HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

BACHELORARBEIT

Herr
David Eisele

Analyse und forensische Evaluation von Technologien zur Festplattenverschlüsselung

Mittweida, 2023

Fakultät
Angewandte Computer- und Biowissenschaften

BACHELORARBEIT

Analyse und forensische Evaluation von Technologien zur Festplattenverschlüsselung

Autor:
Herr

David Eisele

Studiengang:
Allgemeine und Digitale Forensik

Seminargruppe:
FO20w1-B

Erstprüfer:
Prof. Ronny Bodach

Zweitprüfer:
M. Sc. Stefan Schildbach

Einreichung:
Mittweida, 24.09.2023

Verteidigung/Bewertung:
Mittweida, 2023

Faculty Applied Computer Sciences

BACHELOR THESIS

Analysis and Forensic Evaluation of Hard Disk Encryption Technologies

author:
Mr.

David Eisele

course of studies:
General and Digital Forensics

seminar group:
FO20w1-B

first examiner:
Prof. Ronny Bodach

second examiner:
M. Sc. Stefan Schildbach

submission:
Mittweida, 24.09.2023

defence/ evaluation:
Mittweida, 2023

Bibliografische Beschreibung:

Eisele, David:
Analyse und forensische Evaluation von Technologien zur Festplattenverschlüsselung. - 2023. - VI, 49 S.
Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2023

Referat:

Die vorliegende Arbeit beschäftigt sich mit der Analyse von Festplattenverschlüsselungssoftware. Dabei werden BitLocker von Microsoft, FileVault von Apple unter MacOS und LUKS unter Linux betrachtet. Des Weiteren wird die Verschlüsselung Data Protection der iPhones und Android Encryption von Googles Betriebssystem analysiert.
Bei der Untersuchung werden die verwendeten kryptographischen Algorithmen, die Schlüsselhierarchie und hardwarebasierenden Technologien behandelt. Zudem werden Angriffsmöglichkeiten präsentiert, die bei einer forensischen Analyse verwendet werden können, um die Verschlüsselung zu umgehen.

Inhalt

Inhalt I

Abbildungsverzeichnis V

Tabellenverzeichnis VI

1 Übersicht 1

1.1 Motivation 1

1.2 Zielstellung 1

1.3 Kapitelübersicht 1

2 Grundlagen 3

2.1 Verschlüsselung 3
2.1.1 Algorithmen 3
2.1.1.1 RSA 4
2.1.1.2 AES 5
2.1.2 Block-Modus 5
2.1.3 Initialisierungsvektor 6

2.2 Full-Disk-Encryption 6

2.3 File-Based-Encryption 7

2.4 Key-Derivation-Function 7
2.4.1 PBKDF2 7
2.4.1.1 Aufbau 7
2.4.1.2 Funktion 8
2.4.2 Alternative Schlüsselableitungsfunktionen 8

2.5 Entropie 9

2.6 Hashfunktionen 9

2.7 HMAC 9

2.8 TPM 9
2.8.1 dTPM 10
2.8.2 fTPM 10

2.9 Secure Enclave 11

3 Angriffsmöglichkeiten 12

3.1 TPM-Angriff ... 12
3.1.1 dTPM-Angriff ... 12
3.1.2 fTPM-Angriff ... 12

3.2 Brute-Force-Angriff ... 14
3.2.1 Wörterbuch-Angriff ... 14
3.2.2 Anti-Hammering-Maßnahmen ... 14

3.3 Rainbowtable-Angriff ... 14

3.4 Cold-Boot-Angriff ... 15
3.4.1 Funktionsweise des Arbeitsspeichers ... 15
3.4.2 Temperatureinfluss auf den Arbeitsspeicher ... 15
3.4.3 Abbildung des Arbeitsspeichers ... 16
3.4.3.1 PXE Network Boot ... 16
3.4.3.2 USB-Drive Boot ... 17
3.4.4 Angriff auf Schlüssel im Arbeitsspeicher ... 17
3.4.4.1 Abbild des Arbeitsspeichers erstellen ... 17
3.4.4.2 Schlüssel-Rekonstruktion ... 17
3.4.4.3 Schlüsselsuche ... 18
3.4.4.4 Gegenmaßnahmen ... 18
3.4.4.5 Zusammenfassung ... 19

4 Verschlüsselungssoftware ... 21

4.1 BitLocker ... 21
4.1.1 Voraussetzung ... 21
4.1.2 Funktionsweise ... 21
4.1.3 Schlüsselhierarchie ... 21
4.1.4 Verschlüsselungsart ... 22
4.1.4.1 BitLocker mit TPM-Only ... 22
4.1.4.2 BitLocker mit TPM und PIN ... 22
4.1.4.3 BitLocker mit TPM und USB ... 23
4.1.4.4 Bitlocker ohne TPM ... 23

4.2 FileVault ... 23
4.2.1 Erweiterung durch den T2-Chip ... 23
4.2.2 Schlüsselhierarchie ... 24

4.3 LUKS ... 25
4.3.1 Schlüsselhierarchie ... 25
4.3.2 LUKS-Partition ... 25
4.3.3 Hauptschlüssel-Wiederherstellung ... 26
4.3.4 Schwachstelle von LUKS ... 27

4.4 Data Protection ... 27
4.4.1 Aufgaben de Secure Enclave ... 27
4.4.2 Schlüsselhierarchie ... 27

4.4.3 Biometrische Authentifizierung ... 28

4.5 Android Encryption ... 28
4.5.1 Schlüsselhierarchie ... 29
4.5.2 File-Based Encryption ... 29

5 Forensische Analyse ... 31

5.1 BitLocker ... 31
5.1.1 TPM-Analyse ... 31
5.1.1.1 fTPM-Only-Variante ... 31
5.1.1.2 fTPM-PIN-Variante ... 31
5.1.1.3 dTPM-Only-Variante ... 32
5.1.1.4 Zusammenfassung ... 32
5.1.2 Cold-Boot-Angriff Analyse ... 33

5.2 FileVault ... 34
5.2.1 Brute-Force Analyse ... 34
5.2.2 Cold-Boot-Angriff Analyse ... 34

5.3 LUKS ... 35
5.3.1 Brute-Force Analyse ... 35
5.3.1.1 Brute-Force-Angriff mittels Elcomsoft ... 35
5.3.1.2 PBKDF2 Schwachstelle ... 36
5.3.1.3 Verbesserung durch Argon2 ... 36
5.3.2 Cold-Boot-Angriff Analyse ... 36

5.4 Data Protection ... 36
5.4.1 Forensische Analyse ... 37
5.4.2 Secure Enclave Analyse ... 37

5.5 Android Encryption ... 37
5.5.1 Forensische Analyse ... 37
5.5.2 Cold-Boot-Angriff Analyse ... 38
5.5.2.1 Ablauf ... 38
5.5.2.2 Schwierigkeiten ... 38

5.6 PBKDF2 ... 39

6 Vergleich der Festplatten- Verschlüsselungssoftware ... 41

6.1 Anwendbarkeit ... 41
6.1.1 Vergleich ... 41
6.1.2 Ergebnis ... 41

6.2 Verschlüsselungsalgorithmen ... 42
6.2.1 Vergleich ... 42
6.2.2 Ergebnis ... 42

6.3 Wiederherstellungsoption ... 43

6.3.1 Vergleich ... 43
6.3.2 Ergebnis ... 43
6.4 Performanz ... 43
6.4.1 Vergleich ... 44
6.4.2 Ergebnis ... 44
6.5 Benutzerfreundlichkeit ... 44
6.5.1 Vergleich ... 45
6.5.2 Ergebnis ... 45
6.6 Schutz vor Angriffen ... 45
6.6.1 Vergleich ... 46
6.6.2 Ergebnis ... 46
7 Schluss ... 47
7.1 Ergebnisse ... 47
7.2 Bewertung der Arbeit ... 48
7.3 Ausblick ... 48
Literatur 51
Selbstständigkeitserklärung ... 57

Abbildungsverzeichnis

Abbildung 1: HMAC-Funktion 9

Abbildung 2: Ableitung des Storage Keys 13

Abbildung 3: Schlüsselhierarchie BitLocker 22

Abbildung 4: Schlüsselhierarchie bei deaktiviertem FileVault 24

Abbildung 5: Schlüsselhierarchie bei aktiviertem FileVault 24

Abbildung 6: LUKS Partitionsheader 25

Abbildung 7: Hauptschlüssel-Wiederherstellung 26

Abbildung 8: Schlüsselhierarchie Data Protection 28

Abbildung 9: Schlüsselhierarchie Android 29

Abbildung 10: Schlüsselhierarchie Android mit TEE 29

Abbildung 11: PBKDF2 Schema 40

Tabellenverzeichnis

Tabelle 1: Brute-Force-Angriff fTPM und dTPM [30] 32

Tabelle 2: Anwendbarkeit..... 41

Tabelle 3: Verschlüsselungsalgorithmen..... 42

Tabelle 4: Wiederherstellungsoptionen 43

Tabelle 5: Leistungsverlust 44

Tabelle 6: Benutzerfreundlichkeit 45

Tabelle 7: Schutzmaßnahmen 46

1 Übersicht

Im einleitenden Kapitel werden die Motivation und die Aufgabenstellung dieser Bachelorarbeit besprochen. Gleichzeitig erfolgt ein kurzer Überblick zu den einzelnen Kapiteln dieser Arbeit.

1.1 Motivation

In der heutigen Zeit ist Homeoffice nur schwer wegzudenken. Die Pandemie im Jahre 2020 hat gezeigt, dass viele Berufstätige ihre Arbeit auch von zuhause aus verrichten können. Dies in Kombination mit der fortscheitenden Digitalisierung sorgt dafür, dass Unternehmen ihre Angestellten mit Firmenlaptops und -smartphones ausstatten. Auf diesen Geräten werden sensible Daten und Betriebsgeheimnisse verarbeitet. Um Schutz dieser Informationen vor Diebstahl zu gewährleisten, sollten deshalb die Systeme über eine Festplattenverschlüsselung verfügen. Die führenden Betriebssysteme stellen demnach alle eine Festplattenverschlüsselungssoftware zur Verfügung, welche inzwischen auch standardmäßig eingerichtet ist [Abschnitt 4].
Nicht nur Arbeitgeber wollen ihre Geräte schützen. Private Nutzer, aber auch Kriminelle, sind sehr daran interessiert, dass ihre Daten für eine außenstehende Person oder ein Strafverfolgungsorgan unzugänglich sind.

1.2 Zielstellung

Die vorliegende Arbeit beschäftigt sich mit der Analyse von Festplattenverschlüsselungssoftware. Dabei werden BitLocker von Microsoft, FileVault von Apple unter MacOS und LUKS unter Linux betrachtet. Des Weiteren wird die Verschlüsselung Data Protection der iPhones und Android Encryption von Googles Betriebssystem analysiert.

Bei der Untersuchung werden die verwendeten kryptographischen Algorithmen, die Schlüsselhierarchie und hardwarebasierenden Technologien behandelt. Zudem werden Angriffsmöglichkeiten präsentiert, die bei einer forensischen Analyse verwendet werden können, um die Verschlüsselung zu umgehen.

1.3 Kapitelübersicht

Die Bachelorarbeit besteht aus 7 Kapiteln.

Nach der allgemeinen Einleitung des ersten Kapitels werden im Kapitel 2 die Grundlagen der Verschlüsselungstechnik erläutert. Dieses Elementarwissen soll der gesamten Arbeit als Grundlage dienen.

Anschließend werden im Kapitel 3 allgemeine Angriffsmöglichkeiten auf die Festplattenverschlüsselung vorgestellt.

Hinterher wird im Kapitel 4 die Verschlüsselungssoftware der verschiedenen Betriebssysteme präsentiert. Dabei werden die Funktionsweisen, die Schlüsselhierarchien, die kryptographischen Algorithmen und die hardwarebasierenden Technologien betrachtet.

Im Kapitel 5 wird die forensische Analyse beschrieben. Hierbei werden Methoden erklärt, die ein Forensiker anwenden kann, um die Verschlüsselungssoftware zu umgehen, mit dem Ziel Zugriff auf die Daten zu erlangen.

Danach werden im Kapitel 6 die verschiedenen Verschlüsselungssoftwares miteinander verglichen. Hierunter werden die Anwendbarkeit, die verwendeten kryptographischen Algorithmen, die Wiederherstellungsoptionen, die Performanz, die Benutzerfreundlichkeit und der Schutz vor Angriffen beschrieben.

Schließlich werden im Kapitel 7 die Resultate der Bachelorarbeit zusammengefasst. Hierbei werden auch die Leistungen des Bacheloranden aus seiner Sicht skizziert. Zusätzlich wird ein Ausblick auf mögliche Weiterentwicklungen gegeben.

2 Grundlagen

In diesem Kapitel werden die Grundlagen der Verschlüsselungstechnik erörtert, sowie weitere informationstechnische Aspekte behandelt.

2.1 Verschlüsselung

Die Verschlüsselung hat eine lange Geschichte, die mehrere Jahrhunderte zurückreicht. Ursprünglich wurde sie vor allem verwendet, um Nachrichten sicher auszutauschen. Dies hat sich auch bis heute nicht verändert. Fast jeglicher Informationsaustausch über das Internet ist verschlüsselt. Mit der zunehmenden Verbreitung digitaler Speichermedien, auf denen Daten und Dokumente gespeichert werden, hat sich die Bedeutung von Verschlüsselung auch in diesem Bereich verstärkt. Gerade im geschäftlichen Sinne, wo Mitarbeiter mit Laptops und Smartphones von der Firma ausgerüstet werden, müssen die Daten geschützt werden, dass auch im Falle eines Diebstahls keine Firmengeheimnisse veröffentlicht werden.

2.1.1 Algorithmen

Es gibt zwei Arten der Verschlüsselung: die asymmetrische Verschlüsselung und die symmetrische Verschlüsselung [1].

