloud-Umgebungen ihr volles Potenzial ausschöpfen.

Darüber hinaus wird auch das Prinzip der Delegation mit begrenztem Wissen im Laufe des Kapitels regelmäßig bestätigt, wie die Zusammenfassungen der Abschnitte Datenbanken, Dateisystem und Datenzugriff verdeutlichen.

Trusted Cloud Computing

Im Kapitel Trusted Cloud Computing werden Cloud-Dienste beschrieben, die durch Unterstützung von Hardware Modulen zusätzliche Sicherheit bieten. Wie Santos et al. [SRGS12] darlegen, besteht ein Schlüsselfaktor im Vertrauen des Nutzers darin, hohe Garantien über die Integrität der Cloud-Infrastruktur zu bieten. Diese durch formale und kryptographische Methoden zu beweisen und dabei Hardware zu nutzen¹, ist die Grundlage des Trusted Cloud Computing. Abbadi [Abb12] erörtert in seiner Arbeit zwei grundlegende Problematiken des Trusted Computings in Cloud-Umgebungen. Zum einen das Thema Transparenz vs. Vertrauensevaluierung und zum anderen die Komposition von Vertrauensankern, so genannten Roots-of-Trust, in Hardware. Beide Probleme resultieren aus der Tatsache das Cloud-Umgebungen grundsätzlich abstrahieren. Der Grundgedanke des Trusted Cloud Computing ist: Cloud-Ressourcen als Service mit definierten Schnittstellen zu nutzen ohne deren exakte innere Funktionsweise zu kennen oder kennenlernen zu wollen (Black Box Prinzip). In diesen Punkt widerspricht Vertrauen in Hardware den Transparenzeffekten, wie auch Abbadi [Abb12] darlegt. Die Abbildung 5.1 illustriert die Unterteilung dieses Themengebiets. Dieses Kapi-

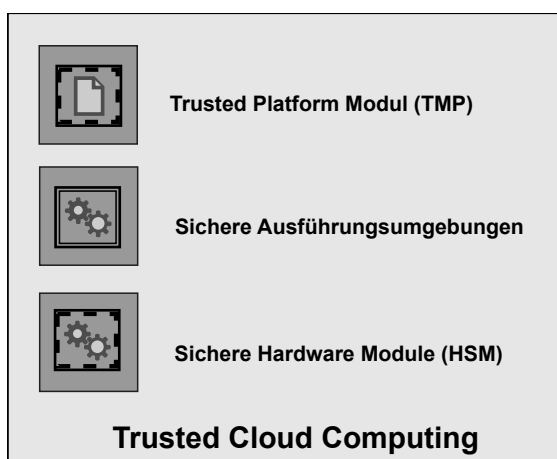


Abbildung 5.1.: Die Bestandteile des Trusted Cloud Computing im Überblick. Die Abbildung visualisiert den Aufbau dieses Kapitels und ist die detaillierte Darstellung der Trusted Cloud Computing-Box aus dem Kapitel Taxonomie der Cloud-Sicherheit.

tel unterteilt sich in drei Abschnitte: Im ersten Abschnitt werden *Trusted Plattform Module* als eine einfache und kostengünstigen Realisierung des Trusted Cloud Computings vorgestellt. Im zweiten Abschnitt geht es um *Sichere Ausführungsumgebungen*, welche einen hardwareseitigen Vertrauensanker in abgesicherte Software-Ausführungsumgebungen erweitern. Abschließend werden im dritten Abschnitt speziell gefertigte Hardwaremodule (Hardware Security Module – HSM) erörtert, die unterschiedliche Aufgaben, wie Schlüsselmanagement oder sicheren Ausführung von Programmcode, wahrnehmen können.

¹Der Vorteil bei der Nutzung von Hardware ist, dass dieser mehr zu vertrauen ist als leichter veränderbare Software. Dies erörtert Schneier [Sch11b] in seiner Arbeit.

5.1. Trusted Plattform Module

Trusted Plattform Module (TPM) wurden von der Trusted Computing Group [TCG14a] als sichere Module eingeführt und setzen sich aus einem Hardware- und einem Softwareteil zusammen. Der Hardwareteil besteht laut der Spezifikation² aus einem passiven Chip, vergleichbar mit einer Smartcard. Der Softwareteil wird als Trusted Software Stack bezeichnet und stellt im Wesentlichen eine API des TPM dar. Die Abbildung 5.2 illustriert ein solches TPM sowie dessen internen Aufbau schematisch.

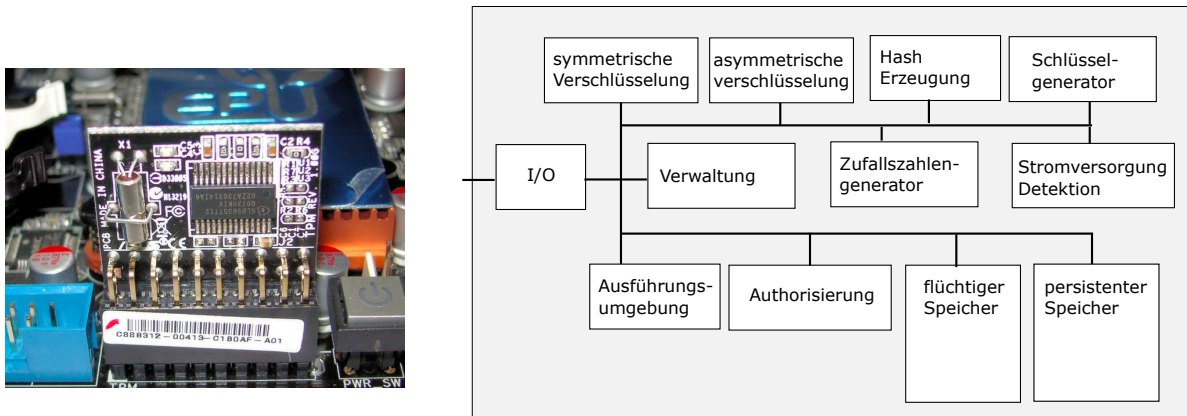


Abbildung 5.2.: Ein Trusted Plattform Modul Chip auf einem Motherboard (links) sowie die TPM-Architektur nach der Spezifikation [TCG14b] (rechts)

Die Ziele, die mit einer Verwendung von TPM in Verbindung gebracht werden, sind digitales Rechtemanagement (DRM) und Schutz der Durchsetzung von Softwarelizenzen. Zudem sieht die Spezifikation der TCG eine Überwachung des Boot-Vorganges vor, um besseren Schutz gegen Maleware zu erhalten. Wie Eckert [Eck13, S.626] hervorhebt, handelt es sich hierbei nur um eine Unterstützung für einen sicheren Bootvorgang in Form einer vertrauenswürdigen Protokollierung des Startvorgangs, die Reaktion auf Verstöße wird in der Spezifikation nicht festgelegt. Nach Francis et al. [FDEM13] setzen TPM die Kernziele sicheren Speicher, Auswertung der Plattform-Integrität und Authentisierung um. Zur Umsetzung dieser Ziele bieten TPM folgende grundlegenden Funktionen (in Klammern finden sich die Bezeichnungen aus Abbildung 5.2):

- kryptographische Berechnungen ((a)symmetrische Verschlüsselung und Hash Erzeugung)
- Ausführungsumgebung für TPM interne Berechnungen (Ausführungsumgebung)
- Schlüsselgenerierung (Schlüsselgenerator) mit Zufallsgeneratoren (Zufallszahlengenerator)
- Schlüsselspeicherung TPM intern/extern (persistenter/flüchtiger Speicher)
- Attestierung des TPMs und von Daten gegenüber Dritten (Authentisierung)

TPM bieten einen hardwareseitigen Vertrauensanker für die sichere Speicherung von Daten und die sichere Ausführung von Programmen. So bietet die Intel Trusted Execution Technology (Intel TXT)³

²Die aktuell (Stand Sep. 2014) gültige Version ist 1.2 von 2011 [TCG11], im März 2014 wurde Version 2.0 zur öffentlichen Besichtigung freigegeben [TCG14b]. Unterschiede werden unter http://en.wikipedia.org/wiki/Trusted_Platform_Module aufgelistet

³Vgl. [SRW⁺08, S.46]

eine Verlängerung der Vertrauenskette für sichere Ausführungsumgebungen (in Hardware) für Betriebssysteme und deren Applikationen. Für detaillierte Informationen hierzu sei auf Literatur von Yeluri und Castro-Leon [YCL14] verwiesen. Die Kosten von TPM-Chipsätzen liegen bei etwa 15\$⁴. Die Chips können beispielsweise für die Data-on-Rest Verschlüsselung in aktuellen Windows Betriebssystemen über Bitlocker genutzt werden. Basierend auf dieser Grundlage existieren zahlreiche Forschungsansätze, die nachfolgend diskutiert werden.

Santos et. al. [SRGS12] haben die Eignung von TPM für Cloud Computing Umgebungen untersucht und gelangten zum Schluss, dass diese nur ungenügend für den Einsatz in Cloud-Umgebungen geeignet sind. Für die Feststellungen werden folgende Gründe genannt:

Erstens fokussieren TPM das Schützen von sensiblen Informationen auf einzelnen Geräten. Multi-Rechnerarchitekturen, wie Cloud- Umgebungen, spielen eine ungeordnete Rolle, entsprechend eignen sich TPM nur bedingt zur parallelen und verteilten Berechnung. So ist die von der Spezifikation beschriebene Datenversiegelung, welche neben einem Schlüssel, Parameter der aktuellen Ausführungsumgebung des TPM einbeziehen, für Multi-Rechner Umgebungen, mit einer Vielzahl an TPM, praktisch nicht einsetzbar.

Zweitens führen TPM zur Identifizierung konkreter Cloud- Rechnerknoten. Diese hochsensiblen Informationen bieten völlig neue Angriffsszenarien gegen CSP. Die Arbeiten von Santos et.al. schlagen zur Lösung dieses Problems die Verwendung einer Datenversiegelung vor, die neben einer Datenverschlüsselung allgemeine Parameter der Umgebung des Verschlüsselungssystem mit einbezieht. Möglich wird dies durch die Verwendung einer Ciphertext Policy Attribute-Based Encryption⁵ [BSW07] und des TPM Technologiestacks. Zudem existiert in ihrem Ansatz eine zentrale Monitoring Instanz, welche die Vorgänge in der Cloud-Umgebung in vertrauenswürdiger Form überwacht. Die Autoren haben in ihrer Arbeit festgestellt, dass die Implementierung der TPM Schnittstellen ineffizient seien und so ein wesentliches Hindernis für die Einbettung von Cloud-Umgebungen sind. Dennoch wurde gezeigt, dass die Verwendung von TPM in dynamischen Cloud-Umgebungen, potenziell geeignete Mittel darstellen, einen hohen Grad an Sicherheit zu bieten.

Die Arbeit von Schiffman et al. [SMV⁺10] versucht die Transparenz innerhalb von IaaS-Services zu verbessern, in dem Nutzer Integritätsbeweise zu deren virtuelle Maschinen und dem unterliegenden Hypervisor geboten werden. Dazu besteht auch in diesem Ansatz eine zentrale Komponente, der Cloud-Verifier (CV), der die Attestierung der Cloud- Knoten vermittelt und dazu Eigenschaften der Cloud-Knoten nutzt. Die Arbeit der Autoren erreicht die folgenden drei Ziele: Erstens bietet der Cloud-Anbieter Beweise zur Datensicherheit. Zweitens haben diese eine klare Bedeutung für das Vertrauen der Nutzer und drittens können die Beweise effektiv und effizient in der Cloud-Umgebung generiert werden.

Zur Integration von TPM in IT-Systeme wurde das Forschungsprojekt *European Multilaterally Secure Computing Base* durchgeführt, in welchem Turaya [EMSCB06] entwickelt wurde. Turaya soll, laut EU Ziel, die Sicherheitsprobleme bereits vorhandener Betriebssysteme minimieren und eine technologische Grundlage für zukünftige innovative Geschäftsideen bilden. Die offene und sichere Architektur erlaubt es, *Turaya bereits auf vorhandenen Rechnersystemen unabhängig von Betriebssystem und Plattform einzusetzen. Überprüfbarkeit der Vertrauenswürdigkeit fremder Rechnersysteme steht dabei im Vordergrund. Turaya überprüft und beglaubigt Rechnerkonfigurationen und sorgt dafür, dass die für moderne Geschäftsabläufe benötigten Sicherheitsregeln in DRM-Anwendungen auf Seiten der Nutzer und der Anbieter durchgesetzt werden. Dabei werden auch mögliche gegenseitig auftretende Konflikte durch effektives Regelmanagement verhindert* [IS15]. Dem Autor liegen jedoch keine Informationen zu Veröffentlichungen über die praktische Nutzung

⁴Stand Sep. 2014 Quelle Amazon.com [TPM15]

⁵Details werden unter 4.2.4 beschrieben.

von Turaya vor.

5.1.1. Virtuelle Trusted Plattform Module

Auf Grund das Cloud-Umgebungen auf Virtualisierungstechnologien beruhen und auf einem physikalischen Knoten häufig mehrere virtuelle Maschinen (VMs) gehostet werden, liegt es nahe TPM ebenfalls zu virtualisieren.⁶ Die grundlegende Idee hinter eines virtuellen TPM ist die Verfügbarkeit eines solchen in jeder virtuellen Maschine. Berger et al. [BSG06] legen die Basis für das Gebiet der *Trusted Virtualization*⁷. Scarlata et. al. [SRW⁺08, S.46] weisen darauf hin, dass das Verständnis der Unterscheidung beider TPM Virtualisierung zwischen TPM Sharing und vTPM wichtig ist. Während im ersten Fall VMs nur Teile des TPM gemeinsam nutzen, wird bei vTPM der gesamte TPM-Funktionsumfang von verschiedenen VMs genutzt. Letzteres ist nach Meinung der Autoren für die praktische Anwendung vorzuziehen. Die Abbildung 5.3 verdeutlicht zwei mögliche Ansätze des virtualisierten TPMs (vTPM).

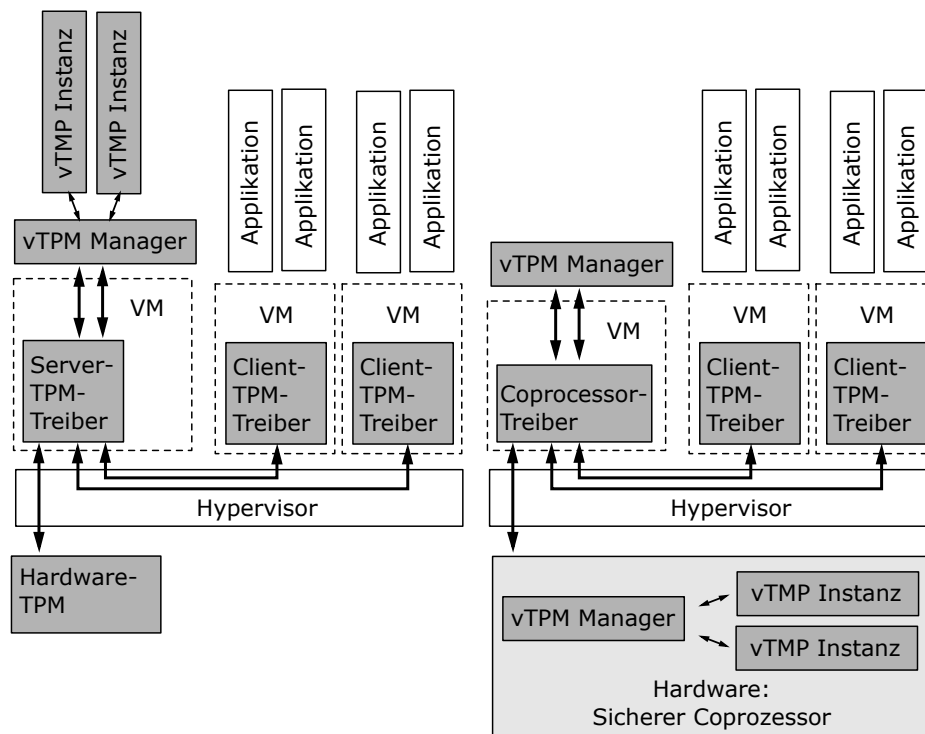


Abbildung 5.3.: Möglichkeiten der Realisierung einer virtualisierten TPM-Architektur (vTPM) nach von Berger et al. [BSG06]. Die Kommunikationswege werden durch Pfeile, Sicherheitskomponenten dunkler, dargestellt.

Deutlich erkennbar ist, dass bei vTPM der Hypervisor eine zentrale Rolle spielt und dass die Verwaltung der vTPM in eine eigene VM ausgelagert wird, an welchem alle Anfragen gesendet werden. Weiterhin ist zu erkennen, dass die Sicherheit der vTPM in sicherer Hardware, wie TPM oder sicherer Coprozessor⁸, verankert ist. Dabei kann laut Berger et. al. [BSG06] die gebotene Sicherheit eines

⁶Die Grundlagen der Virtualisierung werden im Abschnitt 6 zusammengefasst.

⁷Auf dieses wird in Abschnitt 6.2.1 eingegangen

⁸Ein Beispiel ist der IBM 4764 PCI-X <http://www-03.ibm.com/security/cryptocards/pcixcc/overview.shtml>, letzter Zugriff 22.09.2014

Software-TPMs genau so hoch sein wie die eines Hardware TPMs. Dafür müssen nach Berger et al. und Sadeghi et al. [SSW08] folgende Anforderungen erfüllt sein:

1. Ein vTPM muss für das Gastsystem in der gleichen Weise verwendbar sein und den gleichen TPM Befehlssatz bieten, wie einem System, das die Hardware direkt verwendet.
2. Die Verbindung zwischen VM und seinem vTPM muss über den gesamten VM-Lifecycle verwaltet werden. Dies umfasst zudem die Migration einer VM auf einen anderen physikalischen Host.
3. Die Verbindung zwischen vTPM und der unterliegenden Trusted Computing Base muss verwaltet werden.
4. Ein vTPM muss klar von einem Hardware TPM unterscheidbar sein, da beide unterschiedliche Sicherheitseigenschaften besitzen.
5. Der Zustand eines vTPM darf nicht anfällig für Replay-Attacken sein. (Key-Freshness)
6. Versiegelte Daten eines vTPM müssen auch nach einer VM Migration oder nach einem Software Update der Plattform weiterhin verfügbar sein.
7. Der Nutzer sollte entscheiden können, welche Informationen der Plattform (Konfiguration der Hardware und des Hypervisors) an die VM bzw. Dritte weitergereicht wird.
8. Verschiedene Sicherheitslevel und Kryptoverfahren sollten unterstützt werden.

Zusammengefasst kann festgestellt werden, dass der Vorteil gegenüber der direkten Verwendung von TPM die erhöhte Hardwareunabhängigkeit von den physikalischen Rechnern ist. Die Verwaltung der vTPM kann über einen zentralen, mit dem Hypervisor verbundenen, TPM oder sicheren Coprozessor abgebildet werden. Die Sicherheit kann durch erhöhten Integritätsschutz in den VMs verbessert werden und bietet damit eine verbesserte Transparenz der gesamten Cloud-Umgebung. Dennoch ergeben sich aus den genannten Vorteilen gleichzeitig einige Nachteile. So stellt die zentrale Verwaltung der vTPM einerseits einen Single-Point-of-Failure und andererseits einen Engpass bei der Skalierung dar. Durch Verteilung kann dem zwar effektiv entgegengewirkt werden, aber damit erhöht den Aufwand wesentlich. Zudem wird durch Redundanz und Verteilung des vTPM Managers, die anfangs geringe Hardwareabhängigkeit schlussendlich wieder erhöht. Ferner ergeben sich durch die Virtualisierung Einbußen in der Performance, sowohl auf der Seite der Applikation als auch in der Verwaltung der VMs. Letzteres auch bedingt durch die erhöhte Komplexität der Virtualisierung.

5.2. Sichere Ausführungsumgebungen

Dieser Abschnitt beschreibt sichere Ausführungsumgebungen (Secure Execution Environment, SEE) innerhalb eines nicht vertrauenswürdigen Betriebssystems. Somit wird weder der virtuellen Maschine, noch dem Gastsystem in dieser Maschine, Vertrauen geschenkt und der auszuführende Code dementsprechend abgesichert.

Die Autoren McCune et al. [MPP⁺08, MLQ⁺10] haben Ansätze zur Gewährleistung einer isolierten Ausführung von sicherheitskritischen Quellcode entwickelt. So sichert das Flicker-System [MPP⁺08] feingranular den sicherheitskritischen Applikationscode im System und fügt der Trusted Code Base (TCB)⁹ nur ein wenige hundert Codezeilen hinzu. Es zeigt sich zudem, dass die verbreiteten Systeme in der Lage sind, sicher Code auszuführen ohne Vertrauen in das unterliegende Betriebssystem zu benötigen. Ferner wird im Proof-of-Concept aber deutlich, dass die Leistungsfähigkeit (Latenz und Durchsatz) für viele Szenarien unpraktikabel ist. Jede Flicker-Session benötigt zur Ausführung einer Applikation eine signifikante Zeitspanne, da die langsamen TPM-Operationen auf dem systemkritischen Pfad sind, dass heißt bei jeder Operation durchlaufen werden müssen. Dies ist auch für moderate Serverlast (10-100 Nutzer) inakzeptabel. Ferner setzt Flicker voraus, sicherheitskritischen Code speziell zu kompilieren und mit sehr wenigen externen Abhängigkeiten zu verknüpfen, welches den Entwicklungsprozess verkompliziert und den Debugging-Prozess bei der Fehlersuche schwieriger macht.

In einem zweiten Ansatz zeigen McCune et.al. [MLQ⁺10] in ihrer TrustVisor-Lösung, dass eine viel bessere Performance mit aktueller Hardware möglich ist, wenn die Größe des vertrauenswürdigen Codes (TCB) geringfügig erhöht wird, aber weit unterhalb der von verbreiteten Hypervisoren bleibt. Die vorgeschlagene Lösung wird in Abbildung 5.4 dargestellt.

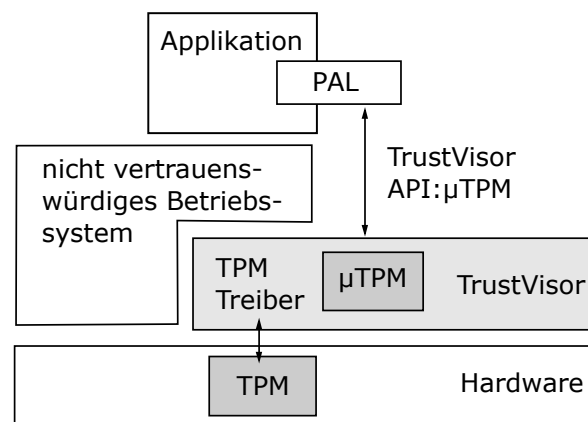


Abbildung 5.4.: Die Darstellung der Architektur von TrustVisor nach McCune et.al.[MLQ⁺10]. Der zu schützenden Programmcode (PAL) wird vom TrustVisor verwalteten Micro-TMP geschützt. Dieser stellt über TPM Treiber eine Verbindung zum Hardware TPM her. Die Sicherheitskomponenten werden dunkler dargestellt.

TrustVisor nutzt einen Registrierungsprozess, der mit dem existierenden Code kompatibel ist, welcher seinen Vorteil aus der Fähigkeit zieht, die Hauptspeicherverwaltung (Memory Paging Struktur) des bestehenden Betriebssystems zu verstehen. Dabei stellt der Lösungsansatz eine Verbindung zwischen dem zu schützenden Programmcode (PAL) und der vom TrustVisor verwalteten Micro-TMP.

⁹Als TCB wird der Programmcode bezeichnet der zur Ausführung des Programms nötig ist und potenzielle Sicherheitslücken enthalten kann. Eine kleine TCB ist aus Sicherheitssicht daher wünschenswert, da hier die Wahrscheinlichkeit von Programmierfehlern geringer ist.

Dieser Prozess stellt zugleich den wesentlichen Nachteil dar, da in einer bestehenden Anwendung Anpassungen vorgenommen werden müssen.

Das Ziel einer SEE sollte es laut Maniatis et al. [MAF⁺11] dagegen sein, dass keine Anpassungen an der Applikation notwendig sind. Laut dieser Autoren führen die Grenzen in Praktikabilität, Effizienz und Sicherheit zu einer Anzahl an obligatorischen Voraussetzungen für eine solche Lösung:

- Der Umgang mit beliebigen Altsystemen bedingt Überwachungs- und Isolationsmechanismen, da die Neugestaltung oder Neu-Kompilierung nicht möglich sind.
- Die effiziente Unterstützung von Applikationen, zur komplexen und vielfältigen Verarbeitung sensibler und nicht-sensibler Daten erfordert die Überwachung des Informationsflusses.
- Die ausführlich durch Schneier [Sch11b, S.186] erörterte Einbeziehung von Betriebssystemen und Massensoftware in die TCB, ist inkompatibel mit den Zielen einer hohen Sicherheit.
- Der kryptographische Schutz der Vertraulichkeit und Integrität von gespeicherten Daten ist notwendig, um Systeme auszuschließen, die unerlaubt Daten aus der TCB verbreiten wollen.
- Eine vertrauenswürdige und isolierte Komponente wird benötigt, welche die Credentials, Authentifizierung und die Speicherung von Schlüsseln verwaltet, da Kryptographie zwischen verschiedenen Parteien zum Einsatz kommt.

Die aufgezeigten Eingrenzungen führen nach Maniatis et al. zu dem in Abbildung 5.5 dargestellten Architekturkonzept.

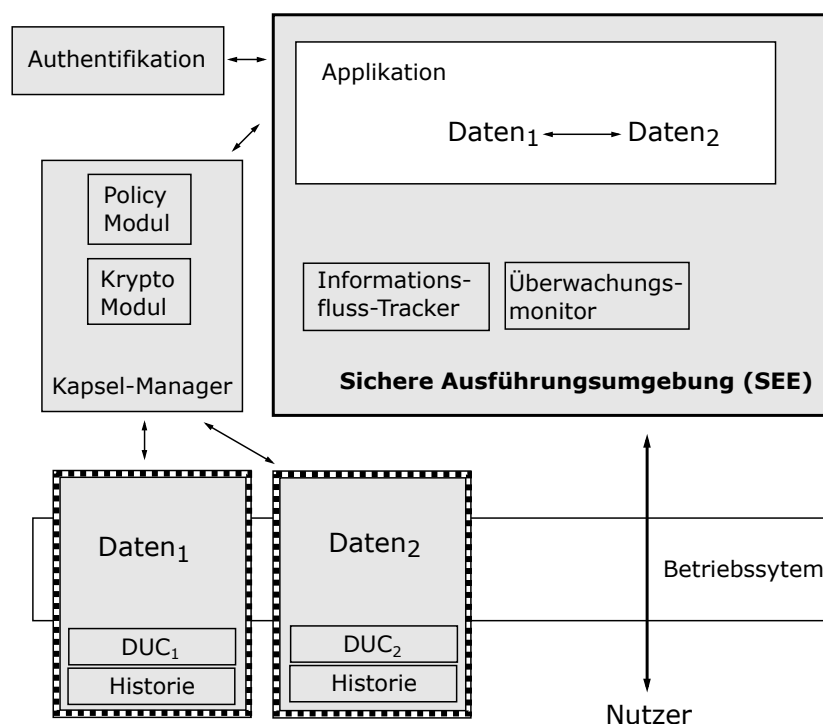


Abbildung 5.5.: Die konzeptionelle Architektur nach Maniatis et al. [MAF⁺11]. Die grau markierten Komponenten gelten als vertrauenswürdig, punktierte Ränder illustrieren die geschützten Datenkapseln.

Die Dateneinheiten in dieser Architektur sind sichere Datenkapseln, ein Container aus dem Datenobjekt (D), seiner Data-Use Controls (DUC_i) und seiner Historie. Diese Kapseln sind mit kryptographischen Mitteln zur Wahrung der Vertraulichkeit und der Integrität geschützt. Die sichere Ausführungsumgebung stellt den Überwachungsmonitor und Informationsfluss-Tracker dar, der eine kontrollierte Ausführung nicht vertrauenswürdiger Applikationen erlaubt. Diese greift in Systemaufrufe ein und interagiert mit dem Nutzer über vertrauenswürdige I/O Kanäle. Außerhalb der Trusted Computing Base befinden sich das Betriebssystem, der Nutzer und die Applikation. Da es sich nur um ein theoretische Konzept handelt, kann über die praktische Anwendung bzw. Leistungsfähigkeit keine Aussage gemacht werden.

5.3. Hardware-Sicherheitsmodule

Unter Hardware-Sicherheitsmodule (HSM) sind im Gegensatz zu den vorgestellten TPM, aktive Prozessoren oder komplexere Recheneinheiten zu verstehen. Im ersten Teil dieses Abschnitts werden HSM in Sinne von kryptografischen Coprozessoren und den entsprechend Lösungsansätzen im Cloud-Umfeld beschrieben. Abschließend erfolgt am Ende des Abschnittes eine kurze Diskussion von HSM im Sinne von Hardware Modulen zum Schlüsselmanagement.

Die Beschreibung des verbreiteten Coprozessors IBM 4758 dargestellt durch Dyer et.al [DPS01] zeigt als wesentliche Unterschiede zu TPM die hohe Leistungsfähigkeit und die Tatsache, dass Coprozessoren häufig tramper-proof sind, d.h. auch gegen einen physikalischen Angriff geschützt. Coprozessoren stellen nach Itani et al. [IKC09] ein vollständiges Rechnersystem mit Prozessor, RAM, ROM, Backup Batterien und nicht flüchtigem Speicher dar, deren Kosten sich laut Berger et al.[BSG06] auf über 1000\$ belaufen. Dennoch finden derartige Systeme Berücksichtigung in den Forschungsarbeiten von [BSG06], Bajaj et.al. [BS11] und Arasu et.al [ABE⁺13], da diese eine effektive Möglichkeit darstellen einen leistungsfähigen Vertrauensanker aufzubauen. Die Abbildung 5.6 veranschaulicht einen solchen sicheren Coprozessor.

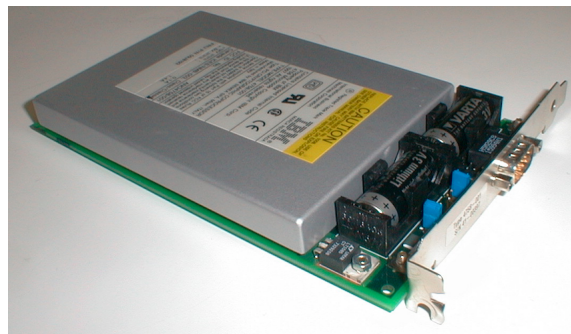


Abbildung 5.6.: Die Abbildung eines IBM 4758 aus der Arbeit von Anderson et al. [ABC05]

Neben Coprozessoren existieren in modernen Prozessoren immer häufiger spezielle Bereiche für kryptographische Funktionalitäten bzw. in dem besonders schützenswerter Code ausgeführt werden kann. Technologie-Beispiel sind die Intel TXT, die AMD Secure Virtual Machine oder die Trustzone von ARM¹⁰. Letztere wird in aktuellen Smartphones genutzt, um hochsensible Daten abzulegen und den Zugriff auf sicherheitskritische Sensoren zu regulieren, für Details sei auf die Arbeiten von Gualti et.al. [GSY14] und Polzin et.al. [PGA⁺14] verwiesen.

Eine Integration von HSM in das Cloud-Umfeld beschreiben Bajaj et al. [BS11] mit ihrem Ansatz

¹⁰<http://www.arm.com/products/processors/technologies/trustzone/index.php>, letzter Zugriff 21.01.2015

TrustDB in einer Kombination aus sicheren Servern und persistenter Datenverschlüsselung. In dieser kommen sichere Coprozessoren und gewöhnliche Server zum Einsatz. Innerhalb der SCP wird eine leichtgewichtige SQLite Datenbank, wogegen auf den gewöhnlichen Servern eine MySQL Datenbank genutzt wird. Begründet ist dies durch die eingeschränkten Möglichkeiten eines Coprozessors, eine vollständige SQL Datenbank mit allen Funktionalitäten zu betreiben. Die Query-Verarbeitung wird auf die beiden Datenbanken aufgeteilt: Sensible, verschlüsselte Daten werden vom Coprozessor verarbeitet und nicht sensible, unverschlüsselte Daten vom MySQL-Datenbankserver. Arasu et.al. [ABE⁺13] bezeichnen diesen Ansatz als die beste Verwendung aktuell verfügbarer Technologien aus sicherer Hardware und Standardhardware.

In ihrer Lösung beschreiben Arasu et al. einen weiteren Ansatz mithilfe einer FPGA¹¹ basierten Datenbank *Chipperbase*. Die Autoren nutzen im Gegensatz zur *TrustDB* eigens entwickelte FPGA als Vertrauensanker. Das Verarbeitungsprinzip entspricht dem von *TrustDB*. Erweiterte ODBC-Treiber für die Kommunikation mit Microsoft SQL Servern ermöglichen sichere Verarbeitung der vom Nutzer verschlüsselten SQL-Queries. Laut den Autoren Arasu et al. soll dadurch eine bessere Leistungsfähigkeit als durch sichere Coprozessoren erreicht werden können. Diese Behauptung wird in der Arbeit jedoch durch die Autoren nicht validiert. Kritisch muss ebenfalls die Entwicklung der kryptographischen Funktionen in Form der FPGA-Chipsatzes betrachtet werden, welche den Entwicklungsaufwand für diese Lösung vervielfacht und als unpraktikabel scheinen lässt.

Itani et al. [IKC09] verfolgen einen anderen Ansatz bei der Integration von sicherer Hardware. Die Autoren führen eine vertrauenswürdige Drittpartei ein, welche die Sicherheit der Daten sicherstellen soll. Dadurch ist es möglich die sicheren Coprozessoren auf der Seite des CSP zwischen verschiedenen Cloud-Nutzern zu teilen. Dies stellt einen erheblichen Kostenvorteil gegenüber einer exklusiven Nutzung dar. Die vertrauenswürdige Partei übernimmt dabei wesentliche Aufgaben des Schlüsselmanagements, die zur Datenverarbeitung beim CSP nötig sind. Voraussetzung des Ansatzes ist die vorherige Klassifikation von ausführbarer Software und verarbeiteter Daten in sensibel und nicht sensibel.

Die Lösung von Anciaux et al. [ABB⁺07] nutzt für ihren Ansatz erweiterte USB-Sticks mit sicheren Coprozessoren. Sensible Daten werden lokal auf dem USB Stick abgespeichert und sind so von nicht sensiblen Daten, die auf gewöhnlichen Servern der Cloud Umgebung abgelegt werden, getrennt. Herausforderung ist hierbei die Datenzusammenführung über die physikalisch getrennten Daten über eine gemeinsame Schnittstelle auf dem Client.

Neben der direkten Nutzung von sicheren Coprozessoren, ist es zudem in der Praxis verbreitet, HSM zu spezifischen Aufgaben, wie dem Schlüsselmanagement, zu nutzen. Die Module, welche in diesem Anwendungsfall als HSM bezeichnet werden, stellen in der Regel vollständige Server oder Blades dar. Innerhalb dieser werden sichere Coprozessoren verwendet, um kryptografischen Operationen und eine sichere Datenspeicherung zu ermöglichen. Der CSP Amazon bietet mit seinem Service AWS CloudHSM¹² zudem die Möglichkeit derartige HSM auch innerhalb einer öffentlichen Cloud-Umgebung zu verwenden. Der Service bietet verschiedene Einsatzszenarien an, reicht aber aufgrund der exklusiven Nutzung die Anschaffungskosten für die Hardware an den Nutzer weiter. Der Vorteil ist, dass die Verwaltung weiterhin Aufgabe des CSP bleiben kann, der dennoch keinen Zugriff auf gespeicherten Schlüssel hat.

¹¹Logisch programmierbare Schaltkreise, weitere Informationen bietet: <http://www.mikrocontroller.net/articles/FPGA>, letzter Zugriff 09.06.2015

¹²<http://aws.amazon.com/de/cloudhsm/faqs/>, letzter Zugriff 09.02.2015

5.4. Zusammenfassung

Die in diesem Kapitel vorstellten und diskutieren Maßnahmen dienen vor allem dem Schutz der Cloud-Umgebung. Basierend auf der Integration von sicheren Hardware-Modulen in die Cloud-Infrastruktur, wird die Sicherheit der Umgebung und damit der Cloud-Applikationen erhöht, die innerhalb dieser Cloud-Umgebung ausgeführt werden. Dabei stellt das Kapitel verschiedene Ansätze vor, begonnen bei der Integration von TPM bis zur spezifischer Hardware zum Schlüsselmanagement.

Interessanterweise widerspricht die Forderung vertrauenswürdiger Hardware aus Sicht des Cloud-Nutzers der Black-Box-Charakteristik von Cloud Computing, demnach ist ein Grundgedanke von Cloud-Ressourcen diese zu Nutzen ohne deren exakte inneren Aufbau zu kennen oder kennenlernen zu wollen. In diesem Punkt unterstützt dieses Kapitel die Annahme der These 1 dieser Arbeit in besonderem Maße.

Paulus [Pau11] benennt in seiner Arbeit Information Rights Management und damit weiter gefasst Trusted Cloud Computing als Lösung für das Sicherheitsproblem in der Cloud. Tatsache ist jedoch, dass sich derartige Konzepte auf grund ihrer Kosten, der schwierige Handhabung und Vendor-Lock-Ins bisweilen haben nicht durchsetzen können. Eine Ausnahme bilden sicherheitskritische Infrastrukturen von Banken, Regierungen und dem Militär. Diese werden jedoch im Gegensatz von öffentlichen Cloud-Umgebungen exklusiv genutzt, welches in den obigen Fällen exakt das Ziel der Lösungen war. Inwiefern sich der scheinbare Widerspruch aus sicherer, exklusiv genutzter Hardware und dynamisch geteilten Umgebungen wie die Cloud vereinbaren lassen, werden die nächsten Jahre zeigen. Eine Tendenz ist erkennbar, das zunehmend auch sensible und sicherheitskritische Daten und Prozesse in die Cloud ausgelagert werden sollen. Amazon Web Services bietet mit der GovCloud Lösung¹³ eine geeignete Lösung an.

Das Prinzip der Delegation mit begrenztem Wissen kann auch in diesem Kapitel deutlich beobachtet werden. Dabei wird, im Fall des Trusted Cloud Computing, die Arbeit des Cloud-Nutzers nicht durch die Preisgabe von Informationen verringert, sondern durch die Bereitwilligkeit einen höheren Preis zu zahlen. Denn die Kosten der Hardware werden auf die Ressourcennutzungskosten der Cloud-Nutzer aufgeschlagen. Bei exklusiver Nutzung ist sogar eine vollständige Übernahme der Investitionskosten erforderlich, die sich nur bei langfristiger Nutzung rentieren.

Dies steht offensichtlich im Widerspruch zu den Cloud-Vorteilen, wie diese im Grundlagenteil der Arbeit beschrieben werden und bestätigt damit die These 1 der Arbeit.

¹³<http://aws.amazon.com/de/govcloud-us/>, letzter Zugriff 29.07.2015

Cloud-Virtualisierungssicherheit

Dieses Kapitel beschreibt Maßnahmen zum Schutz von Virtualisierungsumgebungen. Da Virtualisierung ein integraler Bestandteil jeder Cloud- Umgebung ist, bieten die beschriebenen Lösungsansätze vorrangig für den Schutz Cloud-Umgebungen und entsprechend nur indirekt für die Cloud-Applikation. Dennoch werden zudem Ansätze diskutiert, die eine Anwendung auch ohne Zugriff auf die Virtualisierungsschicht ermöglichen. Ziel der Maßnahmen ist es, dass auch Administratoren der Hardware und virtuellen Maschinen keinen oder nur beschränkten Zugriff auf den Inhalt der virtuellen Maschinen der Nutzer haben sollen. Die Abbildung 6.1 zeigt die Unterteilung dieses Kapitels.