Die asymmetrische Verschlüsselung verwendet zwei unterschiedliche Schlüssel. Einen privaten Schlüssel beim Empfänger zum Entschlüsseln und einen öffentlichen Schlüssel beim Sender zum Verschlüsseln. Dies hat seinen Vorteil im Bereich des Datenaustausches, da nicht jeder Kommunikationskanal einen eigenen Schlüssel braucht, sondern nur der Empfänger seinen öffentlichen Schlüssel zur Verfügung stellen muss, der zu seinem geheimen, privaten Schlüssel gehört. Dadurch entfällt die Notwendigkeit, viele Schlüssel zu verwalten. [1]

Ein weitverbreiteter, asymmetrischer Verschlüsselungsalgorithmus ist der Rivest-Shamir-Adleman (RSA) Algorithmus [1], [2]. Dieser wird in Abschnitt 2.1.1.1 genauer behandelt.

Im Gegensatz dazu verwendet die symmetrische Verschlüsselung nur einen Schlüssel, welcher sowohl zum Verschlüsseln als auch zum Entschlüsseln von Daten verwendet wird. Dies hat den Nachteil, dass jeder Kommunikationskanal seinen eigenen Schlüssel benötigt. [1]

Da es bei der Festplattenverschlüsselung keine Kommunikationskanäle gibt, wird die leistungsstärkere, symmetrische Verschlüsselung verwendet [3].

Einer der bekanntesten symmetrischen Verschlüsselungsalgorithmen ist der Advanced Encryption Standard (AES). Hierbei handelt es sich um eine Blockchiffre, die es in drei unterschiedlichen Versionen gibt, bei denen sich nur die Schlüssellänge unterscheiden [4]. AES wird in Abschnitt 2.1.1.2 erklärt.

2.1.1.1 RSA

RSA ist ein asymmetrisches Verschlüsselungsverfahren, welches von den drei Forschern Rivest, Shamir und Adleman entwickelt wurde, dessen Anfangsbuchstaben den Namen bilden. [5]
RSA wird für den Datenaustausch und der Datensignatur eingesetzt. In der Festplattenverschlüsselung findet hauptsächlich die Datensignatur Verwendung, indem es den Hardwareschlüssel schützt. [6]

Das Verfahren beruht auf der Tatsache, dass es aktuell noch keinen effizienten Algorithmus gibt, der Zahlen in Primfaktoren zerlegt [5].
Die verwendeten Schlüssel basieren auf zwei Primzahlen. Je höher die gewählte Primzahl, desto sicherer ist der Schlüssel. Um die Schlüssel zu erstellen, wird das Produkt der beiden Primzahlen berechnet: [5]

$$n = p * q$$

Anschließend wird das Produkt unter Verwendung der Eulerschen-ϕ-Funktion berechnet: [5]

$$\varphi(n) = (p-1) * (q-1)$$

Danach wird eine Zahl $e \in \mathbb{N}$ gesucht die zu $\varphi(n)$ teilerfremd ist.
Der letzte Parameter d wird wie folgt bestimmt: [5]

$$e * d \equiv mod\ \varphi(n)$$

Der öffentlichen Schlüssel besteht aus e, n und der private Schlüssel aus p, q, d. [5]

Die Verschlüsselung der Daten (x) funktioniert mit dem öffentlichen Schlüssel des Empfängers und folgender Formel: [5]

$$y = x^e mod\ n$$

Die Daten kann dann der Empfänger mit seinem privaten Schlüssel und dieser Formel entschlüsseln: [5]

$$x = y^d mod\ n$$

2.1.1.2 AES

AES ist ein Verschlüsselungsalgorithmus, welcher je nach Schlüssellänge in drei Versionen erscheint. Dabei handelt es sich um eine symmetrische Blockchiffre mit wahlweise 128, 192 oder 256 Bits langen Schlüsseln. Im ersten Schritt des Algorithmus werden die Daten in Blöcke unterteilt. Die Blockgröße beträgt 128, 192 oder 256 Bits. Jeder Block wird dann in eine Tabelle mit vier Zeilen und abhängig von der Schlüssellänge in vier, sechs oder acht Spalten eingeteilt. Ein Element (Box) der Tabelle besteht aus einem Byte. Diese Tabelle durchläuft mehrere Runden. Eine Runde besteht aus vier Verfahren: Substitution, Shift Row, Mix Column und Key Addition. [4]

1. Substitution:

Bei der Substitution wird jedes Byte mit einer Substitutionsbox (S-Box) verschlüsselt. Die S-Box definiert dabei, wie ein Byte des Blocks durch einen anderen Wert ersetzt wird.

2. Shift Row:

Im zweiten Verfahren, der Shift Row, werden die Spalten in einer Zeile um einen bestimmten Wert nach links verschoben. Boxen, die links aus der Tabelle rausfallen, werden rechts neu angehängt. Die erste Zeile bleibt unverändert, die zweite Zeile wird um eine Spalte verschoben, die dritte Zeile um zwei Spalten usw.

3. Mix Columns:

Im Schritt Mix Columns werden die Spalten als Vektor betrachtet und mit einem festen Wert multipliziert. Anschließend wird dieser Wert mit dem vorherigen Ergebnis über die XOR-Operation verknüpft.

4. Key Addition:

Im letzten Schritt, der Key Addition, wird jede Box mit dem aktuellen Rundungsschlüssel, welcher aus dem Eingabeschlüssel generiert wird, mittels XOR verknüpft.

Diese Runden werden auf jeden Block mehrfach angewendet und sind umkehrbar. Das bedeutet, man kann die Blöcke auf dieselbe Art und Weise auch wieder entschlüsseln. [4]

2.1.2 Block-Modus

Wie bereits erwähnt, handelt es sich bei AES und einigen anderen Verschlüsselungsalgorithmen um Blockchiffren, welche die zu verschlüsselnden Daten in Blöcke, meist zwischen 64 und 256 Bits, unterteilen und diese dann einzeln verschlüsseln. Bei Blockchiffren benötigt es noch einen Betriebsmodus bzw. Blockmodus. Dieser gibt vor, wie eine Folge von Klartextblöcken mit einem Blockchiffrieralgorithmus verschlüsselt ist. Die bekanntesten sind “Electronic Codeblock Mode” (ECB), “Cipher Block Chaining Mode”

(CBC), “Cipher Feedback Mode” (CFB), “Counter Mode” (CTR) und der auf Festplattenverschlüsselung spezialisierte XTS-Modus. [7][8]

ECB ist der einfachste Modus, hierbei wird jeder Block auf die gleiche Art und Weise verschlüsselt. Dies hat allerdings den Nachteil, dass gleiche Klartextblöcke auch dieselben Ciffretextblöcke hinterlassen und so ein Zusammenhang erkennbar ist [7].
CBC umgeht dieses Problem. Bei diesem Modus wird jeder Block mit dem vorherigen Block verknüpft. Durch diese Verknüpfung wird jeder Block abhängig von seinen Vorgängern. Für den ersten Block benötigt man einen Initialisierungsvektor, welcher quasi die Verknüpfung zum vorherigen Block repräsentiert [7].
Der CFB Modus ist ähnlich dem CBC Modus, allerdings ist dieser so angepasst, dass die Blockgröße nicht einer festen Größe entsprechen muss, sondern variieren kann [7].
Beim CTR Modus ist die größte Besonderheit im Vergleich zu den anderen, dass die Berechnung des Initialisierungsvektors bei jedem Block neu generiert wird. Dieser setzt sich aus einem Zufallswert (Nonce) kombiniert mit einem Zähler zusammen [7].
Im Gegensatz zu den anderen Blockmodi verwendet XTS zwei AES-Schlüssel. Ein Schlüssel wird zur AES-Blockchiffrierung verwendet und der andere verschlüsselt den sogenannten Tweak-Wert. Der verschlüsselte Tweak-Wert wird wiederum durch die Galois-Polynom-Funktion (GF) und der XOR-Operation mit dem Klartext und dem chiffrierten Text jedes Textblocks verknüpft. Die GF-Funktion stellt dabei sicher, dass Blöcke identischer Daten keinen identischen Chiffre enthalten. [9]

2.1.3 Initialisierungsvektor

Der „Initialization Vector“ (IV, Initialisierungsvektor) ist eine Zufallszahl, die als Eingangswert zu einem kryptographischen Algorithmus verwendet wird [10]. Der IV verhindert eine Wiederholung in der Datenverschlüsselung, sodass es einem Angreifer schwerer fällt, Muster zu erkennen [11]. In Blockchiffren wird das damit realisiert, dass ein Wert des ersten Blocks in den zweiten Block miteinbezogen wird. Da der erste Block keinen Vorgänger-Block hat, wird dafür der IV verwendet [10].

2.2 Full-Disk-Encryption

Bei einer Full-Disk-Encryption (FDE, Vollständige Festplattenverschlüsselung) wird die gesamte Festplatte verschlüsselt. Beim Systemstart hat der Nutzer die Möglichkeit, einen Schlüssel bzw. ein Passwort anzugeben [12]. Durch diese Authentifizierung wird der Entschlüsselungsschlüssel in den Arbeitsspeicher geladen [13]. Alternativ können auch Hardwarekomponenten verwendet werden, die den Schlüssel liefern, wie z. B. TPM oder Smartcards [14]. Ab diesem Zeitpunkt kann der Nutzer sein System komplett entschlüsselt nutzen. Allerdings passiert die Entschlüsselung On-The-Fly, das heißt, die Festplatte

wird nicht vollständig entschlüsselt, sondern nur die gewünschten Daten, die in den Arbeitsspeicher geladen werden [15].

2.3 File-Based-Encryption

Bei einer File-Based-Encryption (FBE, Dateibasierte Verschlüsselung) werden einzelne Dateien oder Dateien-Gruppen mit unterschiedlichen Schlüsseln verschlüsselt. Der Schlüssel der Dateien kann beim Aufrufen durch einen passwort-abgeleiteten Schlüssel entschlüsselt werden. Eine andere Methode erlaubt es den Schlüssel der Datei in seinen Metadaten abzulegen und diesen durch den Benutzerschlüssel sichern. Bei einem System mit mehreren Nutzern sorgt dann allein die Benutzerauthentifizierung dafür, dass jeder Benutzer nur auf seine eigenen Dateien zugreifen kann. [16], [17]

2.4 Key-Derivation-Function

Key-Derivation-Function (KDF, Schlüsselableitungsfunktion) ist eine Funktion, welche ein Benutzerpasswort zu einem Verschlüsselungsschlüssel ableitet. Passwörter sind in der Regel kurz und haben nicht genügend Entropie, so dass sie nicht direkt als Schlüssel für die Implementierung sicherer kryptographischer Systeme verwendet werden können. Mittels KDF wird ein Passwort beliebiger Länge zu einem Pseudozufallsschlüssel mit fester Länge berechnet. Bei dieser Funktion handelt es sich um eine Einwegfunktion. Das bedeutet, dass ein Eingabewert zu einem Ausgabewert führt, wobei es unmöglich ist, vom Ausgabewert auf den Eingabewert zu gelangen. Der Sinn und Zweck der KDF besteht darin, dem Benutzer die Möglichkeit zu bieten, ein Passwort zu verwenden, das leicht zu merken ist, während gleichzeitig ein kryptographisch sicherer Schlüssel für die Verschlüsselung generiert wird. [18]

2.4.1 PBKDF2

2.4.1.1 Aufbau

Die meistverwendete KDF ist die PBKDF2 (Password Based Key Derivation Function Version 2, passwortbasierte Schlüsselableitungsfunktion Version 2). Um Brute-Force-Angriffen auf der Grundlage schwacher Benutzerpasswörter zu begegnen, führt PBKDF2 rechenintensive Operationen ein. Solche Operationen basieren auf einer iterierten Pseudozufallsfunktion (PRF) - z. B. einer Hash-Funktion, Chiffre oder HMAC -, die Eingabewerte auf einen abgeleiteten Schlüssel abbildet. Im Gegensatz zu seinem Vorgänger (PBKDF Version 1), bei dem die Länge des abgeleiteten Schlüssels durch die Länge der zugrunde liegenden PRF-Ausgabe begrenzt ist, kann PBKDF2 Schlüssel beliebiger Länge ableiten. Genauer gesagt, erzeugt PBKDF2 so viele Blöcke *Ti*, wie zur Abdeckung der gewünschten Schlüssellänge erforderlich sind. Jeder Block *Ti* wird durch mehrmalige Iteration der

PRF berechnet, die durch eine Iterationszahl festgelegt wird. Die Länge solcher Blöcke ist durch die Länge der zugrunde liegenden PRF-Ausgabe begrenzt. [19]

2.4.1.2 Funktion

Für PBKDF2 sind die Eingabewerte wie folgt definiert: Es werden ein Passwort *p*, ein Saltwert *s*, die Anzahl der Iterationen c und die gewünschte Länge des abzuleitenden Schlüssels *dkLen* benötigt. Zusätzlich ist die Auswahl einer Pseudozufallsfunktion (PRF) erforderlich. Standardmäßig wird HMAC-SHA1 verwendet. Es besteht jedoch die Möglichkeit, alternativ zu HMAC auch andere Hashalgorithmen wie RIPEMD, SHA-256 oder SHA-512 zu verwenden. Der Ausgabewert ist der abgeleitete Schlüssel *dk*. [19]

$$DK = PBKDF2_{(p,\ s,\ c,\ dkLen)}$$

Der abgeleitete Schlüssel wird definiert als Verkettung von [dkLen/hLen] -Blöcken:

$$DK = T_1||T_2|| \ldots ||T_{[dkLen/hLen]}$$

wobei

$$T_1 = Function_{(p,s,c,1)}$$
$$T_2 = Function_{(p,s,c,2)}$$
$$\ldots$$
$$T_{[dkLen/hLen]} = Function_{(p,s,c,[dkLen/hLen])}$$

Jeder einzelne Block Ti wird wie folgt berechnet

$$T_i = U_1 \oplus U_2 \oplus \ldots \oplus U_c$$

wobei

$$U_1 = PRF_{(p,s||i)}$$
$$U_2 = PRF_{(p,U1)}$$
$$\ldots$$
$$U_c = PRF_{(p,Uc-1)}$$