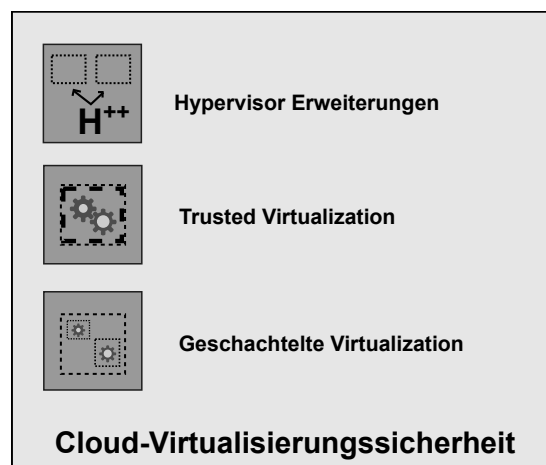


Abbildung 6.1.: Die Bestandteile der Cloud-Virtualisierungssicherheit im Überblick. Die Abbildung visualisiert den Aufbau dieses Kapitels und ist die detaillierte Darstellung der Cloud-Virtualisierungssicherheit-Box aus dem Kapitel Taxonomie der Cloud-Sicherheit.

Im nachfolgenden Abschnitt wird das Grundprinzip der Virtualisierung erläutert zbd niut der Beschreibung der *Hypervisor Erweiterungen* zugleich Maßnahmen zur Verbesserung der Sicherheit vorgestellt und diskutiert. Wie im vorherigen Kapitel beschrieben ist es durch Nutzung von sicherer Hardware möglich den Virtualisierungslayer zu schützen, dies wird als *Trusted Virtualization* bezeichnet. Als letzter Teil dieses Kapitels wird die Bedeutung von *geschachtelter Virtualisierung* in Cloud-Umgebungen und Cloud-Applikationen erörtert.

6.1. Virtualisierung im Zusammenhang mit Cloud Computing

Die Verwendung von Virtualisierung zur effizienteren Ausnutzung physikalischen Ressourcen, ist keine Idee, die mit dem Entstehen des Cloud Computings zusammenhängt. Vielmehr reicht diese Idee bis Mitte der 1960er Jahre zurück, in der es die Virtualisierung ermöglichen sollte große Mainframes effektiver zu nutzen. Lange Zeit war dieser Gedanke wegen möglicher Leistungseinbußen unpopulär. In den Jahren nach 1990 erlebte die Technologie durch hardwareseitige Unterstützung eine erneute Welle der Popularität und hat bis heute für eine hohe Verbreitung dieses Technologieansatzes gesorgt. Durch diese Unterstützung sind heute Einbußen durch Virtualisierung praktisch zu vernachlässigen [BDF⁺03].

Der nachhaltige Erfolg des Cloud Computings basiert in weiten Teilen auf die umfangreiche Verwendung von Virtualisierungstechnologien. Denn Cloud Computing stellt grundlegend die flexible Nutzung bzw. Bereitstellung von virtuellen Maschinen dar. Aus diesem Grund hängt die Sicherheit der Virtualisierung unmittelbar mit der Sicherheit der Cloud-Umgebung zusammen.

Grundlegend stellt die Virtualisierung eine Abstraktionsschicht zwischen der Hardware und dem Betriebssystem dar. Diese ist eine Form der losen Kopplung zwischen einem virtuellem Gastsystem und der physikalischen Hardware. Die Abbildung 6.2 bietet eine einfache Darstellung der Zusammenhänge.

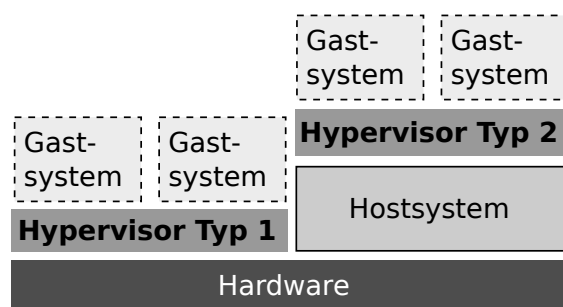


Abbildung 6.2.: Das Grundprinzip der Virtualisierung und die Hypervisor-Klassifizierung nach Popek und Goldberg [PG74]

Wie in der obigen Abbildung erkennbar existiert zwischen Gastsystem und Hardware keine direkte Verbindung. Ein so genannter Hypervisor befindet sich dazwischen, der die Virtualisierung und damit als Aufgaben die Abstraktion von der Hardware und die Isolation der Gastsysteme untereinander übernimmt. Neben diesem Prinzip, wird in zwei Virtualisierungsarten unterschieden. Die volle Virtualisierung (auch Hardware Virtual Machine) bezeichnet die Virtualisierung von Gastsystem ohne dessen Anpassung. Das System erlangt keine Kenntnis, dass es nicht direkt auf physikalische Ressourcen zugreift. Die Paravirtualisierung bezeichnet dagegen die geringfügige Anpassung des virtualisierten Gastsystems, zur Verbesserung der Leistungsfähigkeit. Das Gastsystem ist damit die virtuellen Umgebung bekannt.

Allgemein ermöglicht die Virtualisierung eine Vielzahl von isolierten Gastsystemen mit einer grundlegenden Form der Mehrmandantenfähigkeit. Durch dieses im Cloud-Umfeld intensiv genutzte Prinzip, wird jedoch laut Francis et al. [FDEM13] die Vertrauenswürdigkeit des Gesamtsystems nicht erhöht. Steinberg und Kauer [SK10] weisen zudem darauf hin, dass Virtualisierung die Sicherheit positiv oder negativ beeinflussen kann, abhängig von der Verwendung. Einerseits erhöht sich die Angriffsfläche des Gesamtsystems um den Hypervisor. Andererseits können so sicherheitskritische Funktionen separiert und von Gastsystemen unabhängig gemacht werden. Als Beispiel nennen die Autoren Firewall-Funktionalitäten. Weiterhin weisen Chen et al. [CN01] darauf hin, dass alle Sicher-

heitsmaßnahmen innerhalb des Virtualisierungslayers jedem Gastsystem zur Verfügung stehen bzw. dieses schützen. Als ein Forschungsschwerpunkt der Sicherheit von Virtualisierungsumgebungen ergibt sich daher, der Schutz der Integrität von Gastsystemen und Hypervisoren. Die Autoren Popek und Goldberg [PG74] stellen in ihrer Arbeit grundlegende Virtualisierungsanforderungen dar und klassifizieren Hypervisoren in 2 Typen. Die Abbildung 6.2 verdeutlicht die Unterschiede.

Der Typ 1 bezeichnet Hypervisoren (native oder bare metal Hypervisoren), welche direkt durch die Hardware des Host System ausgeführt werden. Mittels direkten Zugriff auf die Hardware sind diese in der Lage die virtualisierten Gastsysteme zu verwalten. Diese befinden sich wie in Abbildung 6.2 illustriert auf einer Ebene über dem Hypervisor.

Der Typ 2 bezeichnet Hypervisoren (hosted Hypervisoren), welche innerhalb eines Betriebssystems ausgeführt werden. Der Hypervisor stellt in diesem Falle nur eine Softwareschicht dar, die keinen direkten Zugriff auf die Hardware hat. Das Gastsystem befindet sich auch hier auf einer weiteren Softwareebene über den Hypervisor.

Nach Shinagawa et al. [SET⁺09] erlauben es Typ 2 Hypervisoren in einfacher Art und Weise Sicherheitsfunktionalitäten zu entwickeln, so dass auf die Funktionen des Host-Betriebssystems zurückgegriffen werden kann. Jedoch ist damit auch die Trusted Code Base (TCB) um ein Vielfaches größer als bei Typ 1 Hypervisoren. Aus diesem Grund wird in der Praxis häufig Typ 1 verwendet, um darauf basierend, komplexe virtualisierte Systeme, wie Cloud-Umgebungen, zu betreiben.

Der Typ 1 Hypervisor Xen [BDF⁺03] nutzt ein Domain-Konzept zur Verwaltung der VMs. Eine besondere Bedeutung kommt dabei der ersten Domain *Dom0* zu. Diese ist im Gegensatz zu allen anderen Domains privilegiert, kommuniziert direkt mit dem Hypervisor und verwaltet die weiteren Domains (VMs) sowie deren Zugriffe auf die Hardware. Alle weiteren Domains werden als *DomU* bezeichnet und sind unprivilegiert. Xen kann sowohl voll virtuell als auch paravirtuell betrieben werden, abhängig von der unterliegenden Hardware. Zur TCB gehört bei Xen sowohl Hypervisor als auch *Dom0*. Verwendungen findet Xen bei CSP wie Amazon¹ und Rackspace². Ein weiterer OpenSource Hypervisor ist die Kernel-based Virtual Machine, kurz KVM [KKL⁺07], welcher den Linux Kernel insofern erweitert, dass dieser einen Hypervisor zur Verwaltung virtueller Maschinen darstellt. Aufgrund dieser Besonderheit ist KVM nicht abschließend Typ 1 oder Typ 2 zuzuordnen. Ebenso wie Xen kann auch KVM paravirtualisiert betrieben werden.

Bekannte proprietäre Hypervisoren sind der ESX/ESXi von VMware³ und der Hyper-V von Microsoft⁴. Beide sind Type 1 Hypervisoren. Ein äußerst umfangreicher Vergleich existierender Virtualisierungsplattformen findet sich auf der englischen Wikipedia Seite.⁵

¹<http://www.xenproject.org/project-members/141-amazon-web-services.html>, letzter Zugriff 19.04.2015

²<http://www.xenproject.org/project-members/199-rackspace-hosting.html>, letzter Zugriff 19.04.2015

³<http://www.vmware.com/de/products/esxi-and-esx/overview.html>, letzter Zugriff 19.04.2015

⁴<http://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx>, letzter Zugriff 19.04.2015

⁵http://en.wikipedia.org/wiki/Comparison_of_platform_virtualization_software, letzter Zugr. 19.04.2015

6.2. Sichere Virtualisierungslösungen

Der Ansatz Overshadow von Chen et al. [CGL⁺08] ist ein erweiterter Hypervisor, der einen verschlüsselten und integritätsgeschützten Arbeitsspeicher für voll virtualisierte Gastsysteme bietet. Die Voraussetzung ist, dass der Applikation im Gastsystem vertraut wird. Grundlage ist eine als *Multi-Shadowing* bezeichnete Technik, die eine erweiterte Speichervirtualisierung im Hypervisor einführt. Üblicherweise verwaltet dieser eins-zu-ein Beziehungen zwischen der Adresse des Gastsystems und der physikalischen Adresse. Overshadow nutzt kontextabhängige Beziehungen, die Applikationen einen unverschlüsselten Zugriff ermöglichen, dem Gastbetriebssystem jedoch nur einen verschlüsselten. Dies ermöglicht es dem Betriebssystem nach wie vor auf Ressourcen zugreifen zu können, jedoch ohne den Schutz der Applikation zu kompromittieren. Wie Steinberg [SK10] hervorhebt, trifft das ebenfalls auf ein komprimiertes Betriebssystem zu. Zur sicheren Ausführung bestehender Applikationen, sind laut Shinagawa et al. [SET⁺09] einige Anpassungen an diesen nötig, Overshadow muss die Semantik der Applikation erkennen, um zwischen den verschiedenen Kontexten zu unterscheiden.

Der entwickelte Hypervisor Bitvisor von Shinagawa et al. [SET⁺09] kann laut Steinberg und Kauer [SK10] in die Kommunikation mit I/O Geräten eingreifen, um eine für das virtualisierte System transparente Datenverschlüsselung und Intrusion Detection zu implementieren. Im Gegensatz dazu bietet der Hypervisor SecVisor von Seshadri et al. [SLQP07] Schutz gegen die Veränderung des Kernels des Gastsystems, ohne jedoch in den I/O Prozess einzugreifen. Realisiert wird dies durch eine Verifikation der PageTable Modifikationen [SET⁺09].

Wie Schneier [Sch11b] hinweist, ist es für die Sicherheit des Systems entscheidend, dass die TCB minimal ist. Aus diesem Grund kam es zu Entwicklung von Mikrokerneln. Laut Liedtke [Lie95] existieren drei entscheidende Abstraktionen, die ein Mikrokernel bieten sollte: Adressraum, Threads und Inter-Prozesskommunikation. Diesem Grundsatz einer minimalen TCB gehen Steinberg und Kauer [SK10] mit ihrem Microhypervisor NOVA nach. Ihrer Auffassung nach stellt NOVA die Brücke zwischen Hypervisor und Mikrokernel dar. Durch den minimalistischen Ansatz wird die TCB um ein vielfaches reduziert, wodurch die potenziellen Angriffsflächen ebenfalls reduziert werden. Der Microhypervisor bietet ein hypercall-basiertes Interface mit verschiedenen Kernelobjekttypen zur Ausführung, Scheduling und zum Speicherzugriff verschiedener Prozesse. Wie in Abbildung 6.3 zu erkennen ist, unterscheidet sich die Architektur von NOVA von der klassischen aus Abbildung 6.2.

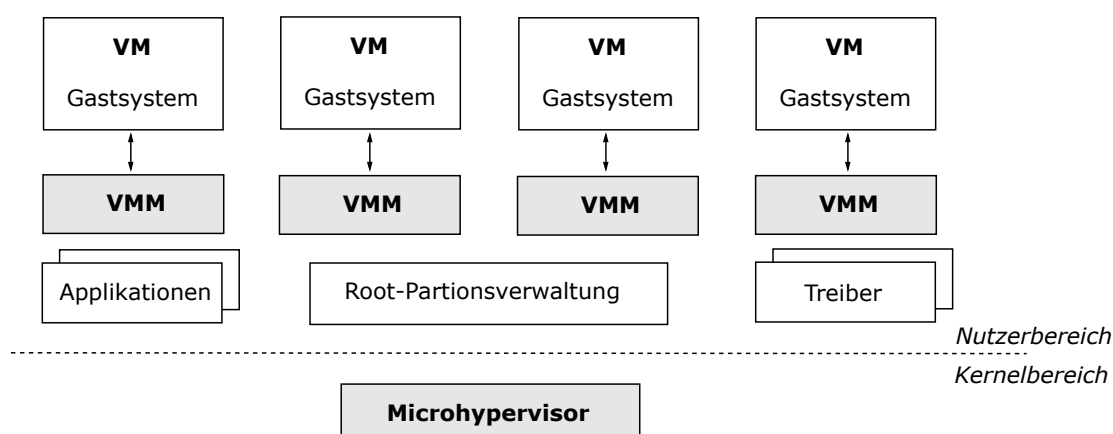


Abbildung 6.3.: Die NOVA Microhypervisor nach Steinberg und Kauer [SK10]. Dunkle Bereiche stellen sicherheitsrelevante Komponenten dar.

Die Autoren unterscheiden zwischen Hypervisor und VMM, wobei Letzterer stets eine VM zuge-

ordnet ist. Diese Aufteilung ermöglicht eine stringenter Segmentierung der Funktionen. Der in der Literatur angegebene Mehraufwand dieses Lösungsansatzes ist mit angegebenen 1-3% als sehr gering einzuschätzen.

Hao et.al. [HLMS10] nutzen in ihrem Ansatz virtualisierte Netzwerke für jeden Nutzer eines Cloud-Services. Dies bietet dabei eine starke Isolation der Nutzer bzw. der Zugriffe über das Netzwerk auf die Ressourcen in der Cloud-Umgebung. Abbildung 6.4 verdeutlicht die Architektur dieses Lösungsansatzes. Obwohl die Lösung nicht unmittelbar die Virtualisierungsumgebung betrifft, ist er

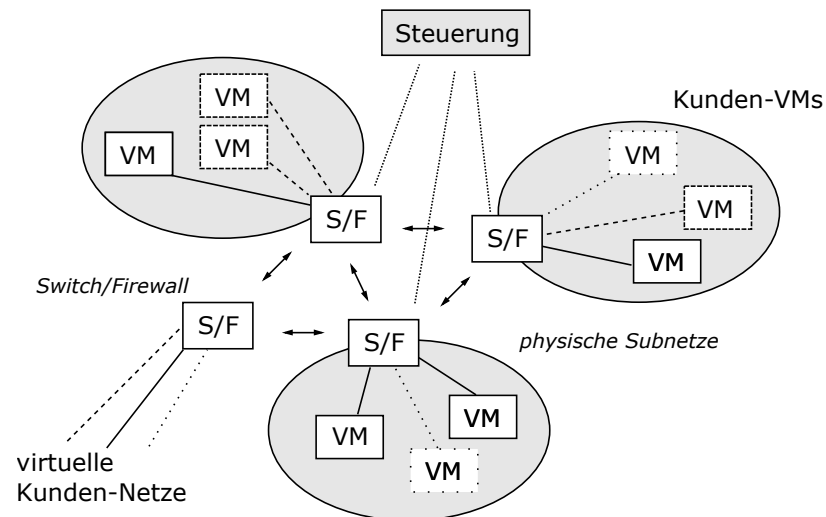


Abbildung 6.4.: SEC2 Architektur nach Hao et.al. [HLMS10]

dennoch für die Betrachtungen der Arbeit interessant, denn der Fokus liegt dabei nicht nur auf der Virtualisierung und Isolierung der Ressourcen, sondern auf dem Zugriff bzw. den Datenströmen zu diesen Ressourcen. Der Ansatz hebt Limitierungen von VLAN Ansätzen, wie in XEN verwendet, auf und bietet eine skalierende Lösung.

Wang und Jiang [WJ10] bieten einen hypervisorbasierenden Integritätsschutz für den vollständigen Lifecycle einer VM. Dieser schützt vor einer Veränderung von außen, d.h. vor externen Angreifern auf die Cloud-Umgebung. Auch die Ansätze von Lombardi und Di Pietro [LDP11] sowie VMwatcher [JWX07] und Lares [PCSL08] bieten Möglichkeiten den Kernel des Gastsystems zu schützen oder ein verlässliches Monitoring des Kernelverhaltens zu bieten. Sie stellen damit Möglichkeiten dar, die Sicherheit des Gesamtsystems durch die im Kapitel 2.2 eingeführte Forderung nach *Detection*-Mechanismen zu erhöhen.

6.2.1. Trusted Virtualization

Die Autoren Garfinkel et al. [GPC⁺03] entwickelten den Prototypen Terra, der einen Trusted Virtual Machine Monitor (TVMM) darstellt. Es war eines der ersten Systeme, welches TPM in einem Hypervisor integriert hat, um die Sicherheit der VMs zu erhöhen. Es bietet jedoch der VM keine virtualisierte Version des TPMs im Sinne eines vTPM von Berger et al. [BSG06]. Auf Grundlage dieser vTPM nach Berger et al. verbessern die Autoren Murray et al. [MMH08] die Xen Sicherheit durch Aufteilung einzelner Architekturkomponenten mit dem Ziel die TCB zu reduzieren. Dazu wird nur noch der Kernel der Dom0 als privilegiert betrachtet und der UserSpace der Dom0 separiert. Diese Arbeit bildet laut Aussagen der Autoren die Grundlage für das Gebiet der Trusted Virtualization. Aufgegriffen und weiterentwickelt wurde dieses Konzept von Bleikertz et al. [BBI⁺13] und Butt et

al. [BLCSG12]. Die beiden unabhängig voneinander entwickelten Lösungsansätze setzen auf die gleiche grundlegende Idee, die in Abbildung 6.5 illustriert wird.

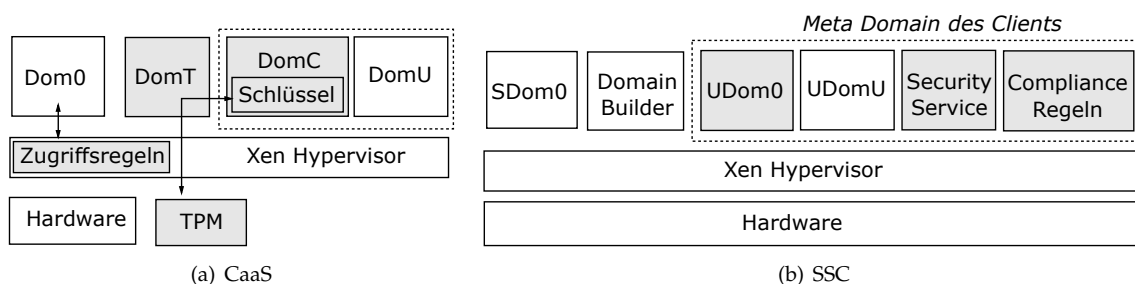


Abbildung 6.5.: Gegenüberstellung: CaaS - Cryptography-as-a-Service aus Bleikertz et al. [BBI⁺13] und SSC - Self-Service Cloud Computing aus Butt et al. [BLCSG12]

Jeder Client hat seine eigene Domain, in der die sicherheitskritischen Operationen durchgeführt werden und die vor externer als auch interner Einsicht geschützt ist. Die Cryptography-as-a-Service [BBI⁺13] Lösung fasst dies unter der Domain mit dem Namen *DomC* zusammen. Der Self-Service-Cloud Computing Ansatz [BLCSG12] teilt diesen in weitere Teile auf. Diese sind in Abbildung 6.5(b) hervorgehoben. Eine weitere Gemeinsamkeit der Konzepte ist die Ausgliederung von VM Verwaltungsaufgaben aus der privilegierten Domain *Dom0* in *DomT* bzw. *Domain Builder* analog zu Murray et al. [MMH08]. Gemeinsames Ziel beider Lösungen ist es, den Self-Service Grundsatz der Cloud auf sicherheitsrelevante Bereiche auszuweiten und damit eine höhere Sicherheit für IaaS-Plattformen bieten zu können. Beide setzen auf die Unterstützung von Vertrauensankern in Form von TPM und der Erweiterung des Xen Hypervisors, mit der Absicht sicheres VM Management, die Integrität der VMs und sensible Nutzerdaten zu schützen. Die Voraussetzung beider Lösungsansätze ist es, dass der CSP diese umsetzt. Anderenfalls sind die Ansätze nur bei Einrichtung und Betrieb einer privaten Cloud-Umgebung interessant.

Ran und Jin [RJ12] bieten eine vereinfachte Lösung für Trusted Virtualisierung basieren auf Open-Nebula und Tahoe. Im Gegensatz zu CaaS und SSC basiert diese nicht auf der Lösung von Murray et al. [MMH08] sondern integriert vielmehr den Overshadow [CGL⁺08] Ansatz. Aus diesem Grund ist dieser Lösungsansatz vor allem für private Cloud-Lösungen interessant, welche Ressourcen nicht unbekannten Externen zur Verfügung stellen soll und dennoch nicht auf eine vertrauenswürdige Umgebung verzichtet.

Catuogno et al. [CDE⁺09] und Berger et al. [BCG⁺09] beschreiben ihren Ansatz *Trusted Virtual Domain* (TVD) als vielversprechendes Konzept zur sicheren Verwaltung von Virtualisierungsplattformen. Eine TVD ist dabei ein Verbund von VMs, welche sich aufgrund einer gemeinsamen Sicherheits-Policy vertrauen und über verschiedene physikalische Systeme verteilt sind. Hauptziel ist auch hier die Kontrolle des Nutzers über Daten und Prozesse zu erhöhen. Dies geht jedoch einen Schritt über die Trusted Virtualization hinaus, in dem diese die Verwaltung der CaaS und SSC Ansätze auf ein gesamtes Datenzentrum erweitern. Das Ziel ist eine bessere Isolierung um die Mehrmandantenfähigkeit, beim Hinzufügen und Entfernen von VMs aus der TVD, zu gewährleisten.

6.2.2. Geschachtelte Virtualisierung

Der Grundstein für geschachtelte (auch rekursive) Virtualisierung legte das Turtles-Project von Ben-Yehuda et al. [BYDD⁺10]. Die Besonderheit dieser Virtualisierung ist der Hypervisor als Gast-System. Üblicherweise werden Hypervisoren (speziell Typ 1) unter der Annahme entwickelt, dass diese di-

rekt auf der Hardware ausgeführt werden. Im Falle des Hypervisors im Gast-System ist dies nicht gegeben. Das Verhalten des Gast-Hypervisors ist demnach laut Ben-Yehuda et al. der maßgebende Faktor. Entscheidend für die Performance ist die Anzahl der weitergereichten Exit-Befehle, die auf Grundlage des *trap-and-emulate* Prinzips von Popek [PG74] vom Gast Hypervisor von dessen VM abgefangen werden, zu minimieren. Alternativ schlägt Ben-Yehuda et al. [BYDD⁺10] vor, den Gast-Hypervisor zu paravirtualisieren. Inwiefern dies jedoch die Performance verbessert, hinge stark von Aufgabe und der spezifischen Last ab.

Mit CloudVisor stellen Zhang et al. [ZCCZ11] einen Ansatz vor, die geschachtelte Virtualisierung zu nutzen, um den Sicherheitskomponenten und die Ressourcenverwaltung zu trennen. Analog zu Murray et al. [MMH08] entfernt auch CloudVisor das VM Management und den Hypervisor aus der TCB, um diese zu minimieren. CloudVisor stellt einen schmalen Security Monitor unterhalb des Hypervisors dar, die Abbildung 6.6(a) illustriert die Funktionsweise.

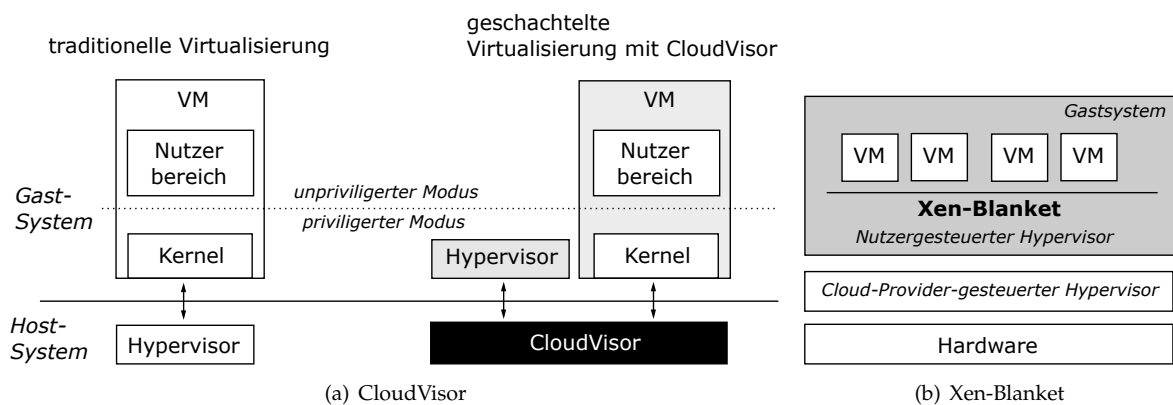


Abbildung 6.6.: CloudVisor Architektur aus Zhang et.al. [ZCCZ11] und Xen-Blanket Architektur aus Williams et al. [WJW12]

Erkennbar ist, dass CloudVisor den Hypervisor (in Abb.6.6(a) VMM) in die Gastrolle verschiebt und damit vom direkten Zugriff auf die Hardware isoliert. Der Ansatz greift direkt in die Speicherverwaltung der VM ein und schützt so den Zugriff seitens des Hypervisors. Weiterhin wird eine transparente Verschlüsselung der I/O Zugriffe auf die virtuellen Festplatten durch CloudVisor gewährleistet. Dieser Lösungsansatz ändert jedoch nichts am Vertrauen, das dem CSP gegenüber aufgebracht werden muss, da sowohl Hypervisor als auch CloudVisor unter der Kontrolle des CSP stehen. Dies impliziert zudem, dass der CSP diese System in seiner Virtualisierungslösung implementieren muss.

Ein nutzerzentrierter Ansatz ist Xen-Blanket, beschrieben von Williams et al. [WJW12]. Dieser bietet eine vollständig geschachtelte Virtualisierung mit dem Fokus einer aus Nutzersicht homogenen Cloud-Umgebung. Im Unterschied zu CloudVisor liegt der Xen-Blanket Hypervisor nicht unter der bestehenden Virtualisierungsschicht, sondern baut auf dieser auf (Abbildung 6.6(b)). So bietet der Xen-Blanket eine vom Nutzer gesteuerte, eigene Virtualisierungsumgebung in der Umgebung des CSP. Damit bietet dieser Ansatz im Gegensatz zu CloudVisor keine Sicherheit im Sinne von Vertraulichkeit oder Integrität, sondern eine Möglichkeit dem Nutzer eine 100%ig transparente Umgebung zu ermöglichen. Dies stellt eine effektive Möglichkeit dar um einen Lock-in Effekt zu vermeiden. Der bei Verwendung entstehende Overhead wird von den Autoren von 3% bei einfachen Prozessoperationen bis zu 70% für komplexe, nebenläufigen Prozesse und Dateierzeugung angegeben. Gleichzeitig wird jedoch darauf hingewiesen, dass durch die eigene Virtualisierungsumgebung bis zu 52% der Kosten gespart werden können, wenn 40 VMs in einer großen VM des CSP virtuali-

siert würden, anstatt 40 kleine Instanzen des CSP zu nutzen. Diese Tatsache ist vor dem Hintergrund nachvollziehbar, dass die Kosten für virtuelle Ressourcen nicht linear mit der Zunahme der Leistung steigen. Dennoch ist dieses von den Autoren angebrachte Beispiel theoretischer Natur, da hier praktisch relevante Kriterien wie Ausfallsicherheit und Redundanz nicht berücksichtigt wurden.

Beide hier beschriebene Ansätze bieten individuelle Lösungen, wünschenswert wäre jedoch eine Kombination aus Beiden. Wie Butt et al. [BLCSG12] hinweisen, sei es theoretisch möglich beide Ansätze zu vereinen. Es ist jedoch fraglich, ob der Aufwand dieser tief geschachtelten VMs und Hypervisoren den Nutzen rechtfertigt. Betrachtete man den ursprünglichen Zweck der Auslagerung in eine Public Cloud aus Kosten- und Effizienzgründen, ist leicht zu erkennen, dass diese Vorteile an der Stelle durch den vielfach erhöhten Aufwand vollständig kompensiert würden.

6.3. Zusammenfassung

Das Kapitel beschreibt Maßnahmen, die zum Schutz der Virtualisierungsschicht innerhalb von Cloud-Umgebungen getroffen werden können. Die Autoren Butt et al. [BLCSG12] fassen die Hauptprobleme der sicheren Virtualisierung in zwei Punkten zusammen: die Datensicherheit und Datenschutz von Client VMs und die unflexible Steuerung/Verwaltung der Client VMs. Da dies im Wesentlichen von der Sicherheit und Integrität des Hypervisoren abhängt, ist diese das zentrale Ziel der Schutzmaßnahmen. Der Fokus ist dabei oft die TCB des Hypervisors minimal zu halten, um dessen Vertrauenswürdigkeit zu erhöhen, oder, im Optimalfall, formal zu beweisen.

Der wesentlicher Vorteil einer sicheren Virtualisierungsumgebung ist, dass die Applikation häufig unverändert bleiben kann. Die Voraussetzung ist jedoch, dass der CSP diese Virtualisierungslösungen bereits anbietet, da der Nutzer keinen Einfluss auf die Virtualisierungslösung des CSP nehmen kann. Besprochene Ansätze wie geschachtelte Virtualisierung können diesen Umstand zwar teils umgehen, jedoch ist die Praxistauglichkeit derartiger Lösungen in Frage zu stellen.

Ein verbreiteter Ansatz ist daher die Virtualisierungsumgebung mit Hilfe von Trusted Computing Ansätzen vertrauenswürdiger zu machen. Dieser als Trusted Virtualization bezeichneter Ansatz ist Gegenstand der aktuellen Forschung, da auch hier wie die gleichen Problemstellungen auftreten wie im Kapitel 5 Trusted Computing beschrieben. Die Angebote von CSP, wie die GovCloud von Amazon⁶, zeigen jedoch die praktische Relevanz dieses Forschungsbereiches.

Dieses Kapitel unterstützt zudem die Annahme der These 2. Da die Lösungen in der Regel einen Einfluss auf niedriger Abstraktionsebene nehmen und die Schutzmaßnahmen zu etablieren, ist deren resultierender Mehraufwand sehr gering. Die Umsetzung der Maßnahmen ist zudem als effizient zu bewerten. Voraussetzung ist jedoch der Zugriff auf die Hardware bzw. Virtualisierungsebene, die dem Betreiber der Cloud-Umgebung vorbehalten ist. Wenn der Nutzer zum Cloud-Betreiber werden muss um diese Lösungsansätze zu nutzen, stellt sich die Frage, vor wem die Informationen, die ausschließlich der Betreiber einsehen kann, geschützt werden sollen. Nutzen weitere Teilnehmer die Cloud-Umgebung, ist die Frage jedoch obsolet.

⁶<http://aws.amazon.com/de/govcloud-us/>, letzter Zugriff 04.09.2014

Teil III

Technologie-Evaluierung nach ausgewählten Kriterien

Technologie Evaluierung

Die Evaluierung der untersuchten Technologien, Forschungs- und Lösungsansätze erfolgt nach einer Anzahl von Kriterien, die in diesem Kapitel eingeführt und definiert werden. Das Ziel der Evaluierung ist es, die verschiedenen, untersuchten Ansätze miteinander vergleichbar zu machen und auf dieser Grundlage eine Schätzung ihrer Praktikabilität zu ermöglichen. Eine Kategorisierung der evaluierten Technologien erfolgt gemäß der eingeführten Taxonomie 2.3 und entsprechend des Aufbaus des vorhergehenden Teils der Arbeit. Insgesamt wurden 96 Forschungsansätze evaluiert.

Im Kontext dieser Arbeit wird der potenzielle, praktische Nutzen $\mathfrak{N}$ definiert, der die Zusammenführung ausgewählter, bewerteter Evaluierungskriterien sowie die zentrale Vergleichsgröße darstellt. Demnach hängt der Nutzen $\mathfrak{N}$ von den drei Größen *Leistungsfähigkeit* L , *Funktionalität* F und *Sicherheit* S ab. Das Volumen des in Abbildung 7.1(a) dargestellten Tetraeders, illustriert den potenziellen Nutzen der entsteht, wenn die untersuchte Technologie die Werte der voneinander abhängen Größen l,s und f besitzt.

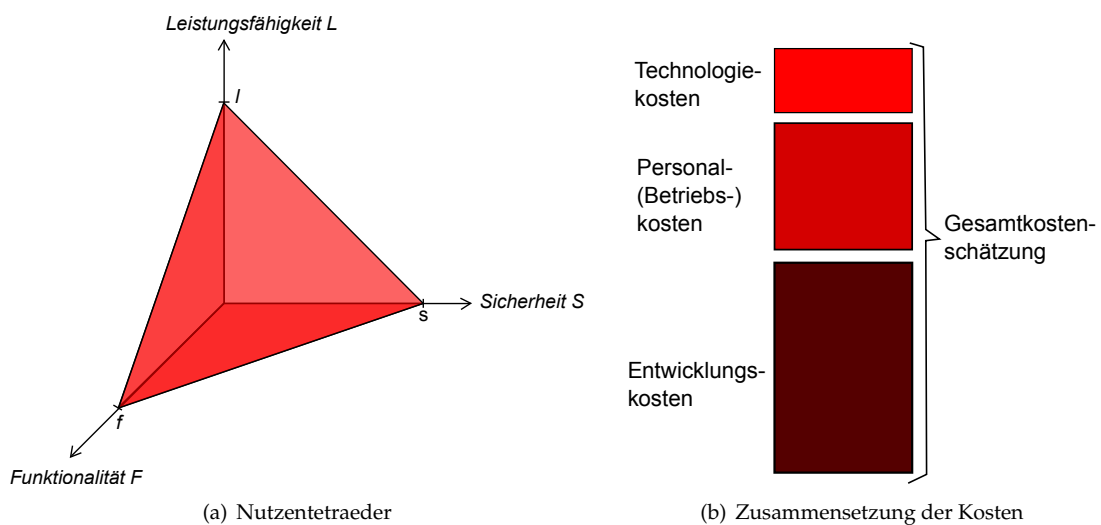


Abbildung 7.1.: Gegenüberstellung von Nutzen- und Kostenzusammensetzung

Dem Nutzen gegenüber stehen die Kosten K die mit dem Einsatz der evaluierten Technologie entstehen. Dazu werden, wie Abbildung 7.1(b) veranschaulicht, die Technologien anhand von Kostenschlüsseln nach ihren entstehenden Kosten für Technologie, Personal und Entwicklung bewertet. Das Verhältnis von Kosten- und Nutzen- Schätzung bildet die Grundlage der in der Arbeit berechneten Praktikabilitätsaussage $\mathfrak{P}$, wobei die folgenden Abschnitte detaillierte Information zur Berechnung des Nutzens $\mathfrak{N}$, der Praktikabilitätsaussage $\mathfrak{P}$, zur Zusammensetzung der Größen P, F und S und zu einzelnen Evaluierungskriterien enthalten. Eine kompakte Form der Ergebnisse der Evaluierung wird in den Tabellen dieses Kapitels im Abschnitt 7.2 dargestellt. und eine vollständige Auflistung aller Evaluierungsergebnisse befindet sich im Anhang. Die Diskussion der erhaltenen Ergebnisse in Bezug auf die in der Arbeit aufgestellten Thesen bildet den Abschluss dieses Kapitels.

7.1. Evaluierungskriterien

In diesem Abschnitt werden die Zusammensetzung, Bewertung und Berechnung des Nutzens $\mathfrak{N}$, der Kosten K und der Praktikabilitätsaussage $\mathfrak{P}$ beschrieben.

7.1.1. Potenzieller Nutzen $\mathfrak{N}$

Der potenzielle praktische Nutzen ergibt sich aus den, im weiteren Verlauf des Abschnitts vorgestellten, Evaluierungskriterien und stellt, neben den Kosten K , die Grundlage für die Vergleichbarkeit der untersuchten Technologien dar. Die Berechnung von $\mathfrak{N}$ erfolgt durch Kalkulation des Rauminhalts des durch die drei Vektoren L , S und F , aufgespannten Tetraeders im $\mathbb{R}_3$. Die Begründung für die Wahl des Rauminhalts ergibt sich anhand der Abhängigkeit der Größen untereinander. Dementsprechend sollen sich zwei Technologien t_1 , t_2 mit gleichen Werten ($S_{t_1} = S_{t_2}, F_{t_1} = F_{t_2}$) mehr voneinander unterscheiden als nur um die Differenz der beiden Werte P_{t_1}, P_{t_2} . Die Technologie, die bei gleicher Sicherheit und Funktionalität eine bessere Performance bietet, soll einen deutlich größeren potenziellen Nutzen aufweisen. Diese Tatsache soll sich schlussendlich auf die Vergleichbarkeit verschiedener Technologieansätze auswirken. Die Berechnung des Volumens erfolgt unter Zuhilfenahme der Vektorrechnung mittels der Gleichung 7.1.

$$\begin{aligned}
 \mathfrak{N} &= \frac{1}{6} \left| \det(\vec{a}, \vec{b}, \vec{c}) \right| \\
 &= \frac{1}{6} \left| (\vec{L} \times \vec{S}) \cdot \vec{F} \right| \\
 &= \frac{1}{6} \left| \left(\begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix} \times \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} \right) \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \right| \\
 &= \frac{1}{6} \left| \begin{bmatrix} L_y \cdot S_z - L_z \cdot S_y \\ L_z \cdot S_x - L_x \cdot S_z \\ L_x \cdot S_y - L_y \cdot S_x \end{bmatrix} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \right| \\
 &= \frac{1}{6} \left| \begin{bmatrix} 0 - 0 \cdot 0 \\ 0 \cdot S - 0 \cdot 0 \\ S - 0 \cdot S \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \right| \\
 &= \frac{1}{6} \left| \begin{bmatrix} 0 \\ 0 \\ L \cdot S \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \right| \\
 &= \frac{1}{6} | (0 \cdot 0 + 0 \cdot 0 + (L \cdot S) \cdot F) | \\
 &= \frac{1}{6} | ((L \cdot S) \cdot F) | \\
 &= \frac{L \cdot S \cdot F}{6}
 \end{aligned} \tag{7.1}$$

Die Berechnung des potenziellen Nutzens erfolgte für alle untersuchten Technologien nach dieser Form. Die berechneten Ergebnisse sind in den Tabellen im Abschnitt 7.2 aufgeführt. Neben den Berechnungswerten sind im Anhang weitere Informationen und Anmerkungen enthalten. Die Bedeutung von $\mathfrak{N}$ wird, nach Einführung aller Evaluierungskriterien, anhand des Encryption Layer von Reinhold et al. [RBK⁺14a] im folgenden Abschnitt erläutert.