2.4.2 Alternative Schlüsselableitungsfunktionen

Schlüsselableitungsfunktionen, wie PBKDF2, verwenden rechenintensive Hash-Algorithmen und eine Vielzahl von Iterationen, um die Rechenzeit von Brute-Force-Angriffen zu verlängern. Aufgrund von immer leistungsstärkeren Grafikkarten und der Möglichkeit der Parallelisierung kann die Verlangsamung kompensiert werden. Da der Arbeitsspeicher sehr teuer ist, nutzen neuere Algorithmen diesen Umstand aus und fokussieren sich auf eine hohe Auslastung des Speichers. Scrypt oder Agon2 benötigen für ihre Berechnung einen Vektor mit Werten, der permanent im Speicher gehalten werden muss und diesen blockiert. Durch diese Vorgehensweise wird ihre Sicherheit gesteigert, da Brute-Force-Angriffe mit erheblich höheren Kosten verbunden sind. [20]–[23]

2.5 Entropie

Die Entropie ist die Informationsdichte des Begriffs. Sie gibt Aufschluss darüber, ob sich in einer Sequenz von Elementen wie Buchstaben, Wörter oder Bytes Wiederholungen zeigen. Ein höherer Entropiewert deutet darauf hin, dass das Auftreten von gleichen Elementen geringer ist. Im Gegensatz dazu bedeutet ein geringer Wert, dass es viele Wiederholungen unter den Elementen gibt. [24]

2.6 Hashfunktionen

Bei einer Hashfunktion handelt es sich um eine Einwegfunktion. Das heißt, dass es nicht möglich ist, vom Ausgabewert auf den Eingabewert zu schließen. Der Hashwert stellt das Ergebnis dar, welcher mittels einer Hashfunktion berechnet wurde. Die Hashfunktion wird für Prüfsummen, digitale Signaturen und Passwörter verwendet.[25]

2.7 HMAC

Keyed-Hashing for Message Authentication Code (HMAC) ist eine Art der Verschlüsselung, welche anstelle eines Verschlüsselungsalgorithmus eine Hashfunktion verwendet. HMAC besitzt zwei Konstante *ipad* und *opad* mit den Werten 0x36 und 0x5c, einen Hash-Algorithmus und einen Geheim-Schlüssel. *ipad* und *opad* werden mittels XOR-Operation mit dem Schlüssel verknüpft und fließen mit dem Klartext in zwei Hashfunktionen. (siehe Abbildung 1) [26], [27]

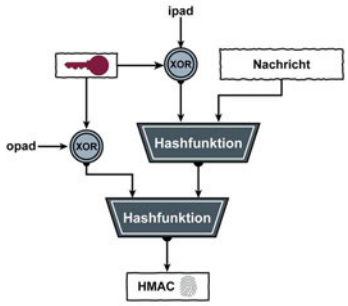

Abbildung 1: HMAC-Funktion

2.8 TPM

Trusted Platform Module (TPM, Vertrauenswürdiges Plattform-Modul) erweitert den Sicherheitsstandard, welcher meist durch Software realisiert wurde, durch eine Hardwarekomponente. In den meisten Fällen ist dies als Co-Prozessorchip auf dem Mainboard platziert. TPM wird zum Ausführen kryptographischer Operationen genutzt, die zum Generieren und Speichern von Verschlüsselungsschlüsseln verwendet werden. Die Anforderungen an TPM werden über eine Reihe von Standards definiert, welche von der Trusted

Computing Group (TCG) definiert wurden. Die TCG ist eine gemeinnützige Organisation, die gegründet wurde, um Industriestandards für vertrauenswürdige Computerplattformen zu entwickeln, die eine hardwarebasierte Vertrauensbasis unterstützen [28]. TPM besitzt einen nichtflüchtigen Speicher, der zum Speichern von kryptographischen Langzeit-Schlüsseln verwendet wird, wovon zwingend zwei Schlüssel hinterlegt sind. Der Endorsement Key (EK, Billigungsschlüssel) und der Storage Root Key (SDK). Der Endorsement Key ist ein Schlüsselpaar, bestehend aus einem privaten und einem öffentlichen Schlüssel. Der private Schlüssel ist im TPM eingebettet und verlässt diesen nie. Der öffentliche Schlüssel ist in einem Zertifikat erhalten und wird nur für wenige Verfahren verwendet, da er einzigartig für jede TPM ist und somit für die Identifizierung des Systems verwendet werden kann. Der Hauptzweck des öffentlichen Schlüssels ist die Verschlüsselung von Daten, die während der Besitzübernahme an den TPM gesendet werden und die Erstellung von Attestation Identity Keys (AIK, Beglaubigungsidentitätsschlüssel). AIK's werden auch als EK aliase bezeichnet. Sie besitzen keinen Zusammenhang zueinander und können somit als Signaturschlüssel verwendet werden, ohne Rückschluss auf die TPM zu geben. Der Storage Root Key hingegen bildet die Grundlage der Schlüsselhierarchie, welche die Festplattenverschlüsselung ermöglicht. Der SDK wird von vielen Festplattenverschlüsselungstools, welche über die TPM laufen, als oberster Schlüssel verwendet. [29]

Auf der Grundlage ihrer Implementierung lassen sich zwei verschiedene Arten von TPMs unterscheiden: diskrete TPMs (dTPMs) und Hardware-TPMs (fTPMs) [30].

2.8.1 dTPM

dTPMs sind dedizierte Hardwarekomponenten, die über eine gewisse physische Manipulationssicherheit verfügen, um gespeicherte Geheimnisse zu schützen. Sie gelten als die sicherste TPM-Variante. Jedoch wurden auch hier physische Angriffe nachgewiesen, die auf den Kommunikationskanal zwischen dem dTPM und dem restlichen System abzielen [30].

2.8.2 fTPM

fTPMs hingegen implementieren die TPM-Funktionen größtenteils über eine Software, welche im Trusted Execution Environment (TEE) einer CPU ausgeführt wird. Da diese Coprozessoren in dem Hauptchip integriert sind, benötigen sie keine externen Busse zur Kommunikation mit der CPU. Somit ist es, im Gegensatz zur dTPM, nicht möglich, den Kommunikationskanal abzuhören. Die TEE bei AMD ist die AMD-SP (AMD Secure Processor) und bei Intel die Intel-ME (Intel Management Engine).

2.9 Secure Enclave

Secure Enclave ist ein dediziertes sicheres Subsystem in den aktuellen iPhones und iPads. Sie ist auf dem SoC (System on a Chip) integriert und vom Hauptprozessor isoliert, um zusätzliche Sicherheit zu gewährleisten. Ihr Aufbau besteht aus einem Boot-ROM, einer AES-Engine und einem True Random Number Generator (TRNG). Der Boot-ROM etabliert durch seinen unveränderlichen Code einen Hardware-Vertrauensanker. Seine Aufgabe besteht darin, die Firmware der Secure Enclave auf Integrität zu prüfen. Die AES-Engine sorgt für eine effizientere und sichere kryptographische Operation. Sie beherbergt die Hard- und Softwareschlüssel. Ein auf dem Gerät laufendes Programm kann Funktionen zur Ver- und Entschlüsselung mittels dem Hardwareschlüssel anfordern. Der Hardwareschlüssel verlässt dabei nicht die Secure Enclave. Der True Random Number Generator (TRNG) dient einzig und allein dazu, sichere Zufallsdaten zu generieren.

3 Angriffsmöglichkeiten

In diesem Abschnitt werden Angriffsmöglichkeiten beschrieben, die bei Festplattenverschlüsselungen Verwendung finden.

3.1 TPM-Angriff

3.1.1 dTPM-Angriff

Da die diskrete TPM eine separate Hardwarekomponente ist, welche außerhalb der CPU steht, wird ein Kommunikationskanal zwischen den beiden Teilen benötigt. TPM-Chips besitzen im Normalfall keine direkten Ports. Anstelle dessen wird das sogenannte Serial Peripheral Interface (SPI) genutzt, welches die Kommunikation ermöglicht. SPI ist die meistverwendete Schnittstelle für die Kommunikation zwischen Mikrocontrollern und Peripheriegeräten. Es besitzt kein festes Protokoll und aufgrund seiner rudimentären technologischen Ausstattung ist die Implementierung von Verschlüsselungsmechanismen nur selten anzutreffen. [31]

Durch die Verwendung eines Logikanalysators, einem elektronischen Messgerät zur Erfassung digitaler Signale [32], und die korrekte Anbindung an die Pins für MOSI, MISO, CS und CLK ist es möglich, die Daten, die über die SPI-Schnittstelle übertragen werden, auszulesen. Die abgefangenen Informationen können dann in einen High Level Analyzer geladen werden. Logic Pro 16, eine hauseigene Software der Marke “Saleae” eignet sich dafür. Um den Schlüssel in den Daten zu finden, entwickelte Henri Numri ein bitlocker-spi-toolkit, welches den Datenfluss decodiert und den Schlüssel extrahiert. Dieses Toolkit ist mit Logic Pro 16 kompatibel und liefert den gefundenen Schlüssel aus. [31]

3.1.2 fTPM-Angriff

Anders als bei dem dTPM kann hier kein Kommunikationskanal abgehört werden, da sich die fTPM im Trusted Execution Enviroment (TEE) des Prozessors befindet. Um die Sicherheitsfunktionen der fTPM zu umgehen, entdeckten Forscher der Technischen Universität Berlin eine Payload, mit dem sie das „Chip Unique Secret“ extrahieren können. Durch das chip-spezifische Geheimnis ist es einem Angreifer möglich, die Verschlüsselungs- und Signaturschlüssel für die nichtflüchtigen Daten der fTPM abzuleiten, die auf dem BIOS-Chip gespeichert sind. Mit den Schlüsseln kann dann der nichtflüchtige Zustand der fTPM entschlüsselt oder verändert werden, um Zugang zu den Speicherschlüsseln der versiegelten TPM-Objekte zu erhalten. Unter TPM-Objekte fallen Storage Keys,

die zum Beispiel von der Microsoft Festplattenverschlüsselungssoftware Bitlocker verwendet werden.

Der Angriff besteht aus einer Voltage Fault Injection (Spannungsfehlerinjektion). Dabei wird ein Mikrocontroller an den SPI-Bus und die Spannungsregler des Prozessors angeschlossen, welche Störungen auf der Stromversorgungsleitung verursachen. Dieser Eingriff behindert den Boot-Prozess des AMD-SP, sodass ein falscher AMD Root Key (ARK) akzeptiert wird. ARK verifiziert alle folgenden Firmwarekomponenten im Bootprozess. Ohne den ist ein Angreifer in der Lage, verschiedene Firmware zu ersetzen oder zu deaktivieren und kann an jeder Stelle seinen eigenen Firmware-Code ausführen.

Ein weiterer Teil ist der oben erwähnte Payload. Er besteht aus einem kurzen ARMv7a-Assemblerteil (der Architektur des AMD-SP), der beliebigen C-Code Payload booten kann.

Abschließend sind die persistierenden Informationen der TPM erforderlich, welche auf dem BIOS-Flash-Chip gespeichert sind. Diese Daten können extrahiert werden und sind durch den Einsatz des PSPTools [33], einem Tool zum Analysieren, Auslesen und Ersetzen von AMD's Firmware [34], interpretierbar. In diesen Daten sind die TPM-Objekte enthalten, welche jedoch mittels eines Storage Keys verschlüsselt sind. Die Ableitung dieses Storage Keys erfolgt gemäß dem folgenden Verfahren.

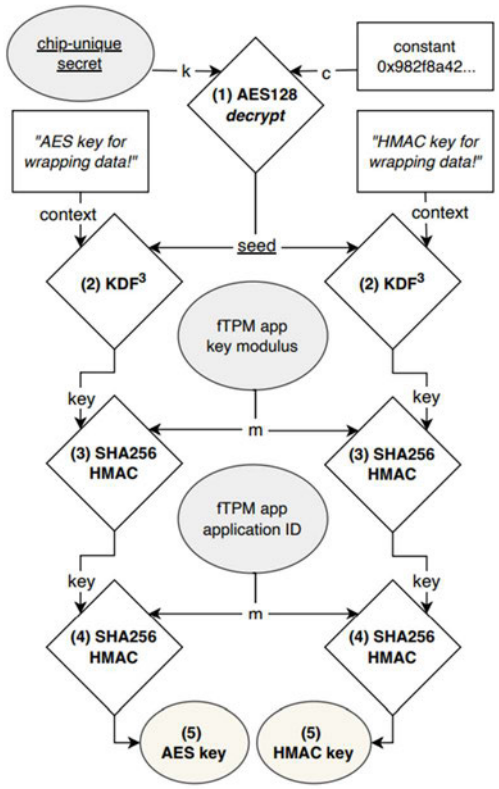

Abbildung 2: Ableitung des Storage Keys

Durch die Spannungsfehlerinjektion lässt sich ein Payload ausführen, welcher das System bootet und Schritt 1 des Schlüsselableitungsprozesses berechnet. Als Ergebnis wird der Seed-Wert auf dem SPI-Bus ausgegeben und kann mit einem Logikanalysator extrahiert werden. Der Seed-Wert ist alles, was benötigt wird, um dann damit den Storage Key und den HMAC Key abzuleiten. Das Chip-Unique-Secret ist damit zwar noch nicht bestimmt, wird aber zum Entschlüsseln der TPM-Objekte nicht benötigt. Jacob et al. beschreiben in ihrem Artikel welche weiteren Schritte zum Bestimmen des Geheimnisses führen.