Bedeutung des potenziellen Nutzens $\mathfrak{N}$

Die Bedeutung des potenziellen Nutzens soll anhand der evaluierten Technologie von Reinhold et. al. [RBK⁺14a] verdeutlicht werden, wobei die einzelnen Evaluierungskriterien in analoger Reihenfolge der nachfolgenden Abschnitte diskutiert werden und damit ihrer ausführlichen Erläuterung vorgreifen. Die Formulierungen sind daher allgemeiner und verständlicher gehalten, als dies nach Einführung der Kriterien notwendig wäre. In Klammern stehen jeweils die Kürzel der Kriterien, wie diese in den folgenden Abschnitten eingeführt werden.

Das entwickelte Verfahren ist effizient und der Mehraufwand wurde im Vergleich zu einer ungeschützten Lösung bei einer Angabe von 75% mit mittel bewertet ($MA = 2$). Dementsprechend ergibt sich nach der Gleichung für die Leistungsfähigkeit $L = 1.71$. Die Bewertung der Sicherheit erfolgt nach sechs Kriterien. Die Vertraulichkeit ist durch eine Verschlüsselung gegenüber dem CSP und Dritten gegeben ($VT = 1$). Es wurden in der Arbeit keine Maßnahmen getroffen, um die Integrität von Daten in der Cloud-Umgebung speziell zu schützen oder zu überwachen ($IG = 0$). Zudem wurden keine Möglichkeiten des CSP genutzt um die Redundanz, und damit die Verfügbarkeit der gespeicherten Daten zu erhöhen ($VF = 0$). Ebenso wenig wurden Maßnahmen getroffen, um eine Unverknüfbarkeit zu realisieren ($UV = 0$). Für Transparenz und Intervenierbarkeit wurde hingegen durch das verwendete Schlüsselmanagement-System gesorgt ($TR, IN = 1$). Daraus ergibt sich nach Gleichung für die Berechnung der Sicherheit $S = 4.25$. Die Bewertung der Funktionalität erfolgt nach fünf Kriterien. Wie bereits hervorgehoben, wird kein Schlüsselmanagement des CSP benutzt, welcher die Daten speichert. Die Schlüssel liegen aber auch nicht beim Nutzer, sondern beim Cloud-Vermittler, der im Encryption Layer als Entwickler und Betreiber der SaaS-Lösung fungiert. Ein Schlüsselmanagement ist demnach integriert ($SM = 1$). Weiterhin legt der Encryption Layer besonderen Wert auf seine Skalierbarkeit, so dass in diesem Zusammenhang wurde eine Vielzahl von Benchmarks und Test unternommen wurden ($SK = 1$). Anforderungen zum Teilen von Daten, für Back-Ups oder Hochverfügbarkeit gab es im Rahmen der Forschungsarbeit nicht und wurden entsprechend nicht umgesetzt ($SH, BA, HA = 0$). Für die Funktionalitätsbewertung ergibt sich nach der Gleichung zur Berechnung der Funktionalität ein Wert von $F = 2.6$.

Konsequenterweise ergibt sich für den potenziellen, praktischen Nutzen $\mathfrak{N}$ nach obig eingeführter Berechnungsvorschrift ein Wert von $\mathfrak{N}_{EL} = 3.14$, der im Verhältnis mit der Abschätzung der Kosten K für die Verwendung des EL zu eine Aussage über dessen Praktikabilität $\mathfrak{P}_{EL}$ führt. Die Kosten des EL wurden wie folgt abgeschätzt: Technologisch setzt der Lösungsansatz OpenSource-Softwarekomponenten und Standardhardware zum Aufbau einer privaten Cloud-Lösung ($K_{tech} = 2$). Die aufgebaute private Cloud-Umgebung benötigt administrative Eingriffe und erfordert eine regelmäßige Wartung ($K_{Pers} = 2$). Der Entwicklungsaufwand für das Etablieren des Ansatzes wird als mittel abgeschätzt ($K_{Entw} = 2$). Für die Kosten K ergibt sich damit durch die Bildung der Summe ein Wert von $K = 6$.

Es gilt daher $\mathfrak{N} < K$, entsprechend ergibt sich $0 < \mathfrak{P} < 1$ und damit die Bewertung *eingeschränkt praktikabel*. Dies ist nach Ansichten des Autoren die korrekte Abschätzung, da bezüglich des Schlüsselmanagement und der redundanten Datenspeicherung weitere Entwicklungsarbeit notwendig ist, um das System produktiv zu betreiben.

7.1.2. Kosten K

Die Tabelle 7.1 zeigt die Zusammensetzung des Evaluierungsfelds der Kosten K , wobei die in der Tabelle aufgeführten Kosten der Darstellung in Abbildung 7.1(b) entsprechen.

Tabelle 7.1.: Darstellung zur Zusammensetzung der Kosten K

Evaluierungsfeld	Kosten K					
Bezeichnung	Technologie		Personal/Betrieb		Entwicklung	
Kriterium	Details	Schlüssel K_{Tech}	Details	Schlüssel K_{Pers}	Details	Schlüssel K_{Entw}

Die konkreten Kriterien unterteilen sich in drei Aspekte: *Technologie*, *Personal* und *Entwicklung*. Neben der jeweiligen Beschreibung werden Kostenschlüssel festgelegt, deren Bewertungsmaßstab in den folgenden Tabellen erläutert wird. Das Ziel dieser Kostenabschätzung ist keine detaillierte Kostenanalyse oder Risikobewertung, sondern eine einheitliche Bewertung, um die zu evaluierenden Technologien kostenseitig miteinander vergleichbar zu machen. Die Unterscheidung in die folgenden Kriterien ist naheliegend, da diese Informationen häufig in den Literaturquellen enthalten sind und damit eine Bewertung ermöglichen.

Die Tabelle 7.2 zeigt die Kostenschlüssel für die Schätzung der Technologiekosten beim Einsatz der untersuchten Technologie. Die Tabelle 7.3 zeigt die Kostenschlüssel für die Schätzung der Perso-

Tabelle 7.2.: Darstellung des Bewertungsmaßstabs für den Schlüssel K_{Tech}

Schlüssel	Bezeichnung	Beschreibung
1	gering	Verwendung von Open Source Technologie
2	mittel	Verwendung einiger proprietärer Software, (günstige) Standard Hardware
4	hoch	Verwendung von proprietärer Software/Hardware, spezielle teure Hardware

nalkosten beim Einsatz der untersuchten Technologie. Vergleichbar ist dies mit der Einteilung der Cloud Deployment Modelle: public, hybrid und private.

Tabelle 7.3.: Darstellung des Bewertungsmaßstabs für den Schlüssel K_{Pers}

Schlüssel	Bezeichnung	Beschreibung
0	keine	keine Aufgaben für den Betrieb
1	gering	Administration (Monitoring)
2	mittel	Administration und Wartung
4	hoch	Administration, Wartung, Verwaltung, Support

Die Tabelle 7.4 zeigt die Kostenschlüssel für die Schätzung der Entwicklungskosten. Dabei wird von geringen (Integration) bis zu sehr hohen Kosten (Neuentwicklung) unterschieden.

Tabelle 7.4.: Die Darstellung des Bewertungsmaßstabs für den Schlüssel K_{Entw} .

Schlüssel	Bezeichnung	Beschreibung
1	gering	Anpassung weniger LoC, Einbindung weniger Bibliotheken
2	mittel	Anpassung weniger Module, Einbindung von Bibliotheken
4	hoch	Anpassung/Austausch von Modulen, Systemteilen, einige Neuimplementierungen
8	sehr hoch	Implementierung neuer Module, Systemteile oder vollständige Neuimplementierung

Die Gesamtkostenabschätzung resultiert aus der Berechnungsvorschrift 7.2.

$$K = K_{Tech} + K_{Pers} + K_{Entw} \quad (7.2)$$

Der Gesamtwert K ergibt sich zu gleichen Teilen aus den Kostenschlüsseln der Einzelschätzungen. Auf Basis der Gleichung 7.2 ergibt sich ein Maximalwert $K_{Max} = 16$, nachdem die Normalisierung des potenziellen Nutzens $\mathfrak{N}$ durchgeführt wird. Dies ist notwendig, um eine korrekte Aussage der Praktikabilitätsaussage zu erlagen, welche das Nutzen-Kosten-Verhältnis darstellt. Es gilt die Normalisierungsgrundlage: $\mathfrak{N}_{Max} = K_{Max} = 16$. Demnach ergibt sich:

$$\begin{aligned}\mathfrak{N}_{Max} &= \frac{L_{Max} \cdot S_{Max} \cdot F_{Max}}{6} \\ 16 &= \frac{L_{Max} \cdot S_{Max} \cdot F_{Max}}{6} \\ 96 &= L_{Max} \cdot S_{Max} \cdot F_{Max}\end{aligned}\tag{7.3}$$

Da den Bestandteile von $\mathfrak{N}$ im Zuge der Untersuchung dieser Arbeit eine unterschiedliche Wichtung zukommen soll, wird folgendes festgelegt:

$$L_{Max} = 2.56\tag{7.4}$$

$$S_{Max} = 7.5\tag{7.5}$$

$$F_{Max} = 5.0\tag{7.6}$$

Nach Gleichung 7.3 wird damit der obigen Normalisierungsgrundlage entsprochen. Die Entwicklung der Berechnungsvorschriften für die Größen L , S und F beziehen, mit entsprechendem Verweis, die obige Normalisierungsvorschrift mit ein.

7.1.3. Praktikabilitätsaussage $\mathfrak{P}$

Die Praktikabilitätsaussage stellt das Nutzen-Kosten-Verhältnis dar und wird nach der Gleichung 7.7 berechnet. Dabei ergibt sich K nach der Gleichung 7.2 und $\mathfrak{N}$ nach Gleichung 7.1.

$$\text{Praktikabilitätsaussage } \mathfrak{P} = \frac{\text{Nutzen } \mathfrak{N}}{\text{Kosten } K} \quad (7.7)$$

Tabelle 7.5.: Die Praktikabilitätskategorien werden in die dargestellten Kategorien unterteilt.

Kategorie	Beschreibung	Bedingung	Symbol
<i>potenziell praktikabel</i>	Die untersuchte Technologie ist potenziell für einen Einsatz im praktischen Umfeld geeignet	$\mathfrak{P} \geq 1$	+
<i>eingeschränkt praktikabel</i>	Die untersuchte Technologie ist eingeschränkt für einen Einsatz im praktischen Umfeld geeignet	$0 < \mathfrak{P} < 1$	○
<i>unpraktikabel</i>	Die untersuchte Technologie ist nicht für einen Einsatz im praktischen Umfeld geeignet	$\mathfrak{P} = 0$	–
<i>nicht bestimmbar</i>	Durch fehlende Informationen kann keine Aussage zur Praktikabilität getroffen werden	–	□

Die Praktikabilitätsaussage entspricht dabei der jeweiligen Kategorie. Die Symbole finden zur kompakten Darstellung der Praktikabilitätsaussage in den Tabellen im Abschnitt 7.2 Anwendung. Die Bedeutung der Symbolik orientiert sich am Nutzen-Kosten-Verhältnis. Nur Technologien mit einem überwiegenden Nutzen sollen in die Kategorie *potenziell praktikabel* eingeordnet werden. Ist der Nutzen daher mindestens so groß wie die Kosten ($\mathfrak{P} \geq 1$) ist die untersuchte Technologie dieser Kategorie zugeordnet. Ist das Verhältnis weniger stark zu Gunsten des Nutzens ausgeprägt ($0 < \mathfrak{P} < 1$), wird diese als *eingeschränkt praktikabel* eingeschätzt. Als *unpraktikabel* wird eine Lösung bezeichnet wenn deren potenzieller Nutzen $\mathfrak{N}$ gleich null ist. Dies ist insbesondere der Fall, wenn die Lösung nicht effizient arbeitet, die Leistungsfähigkeit L null ist, wodurch $\mathfrak{N}$ ebenfalls null wird.

Die Bedeutung der Praktikabilitätsaussage orientiert sich an den Beschreibungen in Tabelle 7.5. Die *potenziell praktikabel*(+) Verfahren stellen Forschungs- und Entwicklungsarbeiten dar, die bereits einen fortgeschrittenen Stand haben und durch geringe Entwicklungs- und Integrationsarbeit in Systeme in der Praxis eingesetzt werden können oder bereits werden.

Als *eingeschränkt praktikabel*(○) werden Technologieansätze bezeichnet, die zwar prinzipiell effizient arbeiten, jedoch weitere Forschungs- oder Entwicklungsarbeit erfordern. Zu diesen Ansätzen gehört die von Reinhold et. al. [RBK⁺14a] entwickelte Lösung des Encryption Layers. Der Prototyp hat seine praktische Verwendbarkeit gezeigt, dennoch ist für die Integration in produktive Systeme weitere Entwicklungsarbeit nötig.

Als *unpraktikabel*(–) werden demnach Verfahren bezeichnet, die nicht im praktischen Umfeld einsetzbar sind. Beispiele für derartige Technologien sind voll homomorphe Verschlüsselungsverfahren. Diese sind nach heutigem Forschungsstand verfahrensbedingt in der Praxis nicht oder nur mit starken Einschränkungen einsetzbar.

Die letzte Kategorie bilden die *nicht bestimmbar*(□) Technologie Ansätze. Hierbei handelt es sich entweder um theoretische Konzepte oder Publikationen einer frühen Forschungsphase. Jedoch existieren auch Arbeiten, wie beispielsweise TwinCloud [BSSS11] oder SEDIC [ZZCW11], die relevant für die Untersuchungen dieser Arbeit sind, jedoch keine ausreichenden Aussagen über praktische Leistungsfähigkeit oder Mehraufwand enthalten.

7.1.4. Leistungsfähigkeit L

Die Leistungsfähigkeit L ist eine der Kriterien, welche in die Bewertung des Nutzens $\mathfrak{N}$ einfließen. Das Evaluierungsfeld *Prototyp* wurde für alle untersuchten Technologien bewertet, fließt sind die Berechnung der Praxistauglichkeit jedoch nicht mit ein. Die Tabelle 7.6 zeigt die Evaluierungsfelder *Leistungsfähigkeit* und *Prototyp*.

Tabelle 7.6.: Zusammensetzung der Evaluierungsfelder *Leistungsfähigkeit* und *Prototyp*

Evaluierungsfeld	Leistungsfähigkeit			Prototyp		
Bezeichnung		Mehraufwand				
Kriterium	effizient? EF	Schätzung MA	Details	existiert? PT	eingesetzte Technologien	Open Source?

Das Kriterium *effizient?* EF von L bewertet, ob das Verfahrens praktisch effizient ist und kann die Werte 0 und 1 annehmen. Die Bedeutung sei an Arbeit von Gentry [Gen09] verdeutlicht: Die voll homomorphe Verschlüsselung bieten eine potenziell große Bedeutung für sicheres Cloud Computing. Dennoch ist diese nur von theoretischer Natur, da selbst für einfache Berechnungen ein enormer Brechungsaufwand nötig ist. Aus diesem Grund wird dieser Lösungsansatz als nicht praktisch effizient bewertet und erhält den Wert $EF = 0$.

Das Kriterium *Schätzung* MA bewertet nach Tabelle 7.7 den Mehraufwand gegenüber der unverschlüsselten/unsicheren Lösung, des Verfahrens. Die *Details* verweisen auf zusätzliche Informationen, die eventuell zum Mehraufwand vorliegen. Vorrangig sind dies die konkreten Angaben der Autoren aus deren Arbeiten.

Tabelle 7.7.: Darstellung des Bewertungsmaßstabs für den Mehraufwand im Kriterium *Schätzung* MA

Schlüssel	Bewertung	Bemerkung
1	niedrig	geringer Mehraufwand, ca. <50%
2	mittel	mäßiger Mehraufwand, höchstens der doppelte Aufwand
3	hoch	mehr als der doppelte Aufwand
4	unbekannt	keine Abschätzung möglich

Die Vergabe der Schlüssel erfolgt nach der Einschätzung des Autoren dieser Arbeit, mit Unterstützung der Angabe in der entsprechenden Literaturquelle, um deren Objektivität zu erhöhen. Die Bewertung sei an folgenden Beispielen verdeutlicht: Die Autoren Popa et al. [PRZB11] geben für Lösung CryptDB einen Mehraufwand von 26% bzgl. des Daten-Durchsatzes an. Gemessen am technologischen Hintergrund der Lösung ist dies, nach Meinung des Autoren, ein geringer Wert. In diesem Zusammenhang seien weitere Beispiele mit der Bewertung *niedrig* ($MA = 1$) aufgelistet, wobei Aussage der Autoren aus der Literaturquelle jeweils in Klammern steht.

- Secure Cloud Maintenance [BKNS12](does not have significant impact on the efficiency of the infrastructure cloud)
- Proxos [TMLL06] (the overhead Proxos introduces is very low (6%))

Die vom Autoren entwickelte Lösung Encryption Layer [RBK⁺14b], mit einem angegebenen Mehraufwand von 50% -75% wird mit dem Schlüssel *mittel* bewertet. Auf Grund der Eigenentwicklung und gemessen an anderen, vergleichbaren Lösungen, ist dies nach Meinung des Autoren angemessen. Nachfolgend werden weitere Beispiele mit der Bewertung *mittel* ($MA = 2$) aufgelistet, wobei mit der Aussage der Literatur erneut in Klammern steht:

- Knox [WJCN09] (*overhead do not depend on number of users ... 100 users and 2GB data [means] 0,33GB signature size, 106,4KB communication costs and 3,4 sec duration for audit*)
- CloudVisor [ZCCZ11] (*moderate slow-down (4.5% - 54.5%) for I/O intensive applications*)

Im Folgenden sind Lösungen aufgelistet die eine hohe Mehraufwandsschätzung $MA = 3$ bekommen haben:

- Tsujii [TDF⁺13] (*calculation of 100 data sets lasts for 20 min*)
- Oblistore [SS13] (*40-50x i/o overhead*)

Sind keinerlei Hinweise auf einen Mehraufwand gegeben oder sind die Aussagen der Autoren zu vage, wurde die Bewertung *unbekannt* $MA = 4$ vergeben. Ein Beispiel für eine solche Bewertung ist:

- CryptoDSP [TPPG11] (*The performance of the preliminary system was encouraging*)

In der Berechnung der Praktikabilitätsaussage wird dieser unbekannte Wert berücksichtigt und bekommt den Status *nicht bestimmbar*. Die Berechnung von L erfolgt unter Berücksichtigung der Bedingung aus Gleichung 7.4 nach der folgenden Gleichung.

$$L = -\frac{L_{Max}}{MA_{Max} - 1} \cdot EF \cdot MA$$

mit $L_{Max} = 2.56$ und $MA_{Max} = 4$

$$L = -\frac{2.56}{3} EF \cdot MA \quad (7.8)$$

mit $EF \in \{0, 1\}$ und $MA \in \{1, 2, 3, 4\}$

Ist die untersuchte Technologie nicht effizient einsetzbar ($EF = 0$), wird auch die Leistungsfähigkeit des Gesamtsystems mit null ($L = 0$) bewertet. Die Abbildung 7.2 visualisiert obige Gleichung im gültigen Wertebereich.

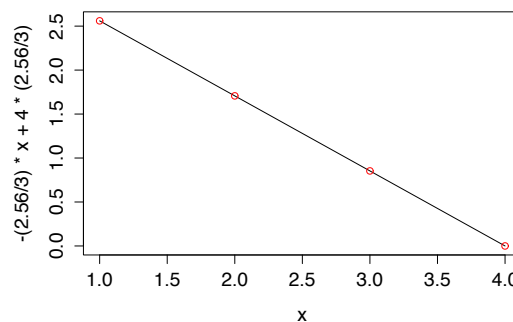


Abbildung 7.2.: Gleichung 7.8 und Hervorhebung der Funktionswerte (mit $EF = 1$)

7.1.5. Sicherheit S

Die Bewertung der *Sicherheit* in Tabelle 7.8 erfolgt nach den in Kapitel 2.4.1 detailliert erläuterten Schutzziele. Diese sind in Datensicherheits- und Datenschutzziele klassifizierbar. Die Bewertung erfolgt boolesch bzw. binär, d.h. nicht in welcher Form, sondern nur ob ein Kriterium vorhanden ist oder nicht.

Tabelle 7.8.: Zusammensetzung des Evaluierungsfelds *Sicherheit*

	Sicherheit					
Bezeichnung	Datensicherheit			Datenschutz		
Kriterium	Vertraulichkeit VT	Integrität IG	Verfügbarkeit VF	Unverknüfbarkeit UV	Transparenz TR	Intervenierbarkeit IN

Das Kriterium Vertraulichkeit VT bewertet die durch die eingesetzte Technologie vorhandene Vertraulichkeit der Daten bei der Speicherung. Üblicherweise wird diese durch die Verschlüsselung der Daten gewährleistet. Das Kriterium bewertet nicht die eingesetzten Verschlüsselungsverfahren, sondern nur ob die Vertraulichkeit der Daten gewährleistet ist oder nicht. Analog erfolgt die Bewertung des Kriteriums Integrität IG.

Das Kriterium Verfügbarkeit VF bewertet die Verfügbarkeit der Daten oder des angebotenen Services. Dabei spielt es insbesondere eine Rolle, ob explizit Maßnahmen getroffen worden diese zu erhöhen. Nur in diesem Fall wird das Kriterium positiv mit $VF = 1$ bewertet. Anderenfalls könnte argumentiert werden, dass die Cloud-Plattform, in welcher der Ansatz betrieben wird, die Verfügbarkeit bereits erhöht. Die Fragestellung, welche das Kriterium VF bewertet, ist, ob die von der Cloud-Plattform angebotenen Möglichkeiten der Verfügbarkeitserhöhung aktiv durch den Ansatz genutzt werden.

Die Kriterien zum Datenschutz sind schwieriger zu bewerten, da in den Literaturquellen bis auf wenige Ausnahmen kein direkter Bezug auf diese genommen wird.

Das Kriterium Unverknüfbarkeit UV bewertet, in welcher Art und Weise Informationen an den CSP selbst und an die Sub-Provider gereicht werden. Insbesondere, ob es den Providern möglich ist, Nutzungsprofile zu erstellen. Dieses Schutzziel ist ein jüngeres Phänomen, welches durch die starke Vernetzung der CSP-seitigen Auslagerung von Services an Bedeutung gewonnen hat. Vor allem im Identitätsmanagement spielt dieses Kriterium eine wichtige Rolle, da hier der Nutzer großes Interesse an einer geregelten Weitergabe seiner Identifikationsmerkmale und Credentials hat.

Ein weiteres Kriterium des Datenschutzes ist die Transparenz TR. Diese bewertet die Informationspreisgabe Cloud-interner Prozesse und Verarbeitungsschritte in der Cloud-Umgebung durch die eingesetzte Technologie. Dieses Kriterium steht häufig im Widerspruch zu der abstrahierenden Servicebereitstellung in der Cloud, nach welcher der Cloud-Nutzer nicht mit technischen Details konfrontiert werden soll, da es für die zu erledigende Kernaufgabe irrelevant ist. Dieses Kriterium bewertet daher, ob die zu evaluierende Technologie die Transparenz (prinzipiell) unterstützt oder dieser widerspricht.

Das letzte Sicherheitskriterium Intervenierbarkeit IN bewertet, ob der Cloud-Nutzer jederzeit seine Rechte (z.B. auf Datensperrung oder Löschung) ausüben kann. Die Berechnung der Sicherheit S erfolgt unter der Berücksichtigung der Bedingung 7.5 nach der folgenden Gleichung 7.9.

$$\begin{aligned}
 S &= \frac{S_{Max} - a}{6} \cdot (VT + IG + VF + UV + TR + IN) + a \\
 &\quad \text{mit } S_{Max} = 7.5 \text{ und } a = 1 \\
 S &= \frac{6.5}{6} \cdot (VT + IG + VF + UV + TR + IN) + 1 \\
 &\quad \text{mit } VT, IG, VF, UV, TR, IN \in \{0, 1\}
 \end{aligned} \tag{7.9}$$

Die Abbildung 7.3 visualisiert die obige Gleichung im gültigen Wertebereich.

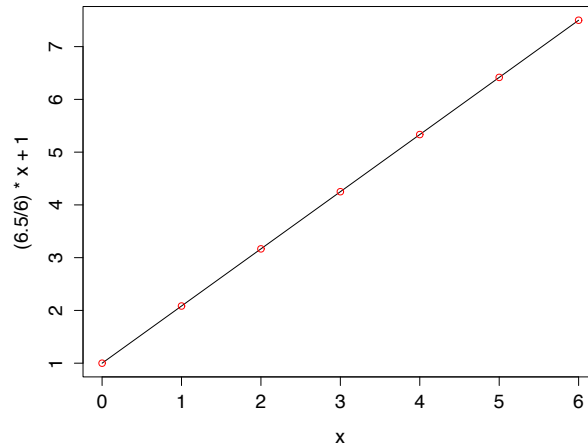


Abbildung 7.3.: Die Visualisierung der Gleichung 7.9 sowie die Hervorhebung der Funktionswerte.

Das Ziel der Gleichung 7.9 ist es die Werte im Wertebereich so zu verteilen, dass die folgenden Bedingungen gelten:

1. Der Minimalwert S_{Min} für $VT, IG, VF, UV, TR, IN \in \{0\}$ ist 1 ($a=1$).
2. Der Maximalwert S_{Max} erfüllt die Gleichung 7.5 ($S_{Max} = 7.5$).

Der Grund für die Bedingung 1 ist es, dass ein System, welches im Sinne der Untersuchung keine Sicherheitsfeatures bietet, dennoch praktikabel sein kann. Es wurden in dieser Arbeit keine derartigen Ansätze bewertet. Dennoch stellt dies einen wesentlichen Unterschied zu L dar. Die geforderte Normierung für die sinnvolle Aussagekraft der Praktikabilitätsaussage als Verhältnis von Nutzen und Kosten, ist der Grund für Bedingung 2.

7.1.6. Funktionalität F

Das Evaluierungsfeld *Funktionalität* beinhaltet die in Tabelle 7.9 aufgelisteten Kriterien, wobei die Art der Ein- und Ausgaben nicht in die Aussage der Praktikabilitätsbewertung einfließt, da ein Vergleich von Datenein- und -ausgaben zwischen verschiedenartigen Technologien wenig Wert hat.

Tabelle 7.9.: Evaluierungsfeld *Funktionalität*

	Funktionalität							
Bezeichnung	Daten-		Schlüsselmanagement					
Kriterium	Eingabe	Ausgabe	existiert? SM	Details	Teilen möglich? SH	Skalierung SK	Backup BA	Hoch- verfüg- barkeit HA

Der Aspekt des Schlüsselmanagement unterteilt sich in die Unterpunkte *existiert? SM* und die Beschreibung *Details* zu den Details, insofern die Schlüsselverwaltung nötig ist. In Letzteren finden sich Informationen, ob es sich um eine nutzer- oder providerseitige Lösung handelt sowie weitere technische Details, insofern angeben. Das Evaluierungskriterium *Teilen möglich? SH* bewertet, ob es mit den evaluierten Technologie möglich ist, Daten zu teilen (Sharing). Insbesondere bei verschlüsselten Daten ist dies eine interessante Fragestellung. Im Kriterium *Skalierung SK* wird bewertet, ob die evaluierte Technologie bzw. das damit zu entwickelnde System in der Lage ist, zu skalieren. Analog zur Bewertung der Verfügbarkeit geht es nicht um die Bewertung der potentiellen Möglichkeiten auf Grund der Cloud-Umgebung sondern ob die bereitgestellten Mittel der Skalierung aktiv genutzt wurden. Auch das Kriterium *Backup BA* nimmt eine solche Bewertung vor, jedoch im Bezug auf das Sichern der Daten in Form von Redundanz oder Backups. Das letzte Kriterium der Evaluierung der Funktionalität ist *Hochverfügbarkeit HA* welches bewertet, ob die betrachtete Technologie oder das damit zu konstruierende System den Anforderungen der Hochverfügbarkeit entspricht bzw. dies beabsichtigt. Die Berechnung der Funktionalität F erfolgt, unter Berücksichtigung der Bedingung 7.6, nach der Gleichung 7.10.

$$F = \frac{F_{Max} - a}{5} \cdot (VT + IG + VF + UV + TR + IN) + a$$

mit $F_{Max} = 5$ und $a = 1$

$$F = \frac{4}{5} \cdot (SM + SH + SK + BA + HA) + 1$$

mit $SM, SH, SK, BA, HA \in \{0, 1\}$

(7.10)

Die Abbildung 7.4 visualisiert die obige Gleichung im gültigen Wertebereich. Das Ziel der Gleichung 7.10 ist es die Werte im Wertebereich so zu verteilen, dass die folgenden Bedingungen gelten:

1. Der Minimalwert F_{Min} für $SM, SH, SK, BA, HA \in \{0\}$ ist 1 ($a=1$).
2. Der Maximalwert F_{Max} erfüllt die Gleichung 7.6.

Der Grund für die Bedingung 1 ist, dass das System, dass im Sinne der Untersuchung keine Funktionsfeatures bietet, dennoch praktikabel sein kann. Im Gegensatz zur Sicherheit S wurden einige Ansätze bei der Evaluierung derartig bewertet.¹ Die geforderte Normierung für die sinnvolle Aussagekraft der Praktikabilitätsaussage als Verhältnis von Nutzen und Kosten, ist der Grund für Bedingung 2.

¹Bsp: APNGS von Zhang et al. [ZZY⁺12], VMwatcher von Jiang et al. [JWX07], Proxos von Ta et al. [TMLL06]

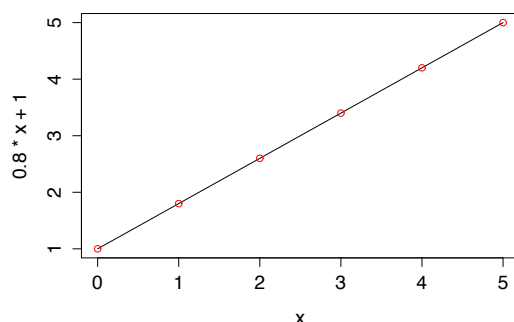


Abbildung 7.4.: Die Visualisierung der Gleichung 7.10 und Hervorhebung der Funktionswerte.

7.1.7. Kategorisierung und Realisierung

Die Kriterien zur Evaluierung Kategorisierung und Realisierung unterteilen sich in zwei Bereiche. Erster Bereich ist die Kategorisierung, welche sich in zwei Evaluierungsfelder aufteilt. Das erste Feld dient der Klassifizierung der untersuchten Technologien, das zweite Feld sind akademische Informationen. Beide Felder haben keinen Einfluss auf die Aussage zur Praktikabilität. Die Tabelle 7.10 zeigt die untersuchten Felder *Klassifikation* und *Akademische Informationen*.

Tabelle 7.10.: Die Zusammensetzung der Evaluierungsfelder *Klassifikation* und *Akademische Informationen*.

	Klassifikation			Akademische Informationen		
Bezeichnung	Technologie			Literatur		
Kriterium	Name, ID	Schlüsselwort, Taxonomie	Grundlagen	Referenz	Link	Anmerkungen

Das Kriterium Name, ID dient der Identifikation innerhalb des Gesamtdatensatzes. Der Inhalt des Evaluierungskriteriums Schlüsselwort, Taxonomie dient der Klassifikation, beinhaltet Schlagwörter aus Literaturquellen sowie die Einordnung in die im Kapitel 2.3 eingeführte Taxonomie². Das letzte Kriterium der Klassifikation Grundlagen beinhaltet grundlegende und verwendete Technologien² der bewerteten Technologieansätze. Die akademischen Informationen beinhalten neben der Literaturangabe und einer URL, unter welchen die Arbeit im Internet zu finden ist, Anmerkungen bzgl. der untersuchten Technologie. Diese können sich auf alle Evaluierungskriterien beziehen, welche keine Felder für detaillierte Informationen besitzen. Die letzten untersuchten Kriterien werden unter *Realisierung* zusammengefasst und haben keine Auswirkungen zur Praktikabilitätsaussage, sondern werden stattdessen für die Berechnung der Realisierungseffizienz im Abschnitt 7.3 verwendet. Untersucht wird zum einen, für welches Cloud Delivery Model die untersuchte Technologie geeignet ist, bzw. welches Modell fokussiert wird. Andererseits in welcher Schicht der Cloud-Applikation beim Einsatz des untersuchten Technologieansatzes Anpassungen vorgenommen werden müssen. Die Kriterien Private PR, IaaS IA, PaaS PA und SaaS SA bewerten die explizite Eignung für das entsprechende Cloud-Deployment- Modell³ und können die Werte 0 bzw. 1 annehmen. Das heißt, dass eine SaaS-Lösung, die auf einer IaaS-Basis fundiert, welche ihrerseits die untersuchte Technologie nutzt, verwendet SaaS-Lösung die Technologie implizit mit, entsprechend gilt $IA = 1, SA = 0$. Das in diesem Kriterium die explizite Verwendung/Eignung bewertet wird. Ferner wird im Kriteri-

²Auf diese wird vor allem im Teil 2 dieser Arbeit Bezug genommen.

³Die Details werden im Abschnitt 2.1 beschrieben.

Tabelle 7.11.: Die Zusammensetzung des Evaluierungsfeld *Realisierung*

Evaluierungsfeld	Realisierung						
Bezeichnung	Cloud Deployment Modell				Anpassungen in		
Kriterium	Private PR	IaaS IA	PaaS PA	SaaS SA	Daten- Layer DL	Business- Layer BL	Präsentation- Layer PL

um PR die Eignung für eine private Cloud-Lösung bewertet. Ist eine private Cloud-Lösung die Voraussetzung, wird dies in den Anmerkungen innerhalb der Akademischen Informationen verzeichnet. Die Kriterien zur Anpassung des Daten Layers (DL), des Business Layers (BL) bzw. des Präsentation Layers (PL) bewerten ob innerhalb der Schicht Veränderungen durch Einsatz der untersuchten Technologie vorgenommen werden müssen. Existieren spezielle Details zu den Anpassungen, ist dies ebenfalls in den Bemerkungen verzeichnet.

7.2. Evaluierungsergebnisse

Die folgenden Tabellen zeigen die Ergebnisse der Evaluierung. Die jeweilige Bezeichnung ist eine Kurzform oder der Name der untersuchten Arbeit sowie die Literaturquelle. Die Kürzel der Kriterien entsprechen denen der Einführung in Abschnitt 7.1. Die Darstellung erfolgt über die vollständigen Doppelseiten.

Sortiert sind die Tabellen aufsteigend nach dem Wert der Praktikabilität. Die Rangliste innerhalb der Praktikabilität ist von untergeordneter Rolle, dient hier ausschließlich der Sortierung und nicht einer Wertung der wissenschaftlichen Arbeit der entsprechenden Autoren. Die nachfolgenden Tabellen unterliegen ebenfalls der beschriebenen Sortierung.

Tabelle 7.13.: Ergebnisse der Evaluierung Cloud-Sicherheitsmanagement-Lösungen Teil 2

MA	L	S	F	$\mathfrak{N}$	K_{Tech}	K_{Pers}	K_{Entw}	K	$\mathfrak{P}$		Bezeichnung
1	2,56	6,42	1,80	4,93	1	0,5	1,5	3	1,64	+	Idemix [IRZ14]
1	2,56	6,42	2,60	7,12	2	0,5	2	4,5	1,58	+	U-Prove [MR14]
1	2,56	5,33	2,60	5,92	2	1	1	4	1,48	+	DIAMETER [Ven01]
1	2,56	4,25	2,60	4,71	2	0,5	1	3,5	1,35	+	Sibboleth
1	2,56	6,42	3,40	9,31	4	1	2	7	1,33	+	Dyn. VM Kerberos [LFB12]
2	1,71	5,33	2,60	3,94	1	0	2	3	1,31	+	SSF DAC [YWRL10]
1	2,56	6,42	2,60	7,12	1	1	4	6	1,19	+	PRAM [XYM ⁺ 13]
1	2,56	5,33	2,60	5,92	2	1	2	5	1,18	+	Kerberos [NT94]
2	1,71	5,33	2,60	3,94	1	1	1,5	3,5	1,13	+	Knox [WJCN09]
1	2,56	6,42	2,60	7,12	2	0,5	4	6,5	1,10	+	SPICE [CHHY12]
1	2,56	5,33	2,60	5,92	2	0,5	3	5,5	1,08	+	PPDIM [BPFS09]
2	1,71	4,25	2,60	3,14	1	0	2	3	1,05	+	AnonymusCloud [KH12]
2	1,71	5,33	2,60	3,94	2	1	2	5	0,79	○	IRM (EDRM)
2	1,71	3,17	2,60	2,34	1	0	2	3	0,78	○	RAFT [BDJ ⁺ 11]
1	2,56	6,42	1,80	4,93	4	2	1	7	0,70	○	SCM [BKNS12]
2	1,71	3,17	1,80	1,62	2	0	1,5	3,5	0,46	○	LOST [WL12]
1	2,56	2,08	1,80	1,60	2	1	1	4	0,40	○	CAD [ZK12]
1	2,56	3,17	1,80	2,43	2	2	2,5	6,5	0,37	○	C3 [BDA ⁺ 10]
3	0,85	5,33	1,80	1,37	2	0	6	8	0,17	○	Log. Attestation [GWP ⁺ 11]
3	0,00	5,33	1,80	0,00	4	0	8	12	0,00	–	PoII [RBO ⁺ 10]
3	0,00	6,42	1,80	0,00	4	0	8	12	0,00	–	Angin [ABR ⁺ 10]
4	0,00	5,33	3,40	0,00	2	1	4	7	0,00	□	SAPPHIRE [PPL12]
4	0,00	2,08	2,60	0,00	1	2	4	7	0,00	□	OASIS [BMY02]
4	0,00	6,42	3,40	0,00	1	0,5	2	3,5	0,00	□	Secure Audit Logs [SK99]
4	0,00	5,33	1,00	0,00	1	0	2	3	0,00	□	ZK [FFS88]
4	0,00	4,25	3,40	0,00	1	0	2	3	0,00	□	CR-Auth
4	0,00	5,33	4,20	0,00	1	0	2,5	3,5	0,00	□	Privacy as a Service [AC11]
4	0,00	6,42	4,20	0,00	2	1	2	5	0,00	□	SaS Cryptoservice [XS07]
4	0,00	7,50	4,20	0,00	1	0	2,5	3,5	0,00	□	PPFSOA [ACEW12]

Tabelle 7.14.: Die Ergebnisse der Evaluierung Cloud-Applikationssicherheit Teil 1

Bezeichnung	EF	DL	BL	PL	PR	IA	PA	SA	KMSH	SC	BA	HA	VT	IG	VT	UV	TR	IN	PT
Sec2 [SMT ⁺ 12]	✓	X	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	✓	✓	X	X	✓	✓
CryptTree [GMSW06]	✓	X	✓	X	✓	✓	X	✓	✓	✓	✓	X	X	✓	X	X	X	✓	✓
Plutus [KRS ⁺ 03]	✓	X	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	✓	✓	X	X	✓	✓
DEPSKY [BCQ ⁺ 13]	✓	X	✓	X	X	✓	X	X	✓	X	✓	✓	✓	✓	✓	✓	X	✓	✓
PKIS [PPL11]	✓	X	✓	✓	✓	✓	X	X	✓	✓	X	X	X	✓	X	X	✓	✓	✓
Relational Cloud [CJP ⁺ 11]	✓	✓	✓	X	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	X	✓	X	X	✓
Mylar [PSV ⁺ 14]	✓	X	✓	✓	✓	✓	✓	X	✓	✓	X	X	X	✓	✓	X	X	✓	✓
CloudProof [PLM ⁺ 11]	✓	X	✓	X	✓	✓	X	X	✓	X	✓	X	X	✓	✓	X	X	✓	✓
SPORC [FZFF10]	✓	X	✓	✓	✓	✓	✓	X	✓	✓	X	✓	X	✓	✓	✓	X	✓	✓
SibF [AAW11]	✓	X	✓	✓	✓	✓	X	X	✓	X	✓	X	X	✓	✓	✓	X	✓	✓
HAIL [BJO09]	✓	X	✓	X	✓	✓	X	X	X	X	✓	✓	✓	X	✓	✓	X	X	✓
Hourglas [DJO ⁺ 12]	✓	X	✓	X	✓	✓	X	X	✓	X	X	X	X	✓	✓	X	X	✓	✓
SUNDR [LKMS04]	✓	X	✓	X	✓	✓	X	X	X	X	✓	✓	X	X	✓	✓	X	X	✓
CryptDB [PZB11]	✓	✓	✓	X	✓	✓	X	X	✓	X	X	✓	X	✓	X	X	X	X	✓
Incognito [LDR05]	✓	✓	X	X	✓	✓	X	X	X	✓	X	X	X	X	X	X	X	✓	✓
Encryption Layer [RBK ⁺ 14a]	✓	X	X	X	✓	✓	X	X	✓	X	✓	X	X	✓	X	X	X	✓	✓
k-Anonymity [Swe02]	✓	✓	X	X	✓	✓	X	X	X	✓	X	X	X	X	X	X	X	✓	✓
VMCrypt [Mal11]	✓	✓	✓	X	✓	✓	✓	X	X	X	✓	X	X	✓	✓	X	X	✓	✓
Venus [SCC ⁺ 10]	✓	X	✓	✓	✓	✓	X	X	X	✓	X	X	X	X	✓	X	X	✓	✓
Oblistore [SS13]	✓	X	✓	X	✓	✓	X	X	✓	X	✓	X	X	✓	✓	✓	X	✓	✓
CS2 [KBR11]	✓	X	X	X	✓	✓	X	X	✓	X	X	X	X	✓	✓	X	X	X	✓
Cloud Filter [PP12]	✓	X	✓	✓	X	X	X	✓	✓	✓	X	X	X	✓	X	X	X	✓	✓
SiRiUS [GSMB03]	✓	X	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	✓	✓	X	X	X	✓
MONOMI [TKMZ13]	✓	✓	X	✓	✓	✓	X	X	✓	X	X	X	X	✓	X	X	X	X	✓
ConfiChair [ABR12]	✓	X	X	X	X	X	X	✓	✓	✓	X	X	X	✓	X	X	X	X	✓
Cloud Storage [ZH10]	✓	X	✓	✓	✓	✓	X	X	✓	X	X	X	X	✓	X	X	X	✓	X
APNGS [ZZY ⁺ 12]	✓	X	✓	X	✓	✓	✓	✓	X	X	X	X	X	X	X	X	X	✓	✓
Client Proof [DLB11]	✓	✓	✓	✓	✓	✓	X	X	X	X	✓	X	X	✓	✓	X	X	✓	X
Key2Cloud [ZYG12]	X	X	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	✓	✓	X	✓	✓	✓
GINGER [SVP ⁺ 12]	X	✓	✓	✓	✓	✓	X	X	X	X	✓	X	X	✓	✓	X	X	✓	✓
SR-ORAM [WS12]	X	X	✓	✓	✓	✓	X	X	✓	X	X	X	X	✓	X	X	X	✓	✓
Shroud [LPM ⁺ 13]	X	X	✓	✓	✓	✓	X	X	✓	X	✓	✓	✓	✓	✓	✓	X	✓	✓
Tsujii [TDF ⁺ 13]	X	✓	✓	✓	✓	✓	X	X	✓	X	X	X	X	✓	X	X	X	✓	✓
Shape CPU [BPS12]	X	✓	✓	✓	✓	✓	X	X	✓	X	X	X	X	✓	✓	X	X	✓	✓
FHE [Gen09]	X	✓	✓	✓	✓	✓	X	X	✓	X	X	X	X	✓	X	X	X	✓	✓
CryptoDSP [TPPG11]	X	✓	✓	✓	✓	✓	✓	X	✓	X	✓	X	X	✓	X	X	X	✓	✓
PP NOSQL [GZLL13]	✓	X	✓	X	✓	✓	X	X	✓	X	X	X	X	✓	✓	X	X	✓	✓
Sealed Cloud [JMR ⁺ 14]	✓	X	X	X	✓	X	X	X	✓	✓	✓	X	X	✓	✓	X	X	✓	✓
CloudSeal [XZY ⁺ 12]	✓	X	✓	✓	X	✓	X	X	✓	✓	✓	X	✓	✓	X	✓	X	X	✓
Twin Clouds [BSSS11]	✓	✓	✓	✓	✓	✓	X	X	✓	X	✓	✓	✓	✓	X	X	X	✓	X
SEDIC [ZCWB11]	✓	✓	✓	✓	✓	✓	X	X	✓	✓	✓	✓	✓	✓	X	✓	X	✓	✓
Kamara [KL10]	✓	X	✓	✓	✓	✓	X	X	✓	✓	X	✓	X	✓	✓	✓	X	✓	X

Tabelle 7.15.: Die Ergebnisse der Evaluierung Cloud-Applikationssicherheit Teil 2

<i>MA</i>	<i>L</i>	<i>S</i>	<i>F</i>	<i>N</i>	<i>C_{tech}</i>	<i>C_{staff}</i>	<i>C_{dev}</i>	<i>C</i>	<i>ϑ</i>		Bezeichnung
1	2,56	5,33	3,40	7,74	2	1	2	5	1,55	+	Sec2 [SMT ⁺ 12]
1	2,56	4,25	3,40	6,17	1	1	2,5	4,5	1,37	+	CryptTree [GMSW06]
1	2,56	5,33	3,40	7,74	2	0	4	6	1,29	+	Plutus [KRS ⁺ 03]
2	1,71	6,42	4,20	7,67	1	1	4	6	1,28	+	DEPSKY [BCQ ⁺ 13]
1	2,56	5,33	2,60	5,92	1	2	2	5	1,18	+	PKIS [PPL11]
1	2,56	4,25	4,20	7,62	1	1	4,5	6,5	1,17	+	Relational Cloud [CJP ⁺ 11]
1	2,56	5,33	2,60	5,92	2	1	2,5	5,5	1,08	+	Mylar [PSV ⁺ 14]
1	2,56	5,33	2,60	5,92	1,5	1	4	6,5	0,91	○	CloudProof [PLM ⁺ 11]
2	1,71	5,33	3,40	5,16	2	0	4	6	0,86	○	SPORC [FZFF10]
2	1,71	5,33	2,60	3,94	1	1	3	5	0,79	○	SibF [AAW11]
2	1,71	4,25	3,40	4,11	1	1	4	6	0,69	○	HAIL [BJO09]
1	2,56	5,33	1,80	4,10	1	1	4	6	0,68	○	Hourglas [DJO ⁺ 12]
1	2,56	4,25	2,60	4,71	2	1	4	7	0,67	○	SUNDR [LKMS04]
1	2,56	3,17	2,60	3,51	1	1	4,5	6,5	0,54	○	CryptDB [PZB11]
1	2,56	3,17	1,80	2,43	1	1,5	2	4,5	0,54	○	Incognito [LDR05]
2	1,71	4,25	2,60	3,14	2	2	2	6	0,52	○	Encryption Layer [RBK ⁺ 14a]
1	2,56	3,17	1,80	2,43	1	1	3	5	0,49	○	k-Anonymity [Swe02]
3	0,85	5,33	1,80	1,37	2	0	1	3	0,46	○	VMCrypt [Mal11]
1	2,56	3,17	1,80	2,43	1	1	4	6	0,41	○	Venus [SCC ⁺ 10]
3	0,85	6,42	2,60	2,37	2	2,5	2	6,5	0,37	○	Oblistore [SS13]
2	1,71	4,25	1,80	2,18	1	1	4	6	0,36	○	CS2 [KBR11]
2	1,71	4,25	2,60	3,14	4	1	4	9	0,35	○	Cloud Filter [PP12]
3	0,85	4,25	3,40	2,06	1	1	4	6	0,34	○	SiRiUS [GSMB03]
1	2,56	3,17	1,80	2,43	2	1	4,5	7,5	0,32	○	MONOMI [TKMZ13]
2	1,71	2,08	2,60	1,54	1	1	3	5	0,31	○	ConfiChair [ABR12]
2	1,71	4,25	1,80	2,18	2	2	4	8	0,27	○	Cloud Storage [ZH10]
1	2,56	2,08	1,00	0,89	2	0	1,5	3,5	0,25	○	APNGS [ZZY ⁺ 12]
3	0,85	5,33	1,80	1,37	2	1	4	7	0,20	○	Client Proof [DLB11]
3	0,00	6,42	3,40	0,00	2	1	1	4	0,00	–	Key2Cloud [ZYG12]
3	0,00	5,33	1,80	0,00	2,5	1	8	11,5	0,00	–	GINGER [SVP ⁺ 12]
3	0,00	4,25	1,80	0,00	2	2	3	7	0,00	–	SR-ORAM [WS12]
3	0,00	6,42	3,40	0,00	4	1	3	8	0,00	–	Shroud [LPM ⁺ 13]
3	0,00	3,17	1,80	0,00	2	2	8	12	0,00	–	Tsujii [TDF ⁺ 13]
3	0,00	5,33	1,80	0,00	4	0	8	12	0,00	–	Shape CPU [BPS12]
3	0,00	4,25	1,80	0,00	2	0	8	10	0,00	–	FHE [Gen09]
4	0,00	4,25	2,60	0,00	2	1	8	11	0,00	□	CryptoDSP [TPPG11]
4	0,00	5,33	1,80	0,00	1	0	2	3	0,00	□	PP NOSQL [GZLL13]
4	0,00	5,33	3,40	0,00	2,5	4	1	7,5	0,00	□	Sealed Cloud [JMR ⁺ 14]
4	0,00	4,25	4,20	0,00	2	1	4	7	0,00	□	CloudSeal [XZY ⁺ 12]
4	0,00	5,33	2,60	0,00	4	2	4	10	0,00	□	Twin Clouds [BSSS11]
4	0,00	5,33	5,00	0,00	4	2,5	4	10,5	0,00	□	SEDIC [ZZCW11]
4	0,00	6,42	3,40	0,00	2	1	2	5	0,00	□	Kamara [KL10]

Tabelle 7.16.: Die Ergebnisse aus dem Trusted Cloud Computing und der Virtualisierungssicherheit Teil 1

Bezeichnung	EF	DL	BL	PL	PR	IA	PA	SA	KMSH	SC	BA	HA	CO	IG	AV	UL	TR	IN	PT
Excalibur [SRGS12]	✓	X	✓	✓	✓	X	X	X	✓	X	✓	X	✓	✓	✓	X	✓	✓	✓
CaaS [BBI ⁺ 13]	✓	X	X	X	✓	X	X	X	✓	X	✓	X	X	✓	✓	✓	X	✓	✓
CloudVerifier [SMV ⁺ 10]	✓	X	✓	✓	✓	X	X	X	✓	X	✓	X	X	✓	✓	✓	X	✓	✓
SDSPF [RJ12]	✓	X	X	X	✓	X	X	X	✓	X	✓	X	X	✓	✓	✓	X	X	✓
GhostDB [ABB ⁺ 07]	✓	✓	✓	✓	✓	✓	X	X	✓	X	X	X	X	✓	✓	X	X	✓	✓
PriaaS [IKC09]	✓	✓	✓	✓	✓	✓	X	X	✓	X	✓	X	X	✓	✓	✓	X	✓	✓
TrustVisor [MLQ ⁺ 10]	✓	X	✓	X	✓	X	X	X	✓	X	X	X	X	✓	✓	X	X	X	✓
CloudVisor [ZCCZ11]	✓	X	X	X	✓	X	X	X	✓	X	✓	X	X	✓	✓	X	X	X	✓
Proxos [TMLL06]	✓	X	✓	X	✓	✓	X	X	X	X	X	X	X	✓	✓	X	X	✓	✓
Flicker [MPP ⁺ 08]	✓	✓	✓	X	✓	X	X	X	✓	X	X	X	X	✓	✓	✓	X	✓	✓
TrustedDB [BS11]	✓	✓	✓	X	✓	X	X	X	✓	X	X	X	X	✓	✓	X	X	X	✓
CipherBase [ABE ⁺ 13]	✓	✓	✓	X	✓	X	X	X	✓	X	X	X	X	✓	✓	X	X	X	✓
Data Capsules [MAF ⁺ 11]	X	X	X	X	✓	X	X	X	✓	✓	X	X	X	✓	✓	✓	✓	✓	X
Terra [GPC ⁺ 03]	✓	X	X	X	✓	X	X	X	✓	X	✓	X	X	✓	✓	✓	X	✓	✓
Xen-Blanket [WJW12]	✓	X	X	X	✓	✓	X	X	X	X	✓	X	X	X	X	✓	X	✓	✓
SSC [BLCSG12]	✓	X	X	X	✓	X	X	X	✓	X	✓	X	X	✓	✓	✓	X	✓	✓
NOVA [SK10]	✓	X	X	X	✓	✓	X	X	X	X	✓	X	X	✓	✓	X	X	✓	✓
ACPS [LDP11]	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	✓	X	X	✓	✓
HyperSafe [WJCN09]	✓	X	X	X	✓	✓	X	X	X	X	✓	X	X	X	✓	X	X	X	✓
BitVisor [SET ⁺ 09]	✓	X	X	X	✓	X	X	X	✓	X	X	X	X	✓	✓	X	X	X	✓
Overshadow [CGL ⁺ 08]	✓	X	X	X	✓	✓	X	X	✓	X	X	X	X	✓	✓	X	X	X	✓
Lares [PCSL08]	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	✓	X	X	X	✓
SecVisor [SLQP07]	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	✓	X	X	X	✓
SEC2 [HLMS10]	X	X	X	X	✓	X	X	X	X	X	✓	X	X	X	X	X	✓	X	X
VMwatcher [JWX07]	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	✓	X	X	X	✓

Tabelle 7.17.: Die Ergebnisse aus dem Trusted Cloud Computing und der Virtualisierungssicherheit Teil 2

MA	L	S	F	$\mathfrak{N}$	C_{tech}	C_{staff}	C_{dev}	C	$\mathfrak{P}$		Bezeichnung
1	2,56	6,42	3,40	9,31	2	0	2	4	2,33	+	Excalibur [SRGS12]
1	2,56	6,42	2,60	7,12	2	1	2	5	1,42	+	CaaS [BBI ⁺ 13]
2	1,71	6,42	2,60	4,75	2	0	2	4	1,19	+	CloudVerifier [SMV ⁺ 10]
1	2,56	4,25	2,60	4,71	2	1	2	5	0,94	○	SDSPF [RJ12]
2	1,71	5,33	1,80	2,73	1	0	3	4	0,68	○	GhostDB [ABB ⁺ 07]
2	1,71	6,42	2,60	4,75	4	0	3	7	0,68	○	PriaaS [IKC09]
1	2,56	3,17	1,80	2,43	1	0	3	4	0,61	○	TrustVisor [MLQ ⁺ 10]
2	1,71	3,17	2,60	2,34	2	1	4	7	0,33	○	CloudVisor [ZCCZ11]
1	2,56	5,33	1,00	2,28	2	1	4	7	0,33	○	Proxos [TMLL06]
3	0,85	6,42	1,80	1,64	2	0	4	6	0,27	○	Flicker [MPP ⁺ 08]
1	2,56	3,17	1,80	2,43	4	1	6	11	0,22	○	TrustedDB [BS11]
1	2,56	3,17	1,80	2,43	4	2	8	14	0,17	○	CipherBase [ABE ⁺ 13]
4	0,00	7,50	2,60	0,00	2	2,5	8	12,5	0,00	□	Data Capsules [MAF ⁺ 11]
4	0,00	6,42	2,60	0,00	2	1	2,5	5,5	0,00	□	Terra [GPC ⁺ 03]
1	2,56	4,25	1,80	3,26	1	1	1	3	1,09	+	Xen-Blanket [WJW12]
1	2,56	6,42	2,60	7,12	2	1	4	7	1,02	+	SSC [BLCSG12]
1	2,56	4,25	1,80	3,26	4	1	1	6	0,54	○	NOVA [SK10]
1	2,56	3,17	1,00	1,35	1	1	1	3	0,45	○	ACPS [LDP11]
1	2,56	2,08	1,80	1,60	2	1	2	5	0,32	○	HyperSafe [WJCN09]
2	1,71	3,17	1,80	1,62	2	1	4	7	0,23	○	BitVisor [SET ⁺ 09]
2	1,71	3,17	1,80	1,62	4	0,5	2,5	7	0,23	○	Overshadow [CGL ⁺ 08]
1	2,56	2,08	1,00	0,89	1	1	4	6	0,15	○	Lares [PCSL08]
2	1,71	2,08	1,00	0,59	2,5	1	1	4,5	0,13	○	SecVisor [SLQP07]
4	0,00	2,08	1,80	0,00	2	1	1	4	0,00	□	SEC2 [HLMS10]
4	0,00	2,08	1,00	0,00	2,5	1		3,5	0,00	□	VMwatcher [JWX07]

7.3. Auswertung und Diskussion

Die Auswertung der Ergebnisse bezieht sich auf die Darstellung in den Tabellen 7.12, 7.13, 7.14, 7.15, 7.16 und 7.17. Zudem fließen die Informationen der ausführlichen Auflistung im Anhang mit ein. Trotz der Vielzahl an Evaluierungskriterien und der geeigneten Parameterwahl zur Vergleichbarkeit der untersuchten 96 Technologien ist die Aussagekraft der Praktikabilität subjektiv und damit vom jeweiligen Betrachter abhängig. Weder Vorkenntnisse, noch Anforderungen oder Prioritäten des Betrachters können im Rahmen dieser Arbeit einbezogen werden. Die Präsentation und Diskussion der Evaluierungsergebnisse bietet jedoch die Form einer Orientierung, auf deren Grundlage weitere Entscheidungen getroffen werden können. Die Tabelle 7.18 verdeutlicht die Verteilung der Praktikabilitätsaussagen für die untersuchten Technologien, wobei die Symbolik analog zur Definition im vorherigen Abschnitt ist.

Tabelle 7.18.: Zusammenfassung der Evaluierungsergebnisse

Symbol	Anzahl	Bemerkung
+	24	Etwa der vierte Teil der untersuchten Technologien wird als potenziell praktikabel bewertet.
○	46	Nahezu die Hälfte der Ansätze sind eingeschränkt praktikablen Lösungen, dies zeigt den weiteren Forschungsbedarf an.
–	9	Nur wenige Ansätze wurden als unpraktikabel bewertet.
□	17	Einige Ansätze bieten nicht genügend Informationen oder sind noch in einem frühen Forschungsstadium.

Resultierend aus der Evaluierungsergebnissen der Arbeit wurden etwa drei Viertel der Lösungen als interessant für eine praktische Umsetzung bewertet. Bemerkenswert ist zudem, dass nur rund ein Fünftel der untersuchten Technologien nicht genügend Informationen für eine Evaluierung boten. Das deutet drauf hin, dass innerhalb der Forschergemeinde ein, wenn man es so bezeichnen möchte, gemeinsamer Qualitätsstandard für Publikationen vorhanden ist. Bestätigt wird dies durch die Tatsache, dass einige der 17 nicht bestimmbar Ansätze nur theoretische Konzepte waren, die noch keine Ergebnisse für die Evaluierung enthalten konnten, aber dennoch für die Untersuchung im Rahmen dieser Arbeit relevant waren. Auf eine detaillierte Analyse, etwa welche der als eingeschränkt praktikablen Lösungen näher am roten oder grünen Bereich liegen, wird verzichtet. Der Grund dafür ist die angedeutete Abhängigkeit der Praktikabilität vom Standpunkt des Betrachters, wodurch eine feingranularere Einteilung nur von geringem Mehrwert wäre. Untersucht wurden dagegen mögliche Zusammenhänge zwischen den Evaluierungskriterien. Eine Regressionsanalyse der Daten ergab jedoch, dass, abgesehen einiger trivialer Zusammenhänge, wie dem Schlüsselmanagement *SM* und der Vertraulichkeit *VT*, keine signifikanten Abhängigkeiten bestehen.

Um die Ergebnisse bezüglich der aufgestellten Thesen besser bewerten zu können, wird folgend eine spezielle Form der Datenvisualisierung vorgestellt. Dazu findet eine Aufteilung der untersuchten Technologieansätze in die vier Teilbereiche Cloud-Sicherheitsmanagement, Cloud-Applikationssicherheit, Trusted Cloud Computing und Cloud-Virtualisierungssicherheit statt. Innerhalb dieser Bereiche werden die Technologien nach ausgewählten Kriterien verglichen. Dies bietet den Vorteil einer höheren Homogenität, wodurch eine verbesserte Vergleichbarkeit zwischen den Teilbereichen erreicht werden soll. Abschließend findet ein Vergleich von, für die Cloud Applikation und für die Cloud Umgebung, geeigneten Ansätzen statt. Damit soll untersucht werden, ob es möglich ist, die Effektivität der Einführung von Sicherheitstechnologien zwischen Cloud Nutzern und CSP vergleichen zu können, um die Fragestellung aus der Einleitung zu beantworten.

In den folgenden Abschnitten werden die untersuchten Lösungen anhand von Kosten-Realisierungseffizienz-Diagrammen diskutiert. Diese Form der Auswertung orientiert sich formal an einer SWOT-Analyse; nach Meffert et al.[MBK09] ein Instrument der strategischen Planung, der Positionsbestimmung und der Strategieentwicklung. Die Verteilung und Bewertung der Quadranten ist jedoch für die Visualisierung der Kosten-Realisierungseffizienz angepasst worden, um die evaluierten Technologien bezüglich ihrer Realisierbarkeit abschätzen zu können. Die Abbildung 7.5 stellt ein solches Diagramm dar.

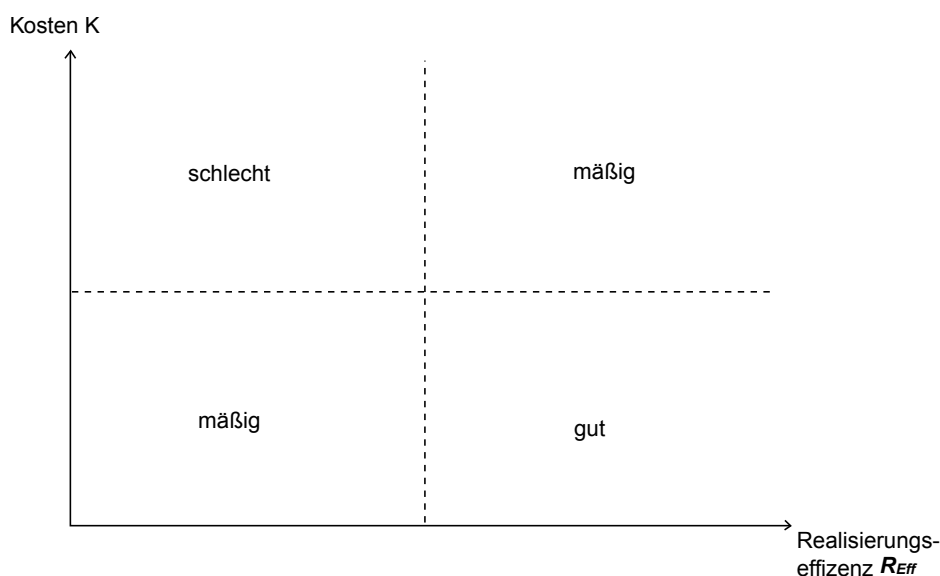


Abbildung 7.5.: Darstellung eines Kosten-Realisierungseffizienz-Diagramms

Die Kennzeichnung der Quadranten dient zur Verdeutlichung der Bewertung der Quadranten. Der mit *schlecht* gekennzeichnete Quadrant spiegelt den Bereich mit hohen Kosten und einer geringen Realisierungseffizienz wieder. Die Lösungen innerhalb dieses Bereichs sind für eine praktische Umsetzung tendenziell ungeeignet. Der als *gut* bezeichnete Bereich dagegen, bietet bei geringen Kosten und einer hohen Realisierungseffizienz, die besten Voraussetzung. Den Mittelweg stellen die mit *mäßig* gekennzeichneten Quadranten dar. Diese charakterisieren sich durch niedrige Kosten, bei gleichzeitig niedriger Realisierungseffizienz oder hohen Kosten bei einer hohen Realisierungseffizienz. Die Darstellung, insbesondere die Abgrenzung der Quadranten ist schematisch zu bewerten, da die Übergänge zwischen den Quadranten fließend sind und nicht, wie in Abbildung 7.5 durch Stichlinien gezeigt, fest eingeteilt.

In den Diagrammen aufgezeichnet sind die Kosten und die Realisierungseffizienz. Die Kosten K fließen direkt aus den Ergebnissen der Evaluierung ein.⁴ Die im Rahmen der Arbeit eingeführte Realisierungseffizienz R_{Eff} ergibt sich aus der Gleichung 7.11 bzw. 7.12.

$$R_{Eff} = \frac{\text{Eignung für Cloud Deployment Modell}}{\text{Anpassung in Applikationsschichten}} \quad (7.11)$$

Diese Effizienz soll die Wirtschaftlichkeit in dem Verhältnis aus Ergebnis und dem dazu nötigem Aufwand repräsentieren. Die verwendeten Werte der Gleichungen entsprechen denen aus dem Evaluierungsfeld *Realisierung*. Als Ergebnis wird dabei die Eignung für ein Cloud-Modell angesehen. Der Aufwand stellt die nötigen Anpassungen in den Applikationsschichten dar. Die Tabelle 7.19

⁴Die Details zur Berechnung finden sich in Abschnitt 7.1.2

verdeutlicht die Bedeutung und Zusammensetzung anhand verschiedener Szenarien.

Tabelle 7.19.: Darstellung von Szenarien zur Beurteilung der Bedeutungen der Realisierungseffizienz

Szenario	Realisierungseffizienz
Eignung für alle Cloud Modelle, keine Anpassungen nötig	maximal
Eignung für viele Cloud Modelle, geringe Anpassungen nötig	hoch
Eignung für IaaS Cloud Modelle, geringe Anpassungen nötig	mittel
Eignung für private Cloud Modelle, viele Anpassungen nötig	niedrig

Ferner erfolgt, wie in Gleichung 7.12 erkennbar, eine Wichtung der eingehenden Werte statt. Eine Eignung für eine private Cloud- Umgebung wird als unbedeutend bewertet, da streng betrachtet jede Technologie in einer solchen Umgebung umgesetzt werden kann.

$$R_{Eff} = \frac{2^0 * PR + 2^1 * IA + 2^2 * PA + 2^3 * SA}{2^0 + 2^1 * PL + 2^2 * BL + 2^3 * DL} \quad (7.12)$$

Ähnliches gilt für die Umsetzung innerhalb einer IaaS-Plattform, durch die bereitgestellten Ressourcen können in dieser Plattform alle Technologien umgesetzt werden die, keinen Zugriff auf die Hardware- oder Virtualisierungsschicht benötigen. Eine höhere Wert erhalten die Technologien welche auch innerhalb einer PaaS oder SaaS Lösung, seitens des Cloud Nutzers, eingesetzt werden können. Dies ist jedoch nur sehr selten der Fall, und nicht Fokus der Untersuchungen dieser Arbeit. Eine analoge Wichtung findet innerhalb der nötigen Anpassungen statt. Um im Falle keiner Anpassungen eine Division durch Null zu vermeiden ist der Wert des Divisors mindestens eins. Die Anpassung Präsentationsschicht ist am geringsten gewichtet, da Anpassungen in diesem Layer zwar ebenfalls aufwändig sein können, jedoch häufig kaum Abhängigkeiten zu unteren Schichten haben oder andere Nebeneffekte hervorrufen. Dies ist im Falle von Anpassungen am Business Layer oder der Datenschicht häufig anders. Daher werden diese höher gewichtet. Eine Normierung der Werte ist durch die Werte der Evaluierungskriterien bereits gegeben.

Auf eine Darstellung der Quadrantenseparierungen wird in den Auswertungen aus Gründen der Übersicht in den nächsten Abschnitten verzichtet. Stattdessen werden die Markierungen der untersuchten Technologien entsprechend ihres Symbols der Praktikabilitätsaussage, aus den Evaluierungsergebnissen im letzten Abschnitt, dargestellt.

7.3.1. Cloud-Sicherheitsmanagement-Lösungen

Die Abbildung 7.6 zeigt die Verteilung der technischen Lösungsansätze die dem Cloud-Sicherheitsmanagement zugeordnet worden. Die Symbolik der gekennzeichneten Werte ergibt sich aus der Praktikabilitätsaussage des untersuchten Lösungsansatzes. Da einige Lösungen mit anderen in Kosten und Realisierungseffizienz übereinstimmen, kann es im Diagramm zu Überschneidungen kommen. Durch die Wahl der Symbolik für die Praktikabilitätsaussage, sind die Bewertungen verschiedenartiger Lösungen meist dennoch erkennbar.

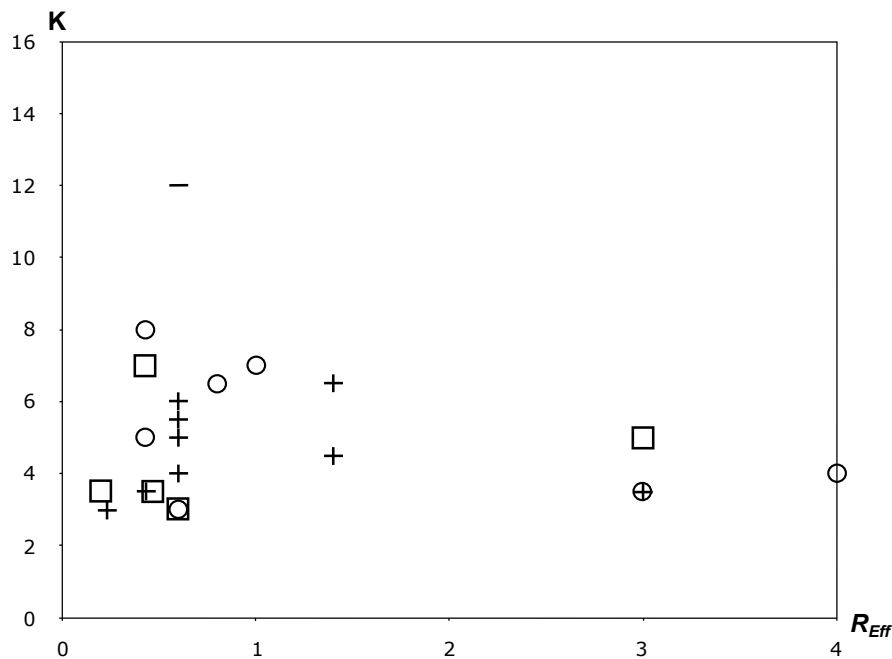


Abbildung 7.6.: Die Verteilung der Praktikabilitätsaussagen im Cloud-Sicherheitsmanagement-Umfeld im Kosten-Realisierungseffizienz-Diagramm. Die Symbolik der Werte orientiert sich an der Praktikabilitätsaussage des untersuchten Forschungsansatzes.

Wie die Abbildung verdeutlicht, befinden sich die potenziell praktikablen Lösungen vorrangig im mittleren Kostenbereich, bei gleichzeitig eher geringer Realisierungseffizienz befinden. Die eingeschränkt praktikablen Lösungen verteilen sich um diese. Die Lösung von Zellwag et al. [ZK12] ($\bigcirc$, ganz rechts) wird am besten bewertet und bietet bei geringen Aufwand die Möglichkeit, Anomalien in redundanten Cloud-Speichern aufzudecken. Ebenfalls gut sind die Lösungen LOST von Watson et al. [WL12] und Sibboleth bewertet (beide überlagern sich bei $R_{Eff} = 3$). Letzte ist in der Praxis bereits etabliert und verdeutlicht die Aussagekraft des obigen Diagramms. Die Lösung von Xu et al. [XS07] ist durch ihre Praxisuntauglichkeit nur von untergeordneter Rolle ($\square$, bei $R_{Eff} = 3$).

Die in dieser Untersuchung am schlechtesten abschneidenden Lösungen sind die Identity Management Systeme von Ranchal et al. [RBO⁺10] und Angin et al. [ABR⁺10] ($-$, überlagern sich bei $K = 12$). Beide Lösungen beruhen auf der Technologie so genannter Active Bundles, welche hohe Kosten verursachen und nur geringe praktische Einsatzfähigkeit zeigen.

Insgesamt ist die Verteilung der untersuchten Lösungen im Vergleich zu Trusted Cloud oder sicheren Virtualisierungsansätzen relativ diffus. Dies begründet sich an der Vielzahl eingesetzter Technologien und der Verschiedenartigkeit der Lösungsansätze. Zudem decken die untersuchten Cloud-

Sicherheitsmanagement- Lösungen eine große Bandbreite von Anwendungsfällen ab.⁵ Die Tabelle 7.20 zeigt die Ergebnisse der Praktikabilitätsbewertung der insgesamt 29 untersuchten Lösungen.

Tabelle 7.20.: Zusammenfassung der Evaluierungsergebnisse für den Bereich Cloud-Sicherheitsmanagement

Symbol	Anzahl	Anteil	Bemerkung
+	12	41%	größter relativer Anteil aller Teilbereiche
○	7	24%	-
-	2	7%	-
□	8	28%	größter relativer Anteil aller Teilbereiche

Dieses Ergebnis verdeutlicht einerseits das bereits viele Ansätze existieren die potenziell praktikabel sind. Dies liegt nicht zuletzt daran, das einige der bewertenden Technologien bereits eingesetzt werden und ohne Anpassungen für Cloud-Umgebungen geeignet sind. Andererseits werden im Cloud-Sicherheitsmanagement-Bereich neue Konzepte entwickelt, um vor allem im Datenschutz Verbesserungen zu erzielen. Entsprechend hoch ist der Anteil jener Lösungen die keine Abschätzung der Praktikabilität erlauben, z.B. weil diese sich in einem frühen Forschungsstadium befinden.

⁵Die Bereiche werden in Kapitel 3 Cloud-Sicherheitsmanagement ausführlich beschrieben.

7.3.2. Cloud-Applikationssicherheit-Lösungen

Die Abbildung 7.7 zeigt die Verteilung der technischen Lösungsansätze zur Cloud- Applikationssicherheit. Die Symbolik der gekennzeichneten Werte ergibt sich aus der Praktikabilitätsaussage des untersuchten Lösungsansatzes. Da einige Lösungen mit anderen in Kosten und Realisierungseffizienz übereinstimmen, kann es innerhalb des Diagramm zu Überschneidungen kommen. Durch die Wahl der Symbolik für die Praktikabilitätsaussage, sind die Bewertungen verschiedenartiger Lösungen meist dennoch erkennbar.

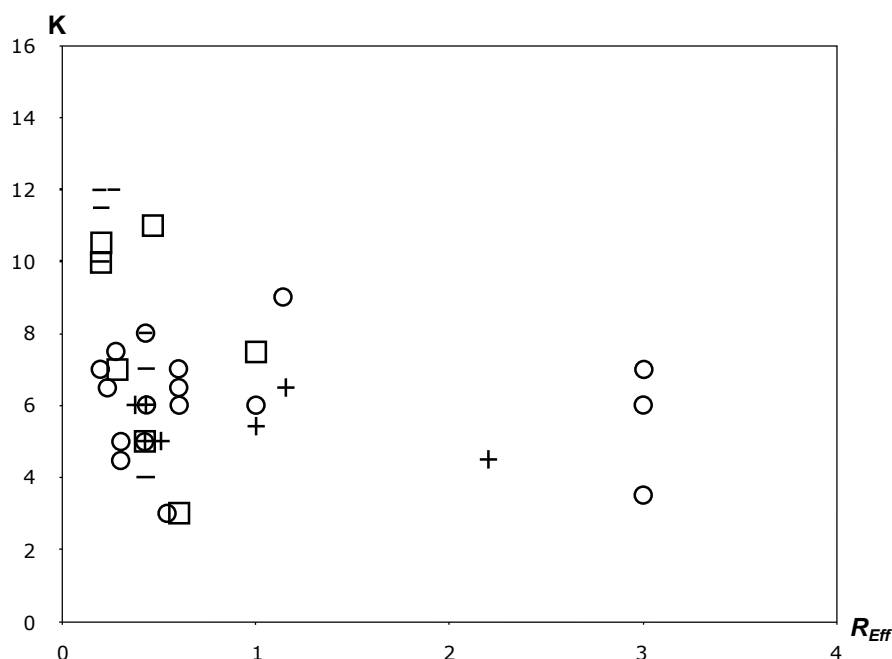


Abbildung 7.7.: Verteilung der Praktikabilitätsaussagen im Cloud-Applikationssicherheit-Umfeld im Kosten-Realisierungseffizienz-Diagramm. Die Symbolik der Werte orientiert sich an der Praktikabilitätsaussage des untersuchten Forschungsansatzes

Analog zu den Lösungen des Cloud-Sicherheitsmanagements, verteilen sich die Ansätze der Cloud-Applikationssicherheit. Die obige Abbildung verdeutlicht, dass sich die potenziell praktikablen Lösungen vorrangig im mittleren Kostenbereich, bei gleichzeitig geringer Realisierungseffizienz befinden. Die eingeschränkt praktikablen Lösungen verteilen sich um diese. Die vier Lösungen die in diesem Teilbereich am besten abschneiden sind CrypTree von Grolimund et al.[GMSW06] (+), Ecrypti-on Layer von Reinhold et al.[RBK⁺14a](○,oben), CS2 von Kamara et al.[KBR11](○, mittig) und die Lösung APNGS von Zhang et al. [ZZY⁺12]. Erstere wird durch einen CSP in der Praxis eingesetzt⁶ und unterstreicht somit die Aussagekraft der Bewertungen in Abbildung 7.7. Die Encryption Layer Lösungen wurde unter Berücksichtigung der praktischen Realisierbarkeit entwickelt, Details hierzu wurden im Abschnitt 7.1.1 erläutert. Die CS2 Lösung setzt, ebenso wie der Encryption Layer, auf den *First-Encrypt-Then-Upload* Ansatz. Beide Lösungsansätze agieren nach dem Prinzip: erst die Daten lokal verschlüsseln und anschließend in die Cloud-Umgebung laden. Diese Vorgehensweise gilt als etabliert und praktikabel, eine solche Umsetzung ist z.B. BoxCryptor⁷.