3.2 Brute-Force-Angriff

Unter einer Brute-Force-Attacke versteht man eine Angriffsmethode, bei der mithilfe hoher Rechenleistung abgesicherte Zugänge durch wiederholte und systematische Eingabe von Nutzer-Passwort-Varianten und Kombinationen aufgebrochen werden [35]. Solch ein Angriff ist abhängig von der Leistung des Computersystems, der Performanz des verwendeten Werkzeugs und der Stärke des Passworts. Umso leistungsstärker und effizienter die Kombination aus Computer und Software ist, desto mehr Passwörter können getestet werden [35]. Aktuelle Brute-Force-Werkzeuge unterstützen GPU- (Graphics Processing Unit, Grafikprozessor) beschleunigte Angriffe, die ungefähr 12.000 Kennwörter pro Sekunde anwenden können [36]. Die Angriffsmethode kann durch Wörterbücher oder Rainbowtables verbessert werden.

3.2.1 Wörterbuch-Angriff

Ein Wörterbuch-Angriff ist eine Erweiterung des Brute-Force-Angriffs. Hierbei werden anstelle von zufälligen Passwörtern Listen genutzt, welche die meistverwendeten Kennwörter enthalten. [37]

3.2.2 Anti-Hammering-Maßnahmen

Anti-Hammering-Maßnahmen sind Vorkehrungen, die einen Brute-Force-Angriff beeinträchtigen. Dies kann zum Beispiel durch Limitierung der Anmeldeversuche in einem bestimmten Zeitraum oder durch kostspielige und zeitintensive Schlüsselableitungsfunktionen (KDF) erreicht werden. [37]

3.3 Rainbowtable-Angriff

Bei einem Rainbowtable handelt es sich um eine Datenbank, die zum Knacken von Hashwerten verwendet wird. Genauer gesagt beinhaltet die Datenbank vorberechnete Hashwerte von Eingabewerten einer bestimmten Hashfunktion. Ist die verwendete Hashfunktion bekannt, so kann der Ausgabewert mit der Datenbank verglichen werden und gibt somit Rückschluss auf den Eingabewert. [38]

3.4 Cold-Boot-Angriff

Bei der Festplattenverschlüsselung ist die gesamte Festplatte durchgehend verschlüsselt. Damit der Benutzer trotzdem auf seine Daten zugreifen kann, muss die Entschlüsselung im laufenden Betrieb geschehen. Um dies zu realisieren, benötigt das System permanenten Zugriff auf den Schlüssel, welcher dafür im Arbeitsspeicher abgelegt wird. Ein Cold-Boot-Angriff zielt darauf ab, den Inhalt des Arbeitsspeichers zu extrahieren. Somit ist es eine valide Angriffsmethode auf die Festplattenverschlüsselung. [13]

3.4.1 Funktionsweise des Arbeitsspeichers

Beim Arbeitsspeicher handelt es sich um einen flüchtigen Speicher. Das heißt, dass Daten nur unter Stromzufuhr erhalten bleiben. Wenn diese unterbrochen wird, kehren die gespeicherten Bits in ihren ursprünglichen Zustand zurück. [13]
Der Verlust der Daten geschieht allerdings nicht sofort und ähnelt eher einem kontinuierlichen Zerfallsprozess [13].

Der Arbeitsspeicher besteht aus mehreren DRAM-Zellen. Eine DRAM-Zelle ist im Wesentlichen ein Kondensator. Jede Zelle kodiert ein einzelnes Bit, indem einer der Leiter des Kondensators entweder geladen oder nicht geladen wird. Der andere Leiter ist je nach Adresse der Zelle innerhalb des Chips entweder mit der Stromversorgung oder mit der Masse verbunden. Über die Zeit verlieren die Kondensatoren ihre Ladung und kehren zum Ursprungswert zurück. Um das zu verhindern, werden die Zellen regelmäßig neu geladen. Diese Periodenzeit, auch Refreshed Time genannt, variiert bei unterschiedlicher Hardware und liegt im Millisekunden-Bereich. Wird eine DRAM-Zelle vom Strom getrennt, ist sie nicht mehr in der Lage, ihre Ladung zu erneuern. [13]

3.4.2 Temperatureinfluss auf den Arbeitsspeicher

Seit geraumer Zeit ist bekannt, dass niedrige Temperaturen einen Einfluss auf die Funktionen von elektronischen Systemen haben. Aus diesem Grund haben Forscher untersucht, wie sich der Datenerhalt im Arbeitsspeicher unter Normalbedingungen im Vergleich zu extremen Kältebedingungen verhält [39].

Für ihren Versuch nutzten sie verschiedene RAM-Modelle, die mit zufälligen Daten beschrieben wurden. Die Daten wurden dann nach unterschiedlichen Zeiträumen und bei unterschiedlichen Temperaturen ausgelesen. Mit den Anfangswerten und den im Experiment erhaltenen Werten, berechnete sich eine Zerfallsrate für jede Probe [13].

Der Zerfall unter normalen Betriebstemperaturen, die zwischen 25,5 °C und 44,1 °C liegen, wurde auf 2,5 bis 32 Sekunden bemessen. Dabei wurde festgestellt, dass ältere Modelle eine signifikant langsamere Zerfallsgeschwindigkeit aufweisen als neuere Modelle.

Um den Zerfall unter Kältebedingungen zu testen wurde der Arbeitsspeicher auf niedrige Temperaturen abgekühlt. Dabei verwendeten die Forscher zwei unterschiedliche Methoden. [13]

In der Ersten wird ein Dosenluft-Spray verwendet, welches üblicherweise aus Fluorkohlenwasserstoff-Kältemittel besteht. Sie drehten die Dose um, damit die Flüssigkeit und nicht das Gas entweicht, und besprühten die DRAM-Zelle. Dieses Vorgehen kühlt den Arbeitsspeicher auf -50 °C herunter. Anschließend wurde die Stromzufuhr unterbrochen und Messungen in unterschiedlichen Zeitabständen durchgeführt. Die Ergebnisse zeigen, dass nach 10 Minuten nur weniger als 1 % der Bits zerfallen sind. [13]

Bei der anderen Variante wurde der Speicher von der Stromquelle getrennt und in einem Behälter mit flüssigem Stickstoff auf eine Temperatur von -196 °C gekühlt. Wieder wurden die Daten nach unterschiedlichen Zeitabständen gemessen. Hierbei zeigt sich, dass nach 60 Minuten nur 0,17 % der Bits zerfallen sind. [13]

In den Experimenten unter Kältebedingungen wurden ausschließlich die Modelle verwendet, bei denen unter Normalbedingungen eine hohe Zerfallsgeschwindigkeit festgestellt wurde.

3.4.3 Abbildung des Arbeitsspeichers

Um einen Angriff auf die Festplattenverschlüsselung zu starten, wird eine Kopie (Image) vom Arbeitsspeicher benötigt. Da es möglich ist, DRAM-Einheiten kurzzeitig vom Strom zu lösen, gibt es mehrere Methoden, davon ein Abbild zu erstellen. Das Abbilden des Speicherinhalts erfordert keine spezielle Ausrüstung. Wenn das System hochfährt, beginnt der Speicher-Controller mit der Auffrischung des DRAM, indem er jeden Bit-Wert liest und neu schreibt. Zu diesem Zeitpunkt sind die Werte fixiert, der Zerfall hält an, und die auf dem System laufenden Programme können alle vorhandenen Daten mit normalen Speicherzugriffsbefehlen lesen. Eine Herausforderung besteht darin, dass beim Booten des Systems zwangsläufig einige Teile des Speichers überschrieben werden. Somit wäre das Laden eines kompletten Betriebssystems destruktiv. Anstelle dessen kann ein winziges Spezialprogramm verwendet werden, das beim Booten ein genaues Abbild des Speicherinhalts auf ein externes Medium erzeugt. [13]

So ein Spezialprogramm kann auf verschiedene Art und Weise gebootet werden.

3.4.3.1 PXE Network Boot

Viele PCs unterstützen Intels Preboot Execution Environment (PXE), welches rudimentäre Start- und Netzwerkdienste bereitstellt. Eine kleine Anwendung kann, nachdem das Zielsystem kurzzeitig heruntergefahren wurde, über PXE gebootet werden und den Speicherinhalt über UDP auf das Analysesystem streamen. [13]

3.4.3.2 USB-Drive Boot

Alternativ dazu kann ein kleines Linux basiertes Betriebssystem gebootet werden, welches den Inhalt des Arbeitsspeichers in eine weitere Partition des USB-Sticks schreibt. [13]

3.4.4 Angriff auf Schlüssel im Arbeitsspeicher

3.4.4.1 Abbild des Arbeitsspeichers erstellen

Der einfachste Angriff ist ein Systemneustart und BIOS-Konfiguration, bei dem ein ausgewähltes Imaging-Tool startet. Ein Warmstart, der über das Neustartverfahren des Betriebssystems ausgelöst wird, stellt normalerweise sicher, dass der Speicher keine Chance hat, zu zerfallen. Allerdings können Vorkehrungen im Betriebssystem getroffen werden, die bei einem Warmstart dafür sorgen, dass der komplette Inhalt des Arbeitsspeichers gelöscht wird. Bei einem Kaltstart ist dies nicht möglich. Hierbei wird der Neustartschalter des Systems betätigt oder die Stromversorgung kurzzeitig ab- und angeschaltet. Das kann, je nach Dauer, aber dafür sorgen, dass Daten zerfallen. [13]

Sollte es einem Angreifer nicht möglich sein, ein Imaging-Tool zu booten, so kann der Arbeitsspeicher physisch entfernt und in ein Analysesystem eingebaut werden. Durch Kühlung der Hardware kann, wie in Abschnitt 3.4.2 beschrieben, der Zerfallsprozess verlangsamt werden. Wenn das Analysesystem über mehrere Steckplätze für den Arbeitsspeicher verfügt, kann ein weiteres Modul eingesetzt werden, welches für den Bereich des Bootens vom BIOS und Betriebssystem verwendet wird. Somit wird kein Teil des zu analysierenden Arbeitsspeichers mit Code überschrieben. [13]

3.4.4.2 Schlüssel-Rekonstruktion

Da es bei einem Imaging Angriff zu einem Zerfall von Informationen kommen kann, besteht die Gefahr, dass Bits in signifikanten Teilen, wie zum Beispiel Verschlüsselungsschlüssel, beschädigt werden. Um diese Fehler zu beheben, erstellten Forscher einen Algorithmus, welcher Verschlüsselungsschlüssel rekonstruieren kann. [13]

Ein naiver Ansatz ist ein Brute-Force Angriff von Schlüsseln mit geringer Varianz zum extrahierten Schlüssel. Solch ein Angriff kann allerdings zeitaufwändig werden, wenn der Zerfall zu hoch gerät. Bei einem 256 Bits langen Schlüssel und einer Zerfallsquote von 10 % liegt die Anzahl an Permutationen bei 256. [13]

Der in Halderman et al. beschriebene Algorithmus der Forscher ist deutlich effektiver. Er beruht darauf, dass viele Verschlüsselungsprogramme durch Vorberechnungen beschleunigt werden. Bei Blockchiffren ist dies meist eine Key-Schedule (Schlüsselplan). Im Schlüsselplan sind die verschiedenen Rundenschlüssel für jede Runde enthalten. Diese verhindern, dass derselbe Schlüssel für jede Runde verwendet wird. Bei RSA ist es eine erweiterte Form des privaten Schlüssels, welcher die Primzahlen und mehrere abgeleitete

Werte enthält. Diese Daten erhalten viel mehr Struktur als der Schlüssel allein, was eine effektivere Schlüsselrekonstruktion ermöglicht. Der Algorithmus funktioniert dabei völlig eigenständig, da der wiederhergestellte Schlüssel ohne die Prüfung der Richtigkeit, durch Entschlüsselung des Chiffreteils erfolgt. [13]

3.4.4.3 Schlüsselsuche

Der zu analysierende Arbeitsspeicher enthält neben den Verschlüsselungsschlüsseln auch viele weitere Daten. Zudem fehlen jegliche Hinweise, die Aufschluss über den Schlüssel geben. Deshalb wird eine Methode benötigt, welche den Verschlüsselungsschlüssel unter der Datenmenge erkennt. [13]

Ein Ansatz zum Auffinden von Schlüsseln ist es, jede Bytesequenz zu prüfen, ob mit dieser eine Entschlüsselung möglich ist. Diese Methode ist schnell, funktioniert allerdings ausschließlich, wenn keine Bitfehler im Schlüssel durch Zerfall beim Imaging entstanden sind. [13]

Forscher erstellten auch für dieses Verfahren einen Algorithmus namens keyfinder. Dieser basiert, wie bereits bei der Schlüsselrekonstruktion, nicht auf dem Schlüssel an sich, sondern auf dem Schlüsselplan. Mit Hilfe dessen ist es möglich, sowohl Schlüssel mit Bitfehler zu rekonstruieren als auch bereits überschriebene Schlüssel zu finden. In Halderman et al. Paper wird dies genauer beschrieben [13]

3.4.4.4 Gegenmaßnahmen

Cold Boot Angriffe sind schwer zu verteidigen, da es notwendig ist, den Verschlüsselungsschlüssel im Hauptspeicher abzulegen, um Zugriff auf die verschlüsselten Daten zu haben. Schutz dagegen bieten Maßnahmen, die den Schlüssel verwerfen oder unkenntlich machen, sobald ein Angreifer in Kontakt mit dem System kommen kann. Verschlüsselungsschlüssel und alle assoziierten Komponenten (Verschlüsselungsplan) sollten vom Arbeitsspeicher gelöscht werden, sobald das System in den Ruhemodus versetzt oder ein Laptop zugeklappt wird. Ein zusätzlicher Schutzmechanismus besteht in der Unterbindung des Netzwerk- oder Wechseldatenträger-Bootvorgangs, da zahlreiche Imaging-Angriffe auf dieser Grundlage beruhen. Ein physischer Eingriff ist trotzdem noch möglich, erfordert aber weitere Kenntnisse und Ressourcen. [13]

Der wahrscheinlich sicherste Schutz bietet eine Verschlüsselung des Arbeitsspeichers, auch Secure Memory Encryption genannt. Die RAM-Verschlüsselung erfolgt in der Regel in hardware- oder firmwaregestützter Form, wobei spezielle Hardwarekomponenten oder spezialisierte Prozessoren für die Verschlüsselung und Entschlüsselung des RAM-Speichers verantwortlich sind. Solche Technologie ist noch in den Anfängen, könnte aber in Zukunft zum Standard werden. [40]
Ein weiterer physischer Schutzmechanismus ist das Verbauen eines Temperatursensors in der Nähe der RAM-Module. Wenn dieser Sensor extremen Temperaturabfall erkennt,

kann das System darauf reagieren und sensible Informationen auf dem Arbeitsspeicher löschen. [41]

3.4.4.5 Zusammenfassung

Mit dem keyfinder-Tool, welches in der Lage ist, Verschlüsselungsschlüssel aufzufinden und zu rekonstruieren, verfügt ein potenzieller Angreifer über ein äußerst effektives Mittel, Angriffe auf die Festplattenverschlüsselung durchzuführen. Beim Zielsystem muss es sich um ein eingeschaltetes oder angehaltenes System (System im Ruhemodus) handeln, bei dem bereits eine Benutzerauthentifizierung dafür sorgte, dass der Verschlüsselungsschlüssel in den Hauptspeicher geladen wurde. Mit erfolgreicher Abbildung des Arbeitsspeichers und unter Verwendung von keyfinder, ist die Schlüsselextraktion fast vollständig automatisiert und dauert nur wenige Minuten. Das genauere Verfahren wird in den Abschnitten 5.1.2, 5.2.2 und 5.3.2 bei den Angriffen auf die jeweilige Festplattenverschlüsselung beschrieben.

4 Verschlüsselungssoftware

In diesem Kapitel werden die unterschiedlichen Verschlüsselungssoftwares analysiert. Dabei werden die Funktionsweisen, die Schlüsselhierarchien, die kryptographischen Algorithmen und die hardwarebasierenden Technologien betrachtet.

4.1 BitLocker

BitLocker ist eine von Microsoft entwickelte Software zur Verschlüsselung von Partitionen und externen Laufwerken, welche unter Windows Vista im Jahre 2007 eingeführt wurde [42]. Als Verschlüsselungsalgorithmus wird AES im Blockmodus XTS in den Versionen 128 und 256, sowie RSA 2048 verwendet [43].

4.1.1 Voraussetzung

Für die Verwendung von BitLocker müssen folgende Voraussetzungen erfüllt sein. Das System, auf dem das Verschlüsselungsprogramm laufen soll, benötigt eine separate Partition, die beim Bootvorgang den BitLocker schon vor dem Betriebssystem startet. Eine weitere Hardwarekomponente ist das Trusted Platform Modul (TPM), welches zur Entschlüsselung der Verschlüsselungsschlüssel benötigt wird. Es ist jedoch möglich, die Verwendung von TPM mithilfe von Gruppenrichtlinien zu umgehen. Bei der Einrichtung von Bitlocker wird ein Wiederherstellungsschlüssel generiert, der es ermöglicht, die Verschlüsselung auch ohne die eigentlichen Sicherheitsvorkehrungen aufzuheben. Dieser Wiederherstellungsschlüssel besteht aus einer 48-stelligen Ziffernfolge und sollte sicher aufbewahrt werden. Um den Schlüssel sicher zu verwahren, wird empfohlen, ihn auf einem weiteren Datenspeichermedium zu sichern, ihn in der Cloud zu speichern oder auszudrucken und in einem physischen Safe zu lagern. [44]

4.1.2 Funktionsweise

Im laufenden Betrieb sorgt der BitLocker für die Ver- und Entschlüsselung. Die Daten sind dabei jederzeit auf der Festplatte in verschlüsselter Form abgelegt. Der autorisierte Nutzer sieht jedoch jederzeit seine Daten in unverschlüsselter Form und kann diese bearbeiten. [44]

4.1.3 Schlüsselhierarchie

Die Verschlüsselung funktioniert dabei mit drei aufeinander bauenden Schlüsseln.
Der Full Volume Encryption Key (FVEK) verschlüsselt die Roh- und Nutzerdaten, mittels

AES 128. Alternativ ist auch AES 256 möglich. Der Volume Master Key (VMK) wird verwendet, um den FVEK zu verschlüsseln. Hierbei wird AES 256 verwendet. Der Storage Root Key (SRK) ist der einzige Schlüssel, der innerhalb der TPM generiert wird und über RSA 2048 den VMK verschlüsselt. [44]

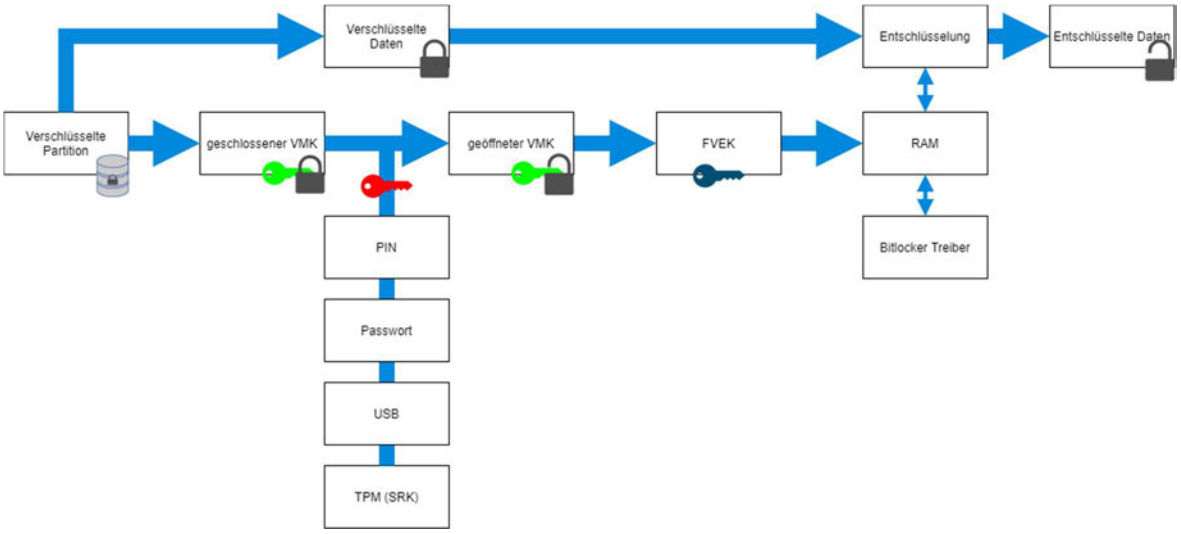

Abbildung 3: Schlüsselhierarchie BitLocker

4.1.4 Verschlüsselungsart

BitLocker bietet folgende Möglichkeiten zur Verschlüsselung der Windows Betriebssysteme:

4.1.4.1 BitLocker mit TPM-Only

Hierbei wird ausschließlich die TPM als Authentifizierung verwendet, welche den SRK ausgibt, der dann die anderen Schlüssel entschlüsselt. Dieses Verfahren sorgt für ein schnelles Booten, da keine weitere Benutzerauthentifizierung benötigt wird. Die Verwendung der TPM gewährleistet dabei, dass die verschlüsselte Festplatte nicht in einem anderen System eingesetzt werden kann, da dort der SRK von der TPM fehlt. Allerdings kann jeder, der über die richtige Hardware verfügt, die Festplatten entschlüsseln. [44]

4.1.4.2 BitLocker mit TPM und PIN

Die reine TPM-Variante wird bei diesem Verfahren durch eine PIN erweitert. Damit der VMK entschlüsselt werden kann, wird eine Kombination aus dem SRK vom TPM und der PIN benötigt. Die PIN besteht aus einer vom Benutzer festgelegten Zahlenkombination, welche im Bootvorgang abgefragt wird. Diese Kombination erhöht die Sicherheit im Vergleich zur Variante ohne PIN, da mehr als die richtige Hardware zur Entschlüsselung der Festplatte benötigt wird. [44]

4.1.4.3 BitLocker mit TPM und USB

Ähnlich wie bei der Variante von TPM und PIN wird hier anstelle eines PINs eine Schlüsseldatei auf einem USB-Stick verwendet, um den VMK in Kombination mit dem SRK zu entschlüsseln. Auch bei dieser Variante wird die Sicherheit im Vergleich zur reinen TPM-Methode erhöht. Ein Vorteil des USB-Sticks ist, dass keine PIN gemerkt werden muss. Allerdings können USB-Sticks verloren gehen, beschädigt oder entwendet werden. [44]

4.1.4.4 Bitlocker ohne TPM

Wenn die Verwendung des TPM nicht erwünscht ist oder dieser schlichtweg fehlt, kann der Schlüssel zum Entschlüsseln des VMK auf einem USB-Stick abgelegt werden. Ein Vorteil dieser Variante ist es, dass Festplatten auch auf verschiedenen Systemen mit unterschiedlicher TPM verwendet werden können. Allerdings kann so eine Funktion auch nachteilhaft sein. Ist ein Angreifer im Besitz des USB-Sticks, so kann er einen Brute-Force-Angriff auf beliebigen Geräten mit deutlich höherer Rechenleistung durchführen. Zudem kommt hinzu, dass, wie bereits erwähnt, USB-Sticks verloren gehen, beschädigt oder entwendet werden können. [44]

4.2 FileVault

FileVault ist das File-Encryption Tool von Apple für deren Laptops und unterstützt seit macOS X 10.7 (20. Juli 2011) FDE unter dem neuen Namen FileVault 2. FileVault ist nicht standardmäßig von Beginn an aktiv und muss vom Nutzer aktiviert werden [45].

Es verwendet das Benutzer-Passwort, sowohl zum Entschlüsseln der Festplatte als auch zur Anmeldung. Bei der Einrichtung von FileVault wird auch automatisch ein Wiederherstellungsschlüssel generiert. Dieser besteht aus 120 Bits, kodiert in Buchstaben und Zahlen von 1-9. Dieser Schlüssel kann ausgedruckt, aber auch in der iCloud gespeichert werden. [45][46]

FileVault 2 nutzt bis 2013 AES-XTS-128 und seit 2013 standardmäßig AES-XTS-256. Die Verwendung eines 128 Bits langen Schlüssels ist nicht empfohlen, wird aber immer noch unterstützt. Zudem ist die Secure Enclave verbaut. Sie sorgt mittels hardwarebasierten Sicherheitsfunktionen und der AES-Engine für sichere und schnelle Verschlüsselung. Ebenso verwaltet die Secure Enclave alle FileVault-Schlüssel. Die für die Verschlüsselung eingesetzten Schlüssel werden der CPU nie direkt offengelegt. [47]

4.2.1 Erweiterung durch den T2-Chip

Seit Oktober 2018 wurden die Laptops mit einem Intel Prozessor mit dem T2 Chip ausgestattet. Dieser Chip ist ein Co-Prozessor, welcher die Sicherheit des Systems verstärkt und den T1 Chip ersetzt. Seine Hauptaufgaben sind Frühstart-Aufgaben, Datenspeicherung im Ruhezustand und die Verschlüsselung der Daten auf der SSD. Weitere

Merkmale, die mit dem T2-Chip einhergehen sind, dass die Festplattenverschlüsselung standardmäßig aktiviert ist und es das Booten von einem externen Gerät verhindert. Letzteres kann allerdings vom Administrator in den Sicherheitseinstellungen geändert werden. Zudem schützt der T2-Chip vor Brute-Force-Angriffen, indem die Benutzeranmeldung auf 30 Anmeldeversuche beschränkt wird. Sind alle Versuche ausgeschöpft, bleibt der Datenträger unbrauchbar und kann nicht wiederhergestellt werden. [48]

4.2.2 Schlüsselhierarchie

Bei deaktivierter FileVault-Funktion werden die Festplatten mittels eines Schlüssels für die Festplattenverschlüsselung über AES-XTS-256 verschlüsselt. Dieser Schlüssel setzt sich aus dem Hardwareschlüssel und einem xART-Schlüssel zusammen. [47]
xART steht für eXtended Anti-Replay Technology und verhindert Replay-Angriffe. Bei diesem Angriff wird der Prozess bei der Verschlüsselung aufgezeichnet und kann abgespielt werden, um Zugriff zu erhalten [49].