Die Obfuskierungslösung von Zhang et al. [ZZY⁺12] wird ebenfalls gut bewertet, da dieser Ansatz

⁶<https://www.wuala.com> (letzter Zugriff 25.03.2015)

⁷<https://www.boxcryptor.com/en> (letzter Zugriff 24.06.2015)

prinzipiell in allen Cloud-Modellen Einsatz finden kann. Der wesentliche Nachteil der Lösung sind jedoch die (unter Umständen stark) erhöhen Betriebskosten beim CSP.

Die am schlechtesten abschneidenden Lösungsansätze sind jene, die auf voll homomorphen Verschlüsselungsverfahren und ORAM beruhen. Erwartungsgemäß sorgen diese für hohe Kosten und sind nur schwer effizient umsetzbar.⁸

Die Gesamtverteilung ist ähnlich der Verteilung der Managementlösungen relativ diffus. Dies begründet sich an der Vielzahl der eingesetzten Technologien und Verschiedenartigkeit der Lösungsansätze. Auch hier begründet sich dies durch die große Bandbreite an verschiedenen, eingesetzten Technologien. Die Tabelle 7.21 zeigt die Bewertung der Praktikabilität der 42 untersuchten Lösungen.

Tabelle 7.21.: Zusammenfassung der Evaluierungsergebnisse für den Bereich Cloud-Applikationssicherheit

Symbol	Anzahl	Anteil	Bemerkung
+	7	17%	-
○	21	49%	eine hohe Anzahl an Lösungen
-	7	17%	größter relativer Anteil aller Teilbereiche
□	7	17%	-

Das Ergebnis zeigt einerseits, dass einige Ansätze existieren die potenziell praktikabel sind, es aber andererseits im Vergleich zu den Sicherheitsmanagement weniger Lösungen sind, da bestehende Lösungen ohne Anpassungen in Cloud-Umgebungen nicht das gleiche Sicherheitsniveau bieten. Folglich gibt es in der Forschung zahlreiche Weiterentwicklungen, die jedoch noch nicht für einen direkten Einsatzes in produktiven Umgebungen geeignet sind. Ferner zeigen die Lösungsansätze die als nicht praktikabel bewertet wurden, dass neben Weiterentwicklungen von bestehenden Lösungen auch neue Forschungsansätze verfolgt werden. Der große Anteil dieser Lösungen deutet auf ein hohes Potenzial für die Entwicklung der Cloud Computing Sicherheit in diesem Teilbereich hin. Entsprechend hoch ist auch der Anteil der Lösungen die keine Abschätzung der Praktikabilität erlauben, etwa weil diese sich in einem zu frühen Forschungsstadium befinden.

⁸Detaillierte Informationen zu diesen Technologieansätzen finden sich in Abschnitt zur 4.2 Applikationslogik, bzw. Abschnitt 4.6 zum Datenzugriff innerhalb der Cloud-Applikationssicherheit-Lösungen.

7.3.3. Lösungen für Trusted Cloud Computing

Die Abbildung 7.8 zeigt die Verteilung der technischen Lösungsansätze zur im Trusted Cloud-Umfeld. Die Symbolik der gekennzeichneten Werte ergibt sich aus der Praktikabilitätsaussage des untersuchten Lösungsansatzes. Da einige Lösungen mit anderen in Kosten und Realisierungseffizienz übereinstimmen, kann es innerhalb des Diagramm zu Überschneidungen der Symbole kommen. Durch die Wahl der Symbolik für die Praktikabilitätsaussage, sind die Bewertungen verschiedenartiger Lösungen meist dennoch erkennbar.

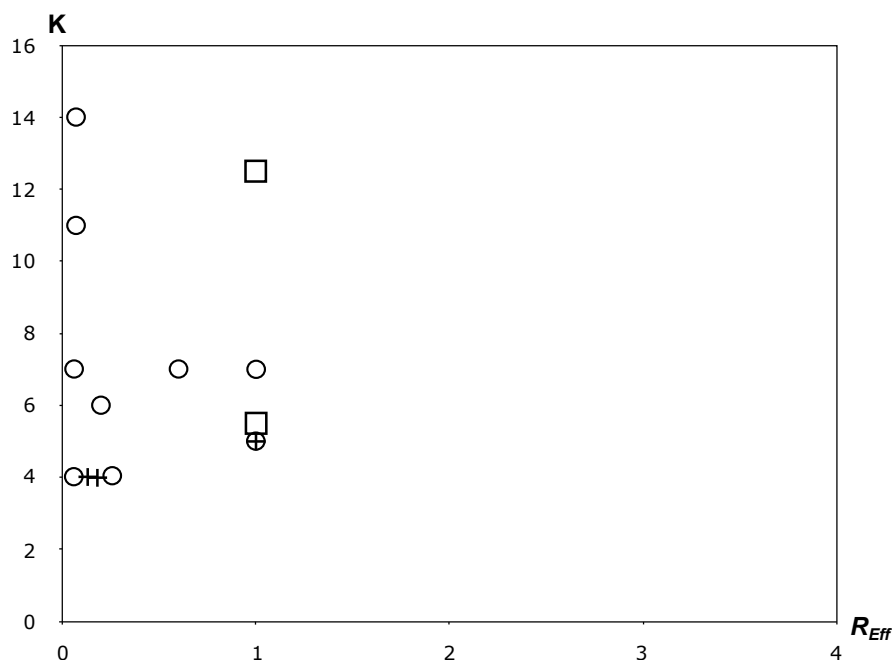


Abbildung 7.8.: Verteilung der Praktikabilitätsaussagen im Trusted Cloud Computing Umfeld im Kosten-Realisierungseffizienz-Diagramm. Die Symbolik der Werte orientiert sich an der Praktikabilitätsaussage des untersuchten Forschungsansatzes

Wie die obige Abbildung erkennen lässt, deutet sich hier neben der geringeren Anzahl, eine Teilung der Technologien in zwei Klassen an. Die erste zeichnet sich durch eine sehr geringe Realisierungseffizienz nahe $R_{Eff} = 0$ aus und die zweite haben eine Realisierungseffizienz von $R_{Eff} = 1$. Während erstere ausschließlich auf die Verwendungen von Hardware- Sicherheitsmodulen setzt, versuchen Lösungen der Klasse zwei diese zu virtualisieren. Tatsächlich ließen sich diese Lösungsansätze auch im nächsten Abschnitt Cloud- Virtualisierungslösungen einfügen, da diese die Schnittmenge zwischen Trusted Computing und Virtualisierungssicherheit darstellen. Ferner ist festzustellen, dass die Kosten ein etwas höheres Niveau als bei den beiden vorhergehenden Bereichen Applikationssicherheit und Sicherheitsmanagement haben. Die lässt sich mit der benötigten Hardware begründen, auf welcher dieser Technologiebereich fundiert und ist ebenso der Grund für das insgesamt geringe Realisierungseffizienz-Niveau.

Die Lösungen die in Abbildung 7.8 am besten abschneiden, sind aus der Schnittmenge mit den sicheren Virtualisierungslösungen: Terra von Garfikel et al. [GPC⁺03], CaaS von Bleikertz et al. [BBI⁺13], SDPF von Ran et al. [R]12] und CloudVisor von Zhang et al. [ZCCZ11]. Grund für eine bessere Bewertung im Vergleich zu den restlichen Trusted Cloud Lösungen ist der Bezug zur Virtualisierung, welcher die Realisierungseffizienz steigert.

Die Datenbanklösungen der Autoren Arasu et al. [ABE⁺13] bzw. Bajaj et al.[BS11] schneiden im Diagramm am schlechtesten ab, da beide auf proprietärer Hardware basieren und mit einem hohen Entwicklungsaufwand bewertet wurden.

Die Gesamtverteilung zeigt, dass Lösungen im Trusted Cloud-Umfeld mit vergleichsweise höheren Kosten und geringer Realisierungseffizienz verbunden sind. Der Grund dafür ist, wie bereits angedeutet, der Einbezug von Hardware in die Systemsicherheit, der sich auf Kosten und Effizienz niederschlägt. Die Tabelle 7.22 zeigt die Bewertung der Praktikabilität der 14 untersuchten Lösungen.

Tabelle 7.22.: Zusammenfassung der Evaluierungsergebnisse für den Bereich Trusted Cloud Computing

Symbol	Anzahl	(Anteil)	Bemerkung
+	3	21%	-
○	9	65%	größter relativer Anteil aller Teilbereiche
-	0	0%	keine Lösungen werden als unpraktikabel eingestuft
□	2	14%	-

Bemerkenswert an der Zusammenfassung dieses Teilbereichs ist die Tatsache, dass keine Lösungen als unpraktikabel eingestuft werden. Die Ursache dafür ist möglicherweise der starke Einbezug von spezieller Hardware, der die Ansätze mit höheren Kosten sowie höherem Entwicklungs- und Forschungsaufwand verbindet, jedoch für ein Mindestmaß an Effizienz in der Verarbeitungsleistung sorgt. Im Bereich der Softwareentwicklung ist die Hemmschwelle geringer, in völlig neue Ansätze zu investieren als in der Hardwareentwicklung, bzw. der hardware-unterstützten Softwareentwicklung. Die beiden nicht bewertbaren Lösungen zeigen jedoch das Bemühen der Forschung in diesem Bereich neuartige Konzepte zu entwickeln.

7.3.4. Cloud-Virtualisierungssicherheit-Lösungen

Die Abbildung 7.9 zeigt die Verteilung der technischen Lösungsansätze zur Cloud- Virtualisierungssicherheit. Die Symbolik der gekennzeichneten Werte ergibt sich aus der Praktikabilitätsaussage des untersuchten Lösungsansatzes. Da einige Lösungen mit anderen in Kosten und Realisierungseffizienz übereinstimmen, kann es im Diagramm zu Überschneidungen kommen. Durch die Wahl der Symbolik für die Praktikabilitätsaussage, sind die Bewertungen verschiedenartiger Lösungen meist dennoch erkennbar.

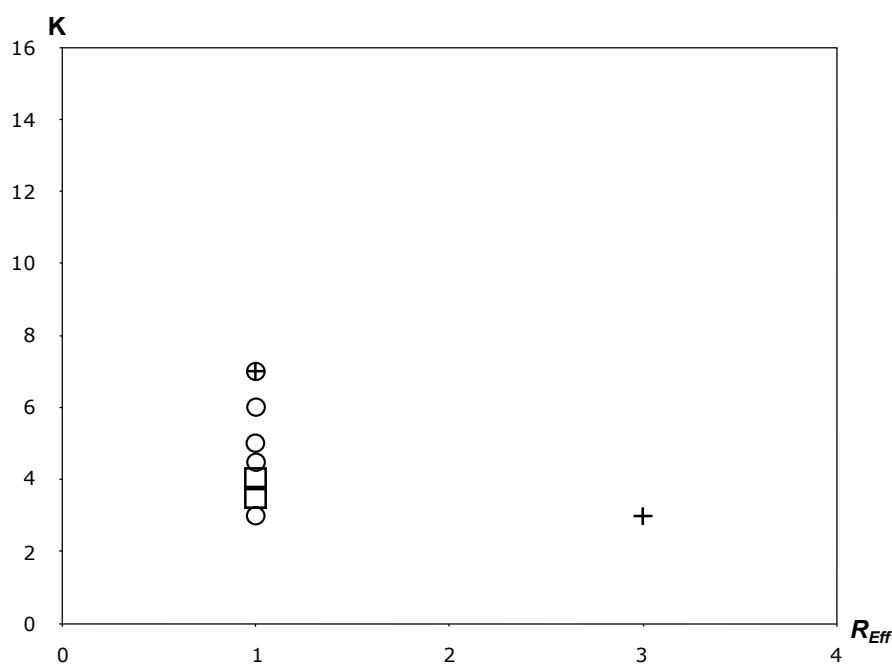


Abbildung 7.9.: Verteilung der untersuchten Lösungen im Cloud-Virtualisierungssicherheit-Umfeld im Kosten-Realisierungseffizienz-Diagramm. Die Symbolik der Werte orientiert sich an der Praktikabilitätsaussage des untersuchten Forschungsansatzes.

In der Abbildung deutlich erkennbar ist eine, im Vergleich zu den anderen Bereichen, eine deutliche Konzentration innerhalb eines Bereichs im Diagramm. Die einzige Ausnahme stellt eine Lösung dar, die nicht nur in privaten Cloud-Umgebungen, sondern auch innerhalb von öffentlichen IaaS-Lösungen eingesetzt werden kann. Dieser als geschachtelte Virtualisierung bezeichnete Ansatz Xen-Blanket von Williams et al. [WJW12] schneidet in Abbildung 7.9 auf Grund seines breiteren Einsatzspektrums am besten ab. Die anderen untersuchten Lösungen sind durch eine geringere Realisierungseffizienz gekennzeichnet, da hier der Zugriff auf die Virtualisierungsumgebung Voraussetzung für die Wahl dieser Ansätze ist. Im Vergleich zu den Cloud-Sicherheitsmanagement und den Applikationssicherheit Lösungen, ist die Gesamtverteilung ist deutlich geordneter. Der Grund ist die Homogenität der eingesetzten Technologien und die Gleichartigkeit der Ansätze, die Virtualisierungsumgebung zu schützen. Die Tabelle 7.23 zeigt die Bewertung der Praktikabilität der 15 untersuchten Lösungen.

Analog zu den Trusted Computing Lösungen zeichnen sich die Virtualisierungslösungen dadurch aus, dass keine als unpraktikabel eingestuft werden. Der Grund dafür ist jedoch weniger durch den starke Einbezug von spezieller Hardware zu suchen. Vielmehr ist es die Tatsache, dass für den Betrieb einer Virtualisierungsumgebung ein Mindestmaß an Ressourcen zur Verfügung stehen muss.

Tabelle 7.23.: Zusammenfassung der Evaluierungsergebnisse für den Cloud-Virtualisierungssicherheit-Lösungen

Symbol	Anzahl	Anteil	Bemerkung
+	2	18%	-
○	9	64%	analog zum Trusted Cloud Computing findet sich hier der Großteil der untersuchten Lösungen
–	0	0%	auch in diesem Teilbereich keine als unpraktikabel bewerteten Lösungen
□	2	18%	-

Obgleich nicht so hoch wie im Trusted Cloud Computing, sind die Investitionskosten dafür nicht zu vernachlässigen. Zudem existieren in diesem Teilbereich Ansätze, deren Sicherheit schlussendlich auf virtualisierten Sicherheitsmodulen in Hardware beruhen, dass als Trusted Virtualization bezeichnet wird.⁹

⁹Die Details zu diesem Themen enthält der Abschnitt 6 Cloud-Virtualisierungssicherheit.

7.3.5. Cloud-Applikation und Cloud-Umgebung

Die Abbildung 7.10 stellt den Vergleich der Verteilung der Lösungen dar, welche für die Cloud-Applikation bzw. die Cloud-Umgebung geeignet sind. Die Symbolik der Praktikabilitätsaussage findet hierbei keine Anwendung. Stattdessen unterscheiden die Symbole $\circ$ und $\times$, die Lösungen die für Cloud-Applikation bzw. Cloud-Umgebung geeignet sind. Ferner sind Überschneidungen erkennbar, die solche Lösungen charakterisieren, welche in beiden Szenarien einsetzbar sind.

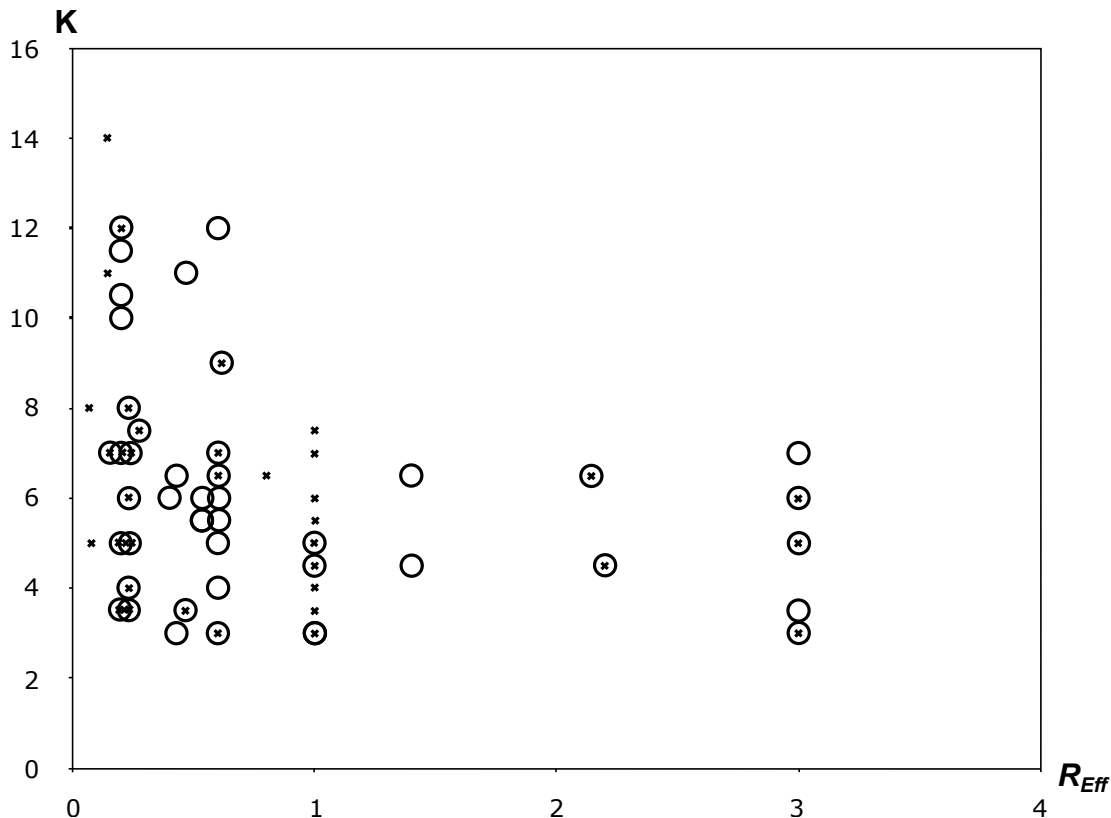


Abbildung 7.10.: Vergleich der Lösungsverteilung geeignet für Cloud-Applikation und Cloud-Umgebung. Mit $\circ$ sind die Ansätze gekennzeichnet, die sich potenziell für Cloud-Applikationen eignen, also vom Cloud-Nutzer eingesetzt werden können. Mit $\times$ werden die Lösungen für Cloud-Umgebungen dargestellt. Erkennbar sind Überschneidungen, die Lösungen charakterisieren, welche in beiden Szenarien einsetzbar sind.

Auffällig ist der Bereich der Realisierungseffizienz $R_{Eff} = 1$, bei dem eine Häufung von Lösungen zu erkennen ist, die nur für Cloud- Umgebungen geeignet sind. Konkret handelt es sich dabei um Ansätze zur Cloud- Virtualisierungssicherheit. Eine andere Auffälligkeit ist die Häufung von Lösungen, die sich für Cloud-Applikationen eignen. Diese befinden sich um eine Realisierungseffizienz von $R_{Eff} = 0,5$. Im übrigen Diagramm ist eine hohe Überschneidungsrate festzustellen, die insbesondere Lösungen für das Management, Datenbanken und Dateisystem einschließt. Die Verbindungslinien in der Taxonomieabbildung 2.6 im Abschnitt 2.3 illustrieren diese.

Ausgehend von diesen Fakten ist als Trend erkennbar, dass Technologien, die exklusiv der Cloud Umgebung vorbehalten sind, effizienter realisierbar sind. Dies ist insofern ein zu erwartendes Ergebnis, da Realisierungen auf höheren Abstraktionsebenen dazu neigen weniger effizienter zu sein

als Lösungen auf niedrigen Abstraktionsebenen, z.B. in Hardware. Damit wird die Annahme der These 2 der Arbeit unterstützt, dass der notwendige Mehraufwand durch die Schutzmaßnahme vom Einfluss auf das gesamte Cloud-System abhängig ist. Neben dem höheren Mehraufwand zeigt sich somit, dass auch die Realisierungseffizienz innerhalb der Cloud-Umgebung höher ist. Resultierend daraus sind Sicherheitslösungen in der Cloud-Umgebung aus Nutzersicht erstrebenswert. Tatsächlich bieten CSP im Service Portfolio zunehmend Sicherheitslösungen an um die Datensicherheit und den Datenschutz von Cloud-Applikationen zu erhöhen. Jedoch dienen diese in der Regeln nur dazu, die Sicherheit gegenüber Dritten zu erhöhen. Der CSP hat in der Regel ähnliche Möglichkeiten zur Einsicht in Daten und Prozesse wie in ungeschützter Form.

Zusammenfassung

In dieser Dissertation wurden umfassend Anforderungen an sicheres Cloud Computing untersucht. Insbesondere wurden 96 bestehende Forschungs- und Lösungsansätze zum Schutz von Daten und Prozessen in Cloud- Umgebungen untersucht und nach ihrer Praxistauglichkeit bewertet. Die Basis für die Vergleichbarkeit stellten spezifizierte und definierte Kriterien dar, nach denen die untersuchten Technologien evaluiert wurden.

Ein Ergebnis dieser Arbeit war eine Bewertungsmethode für technische Forschungsansätze im Cloud Sicherheitsumfeld und der Evaluierung von 96 Forschungs- und Lösungsansätzen nach ihrer Praxistauglichkeit.

Hierzu wurden die Teilbereiche Sicherheitsmanagement, Applikationssicherheit, Trusted Cloud Computing und Virtualisierungssicherheit identifiziert. Im zweiten Teil der Arbeit, den Kapiteln 3 bis 6, wurden Inhalt und Lösungsstrategien der vier Bereiche diskutiert. Die Evaluierung der untersuchten Ansätze und State of the Art Methoden nach spezifizierten Kriterien erfolgte in Kapitel 7. Die in diesem Kapitel ebenfalls eingeführte Aussage zur Praxistauglichkeit ergab sich aus dem Verhältnis des *Nutzens* zu den zu *erwarteten Kosten*.¹ Die Ergebnisse der Evaluierung ergaben, dass rund ein Viertel der untersuchten Forschungsansätze für den potenziellen praktischen Einsatz geeignet sind. Unter diesen Ansätzen befinden sich zudem bereits in der Praxis eingesetzt Lösungen, wie Crypt-Tree [GMSW06] oder Shibboleth². Dies erlaubt den Rückschluss, dass andere ähnlich bewertete Ansätze gleichfalls eine Praxistauglichkeit inne haben. Ferner wird die in der Arbeit vorgenommene Evaluierung durch das korrektes Bewerten obig aufgeführter Technologien in ihrer der Aussagekraft untermauert. Den Abschluss der Evaluierung bildete die Analyse der Teilbereiche und deren Vergleich untereinander. Hierzu wurde in der Arbeit eine angepasste SWOT-Analyse in Form von Kosten-Realisierungseffizienz-Diagrammen durchgeführt.³

Zum Zeitpunkt der Arbeit existieren nach Kenntnis des Autors keine vergleichbaren Auflistung und Evaluierung von Technologien im Cloud Computing Sicherheitsumfeld. Dies betrifft den Umfang der untersuchten Technologien sowie der bewerteten Kriterien. Andere vergleichbare Ergebnisse in der Literatur betreffen entweder einen konkreten Technologie- oder Anwendungsbereich wie Dateisysteme bei Albeshri et al.[ABGN12] oder sind allgemeiner Natur wie bei Fernandes et.al [FSG⁺14].

Ein weiteres Ergebnis der Arbeit ist die Einführung und Bestätigung des Prinzips der Delegation mit begrenztem Wissen. Das aus den Thesen der Arbeit abgeleitete Prinzip verdeutlicht den Widerspruch zwischen dem Auslagern von Daten und Prozessen in Cloud-Umgebung bei gleichzeitiger Wissensbegrenzung bzgl. der Daten und Prozesse gegenüber der Cloud-Umgebung. Die Formalisierung dieses Prinzips bildet im folgenden Abschnitt den Abschluss dieser Arbeit.

¹Die Definition dieser Größen erfolgt in Abschnitt 7.1.

²Vergleiche <http://shibboleth.net>, letzter Zugriff 23.04.2015

³Die Details finden sich in Abschnitt 7.3.

8.1. Fazit

Anhand der aufgestellten Thesen im Kapitel 1.1 wird nachfolgend das Prinzip der Delegation mit begrenztem Wissen formalisiert.

Thesen der Arbeit

Die eingeführte erste These, kann durch die in der Arbeit gewonnenen Erkenntnisse als bestätigt angesehen werden

Die Charakteristiken des Cloud Computings stehen im Widerspruch zu bestehenden Datensicherheits- und Datenschutzerfordernungen.

Klassische Sicherheitsmaßnahmen wie Authentifizierung und Identitäts- oder Schlüsselmanagement bieten häufig zu starre Konzepte, um der hohen Dynamik und starken Vernetzung von CSP in Cloud-Umgebungen die gleiche Sicherheit bieten zu können wie in klassischen Client-Server-Anwendungen. Es ergeben sich zwei Alternativen: Entweder wird die Flexibilität und Dynamik der Cloud-Umgebung reduziert, um bisherige Maßnahmen weiterhin einsetzen zu können, oder die bestehenden Maßnahmen werden erweitert. Letzteres führt jedoch häufig zu einem sprunghaften Anstieg der Komplexität und ist aus diesem Grund sicherheitstechnisch kritisch zu bewerten.

Im Fall des Trusted Cloud Computing wird die durch die gewonnene Sicherheit zusätzliche Arbeit des Cloud-Nutzers nicht durch Informationspreisgabe verringert, sondern durch die Bereitwilligkeit einen höheren Preis zu zahlen. Das im folgenden Abschnitt formalisierte Prinzip der Delegation mit begrenztem Wissen liegt dieser Situation zu Grunde. Die Kosten der Hardware werden auf die Ressourcennutzungskosten der Cloud-Nutzer aufgeschlagen. Bei exklusiver Nutzung ist eine vollständige Übernahme der Investitionskosten erforderlich, die sich nur bei langfristiger Nutzung rentieren. Dies steht offensichtlich im Widerspruch zu den Cloud-Charakteristiken.⁴

Ferner wird die These durch Erkenntnisse der funktionalen Verschlüsselung⁵ deutlich, die für das Cloud-Umfeld aufgrund der potenziell hohen Praxistauglichkeit eine große Bedeutung hat. Die gebotenen Sicherheitsgarantien durch funktionale Verschlüsselung unterscheiden sich von denen der klassischen Verschlüsselungsverfahren. So ist es für den CSP möglich, dass nach vielen Suchanfragen zu einer bestimmte Menge von Dokumenten Informationen preisgegeben werden. Resultierend daraus kann der CSP Annahmen über die Suchmuster des Cloud-Nutzers machen, um Suchbegriffe zu erraten oder Zusammenhänge zwischen Daten zu erlernen. Es ist entscheidend zu verstehen, dass der Suchprozess die Informationen an den Provider weitergibt und dies keine Lücke im Sicherheitsverfahren darstellt. Das Weitergegebene ist exakt das, was der CSP beim Prozess des Datenbereitstellens erfährt, beispielsweise, dass die als Ergebnis gelieferte Dateien einen gemeinsamen Suchbegriff haben. Eine Informationspreisgabe ist folglich Bestandteil eines effizienten Cloud-Services. Dies steht offensichtlich im Widerspruch zu klassischen Anforderungen der Datensicherheit- und des Datenschutzes, Unbefugten keinerlei Informationen preiszugeben. Ferner kann die eingeführte zweite These als bestätigt angesehen werden.

Der notwendige Mehraufwand der Maßnahme ist vom Einfluss auf das gesamte Cloud System abhängig.

Die Schutzmaßnahmen für die Cloud-Umgebung, wie Trusted Cloud Computing oder Cloud-Virtualisierungssicherheit -Lösungen nehmen einen Einfluss auf niedriger Abstraktionsebene. Der daraus resultierende Mehraufwand fällt im Vergleich geringer aus als bei Cloud-Applikationssicherheit-Lösungen. Voraussetzung dafür ist jedoch der Zugriff auf die Hardware bzw. Virtualisierungsebene,

⁴Eine diesbezügliche Auflistung ist Bestandteil des Abschnitts 2.1.

⁵Detaillierte Information zu funktionaler Verschlüsselung bietet Abschnitt 4.2.4.

die dem Betreiber der Cloud-Umgebung, also in der Regel dem CSP, vorbehalten ist. Dadurch ergibt sich der Trend, dass Technologien, die exklusiv der Cloud-Umgebung vorbehalten sind, effizienter realisierbar sind.⁶ Dies ist insofern ein zu erwartendes Ergebnis, da Realisierungen auf höheren Abstraktionsebenen dazu neigen weniger effizienter zu sein als Lösungen auf niedrigen Abstraktionsebenen.

Prinzip der Delegation mit begrenztem Wissen

Das aus den Thesen abgeleitete *Prinzip der Delegation mit begrenztem Wissen* kann im Rahmen der Arbeit als bestätigt angesehen werden.

Je mehr Arbeit delegiert werden soll, aber je weniger Wissen darüber, desto mehr Arbeit hat der Delegierende. Diese Mehrarbeit kann durch Informationspreisgabe reduziert werden.

Das eingeführte Prinzip verdeutlicht, dass die Auslagerung von Arbeit bei gleichzeitiger Wissensbegrenzung zum Anstieg der Arbeitslast des Auslagernden führt, um die Arbeit weiterhin vollständig zu erledigen. Hierzu gibt es im Verlauf der Arbeit regelmäßig Bezug auf das postulierte Prinzip.

Das Prinzip bestätigt sich im obig angedeuteten Szenario der funktionalen Verschlüsselung. Das Ergebnis der Sicherheitsbetrachtung war die Aussage, dass eine Informationspreisgabe in gewissem Sinne einem effizienten und verlässlichen Cloud Service innewohnt. Eine bekannte Alternative ist dem Provider eine falsch positive Menge an Informationen schicken zu lassen, damit das Wissen zu begrenzen, und das Ergebnis durch den Cloud-Nutzer zu filtern. Letzteres ist die im Prinzip angedeutete Mehrarbeit des Delegierenden. Neben dieser Mehrarbeit bedeutet eine solche Vorgehensweise jedoch häufig zusätzlich Ineffizienz in der Cloud-Umgebung und Mehraufwand in der Kommunikation. Letztendlich steigen dadurch die Kosten für die Auslagerung von Daten und Prozessen. Im Extremfall, bei idealisierter Informationspreisgabe von 0, müssten alle Arbeiten vom delegierenden Cloud-Nutzer erledigt werden, da ohne Informationen offensichtlich keine Arbeit vom CSP verrichtet werden kann. Die Autoren Hacigümüs et al. [HILM02] und Bajaj et al. [BS11] beschreiben diese Thematik anhand der Ausführung von SQL Queries,⁷ ohne ihr jedoch einen konkreten Namen zu geben.

Ferner findet das Prinzip der Delegation mit begrenztem Wissen Bestätigung im Cloud-Sicherheitsmanagement, in der Cloud- Applikationssicherheit und im Trusted Cloud Computing. In den jeweiligen Abschnitten zur Zusammenfassung wird das Prinzip diskutiert. Daher wird auf eine erneute Erörterung verzichtet und stattdessen auf diese Abschnitte verwiesen. Das Ziel dieses Abschnitts besteht vielmehr darin, das Prinzip der Delegation mit begrenzten Wissen als bestätigt anzusehen und dieses mit den Mitteln der theoretischen Informatik zu formalisieren.

⁶Weitere Informationen bietet der Abschnitt 7.3.

⁷Im Detail wird dies im Abschnitt 4.3 beschrieben.

Gegeben seien zwei TURING-Maschinen $TM_{1,2}$. TM_1 ist dabei der delegierende Cloud-Nutzer und TM_2 die Cloud-Umgebung des CSP. Konkreter ausgedrückt; es handelt sich bei $TM_{1,2}$ um den Aufgabenumfang, den die jeweiligen Turing-Maschinen potenziell wahrnehmen können. Die Begrenzung des Wissens wird durch den Parameter k dargestellt.

$$TM := (\Sigma, Z, \delta, \square, z_0, E) \quad (8.1)$$

$$TM_1 := \{\delta_1, \dots, \delta_n\}, n \in \mathbb{N}$$

$$TM_2 := \{\delta_1, \dots, \delta_{n-k}\}, k = 0 \dots n$$

$$TM_2 \subseteq TM_1$$

$$\begin{array}{ll} \text{Fall 1} & k=0 \\ TM_2 = TM_1 & \text{keine Begrenzung des Wissens} \end{array} \quad (8.2)$$

$$\begin{array}{ll} \text{Fall 2} & k=n \\ TM_2 \cap TM_1 = \emptyset & \text{vollständige Wissensbegrenzung} \end{array} \quad (8.3)$$

$$\begin{array}{ll} \text{Fall 3} & 0 < k < n \\ TM_2 \subset TM_1 & \end{array} \quad (8.4)$$

$$\max_{0 < k < n} f_1(k) \quad \text{Bestrebung des Cloud Nutzers} \quad (8.5)$$

$$\min_{0 < k < n} f_2(k) \quad \text{Bestrebung des Cloud Anbieters} \quad (8.6)$$

$$\max_{0 < k < n} f_1(k) \Rightarrow \Leftarrow \min_{0 < k < n} f_2(k) \quad (8.7)$$

Die Definition der TURING-Maschinen $TM_{1,2}$ erfolgt nach der deterministischen Turingmaschine aus Gleichung 8.1. Für die Betrachtung des Prinzip der Delegation mit begrenztem Wissen erfolgt eine Reduktion dieser Gleichung auf die relevanten Bereiche, wobei die Bestandteile beider Maschinen die exakt übereinstimmen, nicht aufgezeigt werden. Dagegen wird die verschiedene Anzahl der Überföhrungsfunktionen aus δ dargestellt. Die Parameter n , als Gesamtheit des Wissens, und k , als Wissensbegrenzung, sind von zentraler Bedeutung. Der Aufgabenumfang den die Cloud-Umgebung wahrnehmen kann, wird dabei als Teilmenge der Aufgabenumfangs des Cloud-Nutzers angesehen. Die Fallunterscheidung macht zudem deutlich, dass es sich um eine echte Teilmenge handelt. Die Gleichung 8.2 zeigt den Fall, dass keinerlei Begrenzung des Wissens $k = 0$ stattfindet. Beide Turing Maschinen sind identisch. Dies entspricht einer 100%igen Auslagerung mit vollständiger Informationspreisgabe ohne Sicherheitsmaßen gegen die Cloud-Umgebung. Dieser Grenzfall ist daher nur von geringer Bedeutung für das sichere Auslagern von Aufgaben. Der zweite Fall ist das andere Extrem, in welchem die Schnittmenge der Übergangsfunktion δ von $TM_{1,2}$ der leeren Menge entspricht 8.3. Konsequenterweise kann TM_2 keinerlei Aufgabe übernehmen, die sich im Funktionsumfang von TM_1 befindet, und dem Cloud-Nutzer keinerlei Mehrwert bieten. Der in 8.4 dargestellte dritte Fall ist daher der für das Prinzip der Delegation mit begrenztem Wissen der entscheidende Fall, für den TM_2 eine echte Teilmenge von TM_1 ist. Abstrakt ausgedrückt ist die Cloud-Umgebungen in der Lage, einen Teil der Aufgaben des Cloud-Nutzers zu übernehmen, dabei jedoch nicht das vollständige Wissen zu erhalten. Der Cloud-Nutzer zielt demnach darauf ab das Wissen weitestgehend zu begrenzen, wie die Funktion f_1 in Gleichung 8.5 verdeutlicht. Die

Cloud-Umgebungen benötigt dagegen zur Ausführung der aufgetragenen Aufgaben Wissen über deren Umfang und strebt im Zuge der Effizienzmaximierung die Begrenzung k zu minimieren. Die Gleichung 8.6 verdeutlicht dieses Ziel.

Das Prinzip der Delegation mit begrenztem Wissen wird durch das Gleichsetzen dieser beiden Funktionen in Gleichung 8.7 deutlich. Diese Gleichung formalisiert den Widerspruch, dass mit wachsenden Aufgabenumfang n , bei konstanten Begrenzung des Wissens k , der Arbeitsumfang von TM_1 steigen muss, um den vollständigen Aufgabenumfang δ_n zu erfüllen. Diese Mehrarbeit kann durch die Reduktion von k reduziert werden.

Nicht in die Betrachtung in Gleichung 8.7 eingeschlossen, ist der erhöhte Aufgabenumfang auf der Seite des Cloud-Nutzers, der für die Begrenzung des Wissens nötig ist. Dieser ist unabhängig von δ_n , steigt analog zu k und steht ebenfalls im Widerspruch zur Bestrebung einer Delegation.

Das postulierte Prinzip der Delegation mit begrenztem Wissen gilt damit in Rahmen der Arbeit als formal bestätigt. Wer gegen dieses Prinzip verstößt, wäre in der Lage, einen Arbeitsumfang ohne Wissen über diesen, korrekt und jederzeit reproduzierbar auszuführen. Dabei handelt es sich offensichtlich um Zufall oder für deterministisch arbeitende Systeme, wie einer Cloud-Umgebung oder einem Computer, um einen Widerspruch.

Folglich gilt:

Deterministische Systeme unterliegen stets dem Prinzip der Delegation mit begrenztem Wissen.

Teil IV

Anhang

Auflistung der Evaluierungsergebnisse

Im Folgenden werden die Ergebnisse der Evaluierung vollständig aufgelistet. Jede Tabelle stellt dabei eine der untersuchten Technologien dar. Der Name der Technologie sowie die Referenz finden sich innerhalb der Tabellenbeschriftung.