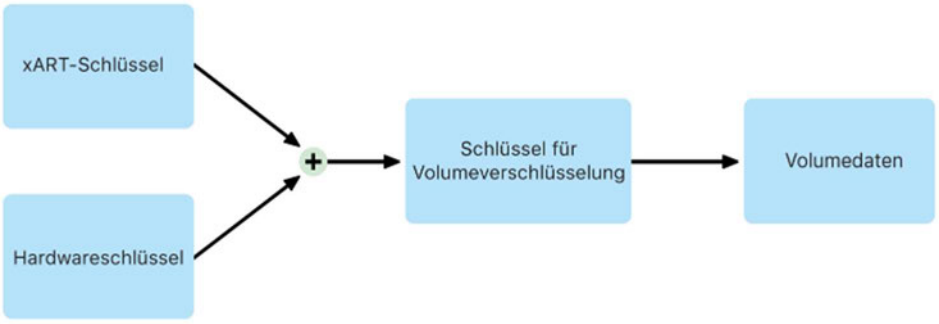

Abbildung 4: Schlüsselhierarchie bei deaktiviertem FileVault

Bei aktivierter FileVault-Funktion wird zu dem Hardwareschlüssel und dem xART-Schlüssel ein Schlüssel für Key Encryption hinzugezogen. Dieser setzt sich aus Benutzerpasswort und Hardwareschlüssel zusammen. [47]

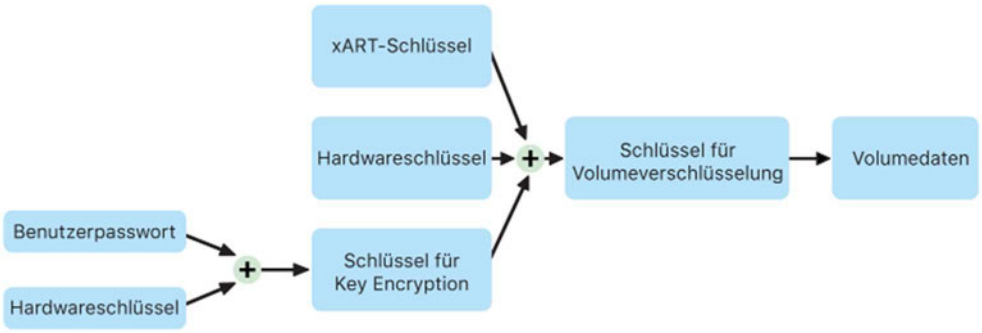

Abbildung 5: Schlüsselhierarchie bei aktiviertem FileVault

4.3 LUKS

Linux Unified Key Setup (LUKS) ist eine Spezifikation zur Festplattenverschlüsselung, die häufig in Linux-basierten Betriebssystemen eingesetzt wird. Es ist ein plattformunabhängiges Standardformat für Festplatten, das 2004 von Clemens Fruhwirth entwickelt wurde. [19]

4.3.1 Schlüsselhierarchie

LUKS basiert auf einer Zwei-Stufen-Schlüsselhierarchie, mit einem Master-Key und einem Benutzerschlüssel. Der Master-Key ver- und entschlüsselt die gesamte Festplatte, wird zufällig vom System erstellt und auf der Festplatte abgelegt. Der Benutzerschlüssel wird hingegen vom Nutzer selbst gewählt und verschlüsselt den Master-Key. Diese Schlüsselhierarchie erlaubt es, einen robusten Verschlüsselungsschlüssel für Festplatten zu generieren, wobei der Benutzer sich lediglich an ein leicht zu merkendes Passwort erinnern muss. Ein weiterer Vorteil dieser Variante ist die Passwortänderung. Um eine höhere Sicherheit zu gewährleisten, ist es ratsam, sein Passwort regelmäßig zu ändern. Der Benutzerschlüssel kann einfach angepasst werden, da dieser nur den Master-Key verschlüsselt, welcher kurz ist. Der Master-Key hingegen sollte sich nicht ändern, denn dann müsste die gesamte Festplatte neu verschlüsselt werden. Dieser Prozess kann sehr zeitaufwendig sein und die Gefahr eines Datenverlusts ist nicht auszuschließen. [19]

Der Nachteil besteht allerdings darin, dass der Benutzerschlüssel tendenziell kurz ist und möglicherweise nur eine geringe Entropie aufweist, damit er sich leicht merken lässt. Wird dieser Schlüssel erraten, so gelangt ein Angreifer an den Master-Key und kann somit die gesamte Festplatte entschlüsseln.
Eine valide Lösung dazu ist es, den Schlüssel mit Hilfe einer Schlüsselableitungsfunktion (KDF, Key Derivation Function) abzuleiten. KDF ist eine Funktion, die aus einer Quelle von ursprünglichem Schlüsselmaterial einen oder mehrere Pseudozufallsschlüssel ableitet. Zudem verlangsamt die KDF den Prozess, Passwörter zu testen, indem der Algorithmus extrem CPU intensiv ist und somit sehr zeitaufwändig ist. Dadurch wird eine Brute-Force-Attacke deutlich verzögert, was das System sicherer macht. [19]

4.3.2 LUKS-Partition

LUKS besitzt eine eigene Partition mit einfachem Layout. (Abbildung 6)

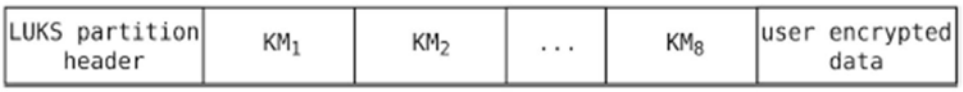

Abbildung 6: LUKS Partitionsheader

Der Partitionsheader beinhaltet Informationen im Klartext über den verwendeten Verschlüsselungsalgorithmus, den Salz-Wert, die Anzahl der Iterationen, die Schlüsselfelder sowie einige weitere Daten. Der Hauptschlüssel kann mit bis zu acht verschiedenen Benutzerschlüsseln verschlüsselt und in einem der acht Schlüsselfelder (KM_{1-8}) gespeichert werden.

4.3.3 Hauptschlüssel-Wiederherstellung

Um an den Hauptschlüssel zu gelangen, wird der Partitionsheader benötigt. Mit dem vorliegenden Benutzerschlüssel wird dann einer der acht Schlüsselmaterialabschnitte entschlüsselt, welche den Hauptschlüssel enthalten (siehe Abbildung 7).

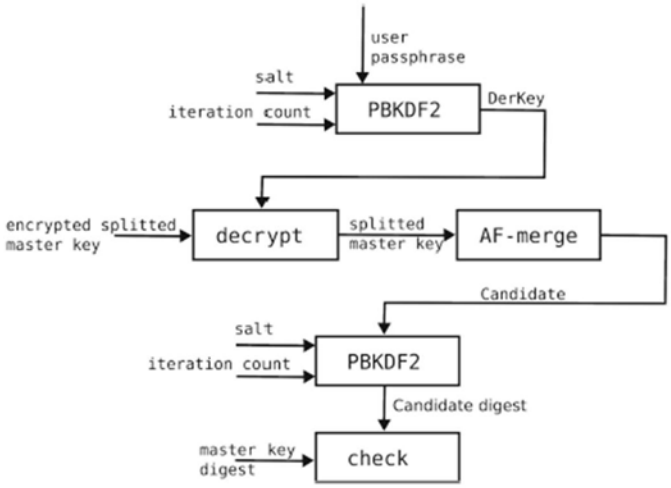

Abbildung 7: Hauptschlüssel-Wiederherstellung

1. Nutzerpasswort p lesen
2. Salt-Wert s aus dem aktiven Schlüsselslot lesen
3. Iterationszahl c aus dem aktiven Schlüsselslot lesen
4. Abgeleiteter Schlüssel (DerKey) mittels PBKDF2 berechnen
5. Startsektor des Schlüsselmaterials aus dem aktiven Schlüsselslot lesen
6. Hauptschlüssel aus dem Schlüsselmaterial lesen
7. Hauptschlüssel mittels abgeleitetem Schlüssel DerKey entschlüsseln
8. Zusammenführen der verschlüsselten Hauptschlüsselteile
9. Iterationszahl c für die Berechnung des Hauptschlüssels lesen
10. Hauptschlüssel mittels PBKDF2 berechnen
11. Vergleich mit dem Ergebnis aus Schritt 10 mit den im Partitionsheader gespeicherten Werten
12. Bei Übereinstimmung war die Wiederherstellung erfolgreich. Ansonsten wird der nächste Schlüsselslot verwendet

4.3.4 Schwachstelle von LUKS

Bei der in Reiter et al. beschriebenen Funktion werden Brute-Force-Attacken zweimal durch die KDF verzögert. Die Erste findet in Schritt 4 statt und leitet den Schlüssel ab, während die Zweite in Schritt 10 die Hauptschlüssel-Prüfsumme ableitet.
Tests zeigen, dass etwa 75-80 % des für die Berechnung eines abgeleiteten Schlüssels erforderlichen Rechenaufwands durch die erste KDF erzeugt wird, während die restlichen 20-25 % durch die Zweite erreicht werden. [19]

Aufgrund der Art und Weise, wie EXT-Dateisysteme aufgebaut sind, befindet sich in den ersten 1024 Bytes der Bootsektor. Mit Hilfe dessen lässt sich testen, ob der abgeleitete Schlüssel passend ist, indem die ersten 1024 Bytes entschlüsselt und auf Merkmale des Bootsektors getestet werden. Somit kann der Schritt 10-12 umgangen werden, womit die Durchführung einer zweiten, zeitaufwändigen KDF entfällt. [19]

4.4 Data Protection

Apples Verschlüsselungssystem für iPhones und iPads wurde unter dem Namen Data Protection mit dem iPhone 3GS und iOS 3.0 veröffentlicht. Der Kern der Verschlüsselung basiert auf einer eindeutigen Gerätekennung, die in Form eines Hardwareschlüssels fungiert und in der Secure Enclave implementiert ist. Diese Gerätekennung wird sicher vor externen Zugriffen sowohl durch Hardware- als auch durch Softwaremechanismen geschützt. [50]

4.4.1 Aufgaben de Secure Enclave

Die Secure Enclave erfüllt mehrere Aufgaben auf dem Gerät. Die Hauptaufgabe ist es, durch isolierte Verschlüsselungsoperationen sensible Benutzerdaten zu schützen. Zudem wird in ihr der Hardwareschlüssel generiert. Zur Fertigungszeit erstellt der TRNG einen zufälligen Schlüssel, welcher über einen Softwareprozess innerhalb der Secure Enclave gespeichert wird. Dieser Prozess stellt sicher, dass der Schlüssel weder von Apple noch von einem Zulieferer eingesehen werden kann. [51]

4.4.2 Schlüsselhierarchie

Jede Datei wird mit einem Dateischlüssel verschlüsselt. Der Schlüssel befindet sich in den Metadaten der Datei. Die Metadaten werden über den File System Key (FSK, Dateisystemschlüssel) und einem Class Key (Klassenschlüssel) geschützt. Der FSK wird bei der Installation von iOS erzeugt und bei jeder Wiederherstellung des Betriebssystems geändert. Das bietet die Möglichkeit, dass beim Zurücksetzen des Gerätes der ursprüngliche FSK verloren geht und somit alle verbleibenden Daten unkenntlich werden. Um den Dateisystemschlüssel zu schützen, wird er durch den Hardwareschlüssel verschlüsselt. Der

FSK befindet sich im eingeschalteten Zustand des Gerätes im Arbeitsspeicher. Zur Abwehr von Angriffen auf den Hauptspeicher wird der Klassenschlüssel hinzugezogen. Dieser Klassenschlüssel verschlüsselt die Metadaten direkt und die resultierende Chiffre wird über den FSK geschützt. Eine Kombination des Hardwareschlüssels und Passwortschlüssel sichert den Klassenschlüssel. Der Passwortschlüssel wird vom Benutzerkennwort mittels PBKDF2 abgeleitet. Durch Richtlinien im Betriebssystem wird dieser gelöscht, sobald die Bildschirmsperre aktiviert wird. Dadurch gewährleistet das System, dass kein Schlüssel zum Entschlüsseln der Dateien auf dem Arbeitsspeicher zu finden ist, sobald die Bildschirmsperre aktiviert wird. [6]

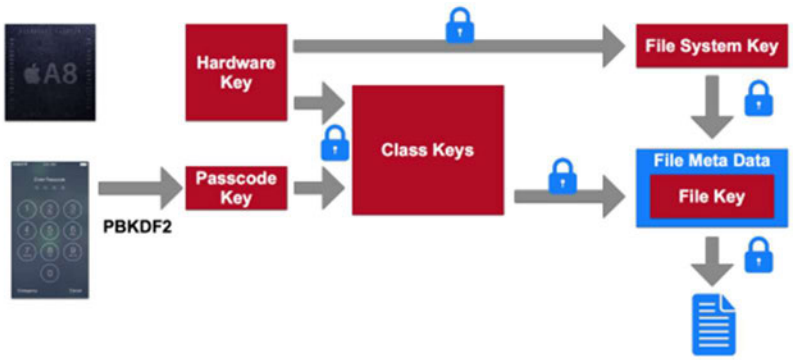

Abbildung 8: Schlüsselhierarchie Data Protection

4.4.3 Biometrische Authentifizierung

Die Sicherheit eines Verschlüsselungssystems bei Data Protection liegt am verwendeten Passwort. Ein starkes Passwort ist effektiv gegen Brute-Force-Angriffe. Allerdings ist deren Eingabe bei regelmäßigem Entsperren des Gerätes sehr aufwändig. Deshalb wird Abhilfe mittels biometrischer Authentifizierung über TouchID und FaceID geschaffen. Diese Anmeldemethode schützt den durch das Passwort abgeleiteten Passwortschlüssel über die Secure Enclave, wo der Passwortschlüssel nach dem erstmaligen Entsperren über das Benutzerpasswort abgelegt wurde. Aufgrund der sowohl software- als auch hardwaretechnischen Isolation der Daten innerhalb der Secure Enclave ist es nicht möglich, diese zu extrahieren. [6]

4.5 Android Encryption

Google veröffentlichte mit Android 4.4 die Full-Disk-Encryption (FDE, Festplattenvollverschlüsselung). Dabei wird tatsächlich nicht der vollständige Speicher verschlüsselt, sondern ausschließlich die Partition mit den Benutzerdaten. Das Verschlüsselungssystem basiert auf dem blockbasierten Festplattenverschlüsselung-Subsystem des Device-Mapper-Systems dm-crypt des Linux-Kernels. Android verwendet dabei AES 128-Bit im CBC-Modus zusammen mit ESSIV:SHA256 für die Generierung des Initialisierungs-Vektors. [52]

4.5.1 Schlüsselhierarchie

Für die Verschlüsselung der Partition der Benutzerdaten ist der Device-Encryption-Key (DEK, Geräte Verschlüsselungsschlüssel), auch „Master-Key“ genannt, zuständig. Die Verschlüsselung läuft dabei mittels aes-cbc-essiv:sha256 und einer Schlüssellänge von 128 Bits. Ein Benutzerpasswort wird mittels scrypt abgeleitet und fungiert als Key-Encryption-Key (KEK, Schlüssel Verschlüsselungsschlüssel). Dieser sichert den DEK mittels aes-cbc und einem Initialisierungs-Vektor, der bei der Erstellung des Master-Keys generiert wurde. [6]