Tabelle A.1.: Encryption Layer [RBK⁺14a]

keywords, taxonomy	Secure Data Storage	
basics	Hybrid Cloud, OpenNebula, JBoss Cluster, AES	
data in- //output	data // encrypted data	
is efficient?	true	
adjustments in Data // BL // UI	no* // no** // no	
overhead	estimation	medium
	details	50%-75% overall system
Cloud Model (Private // IaaS // PaaS // SaaS)	yes***** // yes***** // no // no	
cost	Technology (key)	medium (2)
	Details	JBoss Cluster, OpenNebula, private cloud
	Operation (key)	medium (2)
	Details	administration of the private cloud
	Development (key)	medium-high**** (3)
	Details	-
	Overall Estimation	7
Key-Mgmt	exist?	yes
	Details	master key on client and data keys at provider, no keys at public CSPs
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	OpenNebula, WebDav, JBoss Cluster, AES, CryptDB, Bouncy Castle
	Open Source?	no

*database schema annotations for databases, non for files, **just server addressing, ****mostly for private cloud environment, *****hybrid cloud both are necessary

Tabelle A.2.: CryptDB [PZB11]

keywords, taxonomy	data storage, data retrieval	
basics	SQL aware Encryption, Adjustable Query Encryption, RND, DET, OPE, HOM, multilevel Encryption	
data in- //output	SQL queries, DBMS // gesichertes DBMS	
is efficient?	true	
adjustments in Data	// BL // UI	yes* // yes** // no
overhead	estimation	low
	details	27% throughput, 0,64ms latency
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // yes // no // no
cost	Technology (key)	low (1)
	Details	multilevel encryption schema, order preserving encryption, AES,
	Operation (key)	low (1)
	Details	administration Proxy + Cloud Server mit DBMS
	Development (key)	low - very high (4,5)
	Details	actual implementation lacks in practicability, an re-implementation is necessary
	Overall Estimation	6,5
Key-Mgmt	exist?	yes
	Details	at Proxy, or Proxy(data key)+User (master key)
sharing possible	no	
native scaling included	no	
native backup included	yes	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes
	Technology	mysql, c++, lua,
	Open Source?	git://g.csail.mit.edu/cryptdb
*database schema annotations, **very low, Proxy connection, instead of DB Server		

Tabelle A.3.: Relational Cloud [CJP⁺11]

keywords, taxonomy	Database-as-a-Service	
basics	CryptDB, efficient multi-tenancy, elastic scalability, Privacy	
data in- // output	SQL queries // secure DB-as-a-Service	
is efficient?	true	
adjustments in Data // BL // UI	yes* // yes** // no	
overhead	estimation	low
	details	Overall throughput drops by an average of 22.5%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // yes // yes	
cost	Technology (key)	low (1)
	Details	database-as-a-service, JDBC, MySQL, PostgreSQL
	Operation (key)	low (1)
	Details	-
	Development (key)	low - very high (4,5)
	Details	in theory it is low but implementation of CryptDB is not suitable for practical use
	Overall Estimation	6,5
Key-Mgmt	exist?	yes
	Details	client
sharing possible	no	
native scaling included	yes	
native backup included	yes	
natively high available	yes	
Data Security	C // I // A	yes // no // yes
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes
	Technology	java, jdbc
	Open Source?	no
*database schema annotations, **usage of JDBC client		

Tabelle A.4.: PKIS [PPL11]

keywords, taxonomy	document storage, document management, data quering	
basics	keyword index search in relational DB for documents, AES, SHA-1	
data in- //output	keywords, DBMS // secure DBMS for documents	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes**	
overhead	estimation	low
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	MS-SQL
	Operation (key)	medium (2)
	Details	administration, group manager necessary
	Development (key)	medium (2)
	Details	-
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	Group Manager und User (on client)
sharing possible	yes	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	yes
	Technology	c, MS-SQL, SHA-1, AES, MS-SQL Server 2000
	Open Source?	no
**medium, *Group Manager necessary		

Tabelle A.5.: SR-ORAM [WS12]

keywords, taxonomy	data storage, data retrieval	
basics	ORAM, Bloom Filter	
data in- // output	data // secure data access, secure data storage	
is efficient?	false	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	high
	details	$O(\log n) + \text{Query}, O(\log^2 n \log \log n) + \log n$ Clientstorage
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	Client Speicher, for a 1TB DB approx. 2GB RAM for bloom filter creation
	Operation (key)	medium (2)
	Details	-
	Development (key)	medium - high (3)
	Details	-
Key-Mgmt	Overall Estimation	7
	exist?	yes
	Details	client bloom filter generation
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	bloom filter
	Open Source?	no

Tabelle A.6.: Shroud [LPM⁺13]

keywords, taxonomy	data storage, data retrieval	
basics	ORAM, secure hardware, PIR, binary tree algorithm	
data in- //output	data // secure data access, secure data storage	
is efficient?	false	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	high
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes* // yes* // no // no	
cost	Technology (key)	high (4)
	Details	secure coprocessors
	Operation (key)	low (1)
	Details	-
	Development (key)	medium - high (3)
	Details	-
Overall Estimation		8
Key-Mgmt	exist?	yes
	Details	in secure coprocessors
sharing possible	no	
native scaling included	yes	
native backup included	yes	
natively high available	yes	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	C (2898 lines), 3DES, SHA-1, HMAC, UMAC, C# server
	Open Source?	no
*with secure coprocessors		

Tabelle A.7.: Oblivstore [SS13]

keywords, taxonomy	data storage, data retrieval	
basics	ORAM, AES, secure hardware, TPM	
data in- //output	files // secure data access, secure data storage	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	high
	details	40-50x i/o overhead**
Cloud Model (Private // IaaS // PaaS // SaaS)	yes* // yes* // no // no	
cost	Technology (key)	medium (2)
	Details	TPM and secure coprocessors (optional), private cloud (optional)
	Operation (key)	low or high (2,5)
	Details	depends on chosen scenario With or without a private cloud
	Development (key)	medium (2)
	Details	-
Key-Mgmt	Overall Estimation	6,5
	exist?	yes
	Details	on client
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	C#, 9k LOC, Amazon EC2
	Open Source?	yes***

*two different scenarios are possible, one hybrid cloud model and one public cloud with trusted hardware (TPMs or secure co-processors), **I/O overhead. OblivStore incurs about 40X-50X I/O overhead under parameters used in our experiments, i.e., to access one data block, on average 40-50 data blocks need to be accessed. ***If you are looking for the OblivStore or Multi-Cloud ORAM implementations, you can obtain them by contacting Emil Stefanov.

Tabelle A.8.: CryptTree [GMSW06]

keywords, taxonomy	data storage	
basics	encrypted file system	
data in- //output	files // secure file system	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // yes*	
cost	Technology (key)	low (1)
	Details	-
	Operation (key)	low (1)
	Details	administration
	Development (key)	low - high (2,5)
	Details	depends on usage of wuala storage or not
Key-Mgmt	Overall Estimation	4,5
	exist?	yes
	Details	classic access control list
sharing possible	yes	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Java, RSA-1024, AES 128, Java Cryptography Extension(JCE)
	Open Source?	no, comercial usage: Wuala Cloud Storage
*wuala cloud storage: https://www.wuala.com		

Tabelle A.9.: CS2 [KBR11]

keywords, taxonomy	data storage	
basics	symmetric searchable encryption, search authenticators, proof of storage	
data in- //output	files, keywords // secure filesystem with keyword search	
is efficient?	true	
adjustments in Data // BL // UI		no // no // no
overhead	estimation	medium
	details	indexing high, otherwise low
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // no // no
cost	Technology (key)	low (1)
	Details	-
	Operation (key)	low (1)
	Details	-
	Development (key)	high (4)
	Details	no source code, reimplementation
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	client
sharing possible		no
native scaling included		no
native backup included		no
natively high available		no
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes
	Technology	C++, Microsoft Cryptography API: Next Generation (CNG), Microsoft Bignum lib, AES
	Open Source?	-

Tabelle A.10.: Cloud Filter [PP12]

keywords, taxonomy	data separation, data distribution	
basics	data separation, multiple cloud	
data in- //output	data // separated data	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	medium
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	no // no // no // yes	
cost	Technology (key)	high (4)
	Details	DLP, Surveillance, high investment costs for DLP
	Operation (key)	low (1)
	Details	administration
	Development (key)	high (4)
	Details	-
	Overall Estimation	9
Key-Mgmt	exist?	yes
	Details	internal
sharing possible	yes	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	XMP
	Open Source?	no

Tabelle A.11.: MONOMI [TKMZ13]

keywords, taxonomy	data storage, data retrieval	
basics	CryptDB, split client/server execution	
data in- // output	SQL Queries, DBMS // secure DBMS	
is efficient?	true	
adjustments in Data // BL // UI	yes* // no // yes**	
overhead	estimation	low
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	proxy server for SQL query encryption, design, plan execution
	Operation (key)	low (1)
	Details	administration Proxy + Cloud Server mit DBMS
	Development (key)	low - very high (4,5)
	Details	-
Key-Mgmt	Overall Estimation	7,5
	exist?	yes
sharing possible	Details	at Proxy, or Proxy(Datakey)+User (Masterkey)
		no
native scaling included	no***	
native backup included	no***	
natively high available	no***	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes
	Technology	?
	Open Source?	no
*Datenbank Schema Annotationen, **Datenbank schema Annotationen, *** nicht implizit, aber moeglich vergleiche Relational Cloud		

Tabelle A.12.: ConfiChair [ABR12]

keywords, taxonomy	conference management system	
basics	symmetric/asymmetric Encryption, Key Management, Unlinkability	
data in- //output	documents // secure document management system	
is efficient?	true	
adjustments in Data // BL // UI	no* // - // -	
overhead	estimation	medium
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	no // no // no // yes	
cost	Technology (key)	low (1)
	Details	SaaS
	Operation (key)	low (1)
	Details	administration, data migration
	Development (key)	medium - high (3)
	Details	depends on data migration
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	user, not CSP
sharing possible	yes	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	?
	Open Source?	no
*upload documents		

Tabelle A.13.: Plutus [KRS⁺03]

keywords, taxonomy	data storage	
basics	grouping files	
data in- //output	files // secure file system	
is efficient?	true	
adjustments in Data // BL // UI		no // yes // yes*
overhead	estimation	low
	details	comparable with SFS (Self-certifying File System)
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // no // no
cost	Technology (key)	medium (2)
	Details	client resources
	Operation (key)	no (0)
	Details	-
	Development (key)	high (4)
	Details	on clientside
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	client
sharing possible		yes
native scaling included		yes
native backup included		no
natively high available		no
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	OpenAFS, RSA, SHA-1, 3DES
	Open Source?	no
*Workflows Key Exchange		

Tabelle A.14.: Key2Cloud [ZYG12]

keywords, taxonomy	data storage, anonym access	
basics	key-policy attribute based encryption, AB-SIGN Schema	
data in- //output	files // secure cloud files system	
is efficient?	false	
adjustments in Data // BL // UI	no // yes* // yes**	
overhead	estimation	high
	details	1MB 12 sec WriteTime minimum, depends on hierarchy level
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	key server required
	Operation (key)	low (1)
	Details	master key by root user
	Development (key)	low (1)
	Details	open source
Key-Mgmt	Overall Estimation	4
	exist?	yes
	Details	key server, master key by root user
sharing possible	yes	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	yes
	Technology	Java
	Open Source?	http://sourceforge.net/projects/key2cloud/ , Libs: https://sourceforge.net/projects/kpabe , http://acsc.cs.utexas.edu/cpabe
*small, **small because of a master key,		

Tabelle A.15.: Sec2 [SMT⁺12]

keywords, taxonomy	data storage	
basics	XML Encryption, XML Signature, SAML	
data in- //output	data, files // secure cloud files system	
is efficient?	true	
adjustments in Data	// BL // UI	no // yes // yes
overhead	estimation	low
	details	-
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // yes // no // no
cost	Technology (key)	medium (2)
	Details	mobile End devices mit Smartcard, Key Server (trusted)
	Operation (key)	low (1)
	Details	-
	Development (key)	medium (2)
	Details	-
Key-Mgmt	Overall Estimation	5
	exist?	yes
	Details	client and trusted server (key server)
sharing possible	yes	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	XML Encryption
	Open Source?	no
*An optional integrity protection can be achieved by applying XML Signatures.		

Tabelle A.16.: Kamara [KL10]

keywords, taxonomy	data storage	
basics	searchable Encryption, Attribute Based Encryption, Proof of Storage	
data in- //output	files // secure cloud files system	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	-
	details	just a concept, no implementation or prototype
Cloud Model (Private // IaaS // PaaS // SaaS)	yes* // yes* // no // no	
cost	Technology (key)	medium (2)
	Details	dedicated machines Data Processor, Verifier, Token generator
	Operation (key)	low (1)
	Details	-
	Development (key)	medium (2)
	Details	-
Key-Mgmt	Overall Estimation	5
	exist?	yes
	Details	master key on client, dedicated hardware
sharing possible	yes	
native scaling included	no	
native backup included	yes	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	no
	Technology	-
	Open Source?	-
*hybrid cloud, both is necessary		

Tabelle A.17.: CloudProof [PLM⁺11]

keywords, taxonomy	data storage	
basics	broadcast encryption, key rolling	
data in- //output	files // secure cloud files system	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	latency +15%, throughput +10%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low-medium (1,5)
	Details	data reencryption im client, in MS Azure Cloud
	Operation (key)	low (1)
	Details	administration
	Development (key)	high (4)
	Details	no source code, reimplementation
	Overall Estimation	6,5
Key-Mgmt	exist?	yes
	Details	cloud does key distribution, Encryption on clients
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	MS Azure, C#(4kLOC)
	Open Source?	no

Tabelle A.18.: SiRiUS [GSMB03]

keywords, taxonomy	data storage	
basics	hash trees, AES, RSA	
data in- //output	files // secure cloud files system	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes*	
overhead	estimation	high
	details	access control informations (meta data) at the files, 2.3x - 6x slow down
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	-
	Operation (key)	low (1)
	Details	-
	Development (key)	high (4)
	Details	no source code, reimplementation
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	client
sharing possible	yes	
native scaling included	yes***	
native backup included	no**	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes
	Technology	NFS version 3, SFS toolkit
	Open Source?	no
*encryption or key management, **but supported, ***NFS		

Tabelle A.19.: SUNDR [LKMS04]

keywords, taxonomy	data storage, data integrity, access control	
basics	network file system, digital signatures, hash functions	
data in- //output	files // secure cloud files system (integrity+accessability)	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	2 Server for Block storage und consistency
	Operation (key)	low (1)
	Details	administration
	Development (key)	high (4)
	Details	no source code, reimplementation
	Overall Estimation	7
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	yes*	
native backup included	yes	
natively high available	no	
Data Security	C // I // A	no // yes // yes
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes
	Technology	NFS
	Open Source?	no
*NFS		

Tabelle A.20.: Cloud Storage [ZH10]

keywords, taxonomy	data storage, access control	
basics	broadcast encryption, role based encryption, role based access control	
data in- //output	files // secure cloud file system	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes*	
overhead	estimation	medium
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	med (2)
	Details	key management system server, role Management
	Operation (key)	medium (2)
	Details	Group Admin, Role Manager
	Development (key)	high (4)
	Details	-
	Overall Estimation	8
Key-Mgmt	exist?	yes
	Details	client
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
	exist?	no
Prototype	Technology	-
	Open Source?	-
*Rollen von Group Admin und Role Manager		

Tabelle A.21.: CloudSeal [XZY⁺12]

keywords, taxonomy	data storage, content sharing	
basics	proxy reencryption, broadcast revocation, symmetric encryption (AES)	
data in- //output	files // secure cloud file system	
is efficient?	true	
adjustments in Data // BL // UI	no // yes*** // yes***	
overhead	estimation	-
	details	estimation, no comparison without encryption was done, paper calls overhead acceptable
Cloud Model (Private // IaaS // PaaS // SaaS)	no // yes // no // no	
cost	Technology (key)	medium (2)
	Details	content provider, content delivery network
	Operation (key)	low (1)
	Details	content provider (role)
	Development (key)	high (4)
	Details	no source code, reimplemention
	Overall Estimation	7
Key-Mgmt	exist?	yes
	Details	client and content provider
sharing possible	yes	
native scaling included	yes	
native backup included	no	
natively high available	yes****	
Data Security	C // I // A	yes // no // yes****
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes*
	Technology	PHP, Apache Server, Python library boto**, OpenSSL lib, AES
	Open Source?	no

*on AWS and CloudFront, **boto: Python interface to amazon web services. <http://code.google.com/p/boto/>. ***for key management and encryption/decryption, ****content delivery system integration,

Tabelle A.22.: Twin Clouds [BSSS11]

keywords, taxonomy	data storage, data processing	
basics	Garbled Circuits, Multi Cloud approach	
data in- //output	data, processes // secure data storage, secure data processing	
is efficient?	true	
adjustments in Data // BL // UI	yes // yes // yes	
overhead	estimation	-
	details	just a concept, no implementation or prototype
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	high (4)
	Details	Trusted Cloud Umgebung (ggf. Private Cloud)
	Operation (key)	medium (2)
	Details	administration
	Development (key)	high (4)
	Details	-
	Overall Estimation	10
Key-Mgmt	exist?	yes
	Details	in trusted cloud
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	no
	Technology	-
	Open Source?	-

Tabelle A.23.: SEDIC [ZZCW11]

keywords, taxonomy	data separation, data distribution	
basics	Map Reduce, Privacy-preserving DM, Big Data	
data in- //output	data, processes, classification // secure data storage, secure data processing	
is efficient?	true	
adjustments in Data // BL // UI	yes* // yes // yes**	
overhead	estimation	-
	details	no overhead statement is possible, but the performance is evaluated
Cloud Model (Private // IaaS // PaaS // SaaS)	yes*** // yes*** // no // no	
cost	Technology (key)	high (4)
	Details	Private Cloud
	Operation (key)	medium - high (2,5)
	Details	-
	Development (key)	high (4)
	Details	-
Key-Mgmt	Overall Estimation	10,5
	exist?	yes
sharing possible	Details	in private cloud
		yes
native scaling included	yes	
native backup included	yes	
natively high available	yes	
Data Security	C // I // A	yes // no // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	FutureGrid, MapReduce, Hadoop
	Open Source?	-
*classify **classification, ***hybrid cloud, both are necessary		

Tabelle A.24.: SibF [AAW11]

keywords, taxonomy	data storage, data integrity, data retrieval	
basics	IDA, B+ Tree Index, Message Authentication Code (MAC)	
data in- //output	databases, database tables // secure database, secure query processing	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	medium
	details	improvements possible through index caching
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	client storage for root node, caches, und salt
	Operation (key)	low (1)
	Details	-
	Development (key)	medium-high (3)
	Details	-
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	root node on client + salt on client
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
	exist?	yes
Prototype	Technology	C++, Crypto++ Library 5.6.0, 3DES
	Open Source?	-

Tabelle A.25.: k-Anonymity [Swe02]

keywords, taxonomy	anonymization	
basics	k-Anonymity, anonymization	
data in- //output	data, files, database tables // k-anonymous data	
is efficient?	true	
adjustments in Data // BL // UI	yes // no // no	
overhead	estimation	low
	details	depends on data and use case
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	-
	Operation (key)	low (1)
	Details	-
	Development (key)	medium-high (3)
	Details	depends on use case
	Overall Estimation	5
Key-Mgmt	exist?	no
	Details	-
sharing possible	yes	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	different
	Open Source?	-

Tabelle A.26.: Incognito [LDR05]

keywords, taxonomy	anonymization	
basics	anonymization	
data in- //output	data, files, database tables //	k-anonymous data
is efficient?	true	
adjustments in Data // BL // UI		yes // no // no
overhead	estimation	low
	details	depends on data and use case
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // no // no
cost	Technology (key)	low (1)
	Details	k-anonymity
	Operation (key)	low - medium (1,5)
	Details	depends on data amount
	Development (key)	medium (2)
	Details	no source code, reimplementation
	Overall Estimation	4,5
Key-Mgmt	exist?	no
	Details	-
sharing possible		yes
native scaling included		no
native backup included		no
natively high available		no
Data Security	C // I // A	no // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	?
	Open Source?	-

Tabelle A.27.: GINGER [SVP⁺12]

keywords, taxonomy	verification, data processing	
basics	PCP Theorem, Verifier, Prover Konzept, proof-based calculation	
data in- //output	data, processes // result, verification	
is efficient?	false	
adjustments in Data // BL // UI	yes // yes // yes	
overhead	estimation	high
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low - high (2,5)
	Details	Client with GPU for cryptographic operations oder krypto-Hardware
	Operation (key)	low (1)
	Details	-
	Development (key)	very high (8)
	Details	-
Key-Mgmt	Overall Estimation	11,5
	exist?	no
	Details	-
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Linux
	Open Source?	no

Tabelle A.28.: APNGS [ZZY⁺12]

keywords, taxonomy	obfuscation	
basics	noise generation	
data in- //output	data, noise // data obfuscation	
is efficient?	true	
adjustments in Data // BL // UI		no // yes // no
overhead	estimation details	high through the noise
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // yes // yes
cost	Technology (key)	medium (2)
	Details	noise generation, noise create additional costs
	Operation (key)	no (0)
	Details	-
	Development (key)	low-medium (1,5)
	Details	-
	Overall Estimation	3,5
Key-Mgmt	exist?	no
	Details	-
sharing possible		no
native scaling included		no
native backup included		no
natively high available		no
Data Security	C // I // A	no // no // no
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	Simulation
	Open Source?	-

Tabelle A.29.: SPORC [FZFF10]

keywords, taxonomy	collaboration, document sharing, synchronization	
basics	AES, Fork Konsistenz, OT	
data in- //output	files // secure data sharing, synconization	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	medium
	details	depends on number of clients
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // yes // no	
cost	Technology (key)	medium (2)
	Details	Client History download, sync to join, need a core setup
	Operation (key)	no (0)
	Details	-
	Development (key)	high (4)
	Details	no source code, reimplemention
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	client
sharing possible	yes	
native scaling included	no	
native backup included	yes*	
natively high available	no	
Data Security	C // I // A	yes // yes // yes**
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Java, JavaScript, Google Wave, GWT
	Open Source?	no
*every client, **depends on client number		

Tabelle A.30.: CryptoDSP [TPPG11]

keywords, taxonomy	Signal Processing in Encrypted Domain	
basics	Signal Processing, Virtual processing unit,	
data in- //output	data, request // secure processing of data and requests	
is efficient?	false	
adjustments in Data // BL // UI	yes // yes // yes	
overhead	estimation	-
	details	paper cite: The performance of the preliminary system was encouraging
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // yes // no	
cost	Technology (key)	medium (2)
	Details	signal processing
	Operation (key)	low (1)
	Details	administration
	Development (key)	very high (8)
	Details	-
Key-Mgmt	Overall Estimation	11
	exist?	yes
	Details	client plugin
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	C++, Eucalyptus, KVM, libcrypt++, OpenMP
	Open Source?	no

Tabelle A.31.: Mylar [PSV⁺14]

keywords, taxonomy	java script framework, web applications	
basics	searchable encryption over files with different keys, java script meteor framework	
data in- //output	files, data // secure web applications	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	low
	details	+17% throughput +50ms latency
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // yes // no	
cost	Technology (key)	medium (2)
	Details	Business Logic in client web browser, JavaScript Framework
	Operation (key)	low (1)
	Details	administration
	Development (key)	low-high (2,5)
	Details	migration effort is low
Key-Mgmt	Overall Estimation	5,5
	exist?	yes
sharing possible	Details	client
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Java Script
	Open Source?	git://g.csail.mit.edu/mylar

Tabelle A.32.: DEPSKY [BCQ⁺13]

keywords, taxonomy	Secure Data Storage	
basics	Multi Cloud, PlanetLab, cloud-of-clouds storage, AES, RSA	
data in- //output	files // secure, redundant file storage	
is efficient?	true	
adjustments in Data // BL // UI	no // yes*** // no	
overhead	estimation details	medium overhead in form of higher costs, Reads latency is close to best CSP, writes latency is close to worst CSP, throughput overhead depends on file size
Cloud Model (Private // IaaS // PaaS // SaaS)	no // yes* // no // no	
cost	Technology (key)	low (1)
	Details	(only) twice costs compared to single cloud
	Operation (key)	low (1)
	Details	administration
	Development (key)	high** (4)
	Details	-
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	client(s reader, writer)
sharing possible	no	
native scaling included	yes	
native backup included	yes	
natively high available	yes	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	AES, RSA, Java 6, Java Secret Sharing, Jerasure
	Open Source?	no

*use 4 commercial cloud storage provider (Amazon S3, Windows Azure Blob Service, Nirvanic CDN, Rackspace Files)
 protocols were described, but no source code is available, additionally there is a high effort for implementing and updating all 4 interfaces to the cloud storage provider, * meta data handling is necessary

Tabelle A.33.: PP NOSQL [GZLL13]

keywords, taxonomy	Secure Data Storage	
basics	NoSQL, Pallier, Elgamal, Berkley DB	
data in- //output	key, value // privacy preserving NoSQL database	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	-
	details	unknown, authors do not compare with unencrypted solutions
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	Berkley DB
	Operation (key)	no (0)
	Details	no
	Development (key)	medium (2)
	Details	-
Key-Mgmt	Overall Estimation	3
	exist?	yes
	Details	data owner
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Java 1.7, Berkley DB, Bouncy Castle
	Open Source?	no
quality of the paper is sadly not the best, e.g. y-axis is average time uses for a single query, but ist not clear if this seconds or ms		

Tabelle A.34.: Sealed Cloud [JMR⁺14]

keywords, taxonomy	Secure Data Storage, Computation	
basics	Data center security, physical barrier to servers, key distribution, data clean up, execution environment	
data in- //output	data, processes // secure data storage and processing	
is efficient?	true	
adjustments in Data	// BL // UI	no // no // no
overhead	estimation	-
	details	authors do not any performance evaluation
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // no // no // no**
cost	Technology (key)	low - high (2,5)
	Details	TPM, Propitary Sealed Cloud, heavy lock in effekt
	Operation (key)	high (4)
	Details	administration
	Development (key)	low* (1)
	Details	-
	Overall Estimation	7,5
Key-Mgmt	exist?	yes
	Details	user, CSP
sharing possible	yes	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
	exist?	yes
Prototype	Technology	unkown
	Open Source?	no
provides very special security, but heavy vendor lock in effect, **provider offerts some SaaS		

Tabelle A.35.: Tsujii [TDF⁺13]

keywords, taxonomy	Privacy Preserving Data Processing, computing some statistical information	
basics	Pailler Crypto system, addition and multiplication on encrypted data in an computing center	
data in- //output	calculation formula, personal data comes from different client // final result	
is efficient?	false	
adjustments in Data	// BL // UI	yes // yes // yes
overhead	estimation details	high calculation of 100 data sets lasts for 20 min
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes* // yes* // no // no
cost	Technology (key)	medium (2)
	Details	2 trusted cloud Processing Center of Cryptographic Function. environment for encryption/decryption
	Operation (key)	medium (2)
	Details	administration
	Development (key)	very high (8)
	Details	-
	Overall Estimation	12
Key-Mgmt	exist?	yes
	Details	in trusted cloud
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	?
	Open Source?	
*hybrid, Trusted component necessary		

Tabelle A.36.: VMCrypt [Mal11]

keywords, taxonomy	Secure Computation	
basics	Garbled Circuits, library for secure computation, secure function evaluation, secure multiparty	
data in- //output	computation // secure computation	
is efficient?	true	
adjustments in Data // BL // UI	yes // yes // no	
overhead	estimation details	high up to 15%, but this seems not possible for secure computation**
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // yes // no	
cost	Technology (key)	medium (2)
	Details	VMCrypt Library
	Operation (key)	no (0)
	Details	no
	Development (key)	low* (1)
	Details	-
Overall Estimation		3
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Java
	Open Source?	no

*VMCrypt can be integrated into existing projects and customized without any modifications to its source code, **the paper makes not clear if these 15% overhead of VMCrypt is really the overall overhead: Finally, 84% of the running time was consumed by cryptographic operations and communication (this was measured by comparing with the running time of calculating the component). In other words, VMCrypt overhead was only 16%. It seems that 16% is the part of the VMcrypt overhead of the overall overhead.

Tabelle A.37.: Shape CPU [BPS12]

keywords, taxonomy	data processing, execution environment	
basics	secure execution environment, homomorphic encryption	
data in- //output	program in assembly code // result of the program after execution	
is efficient?	false	
adjustments in Data // BL // UI	yes // yes // yes	
overhead	estimation	high
	details	simulated hardware
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	high (4)
	Details	execution environment for simulated CPU
	Operation (key)	no (0)
	Details	-
	Development (key)	very high (8)
	Details	-
Key-Mgmt	Overall Estimation	12
	exist?	yes
	Details	asymmetric homomorphic encryption schema (means server reencryptions on server with public keys)
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Java
	Open Source?	yes, https://hccrypt.com/shape-cpu/

Tabelle A.38.: FHE [Gen09]

keywords, taxonomy	data processing	
basics	fully homomorphic encryption	
data in- //output	data // processed, but encrypted data	
is efficient?	false	
adjustments in Data // BL // UI	yes // yes // yes	
overhead	estimation	high
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	encryption and decryption on client side
	Operation (key)	no (0)
	Details	-
	Development (key)	very high (8)
	Details	-
Key-Mgmt	Overall Estimation	10
	exist?	yes
Key-Mgmt	Details	client
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	C, Java
	Open Source?	yes, https://hccrypt.com/
https://hccrypt.com/ , Workshop on Applied Homomorphic Cryptography was held in association with Financial Crypto and Data Security 2013		

Tabelle A.39.: Hourglas [DJO⁺12]

keywords, taxonomy	data verification	
basics	embedded watermark, encryption with watermarks	
data in- //output	files // encrypted files	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	increased storage
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	
	Operation (key)	low (1)
	Details	-
	Development (key)	high (4)
	Details	no source code, reimplementation
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	provider
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	AWS, AES
	Open Source?	-

Tabelle A.40.: Client Proof [DLB11]

keywords, taxonomy	data storage, data verification	
basics	Zero Knowledge proofs, Encryption, verification	
data in- //output	data, proof // data storage, verified input	
is efficient?	true	
adjustments in Data // BL // UI	yes // yes // yes	
overhead	estimation	high
	details	calculations on client side
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	Client side calculations, generate Zero Knowledge proof at client side
	Operation (key)	low (1)
	Details	-
	Development (key)	high (4)
	Details	on client and server side
	Overall Estimation	7
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	no
	Technology	-
	Open Source?	-

Tabelle A.41.: Venus [SCC⁺10]

keywords, taxonomy	data verification	
basics	client verification, client communication for verification	
data in- //output	files // file verification	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes**	
overhead	estimation	low
	details	client storage for hash root
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	only works with ≥ 2 clients online
	Operation (key)	low (1)
	Details	-
	Development (key)	high (4)
	Details	no source code, reimplemention
	Overall Estimation	6
Key-Mgmt	exist?	no
	Details	-
sharing possible	yes	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // yes // no
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	Phython 2.6.3, Amazon S3, SHA-1
	Open Source?	no
**client adjustments		

Tabelle A.42.: HAIL [BJO09]

keywords, taxonomy	high availability, integrity protection	
basics	proofs of retrievability	
data in- //output	files // integrity and availability proof	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	medium
	details	increased storage through redundancy
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	small pieces of every file for verification
	Operation (key)	low (1)
	Details	-
	Development (key)	high (4)
	Details	no source code, reimplemention
	Overall Estimation	6
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	yes	
native backup included	yes	
natively high available	yes	
Data Security	C // I // A	no // yes // yes
Data Privacy	UL // TR // I	no // no // yes
Prototype	exist?	yes
	Technology	C++, RSA BSAFE C library, Jerasure
	Open Source?	no

Tabelle A.43.: Dyn. VM Kerberos [LFB12]

keywords, taxonomy	authentication, outsourced calculation, trustworthy VMs	
basics	Kerberos, Tomcat, EC2, Eucalyptus, DNS	
data in- //output	login, calculation requests // secure authentication in distributed systems	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	low
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	high (4)
	Details	Kerberos Server, (Private Cloud)
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium (2)
	Details	-
	Overall Estimation	7
Key-Mgmt	exist?	yes
	Details	Kerberos Server
sharing possible	yes	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes* // yes* // yes*
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Java, AWS, Eucalyptus
	Open Source?	no
*not for calculations		

Tabelle A.44.: AnonymusCloud [KH12]

keywords, taxonomy	anonymity, authentication, privacy-preserving computation	
basics	Tor anonymizing circuit, public-key anonymous authentication	
data in- //output	calculation, credentials, access token // anonymous authentication, privacy preserving computation	
is efficient?	true	
adjustments in Data // BL // UI	yes* // yes* // no	
overhead	estimation details	medium communication overhead depends on tor circuit length
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	TTP in form of a manager
	Operation (key)	no (0)
	Details	-
	Development (key)	medium (2)
	Details	CPs provide computation services to customers (C), who submit computations as jobs
	Overall Estimation	3
Key-Mgmt	exist?	yes
	Details	TTP manager
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // no // no
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	yes
	Technology	Java
	Open Source?	no

Tabelle A.45.: SPICE [CHHY12]

keywords, taxonomy	Identity Management	
basics	DIM (digital identity management) group signatures, randomized signatures, anonymous login	
data in- //output	user id, login // secure login, single sign on, secure IDM	
is efficient?	true	
adjustments in Data	// BL // UI	no // yes // no
overhead	estimation	low
	details	-
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // yes // yes // no
cost	Technology (key)	medium (2)
	Details	secure and trusted storage for credentials (called registrar)
	Operation (key)	no - low (0,5)
	Details	administration for inhouse registrar (optional)
	Development (key)	high (4)
	Details	on client side
	Overall Estimation	6,5
Key-Mgmt	exist?	yes
	Details	registrar
sharing possible	-	
native scaling included	yes*	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // yes**
Data Privacy	UL // TR // I	yes // yes // yes
	exist?	no
Prototype	Technology	-
	Open Source?	-
*delegation of DIM intended, **depends on registrar		

Tabelle A.46.: PPDIM [BPFS09]

keywords, taxonomy	Identity Management	
basics	Zero Knowledge Beweise, access control, DIM	
data in- //output	user id, login // secure login, single sign on, secure IDM	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	secure trustworthy storage for credentials (named registrar)
	Operation (key)	no - low (0,5)
	Details	administration for inhouse registrar (optional)
	Development (key)	medium - high (3)
	Details	on client side high, on serverside medium (CSP)
	Overall Estimation	5,5
Key-Mgmt	exist?	yes
	Details	registrar
sharing possible	-	
native scaling included	yes*	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // yes**
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	java, JSP, java servlet
	Open Source?	no
*delegation of DIM intended, **depends on registrar		

Tabelle A.47.: Poll [RBO⁺10]

keywords, taxonomy	Identity Management	
basics	calculation on encrypted data, DIM, active bundle, multiparty computation	
data in- //output	user id, login // secure IDM	
is efficient?	false	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	high
	details	communication, transmitting the hole Active Bundle (incl. VM)
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	high (4)
	Details	environment for VM execution (active bundles)
	Operation (key)	no (0)
	Details	-
	Development (key)	very high (8)
	Details	technology obsolete?
Key-Mgmt	Overall Estimation	12
	exist?	yes
Key-Mgmt	Details	inside the VM
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	active bundle, VM
	Open Source?	no

Tabelle A.48.: Angin [ABR⁺10]

keywords, taxonomy	Identity Management	
basics	Zero Knowledge proofs (Fiat-Shamir), anonym identification, DIM, include VM, active bundles	
data in- //output	user id, login // secure IDM, secure login	
is efficient?	false	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	high
	details	communication, many roundtrips, integrated VM
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	high (4)
	Details	environment for VM execution (active bundles)
	Operation (key)	no (0)
	Details	-
	Development (key)	very high (8)
	Details	technology obsolete?
Overall Estimation	12	
Key-Mgmt	exist?	yes
	Details	inside the VM
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	yes
	Technology	Java mobile agent framework JADE
	Open Source?	no

Tabelle A.49.: U-Prove [MR14]

keywords, taxonomy	anonymization, authentication, Microsoft Research Project	
basics	zero knowledge proofs, unlinkability, WS-federation protocol, token-based	
data in- //output	credentials, (certificate) // anonym authentication	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	additional party for creating anonym credentials
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // yes // no	
cost	Technology (key)	medium (2)
	Details	Invocation Helper, Plugin for InternetExplorer, TTP in Form of an Issuers (provides tokens)*
	Operation (key)	no - low (0,5)
	Details	administration for inhouse issuer (optional)
	Development (key)	medium (2)
	Details	Cloud and Silverlight components
Key-Mgmt	Overall Estimation	4,5
	exist?	yes
sharing possible	Details	Issuer, Client
		-
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	yes
	Technology	C#, Java
	Open Source?	yes, from Microsoft, but OpenSource
*Issuer have not to be online, when client transmitting its ZKP to the SP, https://www.prime-project.eu http://primelife.ercim.eu		

Tabelle A.50.: Idemix [IRZ14]

keywords, taxonomy	anonymization, authentication	
basics	Zero Knowledge Proof	
data in- //output	credentials // anonym authentication	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	additional party for creating anonym credentials
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	TTP in form of an Issuer*
	Operation (key)	no - low (0,5)
	Details	administration for inhouse issuer (optional)
	Development (key)	low - medium (1,5)
	Details	-
Key-Mgmt	Overall Estimation	3
	exist?	yes
Key-Mgmt	Details	Issuer, Client
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	yes
	Technology	Java
	Open Source?	yes, https://prime.inf.tu-dresden.de/idemix/
*Issuer have not to be online, when client transmitting its ZKP to the SP, https://www.prime-project.eu http://primelife.ercim.eu		

Tabelle A.51.: PRAM [XYM⁺13]

keywords, taxonomy	Identity Management, Access Control, SLA, authentication	
basics	blind signature and hash chain technology	
data in- //output	user id, login // secure IDM, secure access management	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	low, but 5 parties: CSP, User, Register, IdP, CSP-Verify
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	5 (different) parties necessary, Registrar stored credentials
	Operation (key)	low (1)
	Details	administration
	Development (key)	high (4)
	Details	on CSP side
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	registrar
sharing possible	-	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // no // yes*
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	no
	Technology	-
	Open Source?	-
*depends on registrar		

Tabelle A.52.: Sibboleth

keywords, taxonomy	authentication, authorization	
basics	privacy-preserving access control, single sign on, SAML	
data in- //output	user id, password // secure authentication, authorization	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	low, Identity provider mandatory
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // yes // yes	
cost	Technology (key)	medium (2)
	Details	identity provider required
	Operation (key)	no - low (0,5)
	Details	administration for inhouse identity provider (optional)
	Development (key)	low (1)
	Details	-
Key-Mgmt	Overall Estimation	3,5
	exist?	yes
Key-Mgmt	Details	identity provider
	Details	identity provider
sharing possible	-	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	java, c++
	Open Source?	yes, opensource

Tabelle A.53.: Kerberos [NT94]

keywords, taxonomy	authentication, authorization, distributed systems, certificates	
basics	symmetric or asymmetric Encryption, Key Management, Needham-Schroeder-Protokoll	
data in- // output	user id, password // secure authentication, authorization in distributed systems	
is efficient?	true	
adjustments in Data	// BL // UI	no // yes // no
overhead	estimation details	low additional server needed
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // yes // no // no
cost	Technology (key)	medium (2)
	Details	Server: Ticket, Authentication, Key Distribution, Ticket Granting
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium (2)
	Details	-
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	Kerberos servers
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	yes	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	different
	Open Source?	yes, http://www.h51.org , http://modauthkerb.sourceforge.net/index.html , http://web.mit.edu/kerberos/www/

Tabelle A.54.: DIAMETER [Ven01]

keywords, taxonomy	authentication, protocol, distributed systems	
basics	RADIUS, preshared Secret, Hashing, IPSec, TLS	
data in- //output	user id, password // secure authentication in distributed systems	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	additional server needed
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	authentication server, Key Storage Server
	Operation (key)	low (1)
	Details	administration
	Development (key)	low (1)
	Details	-
Key-Mgmt	Overall Estimation	4
	exist?	yes
	Details	Diameter servers
sharing possible	-	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	different
	Open Source?	yes, http://www.openimscore.org/docs/JavaDiameterPeer/main.html , http://www.freediameter.net/trac/ , https://github.com/fiorix/go-diameter

Tabelle A.55.: ZK [FFS88]

keywords, taxonomy	authentication	
basics	Fiat Shamir algorithm	
data in- //output	- // secure authentication	
is efficient?	true	
adjustments in Data // BL // UI		no // yes // no
overhead	estimation	-
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // no // no
cost	Technology (key)	low (1)
	Details	store client secret
	Operation (key)	no (0)
	Details	-
	Development (key)	medium (2)
	Details	-
	Overall Estimation	3
Key-Mgmt	exist?	no
	Details	-
sharing possible		-
native scaling included		no
native backup included		no
natively high available		no
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	yes // no // no
Prototype	exist?	yes
	Technology	different
	Open Source?	yes

Tabelle A.56.: CR-Auth

keywords, taxonomy	authentication	
basics	symmetric/asymmetric encryption, oneway function	
data in- //output	user id // secure authentication	
is efficient?	true	
adjustments in Data // BL // UI		no // yes // no
overhead	estimation	-
	details	-
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // no // no
	Technology (key)	low (1)
	Details	-
	Operation (key)	no (0)
	Details	-
cost	Development (key)	medium (2)
	Details	-
	Overall Estimation	3
Key-Mgmt	exist?	yes
	Details	client
sharing possible		-
native scaling included		yes
native backup included		no
natively high available		yes
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // no // no
	exist?	yes
Prototype	Technology	different
	Open Source?	no, Nutzung in GSM Netzen zur Authentifizierung

Tabelle A.57.: Log. Attestation [GWP⁺11]

keywords, taxonomy	authentication, authorization	
basics	credentials-based authorization (CBA), TPM, access control	
data in- //output	credentials // secure authentication, authorization	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	high
	details	1ms example Autorisierung, medium - high, OS overhead + logical attestation overhead
Cloud Model (Private // IaaS // PaaS // SaaS)	yes* // yes* // no // no	
cost	Technology (key)	medium (2)
	Details	Hardware with TPM, Nexus OS
	Operation (key)	no (0)
	Details	-
	Development (key)	high-very high (6)
	Details	-
Key-Mgmt	Overall Estimation	8
	exist?	yes
Key-Mgmt	Details	TPM
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	TPM, hardware based, OS, Python, Java
	Open Source?	no
*with TPM		

Tabelle A.58.: SAPPHIRE [PPL12]

keywords, taxonomy	authentication, authorization, collaboration, patient management	
basics	Kerberos, anonymization, TSL, AES	
data in- //output	health and wellness data // anonymous data, secure authentication	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	Kerberos Server
	Operation (key)	low (1)
	Details	administration
	Development (key)	high (4)
	Details	source code is not available, reimplementation
	Overall Estimation	7
Key-Mgmt	exist?	yes
	Details	master key client, Kerberos Server
sharing possible	yes	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	unkown
	Open Source?	no

Tabelle A.59.: OASIS [BMY02]

keywords, taxonomy	access control, anonymization, certificate, distributed system	
basics	RBAC, SLA, Zertifikate, Public Key Infrastructure (PKI)	
data in- //output	user, roles, SLA, policies // secure access control model	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	RBAC
	Operation (key)	medium (2)
	Details	administration (of the system), role management
	Development (key)	high (4)
	Details	source code is not available, reimplementation
	Overall Estimation	7
Key-Mgmt	exist?	no
	Details	-
sharing possible	yes*	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // no // no
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	PostgreSQL
	Open Source?	no
used in national HealthRecord in UK, *delegation of roles		

Tabelle A.60.: SSF DAC [YWRL10]

keywords, taxonomy	access control, data storage	
basics	ABE, Proxy-Reencryption. Public Key	
data in- //output	user, roles, data // secure access control model, secure data storage	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	medium
	details	$O(N)$, storage overhead= ID + HEADER(includes encrypted attributes) per file
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	TTP Auditor (optional)
	Operation (key)	no (0)
	Details	-
	Development (key)	medium (2)
	Details	-
	Overall Estimation	3
Key-Mgmt	exist?	yes
	Details	client
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	no
	Technology	-
	Open Source?	-

Tabelle A.61.: Secure Audit Logs [SK99]

keywords, taxonomy	Secure Logging, Auditing	
basics	RBAC, hash chaining	
data in- //output	logfile // trustable logging procedure	
is efficient?	true	
adjustments in Data	// BL // UI	yes // yes // yes
overhead	estimation	-
	details	additional traffic since every log entry is sent, authors do not any performance evaluation
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // no // no
cost	Technology (key)	low (1)
	Details	no technical details are described
	Operation (key)	no - low (0,5)
	Details	minor administration tasks
	Development (key)	medium (2)
	Details	-
Key-Mgmt	Overall Estimation	3,5
	exist?	yes
	Details	trusted machine
sharing possible		yes
native scaling included		no
native backup included		yes
natively high available		no
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	unkown
	Open Source?	no
described the theoretical principle and protocols with practical discussions		

Tabelle A.62.: Knox [WJCN09]

keywords, taxonomy	anonymization, integrity, auditing	
basics	homomorphic MACs (Integrity check), Provable data possession (PDP), Identity Privacy	
data in- //output	data, user group // integrity check for cloud storage	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // yes*	
overhead	estimation details	medium Overhead not depends on number of users, 100 users and 2GB data means 0,33GB signature size, 106,4KB communication costs and 3,4 sec duration for audit
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key) Details	low (1) TTP Auditor (semitrusted)
	Operation (key) Details	low (1) group manager
	Development (key) Details	low-medium (1,5) -
	Overall Estimation	3,5
Key-Mgmt	exist?	no
	Details	-
sharing possible	yes**	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	GMP and PBC libraries
	Open Source?	no
*for group mgmt (key mgmt), ** inside of user groups		

Tabelle A.63.: RAFT [BDJ⁺11]

keywords, taxonomy	Reliability, fault tolerance	
basics	Challenge Protocol, POR, PDP	
data in- //output	redundant cloud storage // proofed redundant cloud storage	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	medium
	details	easy integration in Mozy
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	-
	Operation (key)	no (0)
	Details	-
	Development (key)	medium (2)
	Details	-
	Overall Estimation	3
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	no	
native backup included	yes*	
natively high available	yes*	
Data Security	C // I // A	no // no // yes
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	-
	Open Source?	no (Raptor Code software package?)
developed by RSA Laboratories (EMC Cooperation) * tests for redundancy at CSP side		

Tabelle A.64.: CAD [ZK12]

keywords, taxonomy	Reliability, fault tolerance, data concistency	
basics	global dependency graph (GDG), cycle detection in graphs	
data in- //output	arbitrary cloud applications, BusinessLogic/persistence layer in Cloud // quantitative statement about data consistency, number how often anomalies occurs	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	overhead for CollectorAgent <3%
Cloud Model (Private // IaaS // PaaS // SaaS)	no // no // yes* // no	
cost	Technology (key)	medium (2)
	Details	Server for Benchmarks
	Operation (key)	low (1)
	Details	administration
	Development (key)	low (1)
	Details	-
	Overall Estimation	4
Key-Mgmt	exist?	no
	Details	-
sharing possible	-	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // no // yes
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	Java, AspectJ, JMeter
	Open Source?	no
*Cloud Storages(Tests with Google App Engine datastores, Cassandra)		

Tabelle A.65.: C3 [BDA⁺10]

keywords, taxonomy	compliance, data location	
basics	DSL, Middleware, Compliance Level Agreements (CLAs), SLA, Framework/Service	
data in- //output	CLA, requirements specified via DSL // cloud application with DSL specifics	
is efficient?	true	
adjustments in Data	// BL // UI	yes // yes // yes
overhead	estimation details	low -
Cloud Model (Private	// IaaS // PaaS // SaaS)	no // no // yes* // yes
cost	Technology (key)	medium (2)
	Details	C3 PaaS Provider necessary
	Operation (key)	medium (2)
	Details	admin + data model specification via DSL
	Development (key)	low - high (2,5)
	Details	application specific, data model specification via DSL
Key-Mgmt	Overall Estimation	6,5
	exist?	no
sharing possible	Details	-
		-
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // no // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	eigene DSL,
	Open Source?	no
developed by T-Systems, * C3 PaaS Provider only		

Tabelle A.66.: SCM [BKNS12]

keywords, taxonomy	maintenance, privacy	
basics	maintenance agent/ management, Privilege Levels	
data in- //output	own (private) cloud // own (private) cloud with secure maintenance	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	paper cite: does not have significant impact on the efficiency of the infrastructure cloud
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	high (4)
	Details	own (private) cloud, Trust Database, Maintenance Agent
	Operation (key)	medium (2)
	Details	cloud administration
	Development (key)	low (1)
	Details	-
Key-Mgmt	Overall Estimation	7
	exist?	no
sharing possible	Details	-
		-
native scaling included	-	
native backup included	yes	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	python, OpenStack, SELinux
	Open Source?	no

Tabelle A.67.: LOST [WL12]

keywords, taxonomy	data storage, compliance, data location	
basics	POL (proof of location), POR	
data in- //output	redundant cloud storage // proofed redundant cloud storage with files at different locations*	
is efficient?	true	
adjustments in Data	// BL // UI	no // no // no
overhead	estimation details	medium communication, computation and storage overhead
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // yes // no // no
cost	Technology (key)	medium (2)
	Details	redundant storage nodes
	Operation (key)	no (0)
	Details	-
	Development (key)	low-medium (1,5)
	Details	-
Key-Mgmt	Overall Estimation	3,5
	exist?	yes
	Details	client, CSP(only part of the key)
sharing possible	-	
native scaling included	-	
native backup included	-	
natively high available	-	
Data Security	C // I // A	no // yes // yes
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	?
	Open Source?	no, but tests with AWS

*error-prone, In particular we show that in this restricted case, a set of colluding storage servers who can freely copy the files can always succeed in breaking the security guarantee of the system (assurance about file location).

Tabelle A.68.: IRM (EDRM)

keywords, taxonomy	technical principle, encryption, authorization	
basics	encryption, digital right management, digital signatures (watersigns), right definition	
data in- //output	data, rights, license model // encrypted digital signed data	
is efficient?	true	
adjustments in Data	// BL // UI	no // yes // yes
overhead	estimation details	medium encryption/decryption overhead, communication overhead
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // yes // no // no
cost	Technology (key)	medium (2)
	Details	Content Server, License Server*, client encryption/verification
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium (2)
	Details	-
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	Licence Server
sharing possible	yes	
native scaling included	no**	
native backup included	no**	
natively high available	no**	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	different
	Open Source?	-
*see http://de.wikipedia.org/wiki/Digitale_Rechteinverwaltung , **but possible, proprietary software examples: http://www.adobe.com/de/products/livecycle/modules.displayTab3.html , Authentica Active Rights Management (EMC), http://technet.microsoft.com/de-de/library/cc771234(v=ws.10).aspx , http://www.safenet-inc.de/software-monetization/sentinel-rms/?LangType=1031 , http://www.information-rights-management.com		

Tabelle A.69.: Privacy as a Service [AC11]

keywords, taxonomy	privacy	
basics	SLA, Quality of Service	
data in- //output	cloud service // cloud service with privacy enfacements	
is efficient?	unknown	
adjustments in Data // BL // UI	yes*,** // yes** // yes**	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)		yes // yes // yes // no
	Technology (key)	low (1)
	Details	TTP as PaaS Provider
	Operation (key)	no (0)
cost	Details	-
	Development (key)	low-high** (2,5)
	Details	-
	Overall Estimation	3,5
Key-Mgmt	exist?	no
	Details	-
sharing possible		yes
native scaling included		yes***
native backup included		yes***
natively high available		yes***
Data Security	C // I // A	no // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
	exist?	no
Prototype	Technology	-
	Open Source?	-
*data classification, like declared in SLA and privacy rules, **data classification is not mentioned in this work,***provided by the TTP		

Tabelle A.70.: SaS Cryptoservice [XS07]

keywords, taxonomy	authentication, distributed system, key management	
basics	3 factor authentication, key disabling, large scale networks, crypto key distribution	
data in- //output	cryptokeys // secure key protection	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	large scale network
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium* (2)
	Details	-
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	network, distributed
sharing possible	no	
native scaling included	yes	
native backup included	yes	
natively high available	yes	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
	exist?	no
Prototype	Technology	-
	Open Source?	-
*for integration in existing systems		

Tabelle A.71.: PPF SOA [ACEW12]

keywords, taxonomy	privacy, privacy aware architecture	
basics	SOA, Framework, Privacy	
data in- //output	SOA webservice // privacy standard for SOA	
is efficient?	true	
adjustments in Data // BL // UI	yes* // yes* // yes*	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	TTP as Auditing Service (AdS)
	Operation (key)	no (0)
	Details	-
	Development (key)	low-high** (2,5)
	Details	-
	Overall Estimation	3,5
Key-Mgmt	exist?	no
	Details	-
sharing possible	yes	
native scaling included	yes***	
native backup included	yes***	
natively high available	yes***	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	yes
	Technology	SOA
	Open Source?	no
*data classification after TS(top secret), S(secret), C(confidential),U(unclassified) and creating a privacy policy document,		
depending on existing data models, *introduction of an enterprise service bus		

Tabelle A.72.: CipherBase [ABE⁺13]

keywords, taxonomy	Trusted Computing, database	
basics	Microsoft SQL server, FPGA based secure coprocessors, AES	
data in- //output	Database, sql queries // secure database, secure query execution	
is efficient?	true	
adjustments in Data // BL // UI	yes // yes // no	
overhead	estimation	low
	details	no validation, authors said they used FPGAs in order to be faster then TrustedDB, but one can be skeptic
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	high (4)
	Details	FPGAs
	Operation (key)	medium (2)
	Details	administration FPGAs
	Development (key)	very high (8)
	Details	(FPGA programming)**
	Overall Estimation	14
Key-Mgmt	exist?	yes
	Details	Client and FPGA
sharing possible	no	
native scaling included	no*	
native backup included	no*	
natively high available	no*	
Data Security	C // I // A	yes** // yes** // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	Microsoft SQL, ODBC, FPGA
	Open Source?	no, Microsoft research
*high performance is mentioned also scalability, but there are no results at all, not even for FPGA calculation, **FGPA programming includes encryption schema implementation, this is highly critical		

Tabelle A.73.: TrustedDB [BS11]

keywords, taxonomy	Trusted Computing, secure coprocessor, database	
basics	SQL Database, IBM 4764 secure coprocessor, Query Processing, PKI, data classification	
data in- //output	Database, sql queries // secure database, secure query execution	
is efficient?	true	
adjustments in Data	// BL // UI	yes* // yes** // no
overhead	estimation details	low compared to other encryption methods it is quite low (the authors said)
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // no // no // no
cost	Technology (key)	high (4)
	Details	IBM secure coprocessor
	Operation (key)	low (1)
	Details	administration
	Development (key)	high-very high (6)
	Details	-
	Overall Estimation	11
Key-Mgmt	exist?	yes
	Details	IBM secure coprocessor and client
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	mysql, SQLite, C, IBM 4764-001 secure coprocessor
	Open Source?	no
*adding SENSITIVE tag to all sensitiv data fields in database, **encryption and decryption of the SQL queries		

Tabelle A.74.: Excalibur [SRGS12]

keywords, taxonomy	Trusted Computing	
basics	policy-sealed data, TMP, CPABE, trusted cloud services, AES	
data in- //output	data, attributes, policies // secured and sealed data	
is efficient?	true	
adjustments in Data // BL // UI	no // yes* // yes*	
overhead	estimation	low
	details	but TPM operations are inefficient
Cloud Model (Private // IaaS // PaaS // SaaS)	yes** // no** // no // no	
cost	Technology (key)	medium (2)
	Details	TPM on every cloud node in public cloud
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium (2)
	Details	no source code, reimplemention
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	client and trustworthy cloud monitor
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	yes	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	eucalyptus, c code(22k lines), OpenSSL, CPABE toolkit
	Open Source?	no, Microsoft research
*attributes have to assign to the data Keys, **with TMP		

Tabelle A.75.: CloudVerifier [SMV⁺10]

keywords, taxonomy	Trusted Computing	
basics	access control, TPM, PKI	
data in- //output	VM // integrity and access control enforcement	
is efficient?	true	
adjustments in Data // BL // UI	no // yes* // yes*	
overhead	estimation	medium
	details	generating a RSA key pair takes 1 minute, TPM operations are inefficient
Cloud Model (Private // IaaS // PaaS // SaaS)	yes** // no** // no // no	
cost	Technology (key)	medium (2)
	Details	TPM on every cloud node in public cloud
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium (2)
	Details	no source code, reimplemention
	Overall Estimation	5
Key-Mgmt	exist?	yes
	Details	client and trustworthy cloud verifier
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	eucalyptus, KVM, OSLO bootloader, netROTI, ProxyD-HCP, TFTP server
	Open Source?	no
*for key handling, **with TMP		

Tabelle A.76.: TrustVisor [MLQ⁺10]

keywords, taxonomy	Trusted Computing	
basics	Hypervisor, secure execution environment, TPM, AES, RSA	
data in- //output	Application (Pieces of application logic) // secure execution of the code	
is efficient?	true	
adjustments in Data // BL // UI	no // yes* // no	
overhead	estimation	low
	details	less than 7%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes* // no* // no // no	
cost	Technology (key)	medium (2)
	Details	TPM
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium-high** (3)
	Details	-
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	TPM/softwareTPM
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	C (TCP 6351 lines),
	Open Source?	no
*TrustVisor implements an application-level hypercall interface for registering PALs (Pieces of Application Logic), **with TPM		

Tabelle A.77.: Proxos [TMLL06]

keywords, taxonomy	Trusted Computing	
basics	Xen, Process Separation, private VM	
data in- //output	Application // private application execution	
is efficient?	true	
adjustments in Data // BL // UI	no // yes // no	
overhead	estimation	low
	details	the overhead Proxos introduces is very low (6%)
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	private VM with Proxos + commodity OS VM
	Operation (key)	low (1)
	Details	administration
	Development (key)	high* (4)
	Details	-
	Overall Estimation	7
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Xen, Linux Kernel
	Open Source?	no
*see Wang, Xiaoguang, et al. Transparent Security-Sensitive Process Protection via VMM-Based Process Shadowing.		

Tabelle A.78.: Data Capsules [MAF⁺11]

keywords, taxonomy	Trusted Computing, Secure Execution Environment	
basics	cryptography, hardware root of trust, data capsuling, theoretical concept	
data in- //output	data // secure data protection	
is efficient?	unknown	
adjustments in Data	// BL // UI	no* // no // no
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // no // no // no
cost	Technology (key)	medium (2)
	Details	optional secure Hardware
	Operation (key)	low to high (2,5)
	Details	at least administration
	Development (key)	very high (8)
	Details	-
Key-Mgmt	Overall Estimation	12,5
	exist?	yes
	Details	provided by the TCB
sharing possible	yes**	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	yes // yes // yes
Prototype	exist?	no
	Technology	-
	Open Source?	-
*data got encapsulated, **provided by the TCB		

Tabelle A.79.: Flicker [MPP⁺08]

keywords, taxonomy	Trusted Computing, Secure Execution Environment	
basics	TCB, isolation, TPM, execution	
data in- //output	lines of code // secure execution of the code	
is efficient?	true	
adjustments in Data // BL // UI	no // yes* // no	
overhead	estimation	high
	details	250 LoC for TCB, but high throughput and latency overhead
Cloud Model (Private // IaaS // PaaS // SaaS)	yes* // no* // no // no	
cost	Technology (key)	medium (2)
	Details	TPM
	Operation (key)	low (1)
	Details	administration
	Development (key)	high* (4)
	Details	-
Key-Mgmt	Overall Estimation	7
	exist?	yes
sharing possible	Details	TPM
		no
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	C
	Open Source?	no
*Flicker requires the security-sensitive code of interest to be custom-compiled and linked with very few external dependencies.		

Tabelle A.80.: GhostDB [ABB⁺07]

keywords, taxonomy	Trusted Computing, secure Hardware,	
basics	USB key, Database Separation	
data in- //output	data // distributed secure database system	
is efficient?	true	
adjustments in Data // BL // UI	yes* // yes** // yes**	
overhead	estimation	medium
	details	especially data aggregation, indexing, subtrees, bloomfilter
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	medium (2)
	Details	secure USB dongle
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium-high** (3)
	Details	-
	Overall Estimation	6
Key-Mgmt	exist?	yes
	Details	USB dongle
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	C
	Open Source?	no
*data classification **data classification, separation handling and JOIN handling		

Tabelle A.81.: PriaaS [IKC09]

keywords, taxonomy	Trusted Computing, secure coprocessor, privacy	
basics	TTP, secure coprocessors, trusted execution, software classification, data classification	
data in- // output	lines of code (software part), data // secure execution of the code	
is efficient?	true	
adjustments in Data	// BL // UI	yes* // yes* // yes**
overhead	estimation details	medium (through third party roundtrips) + data classification process (logic and data)
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // no // no // no
cost	Technology (key)	high (4)
	Details	secure coprocessor (CSP + TTP), Trusted Third Party
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium-high** (3)
	Details	-
Overall Estimation		8
Key-Mgmt	exist?	yes
	Details	TTP
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	C#, .Net framework,
	Open Source?	no
*classification, **to enable classification		

Tabelle A.82.: Terra [GPC⁺03]

keywords, taxonomy	Trusted Computing, Secure Virtualization	
basics	Trusted virtual machine monitor, TPM	
data in- //output	VM // Trusted VM Environment	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	medium (2)
	Details	TPM
	Operation (key)	low (1)
	Details	administration
	Development (key)	low - high (2,5)
	Details	This depends on the chosen model, confidentiality is only provided by the closed-box VMs.
Key-Mgmt	Overall Estimation	5,5
	exist?	yes
	Details	TPM inside the hypervisor
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes* // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	VMware GSX Server 2.0.1, Debian GNU/Linux OS, Python, OpenSSL
	Open Source?	no
Autors implement a trusted version of the online game quake.		

Tabelle A.83.: CaaS [BBI⁺13]

keywords, taxonomy	Secure Virtualization, Trusted Computing	
basics	Xen, TMP, Access Control, hypervisor Security, AES, small TCB	
data in- //output	Xen // Trusted VM Environment	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	read(MB/s) -17%, write(MB/s) -10%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes** // no // no // no	
cost	Technology (key)	low (1)
	Details	Xen Hypervisor
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium*** (2)
	Details	-
	Overall Estimation	4
Key-Mgmt	exist?	yes
	Details	TPM inside the DomC and DomT,
sharing possible	no	
native scaling included	yes*****	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Xen, Linux, SoftHSM*
	Open Source?	no***

*<http://www.opendnssec.org/softhsm/> **to have access to hypervisor/ cloud vm management, ***adjustment of the xen hypervisor since source code is not available, *****not for one client but for many different

Tabelle A.84.: SDSPF [RJ12]

keywords, taxonomy	Secure Virtualization, Trusted Computing	
basics	Xen, OpenNebula, VM Trust Environment, TPM, Page Table Encryption	
data in- //output	VM // Trusted VM Environment	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	about 20% for application memory encryption
Cloud Model (Private // IaaS // PaaS // SaaS)	yes*** // no // no // no	
cost	Technology (key)	low - medium (1,5)
	Details	OpenNebula, Tahoe, TPM
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium (2)
	Details	-
	Overall Estimation	4,5
Key-Mgmt	exist?	yes
	Details	hypervisor (CSP), integrates an overshadow like approach of memory encryption based in Shadow Page Tables used by Xen hypervisor
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	opennebula, xen, Tahoe**
	Open Source?	no

* about 20% for application memory encryption, **probably this: <http://en.wikipedia.org/wiki/Tahoe-LAFS>, **access to hypervisor/ cloud vm management,

Tabelle A.85.: CloudVisor [ZCCZ11]

keywords, taxonomy	Secure Virtualization, Nested Virtualization, Trusted Computing	
basics	TPM support, Intel TXT support, encrypted VM images, AES, Merkle Hash Tree, Xen support	
data in- //output	VM, hypervisor // Trusted VM Environment	
is efficient?	true	
adjustments in Data	// BL // UI	no // no // no
overhead	estimation details	medium moderate slow-down (4.5% - 54.5%) for I/O intensive applications and very small slowdown for other applications(0.1% - 16.8%)
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // no // no // no
cost	Technology (key) Details	medium (2) support Intel TXT and TPM (optional), Windows and Linux Guest systems
	Operation (key) Details	low (1) administration
	Development (key) Details	high*** (4) -
	Overall Estimation	7
Key-Mgmt	exist? Details	yes yes**
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes***** // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	5.5 KLOC, Xen, AES
	Open Source?	no

inside CloudVisor(at CSP, but protected) CloudVisor controls the private key of the platform and uses it to decrypt the images for booting. With the help of TPMs, *since sourcecode ist not available, ****although the authors describe an application of their approach at public clouds they assume that public CSP introduce their approach, *****with the help of TPM

Tabelle A.86.: Overshadow [CGL⁺08]

keywords, taxonomy	Secure Virtualization, Memory Encryption	
basics	VMM, AES, SHA, memory encryption	
data in- //output	applications, operating system // execution with secured memory	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	medium
	details	performance is uniformly above 80% of the baseline
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	high (4)
	Details	Virtualization Platform
	Operation (key)	low (1)
	Details	administration for inhouse virtualization platform
	Development (key)	low-high* (2,5)
	Details	-
	Overall Estimation	7,5
Key-Mgmt	exist?	yes
	Details	VMM
sharing possible	no	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	VMware
	Open Source?	no, VMware research

*By contrast, providing protection at the process level (e.g., Overshadow) is usually closely coupled with a specific type of operating system and requires non-trivial efforts when being ported to other operating systems. Said by CloudVisor guys.

Tabelle A.87.: NOVA [SK10]

keywords, taxonomy	Secure Virtualization	
basics	small TCB, Microhypervisor, memory protection, Virtualization Plaform, VM Iso-lation	
data in- //output	VM // secure execution	
is efficient?	true	
adjustments in Data	// BL // UI	no // no // no
overhead	estimation details	low 1-3 %
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // no // no // no
cost	Technology (key)	high (4)
	Details	usage of the VMM, usage of a hypervisor, each VM has its own VMM
	Operation (key)	low (1)
	Details	administration
	Development (key)	low (1)
	Details	open source
	Overall Estimation	6
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	yes
	Technology	C
	Open Source?	yes, https://github.com/udosteinberg/NOVA

Tabelle A.88.: HyperSafe [WJCN09]

keywords, taxonomy	Secure Virtualization	
basics	BitVisor, Xen, Memory Protection, Integrity Protection, hypervisor extension	
data in- //output	type 1 hypervisor // integrity protected hypervisor	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	less than 5%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	medium (2)
	Details	type 1 hypervisor
	Operation (key)	low (1)
	Details	administration
	Development (key)	medium (2)
	Details	Hyper- Safes code size is small and its integration with commodity hypervisors is straightforward
Key-Mgmt	Overall Estimation	5
	exist?	no
sharing possible	Details	-
native scaling included	no	
native backup included	yes	
natively high available	no	
Data Security	C // I // A	no // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	tboot, C, Assembly Code
	Open Source?	no

Tabelle A.89.: SSC [BLCSG12]

keywords, taxonomy	Secure Virtualization	
basics	Xen, TPM, hypervisor extension, AES	
data in- //output	Xen // Trusted VM Environment	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	1% - 7%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes** // no // no // no	
cost	Technology (key)	medium (2)
	Details	Cloud Nodes with TPM
	Operation (key)	low (1)
	Details	administration
	Development (key)	high* (4)
	Details	-
Overall Estimation		7
Key-Mgmt	exist?	yes
	Details	stored in client controls VMs, CSP cannot access any keys
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Xen
	Open Source?	no
**access to hypervisor/ cloud vm management, *since source code is not available		

Tabelle A.90.: ACPS [LDP11]

keywords, taxonomy	Secure Virtualization	
basics	Monitoring, Intrusion Detection Systems, Full Virtualization, Eucalyptus, Integrity Protection	
data in- //output	cloud management system // Intrusion Detection Systems for cloud management systems	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation details	low < 10%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key) Details	low (1) Eucalyptus
	Operation (key) Details	low (1) administration
	Development (key) Details	low (1) just setup, configuration, administration
	Overall Estimation	3
Key-Mgmt	exist?	no
	Details	-
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // yes // no
Data Privacy	UL // TR // I	no // yes* // no
Prototype	exist?	yes
	Technology	Eucalyptus, OpenECP
	Open Source?	no
* ACPS is completely transparent to guest machines: it become not clear what this means in case of a security monitoring system, did the guest machines got insight of the monitored data?		

Tabelle A.91.: SEC2 [HLMS10]

keywords, taxonomy	Secure Virtualization	
basics	Network Virtualization, User Isolation, Access Control, VLAN	
data in- //output	cloud network infrastructure // virtualized network infrastructure	
is efficient?	unknown	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	medium (2)
	Details	enhanced network switches
	Operation (key)	low (1)
	Details	administration
	Development (key)	low (1)
	Details	-
	Overall Estimation	4
Key-Mgmt	exist?	no
	Details	-
sharing possible	-	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no* // no // no
Data Privacy	UL // TR // I	no // yes // no
Prototype	exist?	no
	Technology	-
	Open Source?	-
*provide isolation between user		

Tabelle A.92.: VMwatcher [JWX07]

keywords, taxonomy	Secure Virtualization	
basics	virtual machine introspection, malware detection	
data in- //output	VM // monitored VM	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	-
	details	no overhead statement is possible
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	low - high (2,5)
	Details	support: VMWare, QEMU, Xen and UML
	Operation (key)	low (1)
	Details	administration
	Development (key)	unkown (-)
	Details	-
Key-Mgmt	Overall Estimation	3,5
	exist?	no
sharing possible	Details	-
		-
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	-
	Open Source?	no

Tabelle A.93.: Lares [PCSL08]

keywords, taxonomy	Secure Virtualization	
basics	xen, secure active monitoring, memory protection	
data in- //output	VM // secure active monitoring this vm	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	quotation small overhead
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	low (1)
	Details	Xen
	Operation (key)	low (1)
	Details	administration
	Development (key)	high* (4)
	Details	-
	Overall Estimation	6
Key-Mgmt	exist?	no
	Details	-
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	Xen 3.0.4., Fedora 7 security VM, Intel VT-x
	Open Source?	no
*since source code is not available		

Tabelle A.94.: SecVisor [SLQP07]

keywords, taxonomy	Secure Virtualization	
basics	kernel code protection, integrity protection, hardware memory protections, shadow paging	
data in- //output	commodity OS // integrity protected OS	
is efficient?	true	
adjustments in Data	// BL // UI	no // no // no
overhead	estimation details	medium depending on workload: higher overhead from compute-bound, lower from I/O-bound applications
Cloud Model (Private	// IaaS // PaaS // SaaS)	yes // no // no // no
cost	Technology (key)	low - high (2,5)
	Details	accessible OS kernel
	Operation (key)	low (1)
	Details	administration
	Development (key)	low* (1)
	Details	-
	Overall Estimation	4,5
Key-Mgmt	exist?	no
	Details	-
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	AMD SVM
	Open Source?	no
*It is easy to port OS kernels to SecVisor. We port the Linux kernel version 2.6.20 by adding 12 lines and deleting 81 lines, out of a total of approximately 4.3 million lines of code in the kernel.		

Tabelle A.95.: BitVisor [SET⁺09]

keywords, taxonomy	Secure Virtualization	
basics	I/O device security, hypervisor, I/O Monitoring, Shadow Paging, AES	
data in- //output	hypervisor // hypervisor enforcing I/O device security	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	medium
	details	encryption overhead 36%
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // no // no // no	
cost	Technology (key)	medium (2)
	Details	Intel VT processors
	Operation (key)	low (1)
	Details	administration
	Development (key)	high* (4)
	Details	-
	Overall Estimation	7
Key-Mgmt	exist?	yes
	Details	not described, probably on hypervisor
sharing possible	-	
native scaling included	no	
native backup included	no	
natively high available	no	
Data Security	C // I // A	yes // yes // no
Data Privacy	UL // TR // I	no // no // no
Prototype	exist?	yes
	Technology	XTS-AES, supports Intel VT processors, Dr. Brian Gladmans AES engine written in 64 bit assembler
	Open Source?	no
*since source code is not available		

Tabelle A.96.: Xen-Blanket [WJW12]

keywords, taxonomy	Secure Virtualization, Nested Virtualization	
basics	Xen, user-centric cloud deployment, Hypervisor	
data in- //output	IaaS // homogenized cloud deployment	
is efficient?	true	
adjustments in Data // BL // UI	no // no // no	
overhead	estimation	low
	details	3% up to 30% for file creation
Cloud Model (Private // IaaS // PaaS // SaaS)	yes // yes // no // no	
cost	Technology (key)	low (1)
	Details	IaaS*
	Operation (key)	low (1)
	Details	administration
	Development (key)	low (1)
	Details	
	Overall Estimation	3
Key-Mgmt	exist?	no
	Details	-
sharing possible	no	
native scaling included	yes	
native backup included	no	
natively high available	no	
Data Security	C // I // A	no // no // yes
Data Privacy	UL // TR // I	no // yes // yes
Prototype	exist?	yes
	Technology	Xen or KVM
	Open Source?	yes, https://code.google.com/p/xen-blanket/
*No modifications are expected or required to the underlying hypervisor.		

Literaturverzeichnis

- [AAW11] Divyakant Agrawal, Amr El Abbadi, and Shiyuan Wang. Secure Data Management in the Cloud. *DNIS*, pages 1–15, 2011.
- [ABB⁺07] Nicolas Anciaux, Mehdi Benzine, Luc Bouganim, Philippe Pucheral, and Dennis Shasha. Ghostdb: querying visible and hidden data without leaks. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 677–688. ACM, 2007.
- [Abb12] Imad M Abbadi. Clouds trust anchors. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 127–136. IEEE, 2012.
- [ABC05] Ross Anderson, Mike Bond, and Jolyon Clulow. Cryptographic processors – a survey. (641), 2005.
- [ABE⁺13] Arvind Arasu, Spyros Blanas, Ken Eguro, Manas Joglekar, Raghav Kaushik, Donald Kossmann, Ravi Ramamurthy, Prasang Upadhyaya, and Ramarathnam Venkatesan. Secure database-as-a-service with cipherbase. In *Proceedings of the 2013 international conference on Management of data*, pages 1033–1036. ACM, 2013.
- [ABGN12] Aiiad Albeshri, Colin Boyd, and Juan Gonzalez Nieto. A security architecture for cloud storage combining proofs of retrievability and fairness. In *CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 30–35, 2012.
- [ABGP05] Maurizio Atzori, Francesco Bonchi, Fosca Giannotti, and Dino Pedreschi. Anonymity Preserving Pattern Discovery. *Knowledge Creation Diffusion Utilization*, 2005.
- [ABR⁺10] Pelin Angin, Bharat Bhargava, Rohit Ranchal, Noopur Singh, Mark Linderman, Lotfi Ben Othmane, and Leszek Lilien. An entity-centric approach for privacy and identity management in cloud computing. In *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, pages 177–183. IEEE, 2010.
- [ABR12] Myrto Arapinis, Sergiu Bursuc, and Mark Ryan. Privacy Supporting Cloud Computing : ConfiChair , a Case Study. *LNCS*, (7215):89–108, 2012.
- [AC11] David S Allison and Miriam A M Capretz. Furthering the Growth of Cloud Computing by Providing Privacy as a Service. *LNCS*, 6868:64–78, 2011.

- [ACEW12] David S Allison, Miriam AM Capretz, Hany F ELYamany, and Shuying Wang. Privacy protection framework with defined policies for service-oriented architecture. *Journal of Software Engineering and Applications*, 5(3):200–215, 2012.
- [AI09] Nuttapong Attrapadung and Hideki Imai. Conjunctive broadcast and attribute-based encryption. In *Pairing-Based Cryptography–Pairing 2009*, pages 248–265. Springer, 2009.
- [Aim10] Marco Domenico Aime. Trusted distributed log services. In *Wireless Conference (EW), 2010 European*, pages 488–495. IEEE, 2010.
- [AKSX04] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 563–574. ACM, 2004.
- [AWS13] AWS. Aws rds orcale encryption, 2013.
- [AY07] Charu C. (IBM) Aggarwal and Philip S. Yu. Privacy-preserving data mining: models and algorithms. 2007.
- [BA12] Philogene A Boampong and Park Avenue. Different Facets of Security in the Cloud. 2012.
- [Bas12] Salman A Baset. Cloud SLAs : Present and Future. *ACM SIGOPS Operating Systems Review* 46.2, pages 57–66, 2012.
- [BBI⁺13] Sören Bleikertz, Sven Bugiel, Hugo Ideler, Stefan Nürnberger, and Ahmad-reza Sadeghi. Client-controlled Cryptography-as-a-Service in the Cloud. (Vm), 2013.
- [BCG⁺09] Stefan Berger, Ramón Cáceres, Kenneth Goldman, Dimitrios Pendarakis, Ronald Perez, Josyula R Rao, Eran Rom, Reiner Sailer, Wayne Schildhauer, Deepa Srinivasan, et al. Security for the cloud infrastructure: Trusted virtual data center implementation. *IBM Journal of Research and Development*, 53(4):6–1, 2009.
- [BCLO09] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O’neill. Order-preserving symmetric encryption. In *Advances in Cryptology-EUROCRYPT 2009*, pages 224–241. Springer, 2009.
- [BCO11] Alexandra Boldyreva, Nathan Chenette, and Adam O’Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *Advances in Cryptology-CRYPTO 2011*, pages 578–595. Springer, 2011.
- [BCQ⁺13] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando André, and Paulo Sousa. Depsky: dependable and secure storage in a cloud-of-clouds. *ACM Transactions on Storage (TOS)*, 9(4):12, 2013.
- [BDA⁺10] Ivona Brandic, Schahram Dustdar, Tobias Anstett, David Schumm, Frank Leymann, and Ralf Konrad. Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds. *2010 IEEE 3rd International Conference on Cloud Computing*, (i):244–251, July 2010.
- [BDF⁺03] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37(5):164–177, 2003.

- [BDJ⁺11] Kevin D Bowers, Marten Van Dijk, Ari Juels, Alina Oprea, and Ronald L Rivest. How to Tell if Your Cloud Files Are Vulnerable to Drive Crashes. *Proceedings of the 18th ACM conference on Computer and communications security.*, 2011.
- [BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology—CRYPTO 2001*, pages 213–229. Springer, 2001.
- [BGJ⁺13] Jens-Matthias Bohli, Nils Gruschka, Meiko Jensen, Luigi Lo Iacono, and Ninja Marnau. Security and Privacy Enhancing Multi-Cloud Architectures. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2013.
- [BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology—CRYPTO 2005*, pages 258–275. Springer, 2005.
- [BIT10] BITKOM. Cloud Computing – Was Entscheider wissen müssen. page 116, 2010.
- [BJO09] Kevin D Bowers, Ari Juels, and Alina Oprea. HAIL : A High-Availability and Integrity Layer for Cloud Storage. pages 187–198, 2009.
- [BK13] BITOM and KMPG. Cloud-monitor 2013 cloud-computing in deutschland – status quo und perspektiven. KMPG Study, February 2013.
- [BKNS12] Sören Bleikertz, Anil Kurmus, Zoltán a. Nagy, and Matthias Schunter. Secure cloud maintenance. *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security - ASIACCS '12*, page 83, 2012.
- [BLCSG12] Shakeel Butt, H Andrés Lagar-Cavilla, Abhinav Srivastava, and Vinod Ganapathy. Self-service cloud computing. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 253–264. ACM, 2012.
- [Blo70] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.
- [BLS⁺09] David Bernstein, Erik Ludvigson, Krishna Sankar, Steve Diamond, and Monique Morrow. Blueprint for the intercloud-protocols and formats for cloud computing interoperability. In *Internet and Web Applications and Services, 2009. ICIW'09. Fourth International Conference on*, pages 328–336. IEEE, 2009.
- [BM12] Kirsten Bock and Sebastian Meissner. Datenschutz-Schutzziele im Recht. *Datenschutz und Datensicherheit - DuD*, 36(6):425–431, May 2012.
- [BMY02] Jean Bacon, Ken Moody, and Walt Yao. A model of oasis role-based access control and its support for active security. *ACM Transactions on Information and System Security (TISSEC)*, 5(4):492–540, 2002.
- [BOL09] L Ben Othmane and Leszek Lilien. Protecting privacy of sensitive data dissemination using active bundles. In *Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS'09. World Congress on*, pages 202–213. IEEE, 2009.
- [BPFS09] Elisa Bertino, Federica Paci, Rodolfo Ferrini, and Ning Shang. Privacy-preserving digital identity management for cloud computing. *IEEE Data Eng. Bull.*, 32(1):21–27, 2009.