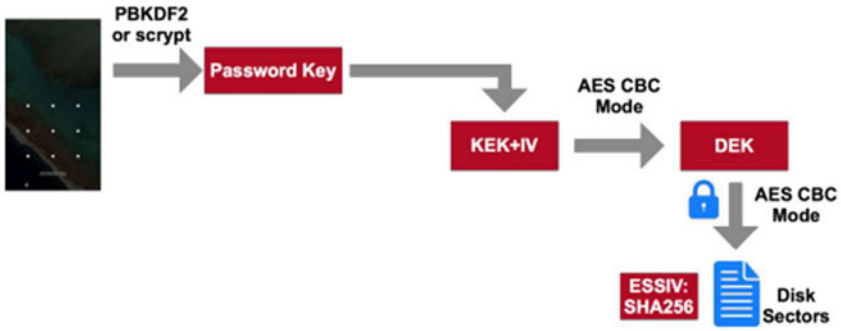

Abbildung 9: Schlüsselhierarchie Android

Mit Android 5.x gab es die Möglichkeit die Generierung des KEK über ein Trusted Execution Environment (TEE, vertrauenswürdige Ausführungsumgebung) zu verbessern, solange die Hardware des Gerätes darüber verfügt. Die Ableitung des KEK besteht seitdem aus dem Benutzerpasswort und einem in der TEE gesicherten Schlüssel, welcher nicht extrahiert werden kann. Diese Erweiterung bindet den Speich an das System und erschwert einen Angriff auf eine ausgebaute Festplatte. [6]

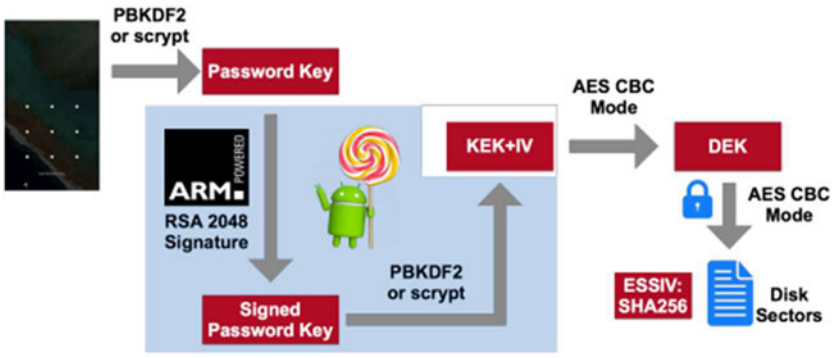

Abbildung 10: Schlüsselhierarchie Android mit TEE

4.5.2 File-Based Encryption

Mit Android 9.0 wurde die File-Based-Encryption (FBE, Dateibasierte Verschlüsselung) eingeführt und ersetzt seit Android 10 die Full-Disk-Encryption. Hierbei wird FDE durch einen File-Encryption-Key (FEK, Datei Verschlüsselungsschlüssel) erweitert, der jedes File

unabhängig voneinander verschlüsselt. Der FEK wird wiederum vom Master-Key gesichert.

Die Einführung der dateibasierten Verschlüsselung ermöglicht die Nutzung von mehreren Benutzern. Jeder Benutzer isoliert dabei seine eigenen Dateien mittels seinem individuellen Benutzerpasswort.

Die Dateien bei der FBE werden zudem in zwei Bereiche eingeteilt. Dem Credential-Encrypted (CE) Storage und dem Device-Encryption (DE) Storage. Beim CE werden die Dateien mit dem FEK und dem Master-Key in Kombination des Benutzerpasswortes verschlüsselt. Somit ist eine Benutzerauthentifizierung nötig, um an die Daten zu gelangen. Die Dateien im DE-Speicher werden nur mit einem FEK verschlüsselt, der ausschließlich über den Master-Key gesichert ist. Dadurch können die Daten auch gelesen werden, wenn sich kein Benutzer angemeldet hat.

Diese Aufteilung in zwei unterschiedliche Verschlüsselungshierachienen bringt folgenden Vorteil mit sich. Benutzerdaten sind während der Bildschirmsperre unzugänglich, da die Daten im CE-Speicher eine Benutzerauthentifizierung benötigen. Die Daten im DE-Speicher sind davon nicht betroffen und stehen auch bei einer aktiven Bildschirmsperre zur Verfügung. Dies ermöglicht es, Telefon- oder Wecker-Diensten, aber auch weitere Anwendungen dauerhaft bereit zu stellen. [16]

5 Forensische Analyse

In diesem Kapitel wird die forensische Analyse beschrieben. Hierbei werden Methoden erklärt, die ein Forensiker anwenden kann, um die Verschlüsselungssoftware zu umgehen, mit dem Ziel Zugriff auf die Daten zu erlangen.

5.1 BitLocker

BitLocker gewährleistet die Sicherheit über die TPM. Diese wird hierbei genauer betrachtet und es werden weitere Angriffsmöglichkeiten beschrieben um die Verschlüsselung zu umgehen.

5.1.1 TPM-Analyse

5.1.1.1 fTPM-Only-Variante

Bei der fTPM-Only Variante wird ausschließlich die TPM als Authentifizierung verwendet. Um den in Absatz 3.1.2 beschriebene Angriff durchzuführen, werden zunächst die Metadaten des BitLocker-Datenträgers benötigt. Diese sind unverschlüsselt im Header der verschlüsselten Partition zu finden. Mittels dem *Dislocker* Werkzeug [53] kann dieser analysiert werden. Bei der TPM-Only Variante bestehen die Metadaten aus einer Datei namens “TPM-encoded”. Bei dieser Datei handelt es sich um ein verschlüsseltes TPM-Objekt. Dieses Objekt enthält den VMK, welcher für die Entschlüsselung benötigt wird. Mit dem im Abschnitt 3.1.2 durchgeführten Angriff kann der Storage Key und der HMAC Key extrahiert werden, mit denen dann das TPM-Objekt entschlüsselt wird. Unter Verwendung der Informationen des TPM-Objekts und dem *Dislocker* Befehl *dislocker-fuse* lässt sich der BitLocker-Datenträger entschlüsseln. [30]

5.1.1.2 fTPM-PIN-Variante

Bei dieser Variante wird das Verfahren durch eine PIN, welche der Nutzer beim Starten des Systems eingeben muss, erweitert.
Im Gegensatz zur TPM-Only Varianten befinden sich in den Metadaten mehrere Einträge. Einen Eintrag, welcher die Streckung des PINs beschreibt und einen weiteren Eintrag, welcher den VMK besitzt. Beim letzten Eintrag ist alles mittels AES-CCM und der abgeleiteten PIN verschlüsselt. Die Schlüsselableitung umfasst einen 128 Bits langen Saltwert, welcher im ersten Eintrag zu finden ist, und 1.048.567 SHA256-Runden, welche die Geschwindigkeit eines Brute-Force-Angriffs bremsen. Die Tabelle 1 veranschaulicht die Geschwindigkeit eines Brute-Force-Angriffs unterschiedlicher PIN-Längen auf fTPM und dTPM. Als GPU wurde die NVIDIA GeForce Titan X Grafikkarte verwendet. [30]

Tabelle 1: Brute-Force-Angriff fTPM und dTPM [30]

Passwortlänge	min-Entropy	fTPM	dTPM
4 Ziffern	2^9	0,5 s	3,5 Tage
10 Ziffern	2^{15}	33 s	7,3 Monate
4 Zeiche	2^{21}	34 min	41 Jahre
10 Zeichen	2^{36}	2,1 Jahre	$1,3 * 10^6$ Jahre

5.1.1.3 dTPM-Only-Variante

Wie bereits in Absatz 3.1.1 beschrieben ist der Kommunikationskanal zwischen der dTPM und dem Prozessor abhörbar. Dabei kann der VMK Schlüssel beim Systemstart abgefangen werden. Dieser kann auf dieselbe Art und Weise wie in 5.1.1 verwendet werden und die Festplatte mittels dem Dislocker Tool entschlüsseln. [30]

5.1.1.4 Zusammenfassung

Zusammengefasst:

Die beschriebenen Angriffe zeigen, dass ein TPM-Only-Schutz unwirksam ist, wenn der interne Status der TPMs extrahiert werden kann. Zusätzlich wird deutlich, dass eine fTPM-PIN Strategie kaum höheren Schutz bietet als ein reiner PIN-Schutz, solange kein weiterer Brute-Force-Schutz verwendet wird.
Ein weiteres Kriterium von BitLocker ist die Komplexität, einen TPM-PIN-Schutz einzurichten. TPM-PIN Kombination ist standardmäßig deaktiviert und der Benutzer muss diese in der Group-Policy-Einstellung finden, bevor er sie aktivieren kann.

Anleitung:
(Computer Configuration → Administrative Templates → Windows Components → BitLocker Drive Encryption → Operating System Drives → Require Additional Authentication at Startup) [30], [54]

Zudem muss zum Einrichten das Command-Line-Tool manage-bde verwendet werden. Ebenso kommt hinzu, dass BitLocker nur alphanumerische Zeichen akzeptiert. Um dies zu umgehen, müssen wieder Änderungen in den Group-Policy-Einstellungen vorgenommen werden.

Anleitung:
(Computer Configuration → Administrative Templates → Windows Components → BitLocker Drive Encryption → Operating System Drives → Allow enhanced PINs for startup) [30], [54]

BitLockers Standard TPM-Only Strategie vermittelt einen falschen Eindruck von Sicherheit und eine TPM-PIN Kombination, mit Anti-Hammering-Mechanismen sollte der Standard werden.

5.1.2 Cold-Boot-Angriff Analyse

Bei BitLocker befindet sich der Schlüssel so lange im Arbeitsspeicher, wie der Datenträger eingebunden ist. Eine Bildschirmsperre oder die Verwendung des Energiesparmodus verhindert diesen Angriff nicht. Bei letzterem wird der Zustand des Arbeitsspeicher auf der Festplatte gespeichert und nach Deaktivierung des Energiesparmodus wiederhergestellt, ohne dass der BitLocker-Schlüssel erneut eingegeben werden muss. [13]

Halderman et al. entwickelten einen vollautomatisierten Demonstrationsangriff namens Bit-Unlocker. Dabei wird ein USB-Stick mit einer kleinen Linux-Distribution, einem SYSLINUX-basierten Bootloader und einem FUSD-Treiber, welcher BitLocker-Volumes unter Linux mounten kann, verwendet. [13]

Im Folgenden wird der Ablauf stichpunktartig beschrieben:

- System ausschalten
- USB-Gerät anschließen
- Booten vom USB-Laufwerk
- Erstellung eines Arbeitsspeicherabbilds
- keyfinder wird ausgeführt
    - dieser testet alle Schlüsselkandidaten
- bei erfolgreicher Entschlüsselung wird das Laufwerk eingehängt
- anschließend kann das Laufwerk normal unter Linux verwendet werden

Die Angriffsdauer bei 2 GB Arbeitsspeicher beträgt etwa 25 Minuten. [13]

Bei BitLocker mit einem TPM-Only-Schutz ist es möglich, den Angriff auch dann durchzuführen, wenn das System längere Zeit ausgeschaltet ist. Das liegt daran, dass TPM-Only keine weitere Benutzerauthentifikation benötigt und den Schlüssel direkt beim Booten in den Arbeitsspeicher schreibt. [13]

5.2 FileVault

FileVault bietet aufgrund ihrer vielen Hardwarekomponenten eine hohe Sicherheit. Trotzdem gibt es Angriffsmöglichkeiten, die die Verschlüsselung umgehen kann.

5.2.1 Brute-Force Analyse

Aktuell sind noch keine Angriffe bekannt, welche den T2-Chip umgehen. Aufgrund dieser Sicherheitsfunktion ist es unmöglich, ein physisches Speicherabbild ohne Benutzerauthentifikation zu erstellen, weshalb die Festplatte zur Analyse immer ausgebaut werden muss. Da Brute-Force-Angriffe auf das Benutzerpasswort durch den T2-Chip massiv begrenzt werden, muss der Entschlüsselungsschlüssel erraten werden, welcher aus einer Kombination aus Hardware-ID und Benutzerpasswort besteht. Diesen Schlüssel zu Brute-Forcen, würde eine nicht praktikable Zeit in Anspruch nehmen.

Allerdings zeigten sich ein paar Erfolge durch das kommerzielle Forensik-Tool Passware. Passware war bereits in der Lage, Passwörter von FileVault-geschützten Festplatten von Macs ohne T2-Chip zu knacken. Dieser Angriff basierte auf einem GPU-beschleunigten Brute-Force-Angriff, mit dem zehntausende Passwörter pro Sekunde getestet wurden. In Modellen, die mit dem T2-Chip ausgestattet sind, ist das Passwort nicht auf dem Solid State Drive (SSD) gespeichert und der Chip begrenzt die Anzahl an Anmeldeversuchen. Somit müsste der Entschlüsselungsschlüssel erraten werden, was äußerst zeitaufwendig ist. Passware wurde seitdem mit einem Zusatzmodul erweitert, welches in der Lage ist, die begrenzte Anzahl an Anmeldeversuchen zu umgehen. Die Software ist allerdings nur in der Lage, 15 Passwörter pro Sekunde zu testen. Diese Geschwindigkeit sollte aber für einen Wörterbuchangriff ausreichen und bietet somit eine solide Angriffsmöglichkeit auf weniger starke Passwörter. Nach Angaben von Passware ist das Zusatzmodul nur für Regierungskunden sowie für Privatunternehmen verfügbar, die eine stichhaltige Begründung für seine Verwendung vorlegen können. [55]

5.2.2 Cold-Boot-Angriff Analyse

Unter Verwendung des von Halderman et al. [vorgestellten EFI-Memory-Imaging Programm lässt sich der Arbeitsspeicher eines Macs extrahieren. Mittels keyfinder waren die Forscher in der Lage, den Schlüssel auszulesen. Allerdings ist es nur möglich, 4.080 Bytes jedes 4.096 Bytes großen Festplattenblocks zu entschlüsseln. Das ist auf die Tatsache zurückzuführen, dass der Initialisierungsvektor (IV) unbekannt ist und Einfluss auf den ersten Teil des Blockes hat. Der IV befindet sich ebenfalls im Arbeitsspeicher und könnte daraus extrahiert werden. Jedoch ist dies nur möglich, solange keine Bitfehler entstanden sind, da dieser nicht rekonstruiert werden kann. Bei einem Testangriff auf Mac OS X 10.4 und 10.5 fanden Forscher mehrere Kopien des Anmeldekennworts des Benutzers. Unter dessen Verwendung ist ein Angreifer in der Lage, das gesamte System zu entschlüsseln. [13]

Dieser Angriff wurde auf Modelle ohne T2-Chip angewendet, welcher das Booten von Software, wie zum Beispiel einem Imaging Tool, verhindert. Bei solchen Modellen müsste der Arbeitsspeicher physisch ausgebaut werden.