- [BPR⁺08] Dan Boneh, Periklis Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 283–292. IEEE, 2008.
- [BPS12] Michael Brenner, Henning Perl, and Matthew Smith. How practical is homomorphically encrypted program execution? an implementation and performance evaluation. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 375–382. IEEE, 2012.
- [BS11] Sumeet Bajaj and Radu Sion. Trusteddb: a trusted hardware based database with privacy and data confidentiality. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 205–216. ACM, 2011.
- [BSG06] Stefan Berger, Reiner Sailer, and Kenneth A Goldman. vTPM : Virtualizing the Trusted Platform Module. In *15th conference on USENIX Security Symposium.*, pages 305–320, 2006.
- [BSI12] BSI. Sicherheitsempfehlungen für Cloud Computing Anbieter. In *Bundesamt für Sicherheit in der Informationstechnik*. 2012.
- [BSSS11] Sven Bugiel, N Stefan, Ahmad-reza Sadeghi, and Thomas Schneider. Twin Clouds : Secure Cloud Computing with Low Latency. *CMS*, pages 32–44, 2011.
- [BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 321–334. IEEE, 2007.
- [BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography*, pages 253–273. Springer, 2011.
- [BT10] René Bröcker and Johannes Tiemeyer. Relationale Cloud-Datenbanken, ein aktueller Vergleich. *Informatik-Spektrum*, 34(1):90–98, October 2010.
- [Bue10] Rene Buest, November 2010.
- [Bue13] Rene Buest, Mai 2013.
- [BWvVS11] Michael Brenner, Jan Wiebelitz, Gabriele von Voigt, and Matthew Smith. Secret program execution in the cloud applying homomorphic encryption. In *Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on*, pages 114–119. IEEE, 2011.
- [BYDD⁺10] Muli Ben-Yehuda, Michael D Day, Zvi Dubitzky, Michael Factor, Nadav Har'El, Abel Gordon, Anthony Liguori, Orit Wasserman, and Ben-Ami Yassour. The turtles project: Design and implementation of nested virtualization. In *OSDI*, volume 10, pages 423–436, 2010.
- [CDE⁺09] Luigi Catuogno, Alexandra Dmitrienko, Konrad Eriksson, Gianluca Ramunno, Ahmad-reza Sadeghi, Steffen Schulz, Matthias Schunter, Marcel Winandy, Jing Zhan, and G Horst. Trusted Virtual Domains – Design , Implementation and Lessons Learned. 2009.

- [CdVFS07] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. 47:1–36, January 2007.
- [CGJ⁺09] Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka, and Jesus Molina. Controlling Data in the Cloud : Outsourcing Computation without Outsourcing Control. 2009.
- [CGL⁺08] Xiaoxin Chen, Tal Garfinkel, E Christopher Lewis, Pratap Subrahmanyam, Dan Boneh, Jeffrey Dworkin, Dan R K Ports, and Carl A Waldspurger. Overshadow : A Virtualization-Based Approach to Retrofitting Protection in Commodity Operating Systems. pages 2–13, 2008.
- [Cha07] Melissa Chase. Multi-authority attribute based encryption. In *Theory of Cryptography*, pages 515–534. Springer, 2007.
- [CHHY12] Sherman S M Chow, Yi-jun He, Lucas C K Hui, and Siu Ming Yiu. SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment. pages 526–543, 2012.
- [CJP⁺11] Carlo Curino, Evan P C Jones, Raluca Ada Popa, Eugene Wu, and Nickolai Zeldovich. Relational Cloud : A Database-as-a-Service for the Cloud Accessed Citable Link Relational Cloud : A Database-as-a-Service for the Cloud. pages 0–6, 2011.
- [CK10] Yanpei Chen and Randy H Katz. What ’ s New About Cloud Computing Security ? 2010.
- [CN01] Peter M Chen and Brian D Noble. When virtual is better than real [operating system relocation to virtual machines]. In *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*, pages 133–138. IEEE, 2001.
- [Com15] European Commission. European commission directorate general communications networks, content and technology unit e2 - software and services, cloud - cloud computing service level agreements exploitation of research results, 2015.
- [Cri14] CrispResearch. Study platform-as-a-service: German sme market survey, 2014.
- [CSA11] CSA. Security guidance for critical areas of focus in cloud computing v3.0. CSA, pages 0–176, 2011.
- [CSA14] CSA. Csa security, trust and assurance registry (star), 2014.
- [CWL⁺11] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. In *INFOCOM, 2011 Proceedings IEEE*, pages 829–837. IEEE, 2011.
- [Del06] Alexander Delp. Privatheit durch Manipulation der Datenqualität. 2006.
- [DH03] Naganand Doraswamy and Dan Harkins. *IPSec: the new security standard for the Internet, intranets, and virtual private networks*. Prentice Hall Professional, 2003.
- [DJ10] Marten Van Dijk and Ari Juels. On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing. 2010.
- [DJO⁺12] Marten Van Dijk, Ari Juels, Alina Oprea, Ronald L Rivest, U C Berkeley, and Nikos Triandopoulos. Hourglass Schemes : How to Prove that Cloud Files Are Encrypted. 2012.

- [DLB11] George Danezis, Benjamin Livshits, and Samuel Beckett. Towards Ensuring Client-Side Computational Integrity (A position paper). pages 1–6, 2011.
- [DPS01] Joan G Dyer, Ronald Perez, and Sean W Smith. Building the IBM 4758 Secure. (October):57–66, 2001.
- [Eck13] Claudia Eckert. *IT-Sicherheit: Konzepte-Verfahren-Protokolle*. Oldenbourg Wissenschaftsverlag, 2013.
- [EEJS⁺06] Khaled El Emam, Sam Jabbouri, Scott Sams, Youenn Drouet, and Michael Power. Evaluating common de-identification heuristics for personal health information. *Journal of Medical Internet Research*, 8(4), 2006.
- [EK11] David Evans and Jonathan Katz. Faster Secure Two-Party Computation Using Garbled Circuits. (August):8–12, 2011.
- [ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology*, pages 10–18. Springer, 1985.
- [EMSCB06] European-Multilaterally-Secure-Computing-Base. Towards trustworthy systems with open standards and trusted computing, 2006.
- [ENI12] ENISA. ENISA Threat Landscape. 2012.
- [Ert03] Wolfgang Ertel. *Angewandte Kryptographie*. Fachbuchverlag Leipzig im Carl Hanser Verlag München Wien, 2. bearbeitete auflage edition, 2003.
- [FDEM13] Angela Francis, Renu Mary Daniel, Vinodh Edwards S E, and A Threat Model. TPM : A More Trustworthy Solution to Computer Security. *IJCSET*, 3(3):99–103, 2013.
- [Fed15] FedRAMP. Fedramp zertifizierung, Juli 2015.
- [FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.
- [FK09] David F. Ferraiolo and D. Richard Kuhn. Role-based access controls. *CoRR*, abs/0903.2171, 2009.
- [FLR⁺14] Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck, and Peter Arbitter. *Cloud Computing Patterns*. Springer-Verlag Wien, 2014.
- [FN94] Amos Fiat and Moni Naor. Broadcast encryption. In *Advances in Cryptology—CRYPTO’93*, pages 480–491. Springer, 1994.
- [FSG⁺14] Diogo AB Fernandes, Liliana FB Soares, João V Gomes, Mário M Freire, and Pedro RM Inácio. Security issues in cloud environments: a survey. *International Journal of Information Security*, 13(2):113–170, 2014.
- [FZFF10] Ariel J Feldman, William P Zeller, Michael J Freedman, and Edward W Felten. SPORC : Group Collaboration using Untrusted Cloud Resources. *Proceedings of the 9th USENIX conference on Operating systems design and implementation.*, 2010.
- [Gen09] Craig Gentry. *A FULLY HOMOMORPHIC ENCRYPTION SCHEME*. PhD thesis, Stanford University, 2009.

- [Gen10] Craig Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97, March 2010.
- [GH11] Craig Gentry and Shai Halevi. Implementing Gentry ' s Fully-Homomorphic Encryption Scheme. *EUROCRYPT*, pages 1–29, 2011.
- [GHS12] Craig Gentry, Shai Halevi, and Nigel P Smart. Fully Homomorphic Encryption with Polylog Overhead. *EUROCRYPT*, pages 465–482, 2012.
- [GMR⁺12] Nelson Gonzalez, Charles Miers, Fernando Redígolo, Marcos Simplicio, Tereza Carvalho, Mats Näslund, and Makan Pourzandi. A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing*, 1(1):1–18, 2012.
- [GMSW06] Dominik Grolimund, Luzius Meisser, Stefan Schmid, and Roger Wattenhofer. Cryptree : A Folder Tree Structure for Cryptographic File Systems. (Section 7), 2006.
- [GO96] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious rams. *Journal of the ACM (JACM)*, 43(3):431–473, 1996.
- [Gol87] Oded Goldreich. Towards a theory of software protection and simulation by oblivious rams. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 182–194. ACM, 1987.
- [GP11] G R Gangadharan and Davide Maria Parrilli. Service Level Agreements in Cloud Computing : Perspectives of Private Consumers and Small-to-Medium Enterprises. *Computer Communications and Networks*, pages 207–225, 2011.
- [GPC⁺03] Tal Garfinkel, Ben Pfaff, Jim Chow, Mendel Rosenblum, and Dan Boneh. Terra: A virtual machine-based platform for trusted computing. In *ACM SIGOPS Operating Systems Review*, volume 37, pages 193–206. ACM, 2003.
- [GPS13] Craig Gentry, Chris Peikert, and Nigel P Smart. Field Switching in BGV-Style Homomorphic Encryption. pages 1–17, 2013.
- [GR95] Barbara Guttman and Edward Roback. *An introduction to computer security: the NIST handbook*. DIANE Publishing, 1995.
- [GSMB03] Eu-Jin Goh, Hovav Shacham, Nagendra Modadugu, and Dan Boneh. Sirius: Securing remote untrusted storage. In *NDSS*, volume 3, pages 131–145, 2003.
- [GSY14] M. Gulati, M.J. Smith, and S.Y. Yu. Security enclave processor for a system on a chip, September 9 2014. US Patent 8,832,465.
- [GT08] Vandana Gunupudi and Stephen R Tate. Generalized non-interactive oblivious transfer using count-limited objects with applications to secure mobile agents. In *Financial Cryptography and Data Security*, pages 98–112. Springer, 2008.
- [GW11] Rüdiger Giebichenstein and Andreas Weiss. Zertifizierte Cloud durch das EuroCloud Star Audit SaaS. *Datenschutz und Datensicherheit - DuD*, 35(5):338–342, May 2011.
- [GWP⁺11] Emin Gün, Sirer Willem, De Bruijn Patrick, Alan Shieh, Kevin Walsh, Dan Williams, and Fred B Schneider. Logical Attestation : An Authorization Architecture for Trustworthy Computing. *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles.*, 2011.

- [GZLL13] Yubin Guo, Liankuan Zhang, Fengren Lin, and Ximing Li. A solution for privacy-preserving data manipulation and query on nosql database. *Journal of Computers*, 8(6):1427–1432, 2013.
- [Han12] Marit Hansen. Vertraulichkeit und Integrität von Daten und IT-Systemen im Cloud-Zeitalter. *Datenschutz und Datensicherheit - DuD*, 36(6):407–412, May 2012.
- [HILM02] Hakan Hacigümüş, Bala Iyer, Chen Li, and Sharad Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 216–227. ACM, 2002.
- [HLMS10] Fang Hao, T Lakshman, Sarit Mukherjee, and Haoyu Song. Secure cloud computing with a virtualized network infrastructure. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 16–16. USENIX Association, 2010.
- [HMT04] Bijit Hore, Sharad Mehrotra, and Gene Tsudik. A privacy-preserving index for range queries. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 720–731. VLDB Endowment, 2004.
- [HS02] Dani Halevy and Adi Shamir. The lsd broadcast encryption scheme. In *Advances in Cryptology—CRYPTO 2002*, pages 47–60. Springer, 2002.
- [HSE⁺11] Yan Huang, Chih-hao Shen, David Evans, Jonathan Katz, and Abhi Shelat. Efficient Secure Computation with Garbled Circuits. pages 28–48, 2011.
- [Hus10] Mohammed Hussain. *The Design and Applications of a Privacy-Preserving Identity and Trust-Management System*. PhD thesis, Queen’s University (Kingston, Ont.), 2010.
- [iC15] Trust in Cloud. Trust in cloud website, 2015.
- [IG08] Ioannis Ioannidis and Ananth Grama. An Efficient Protocol for Yao ’ s Millionaires ’ Problem. *Management*, 2008.
- [IKC09] Wassim Itani, Ayman Kayssi, and Ali Chehab. Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In *Dependable, Autonomic and Secure Computing, 2009. DASC’09. Eighth IEEE International Conference on*, pages 711–716. IEEE, 2009.
- [IRZ14] IBM-Research-Zürich. Idemix reseach projekt, 2014.
- [IS10] Alexander Iliev and Sean W Smith. Small, stupid, and scalable: secure computing with faerieplay. In *Proceedings of the fifth ACM workshop on Scalable trusted computing*, pages 41–52. ACM, 2010.
- [IS15] Internet-Sicherheit.de. Turaya das ziel, 2015.
- [JBU12] Martin Gilje Jaatun, Karin Bernsmed, and Astrid Undheim. Security SLAs – An Idea Whose Time Has Come ? pages 123–130, 2012.
- [JKSS10] Kimmo Järvinen, Vladimir Kolesnikov, Ahmad-Reza Sadeghi, and Thomas Schneider. Garbled circuits for leakage-resilience: Hardware implementation and evaluation of one-time programs. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 383–397. Springer, 2010.

- [JMR⁺14] Hubert A Jager, Arnold Monitzer, Ralf Rieken, Edmund Ernst, and Khiem Dau Nguyen. Sealed cloud—a novel approach to safe guard against insider attacks. In *Trusted Cloud Computing*, pages 15–34. Springer, 2014.
- [JSGI09] Meiko Jensen, Jörg Schwenk, Nils Gruschka, and Luigi Lo Iacono. On technical security issues in cloud computing. In *Cloud Computing, 2009. CLOUD’09. IEEE International Conference on*, pages 109–116. IEEE, 2009.
- [JWX07] Xuxian Jiang, Xinyuan Wang, and Dongyan Xu. Stealthy malware detection through vmm-based out-of-the-box semantic view reconstruction. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 128–138. ACM, 2007.
- [KBR11] Seny Kamara, U C Berkeley, and Tom Roeder. CS2 : A Searchable Cryptographic Cloud Storage System. pages 1–25, 2011.
- [KD10] Ronald L. Krutz and Russel Vines Dean. Cloud Security - A comprehensive guide to secure cloud computing. 2010.
- [Ker12] Florian Kerschbaum. Privacy-preserving computation (position paper), 2012.
- [KH12] Safwan Mahmud Khan and Kevin W Hamlen. Anonymouscloud: A data ownership privacy provider framework in cloud computing. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 170–176. IEEE, 2012.
- [KJGB06] Louis Kruger, Somesh Jha, Eu-Jin Goh, and Dan Boneh. Secure function evaluation with ordered binary decision diagrams. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 410–420. ACM, 2006.
- [KK12] Vladimir Kolesnikov and Ranjit Kumaresan. Improved Secure Two-Party Computation via Information-Theoretic Garbled Circuits. pages 205–221, 2012.
- [KKL⁺07] Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, and Anthony Liguori. kvm: the linux virtual machine monitor. In *Proceedings of the Linux Symposium*, volume 1, pages 225–230, 2007.
- [KL10] Seny Kamara and Kristin Lauter. Cryptographic cloud storage. MICROSOFT RESEARCH CRYPTOGRAPHY GROUP, 2010.
- [KR13] Seny Kamara and Mariana Raykova. Parallel homomorphic encryption. In *Financial Cryptography and Data Security*, pages 213–225. Springer, 2013.
- [Kre10] Düsseldorf Kreis. Safe harbor, April 2010.
- [KRS⁺03] Mahesh Kallahalla, Erik Riedel, Ram Swaminathan, Qian Wang, and Kevin Fu. Plutus: Scalable secure file sharing on untrusted storage. In *Fast*, volume 3, pages 29–42, 2003.
- [KSL⁺08] Vladimir Kolesnikov, Thomas Schneider, Bell Laboratories, Mountain Ave, and Murray Hill. Improved Garbled Circuit : Free XOR Gates and Applications. pages 486–498, 2008.
- [KSS10] Vladimir Kolesnikov, Ahmad-reza Sadeghi, and Thomas Schneider. From Dust to Dawn : Practically Efficient Two-Party Secure Function Evaluation Protocols and their Modular Design Table of Contents. 2010.

- [KW13] Seny Kamara and Lei Wei. Garbled circuits via structured encryption. In *Financial Cryptography and Data Security*, pages 177–188. Springer, 2013.
- [LDP11] Flavio Lombardi and Roberto Di Pietro. Secure virtualization for cloud computing. *Journal of Network and Computer Applications*, 34(4):1113–1122, 2011.
- [LDR05] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 2005.
- [LeF07] Kristen Riedt LeFevre. *Anonymity in data publishing and distribution*. PhD thesis, UNIVERSITY OF WISCONSIN–MADISON, 2007.
- [LFB12] Thomas Ludescher, Thomas Feilhauer, and Peter Brezany. Security concept and implementation for a cloud based e-science infrastructure. In *Availability, Reliability and Security (ARES), 2012 Seventh International Conference on*, pages 280–285. IEEE, 2012.
- [Lie95] Jochen Liedtke. *On micro-kernel construction*, volume 29. ACM, 1995.
- [LKMS04] Jinyuan Li, Maxwell N Krohn, David Mazières, and Dennis Shasha. Secure untrusted data repository (sundr). In *OSDI*, volume 4, pages 9–9, 2004.
- [LPM⁺13] Jacob R Lorch, Bryan Parno, James Mickens, Mariana Raykova, and Joshua Schiffman. Shroud : Ensuring Private Access to Large-Scale Data in the Data Center. pages 199–213, 2013.
- [LS10] Philip Laue and Oliver Stiernerling. Identitäts- und Zugriffsmanagement für Cloud Computing Anwendungen. *Datenschutz und Datensicherheit - DuD*, 34(10):692–697, October 2010.
- [LTM⁺11] Fang Lui, Jin Ting, Jian Mao, Robert Bohn, John Messina, Lee Badger, and Leaf Dawn. Nist cloud computing reference architecture. Technical report, National Institute of Standards and Technology, 2011.
- [LW11] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Advances in Cryptology–EUROCRYPT 2011*, pages 568–588. Springer, 2011.
- [MAF⁺11] Petros Maniatis, Devdatta Akhawe, Kevin Fall, Elaine Shi, Stephen Mccamant, Dawn Song, U C Berkeley, and Intel Labs Berkeley. Do You Know Where Your Data Are ? Secure Data Capsules for Deployable Data Protection. *Proc. 13th Usenix Conf. Hot Topics in Operating Systems.*, 2011.
- [Mal11] Lior Malka. Vmccrypt: modular software architecture for scalable secure computation. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 715–724. ACM, 2011.
- [MBJ⁺12] Per Håkon Meland, Karin Bernsmed, Martin Gilje Jaatun, Astrid Undheim, and Humberto Nicolás Castejón. Expressing cloud security requirements in deontic contract languages. In *CLOSER*, pages 638–646, 2012.
- [MBK09] Heribert Meffert, Christoph Burmann, and Manfred Kirchgeorg. *Marketing Arbeitsbuch*. Springer, 2009.
- [MG11] Peter Mell and Timothy Grance. The nist definition of cloud computing. Technical report, National Institute of Standards and Technology, 2011.

- [MKL09] Tim Mather, Subra Kumaraswamy, and Shahed Latif. *Cloud Security and Privacy*. 2009.
- [MLQ⁺10] Jonathan M. McCune, Yanlin Li, Ning Qu, Zongwei Zhou, Anupam Datta, Virgil Gligor, and Adrian Perrig. TrustVisor: Efficient TCB Reduction and Attestation. *2010 IEEE Symposium on Security and Privacy*, pages 143–158, 2010.
- [MMH08] Derek Gordon Murray, Grzegorz Milos, and Steven Hand. Improving Xen security through disaggregation. *Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments - VEE '08*, page 151, 2008.
- [MPB⁺12] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Avi Patel, and Muttukrishnan Rajarajan. A survey on security issues and solutions at different layers of Cloud computing. *The Journal of Supercomputing*, October 2012.
- [MPP⁺08] Jonathan M Mccune, Bryan Parno, Adrian Perrig, Michael K Reiter, and Hiroshi Isozaki. Flicker : An Execution Infrastructure for TCB Minimization Categories and Subject Descriptors. 2008.
- [MR14] Microsoft-Research. U-prove reseach projekt, 2014.
- [MS11] Ninja Marnau and Eva Schlehahn. Cloud Computing und Safe Harbor. *Datenschutz und Datensicherheit - DuD*, pages 311–316, 2011.
- [MSG13] Murali Mani, Kinnari Shah, and Manikanta Gunda. Enabling Secure Database as a Service using Fully Homomorphic Encryption : Challenges and Opportunities. pages 1–13, 2013.
- [MSS⁺11] Ninja Marnau, Norbert Schirmer, Eva Schlehahn, Matthias Schunter, Das Von Der Europäischen, and Projekt Tclouds. TClouds. *DuD Datenschutz und Datensicherheit*, pages 333–337, 2011.
- [Mül11] Sascha Müller. Data-centric security with attribute-based encryption. 2011.
- [NIS02] NIST. Federal information security management act of 2002. United States federal law, 2002.
- [NLV11] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? *Proceedings of the 3rd ACM workshop on Cloud computing security workshop - CCSW '11*, page 113, 2011.
- [NS78] Roger M Needham and Michael D Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.
- [NT94] B Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, 1994.
- [NT12] David Naccache and Mehdi Tibouchi. Public Key Compression and Modulus Switching for Fully Homomorphic Encryption. *EUROCRYPT*, pages 446–464, 2012.
- [Ora15a] Oracle. Berkleydb overview, Juli 2015.
- [Ora15b] Oracle. Transparent data encryption, 2015.
- [Ost90] Rafail Ostrovsky. Efficient computation on oblivious rams. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 514–523. ACM, 1990.

- [Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin Heidelberg, 1999.
- [Pau11] Sachar Paulus. Standards für Trusted Clouds Anforderungen an Standards und aktuelle Entwicklungen. *Datenschutz und Datensicherheit - DuD*, pages 317–321, 2011.
- [PCSL08] Bryan D Payne, Martim Carbone, Monirul Sharif, and Wenke Lee. Lares: An architecture for secure active monitoring using virtualization. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 233–247. IEEE, 2008.
- [PG74] Gerald J Popek and Robert P Goldberg. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, 17(7):412–421, 1974.
- [PGA⁺14] R.S. Polzin, F.L. Gautier, M.D. Adler, C. Sauerwald, and M.L.H. Brouwer. Key management using security enclave processor, March 27 2014. US Patent App. 13/626,476.
- [PLM⁺11] Raluca Ada Popa, Jacob R Lorch, David Molnar, Helen J Wang, and Li Zhuang. Enabling Security in Cloud Storage SLAs with CloudProof. *Proceedings of 2011 USENIX Annual Technical Conference*, 2011.
- [Pog07] Werner Poguntke. *Basiswissen IT-Sicherheit: Das Wichtigste für den Schutz von Systemen und Daten*. W3l GmbH, 2007.
- [PP12] Ioannis Papagiannis and Peter Pietzuch. CloudFilter : Practical Control of Sensitive Data Propagation to the Cloud Categories and Subject Descriptors. 2012.
- [PPL11] Hyun-A Park, Jae Hyun Park, and Dong Hoon Lee. PKIS: practical keyword index search on cloud datacenter. *EURASIP Journal on Wireless Communications and Networking*, 2011(64):16, 2011.
- [PPL12] John Pecarina, Shi Pu, and Jyh-Charn Liu. Sapphire: Anonymity for enhanced control and private collaboration in healthcare clouds. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 99–106. IEEE, 2012.
- [PR04] Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. *Journal of Algorithms*, 51(2):122–144, 2004.
- [PR10] Benny Pinkas and Tzachy Reinman. Oblivious ram revisited. In *Advances in Cryptology—CRYPTO 2010*, pages 502–519. Springer, 2010.
- [PRZB11] Raluca Ada Popa, Catherine M S Redfield, Nickolai Zeldovich, and Hari Balakrishnan. CryptDB : Protecting confidentiality with encrypted query processing Accessed Citable Link Detailed Terms CryptDB : Protecting Confidentiality with Encrypted Query Processing. 2011.
- [PSV⁺14] Raluca Ada Popa, Emily Stark, Steven Valdez, Jonas Helfer, Nickolai Zeldovich, and Hari Balakrishnan. Securing web applications by blindfolding the server. In *Proceedings of the USENIX Symposium of Networked Systems Design and Implementation, NSDI*, 2014.
- [Pul12] Tobias Pulls. Privacy-preserving transparency-enhancing tools, 2012.
- [PWVG12] Tobias Pulls, Karel Wouters, Jo Vliegen, and Christian Grahm. Distributed privacy-preserving log trails. Technical Report 2012:24, Karlstad University, Department of Computer Science, 2012.

- [PZB11] Raluca Ada Popa, Nickolai Zeldovich, and Hari Balakrishnan. Computer Science and Artificial Intelligence Laboratory Technical Report CryptDB : A Practical Encrypted Relational DBMS CryptDB : A Practical Encrypted Relational DBMS. 2011.
- [RBK⁺14a] Paul Reinhold, Wolfgang Benn, Benjamin Krause, Frank Goetz, and Dirk Labudde. Hybrid cloud architecture for software-as-a-service provider to achieve higher privacy and decrease security concerns about cloud computing. In *CLOUD COMPUTING 2014, The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 94–99, 2014.
- [RBK⁺14b] Paul Reinhold, Wolfgang Benn, Benjamin Krause, Frank Goetz, and Dirk Labudde. Introducing a scalable encryption layer to address privacy and security issues in hybrid cloud environments. *International Journal On Advances in Software*, 7(3 and 4):727–739, 2014.
- [RBO⁺10] Rohit Ranchal, Bharat Bhargava, Lotfi Ben Othmane, Leszek Lilien, Anya Kim, Myong Kang, and Mark Linderman. Protection of identity information in cloud computing without trusted third party. In *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, pages 368–372. IEEE, 2010.
- [RJ12] Longbo Ran and Hai Jin. SDSPF : A Secure Data Storage and Processing Framework for Cloud Computing Systems. *LNEE*, 180:127–133, 2012.
- [RS12] Satyendra Singh Rawat and Nireesh Sharma. A survey of various techniques to secure cloud storage. In *National Conference on Security Issues in Network Technologies (NCSI-2012)*, 2012.
- [RSA78] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [SB12] Johanna Schmidt-Bens. *Cloud Computing Technologien und Datenschutz*. Oldenburger Verlag für Wirtschaft, Informatik und Recht, 2012.
- [SBK⁺14] Andreas Schaad, Anis Bkakra, Florian Keschbaum, Frederic Cuppens, Nora Cuppens-Boulahia, and David Gross-Amblard. Optimized and controlled provisioning of encrypted outsourced data. In *Proceedings of the 19th ACM symposium on Access control models and technologies*, pages 141–152. ACM, 2014.
- [SCC⁺10] Alexander Shraer, Christian Cachin, Asaf Cidon, Idit Keidar, Yan Michalevsky, and Dani Shaket. Venus: Verification for untrusted cloud storage. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pages 19–30. ACM, 2010.
- [Sch08] Thomas Schneider. *Practical Secure Function Evaluation*. diplom, Friedrich-Alexander-Universität Erlangen Nürnberg, 2008.
- [Sch10a] Thomas Schneider. Reden ist Silber – Schweigen ist Gold : Datensparsamkeit durch effizientes Rechnen unter Verschlüsselung. pages 1–8, 2010.
- [Sch10b] Thomas Schneider. Tool for Automating Secure Two-party computations TASTY. 2010.
- [Sch11a] Thomas Schneider. *Engineering Secure Two-Party Computation Protocols Efficient Secure Function Evaluation*. PhD thesis, 2011.

- [Sch11b] Bruce Schneier. *Secrets and lies: digital security in a networked world*. John Wiley & Sons, 2011.
- [Seb10] Sebastian Sebal. k-Anonymity und dessen Einfluss auf die Forschung. 2010.
- [SET⁺09] Takahiro Shinagawa, Hideki Eiraku, Kouichi Tanimoto, Kazumasa Omote, Shoichi Hasegawa, Takashi Horie, Manabu Hirano, Kenichi Kourai, Yoshihiro Oyama, Eiji Kawai, et al. Bitvisor: a thin hypervisor for enforcing i/o device security. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pages 121–130. ACM, 2009.
- [Sha85] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in cryptology*, pages 47–53. Springer, 1985.
- [SK99] Bruce Schneier and John Kelsey. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security (TISSEC)*, 2(2):159–176, 1999.
- [SK09] Rashid Sheikh and Beerendra Kumar. Privacy-Preserving k-Secure Sum Protocol. *Journal of Computer Science*, 6(2), 2009.
- [SK10] Udo Steinberg and Bernhard Kauer. Nova: a microhypervisor-based secure virtualization architecture. In *Proceedings of the 5th European conference on Computer systems*, pages 209–222. ACM, 2010.
- [SLHL14] Wenhai Sun, Wenjing Lou, Y Thomas Hou, and Hui Li. Privacy-preserving keyword search over encrypted data in cloud computing. In *Secure Cloud Computing*, pages 189–212. Springer, 2014.
- [SLQP07] Arvind Seshadri, Mark Luk, Ning Qu, and Adrian Perrig. Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses. *ACM SIGOPS Operating Systems Review*, 41(6):335–350, 2007.
- [SLS13] Stephan Schneider, Jens Lansing, and Ali Sunyaev. Empfehlungen zur gestaltung von cloud-service-zertifizierungen. *Industrie Management-Zeitschrift für industrielle Geschäftsprozesse*, pages 13–17, 2013.
- [SMT⁺12] Juraj Somorovsky, Christopher Meyer, Thang Tran, Mohamad Sbeiti, and Christian Wietfeld. SEC 2 : SECURE MOBILE SOLUTION FOR DISTRIBUTED PUBLIC CLOUD STORAGEES. 2012.
- [SMV⁺10] Joshua Schiffman, Thomas Moyer, Hayawardh Vijayakumar, Trent Jaeger, and Patrick McDaniel. Seeding clouds with trust anchors. *Proceedings of the 2010 ACM workshop on Cloud computing security workshop - CCSW '10*, page 43, 2010.
- [SR15a] Paolo Smiraglia and Gianluca Ramunno. Chapter 9 log service, 2015.
- [SR15b] Paolo Smiraglia and Gianluca Ramunno. Secure logging, 2015.
- [SRGS12] Nuno Santos, Rodrigo Rodrigues, Krishna P Gummadi, and Stefan Saroiu. Policy-Sealed Data : A New Abstraction for Building Trusted Cloud Services. In *21nd USENIX Security Symposium. USENIX*, 2012.
- [SRW⁺08] Vincent Scarlata, Carlos Rozas, Monty Wiseman, David Grawrock, and Claire Vishik. Tpm virtualization: Building a general framework. In *Trusted Computing*, pages 43–56. Springer, 2008.

- [SS98] Pierangela Samarati and Latanya Sweeney. Protecting Privacy when Disclosing Information : k -Anonymity and Its Enforcement through Generalization and Suppression 1 Introduction. pages 1–19, 1998.
- [SS10a] Ahmad-reza Sadeghi and Thomas Schneider. Verschlüsselt Rechnen : Sichere Verarbeitung Verschlüsselter Medizinischer Daten am Beispiel der Klassifikation von EKG Daten. 2010.
- [SS10b] Damien Stehlé and Ron Steinfeld. Faster Fully Homomorphic Encryption. *CRYPTO*, 2010.
- [SS13] Emil Stefanov and Elaine Shi. Oblivstore: High performance oblivious cloud storage. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 253–267. IEEE, 2013.
- [SSS12] Emil Stefanov, Elaine Shi, and Dawn Song. Towards Practical Oblivious RAM . pages 1–40, 2012.
- [SSW08] Ahmad-reza Sadeghi, Christian Stübke, and Marcel Winandy. Property-Based TPM Virtualization. In *11th International Conference on Information Security (ISC'08)*, (Vmm), 2008.
- [SSWH10] Ahmad-reza Sadeghi, Thomas Schneider, Marcel Winandy, and G Horst. Token-Based Cloud Computing Secure Outsourcing of Data and Arbitrary Computations with Lower Latency. 2:417–429, 2010.
- [SV10] N P Smart and F Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. *PKC'10 Proceedings of the 13th international conference on Practice and Theory in Public Key Cryptography*, pages 1–19, 2010.
- [SVP⁺12] Srinath Setty, Victor Vu, Nikhil Panpalia, Benjamin Braun, Andrew J Blumberg, and Michael Walfish. Taking proof-based verified computation a few steps closer to practicality. 2012.
- [SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.
- [Swe02] Latanya Sweeney. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. 10(5):1–14, 2002.
- [SWP00] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 44–55. IEEE, 2000.
- [TCG11] Trusted-Computing-Group. Tpm main specification level 2 version 1.2, revision 116, 2011.
- [TCG14a] Trusted-Computing-Group. Trusted computing group webseite, 2014.
- [TCG14b] Trusted-Computing-Group. Trusted platform module library specification, family 2.0, level 00, revision 01.07, 2014.
- [TDF⁺13] Shigeo Tsujii, Hiroshi Doi, Ryo Fujita, Masahito Gotaishi, Yukiyasu Tsunoo, and Takahiko Syouji. Privacy preserving data processing with collaboration of homomorphic cryptosystems. In *Financial Cryptography and Data Security*, pages 201–212. Springer, 2013.

- [TJA10] Hassan Takabi, James BD Joshi, and Gail-Joon Ahn. Security and privacy challenges in cloud computing environments. *IEEE Security & Privacy*, 8(6):24–31, 2010.
- [TKMZ13] Stephen Tu, M Frans Kaashoek, Samuel Madden, and Nickolai Zeldovich. Processing analytical queries over encrypted data. In *Proceedings of the 39th international conference on Very Large Data Bases*, pages 289–300. VLDB Endowment, 2013.
- [TMLL06] Richard Ta-Min, Lionel Litty, and David Lie. Splitting interfaces: Making trust between applications and operating systems configurable. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 279–292. USENIX Association, 2006.
- [TPM15] TPM. Price of tpm, 2015.
- [TPPG11] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. Cryptodsp for cloud privacy. In *Web Information Systems Engineering–WISE 2010 Workshops*, pages 428–439. Springer, 2011.
- [TR14] TÜV-Rheinland. Cloud sicherheitszertifizierung, 2014.
- [Una08] Unabhängigen Landeszentrum für Datenschutz Schleswig-Holstein. EuroPriSe ® – Europäisches Datenschutz-Gütesiegel - Fact Sheet. 2008.
- [Ven01] Håkan Ventura. Diameter : Next generations aaa protocol, 2001.
- [Wei10] Thilo Weichert. Cloud Computing und Datenschutz. *Datenschutz und Datensicherheit - DuD*, 5:9, 2010.
- [Wik15] Key Management Wikiage. Key management, 2015.
- [Wil12] Bill Wilder. *Cloud Architecture Patterns: Using Microsoft Azure*. O Reilly Media, Inc., 2012.
- [Win10] Michael Winkelmann. Cloud computing: Sicherheit und datenschutz. *Arbeitspapier, Universität Potsdam, Potsdam*, 2010.
- [Win11] Vic J.R. Winkler. *Securing the Cloud*. Elsevier, 2011.
- [WJ10] Zhi Wang and Xuxian Jiang. HyperSafe: A Lightweight Approach to Provide Lifetime Hypervisor Control-Flow Integrity. *2010 IEEE Symposium on Security and Privacy*, pages 380–395, 2010.
- [WJCN09] Zhi Wang, Xuxian Jiang, Weidong Cui, and Peng Ning. Countering kernel rootkits with lightweight hook protection. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 545–554. ACM, 2009.
- [WJW12] Dan Williams, Hani Jamjoom, and Hakim Weatherspoon. The xen-blanket: virtualize once, run everywhere. pages 113–126, 2012.
- [WL12] Gaven J Watson and Michael E Locasto. LoSt : Location Based Storage. *CCSW’12*, pages 59–69, 2012.
- [WLL12] Boyang Wang, Baochun Li, and Hui Li. Knox : Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud. pages 507–525, 2012.
- [WS12] Peter Williams and Radu Sion. Single round access privacy on outsourced storage. *Proceedings of the 2012 ACM conference on Computer and communications security - CCS ’12*, page 293, 2012.

- [WWRL10] Cong Wang, Qian Wang, Kui Ren, and Wenjing Lou. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. *2010 Proceedings IEEE INFOCOM*, pages 1–9, March 2010.
- [XS07] Shouhuai Xu and Ravi Sandhu. A Scalable and Secure Cryptographic Service. In *Data and Applications Security, LNCS*, 4602(XXI):1–16, 2007.
- [XYM⁺13] Jinbo Xiong, Zhiqiang Yao, Jianfeng Ma, Ximeng Liu, Qi Li, and Tao Zhang. Pram: privacy preserving access management scheme in cloud services. In *Proceedings of the 2013 international workshop on Security in cloud computing*, pages 41–46. ACM, 2013.
- [XZY⁺12] Huijun Xiong, Xinwen Zhang, Danfeng Yao, Xiaoxin Wu, and Yonggang Wen. Towards end-to-end secure content storage and delivery with public cloud. In *Proceedings of the second ACM conference on Data and Application Security and Privacy*, pages 257–266. ACM, 2012.
- [Yao82] Andrew C Yao. Theory and Applications of Trapdoor Functions, 1982.
- [Yao86] Andrew Chi-chih Yao. How to Generate an Exchange Secrets. *Exchange Organizational Behavior Teaching Journal*, (1):162–167, 1986.
- [YCL14] Raghuram Yeluri and Enrique Castro-Leon. *Building the Infrastructure for Cloud Security: A Solutions View*. Apress, 2014.
- [YWRL10] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. Ieee, 2010.
- [ZCCZ11] Fengzhe Zhang, Jin Chen, Haibo Chen, and Binyu Zang. CloudVisor : Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization. pages 203–216, 2011.
- [ZH10] Zhibin Zhou and Dijiang Huang. On efficient ciphertext-policy attribute based encryption and broadcast encryption. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 753–755. ACM, 2010.
- [Zie11] Philipp Zieris. Konzeption und prototypische Umsetzung eines k-Anonymity Algorithmus bei verteilten Daten und prototypische Umsetzung eines Algorithmus bei verteilten Daten. 2011.
- [ZK12] Kamal Zellag and Bettina Kemme. How consistent is your cloud application? In *Proceedings of the Third ACM Symposium on Cloud Computing*, page 6. ACM, 2012.
- [ZLS11] Kun Zhang, Qingzhong Li, and Yuliang Shi. Data Privacy Preservation during Schema Evolution for Multi-tenancy Applications in Cloud Computing *. *LNCS*, 6987(90818001):376–383, 2011.
- [ZVH12] Lan Zhou, Vijay Varadharajan, and Michael Hitchens. A flexible cryptographic approach for secure data storage in the cloud using role-based access control. *International Journal of Cloud Computing*, 1(2/3):201, 2012.
- [ZYG12] Saman Zarandioon, Danfeng Daphne Yao, and Vinod Ganapathy. K2C : Cryptographic Cloud Storage With Lazy Revocation and Anonymous Access. pages 1–18, 2012.

- [ZZCW11] Kehuan Zhang, Xiaoyong Zhou, Yangyi Chen, and Xiaofeng Wang. Sedic : Privacy-Aware Data Intensive Computing on Hybrid Clouds Categories and Subject Descriptors. pages 515–525, 2011.
- [ZZY⁺12] Gaofeng Zhang, Xuyun Zhang, Yun Yang, Chang Liu, and Jinjun Chen. An Association Probability Based Noise Generation Strategy for Privacy Protection in Cloud Computing. *LNCS*, (7636):639–647, 2012.