5.3 LUKS

LUKS ist die einzige Verschlüsselungssoftware ohne Hardwareerweiterung. Trotzdem wird ein hohes Sicherheitsniveau erreicht. Im folgenden werden zwei Angriffsmethoden erläutert.

5.3.1 Brute-Force Analyse

Der Aufbau von LUKS ist wenig komplex. Er besteht aus einem Master-Key, der auf der Festplatte abgelegt wird und durch einen Benutzerschlüssel gesichert ist. Diese schlichte Struktur weist daher nur wenige Angriffspunkte auf. Somit ist ein Brute-Force-Angriff die einzige valide Angriffsmöglichkeit. LUKS ist rein softwarebasiert und verfügt über keinen physischen Schutzmechanismus gegen Brute-Force-Angriffe, wie beispielsweise Anti-Hammering-Maßnahmen. Es gibt jedoch Ansätze, LUKS mit einer TPM zu kombinieren, um einen derartigen Schutz zu implementieren. Allerdings sind diese noch in den Anfängen und nicht weit verbreitet.

5.3.1.1 Brute-Force-Angriff mittels Elcomsoft

Es existiert Software, die sich auf Festplattenentschlüsselung von LUKS spezialisiert. Eine dieser Software ist Forensic Disk Decryptor, vom russischen Unternehmen Elcomsoft. Diese Software beschäftigt sich zunächst mit dem Extrahieren der Verschlüsselungsmetadaten, die im Partitionsheader (siehe Abschnitt 4.3.2) enthalten sind, um alle relevanten Werte der angewendeten Verschlüsselung zu gewinnen. Dies kann einerseits auf dem analysierenden System geschehen, indem Elcomsoft System Recovery in einer Windows PE-Umgebung implementiert und von einem USB-Stick aus gebootet wird. Andererseits kann es auch im Labor am Untersuchungssystem angewendet werden, bei dem es möglich ist, ausgebaute Datenträgerobjekte oder extrahierte Datenträgerimages zu untersuchen. [36]
Unter Verwendung dieser Informationen lässt sich dann ein Brute-Force-Angriff starten. LUKS bietet einen robusten Schutz gegen derartige Angriffe, indem es tausende Iterationen einer Hash-Funktion durchführt. Allerdings hat die Integration von GPU-Beschleunigung und verteiltem bzw. Cloud-Compution zu einer signifikanten Steigerung der Leistung eines Brute-Force-Angriffs geführt. Elcomsoft Forensic Disk Decryptor ermöglicht bis zu 10.000 Einheiten, die parallel arbeiten, um ein Passwort zu ermitteln. [36]
Zusätzlich bietet Elcomsoft Forensic Disk Decryptor eine Funktion zur Optimierung von Brute-Force-Angriffen durch intelligente Methoden. Informationen über den Eigentümer

oder früher verwendete Passwörter können verwendet werden, um Schlussfolgerungen über die wahrscheinlich verwendeten Zeichengruppen zu ziehen und in das Programm einzubringen. Diese Vorgehensweise trägt zur Verbesserung der Angriffseffizienz bei. [36]

5.3.1.2 PBKDF2 Schwachstelle

Eine Studie, die sich mit der beliebten Schlüsselableitungsfunktion PBKDF2 beschäftigt, zeigt, dass auch hier Sicherheitslücken bestehen, die einen Brute-Force-Angriff begünstigen. Die Schwachstelle erfolgt aufgrund der Möglichkeit, den ersten Nachrichtenblock vorzuberechnen und diesen Wert in allen nachfolgenden HMAC-Aufrufen wiederzuverwenden. Dadurch vermeidet der Angreifer 50 % der rechenintensiven Operationen von PBKDF. [Abschnitt 5.6]
Elcomsoft Forensic Disk Decryptor, in Kombination mit dieser Schwachstelle beschleunigt einen Brute-Force-Angriff weiterhin.

5.3.1.3 Verbesserung durch Argon2

Aufgrund dieser Angriffsmöglichkeit wurde ein Nachfolger von LUKS erstellt. LUKS2 wurde in der Linux-Kernel-Version 4.12 veröffentlicht [56]. Wesentlicher Unterschied zum Vorgänger ist die verwendete KDF. PBKDF2 wurde durch den Sieger des Passwort-Hashing Wettbewerbs Argon2 ersetzt [21]. Dieser sorgt aufgrund seines Designs für eine hohe Belastung des kostbaren Arbeitsspeichers [20]. Im Gegensatz zu PBKDF2, welches auf CPU-Rechenintensive Iterationen setzt, ist die Verwendung von Hochleistungs-Grafikkarten und Parallelisierung weniger erfolgversprechend. Das Forensik-Unternehmen Elcomsoft hat sich in einem Statement zu der Verbesserung wie folgt geäußert: „LUKS2 macht die Verschlüsselung von Linux-Festplatten mit einer neuen Key Derivation Function noch sicherer und verhindert effektiv die GPU-Beschleunigung bei Angriffen auf LUKS2-Volumes.“ [Übers. d. Verf.] [57]

5.3.2 Cold-Boot-Angriff Analyse

Bei LUKS verwendeten die Forscher ein Imaging Tool, welches über den PXE Network Boot ausgeführt wurde. Das keyfinder-Tool konnte den Schlüssel auf dem Arbeitsspeicherabbild extrahieren und die Festplattenverschlüsselung entschlüsseln. [13]

5.4 Data Protection

Data Protection ist Apples Verschlüsselungssoftware für iPhones und iPads. Im folgenden wird die Analyse aus Sicht eines Forensikers beschrieben.

5.4.1 Forensische Analyse

Die forensische Analyse von iPhones und iPads erweisen sich als schwierig [58]. Aufgrund der Schlüsselhierarchie, bei dem sowohl ein Benutzerschlüssel als auch ein Hardwareschlüssel verwendet wird, werden Angriffsmöglichkeiten begrenzt [6]. Der Hardwareschlüssel wird von der Secure Enclave isoliert und verlässt diesen zu keinem Zeitpunkt [6]. Ein Angriff darauf wurde noch nicht veröffentlicht. Des Weiteren sorgt der Benutzerschlüssel für zusätzliche Sicherheit. Dieser wird beim Sperren des Bildschirms vom System gelöscht. Dadurch verringert es die Wahrscheinlichkeit, einen Verschlüsselungsschlüssel bei einem Cold-Boot-Angriff auf den Arbeitsspeicher zu extrahieren [6]. Durch den Hardwareschlüssel und der fehlenden Möglichkeit diesen auszulesen, werden Brute-Force-Angriffe an das System gebunden. Das iPhone und iPad besitzen softwareseitige Anti-Hammering-Maßnahmen, die die Anmeldeversuche auf einen Zeitraum begrenzen [6], [58]. Zudem kann die Funktion eingerichtet werden, dass der Master-Key nach zehn fehlerhaften Anmeldungen gelöscht wird und somit der Zugriff auf die Daten dauerhaft unzugänglich wird [6].

5.4.2 Secure Enclave Analyse

Das Softwareentwickler Team Pangu aus Asien hat 2020 angeblich eine Möglichkeit gefunden, die Secure Enclave von Apple zu knacken und war in der Lage, den Hardware-Schlüssel auszulesen. Das Pangu Team ist für ihre Jailbreaks von Apple-Geräten bekannt und deren Umgehung von Sicherheitsmechanismen. Genauere Informationen darüber, wie der Angriff funktioniert, wurden nicht genannt. Vermutlich zum Schutz von böswilligen Angreifern, da die Sicherheitslücke aufgrund der Isolation der Secure Enclave, bei verkauften Geräten, nicht behoben werden kann. [59]

5.5 Android Encryption

Smartphone unter Verwendung von Googles Betriebssystem Android haben in den letzten Jahren ihre Sicherheit stark verbessert [60]. In diesem Abschnitt wird die forensische Analyse von Android untersucht.

5.5.1 Forensische Analyse

Die forensische Analyse bei Android-Geräten erweist sich als ebenso schwierig wie die bei iOS [60]. Auch hier werden die Daten über eine Schlüsselhierarchie gesichert, die aus einem Hardware- und einem Benutzerschlüssel bestehen [6]. Die TEE schützt dabei den Hardwareschlüssel. Selbst bei einer kompletten Kompromittierung des Kernels kann kein Schlüsselmaterial ausgelesen werden [61]. Ein Angriff darauf wurde noch nicht

veröffentlicht. Des Weiteren sorgt der Benutzerschlüssel für die gleiche Sicherheit wie bei den Apple-Geräten. Dieser verfügt nicht wie bei iOS über eine standardmäßige, softwareseitige Anti-Hammering-Maßnahme. Anstelle dessen wird die *scrypt* KDF verwendet [6]. Durch ihren speicherintensiven Rechenprozess werden Brute-Force-Angriffe selbst mit GPU-Unterstützung erschwert [23].

5.5.2 Cold-Boot-Angriff Analyse

In einer wissenschaftlichen Untersuchung zur Sicherheit von Android-Geräten wurde ein Cold Boot Angriff auf ein Smartphone analysiert. Der Angriff nutzt das Tool FROST (Forensic Recovery Of Scrambled Telephones) [62] zur Durchführung. FROST ist ein Wiederherstellungsimage, welches in die Wiederherstellungspartition eines Smartphones installiert wird, wodurch es möglich ist, den Arbeitsspeicher zu extrahieren. Die Forscher führten diesen Angriff auf einem Samsung Galaxy-Nexus mit Android 4 durch, mit dem Ziel, ein RAM-Abbild zu erstellen, in dem Verschüsselungsschlüssel enthalten sind. [63]

5.5.2.1 Ablauf

Zunächst wird das gesamte Gerät abgekühlt, da es äußerst aufwändig ist, ein Smartphone auseinanderzubauen, sodass direkter Zugriff auf das RAM-Modul möglich ist. Da das untersuchte Gerät keine Reset-Taste besitzt und ein herkömmliches Herunterfahren des Betriebssystems zu lange dauert, was potenziell Informationsverlust verursachen könnte, wird das Gerät durch das kurzzeitige Entfernen des Akkus neu gestartet. Um den Einschaltvorgang des Gerätes zu beschleunigen, muss während dieses Vorgangs die Einschalttaste gedrückt bleiben. Um das FROST-Image auf das Mobiltelefon zu übertragen, ist es notwendig, während des Bootvorgangs zusätzlich die Lauter- und Leisertasten gedrückt zu halten, um in den Fastboot-Modus zu gelangen. In diesem Modus wird das Gerät über USB an ein Linux-System angeschlossen, auf dem anschließend der Befehl "fastboot flash recovery frost.img" ausgeführt wird, um FROST zu starten. Auf dem Telefon selbst wird die Option "Recovery Mode" im Telefonmenü ausgewählt, welches das FROST GUI startet, mit dem dann über die Option „RAM dump via USB“ ein Abbild des Arbeitsspeichers erstellt werden kann. [63]

5.5.2.2 Schwierigkeiten

Ein Cold Boot Angriff weist jedoch einige Schwierigkeiten auf, in Bezug auf das moderne Design von Smartphones und die Art und Weise der Verschlüsselungstechnik.
Der präsentierte Angriff wurde auf einem älteren System durchgeführt, bei dem das Öffnen des Gehäuses und das Entnehmen des Akkus problemlos möglich ist. Allerdings sind neuere Geräte heute oft fest verklebt und verbaut, teilweise auch aus Gründen der Wasserdichtigkeit. Dies macht das Öffnen moderner Smartphones erheblich komplexer und anfälliger für Fehler, die zu Datenbeschädigungen führen können.
Zudem wird unter Android der sogenannte „Master Key“ vom Benutzerpasswort verschlüsselt. Das Benutzerpasswort muss immer eingegeben werden, sobald die

Bildschirmsperre aktiviert wird, da mit der Aktivierung der Bildschirmsperre der Schlüssel aus dem Arbeitsspeicher entfernt wird. Darüber hinaus ist in den meisten Geräten durch Akkuoptimierungssoftware ein Bildschirm-Timeout aktiv, welche das Gerät nach 30 Sekunden Inaktivität sperrt. [6]
Aufgrund dieser beiden Faktoren ist es äußerst unwahrscheinlich, ein entschlüsseltes Gerät zu stehlen, auf dem dann allerdings nur mit extremem Aufwand ein Cold Boot Angriff durchgeführt werden kann.

5.6 PBKDF2

Die passwort-basierte Schlüsselableitungsfunktion PBKDF2 ist weitverbreitet und findet Verwendung in iOS, Android und LUKS1. In diesem Abschnitt wird die Schwachstelle dieser KDF analysiert.

Die Abbildung 11 zeigt:

a. Der erste Nachrichtenblock einer schlüsselbasierten Hash-Funktion wird c-mal wiederholt (dunkelgraue Rechtecke)
b. Der erste Nachrichtenblock einer zweiten schlüsselbasierten Hash-Funktion wird c-mal wiederholt (hellgraue Rechtecke)
c. Alle dunkelgrauen Rechtecke haben den gleichen Inhalt und bestehen nur aus dem Wert SHA1(P $\oplus$ ipad)
d. Alle hellgrauen Rechtecke haben den gleichen Inhalt und bestehen nur aus dem Wert SHA-1(P $\oplus$ opad)

Es ist also möglich, diese beiden Blöcke im Voraus zu berechnen und somit die Werte: SHA-1(P $\oplus$ ipad) und SHA-1(P $\oplus$ opad) c-mal zu verwenden [19].

Durch das Vorberechnen des ersten Nachrichtenblocks und das Wiederverwenden dieser Werte in allen nachfolgenden HMAC-Aufrufen, vermeidet der Angreifer 50 % der rechenintensiven Operationen von PBKDF2 [19].

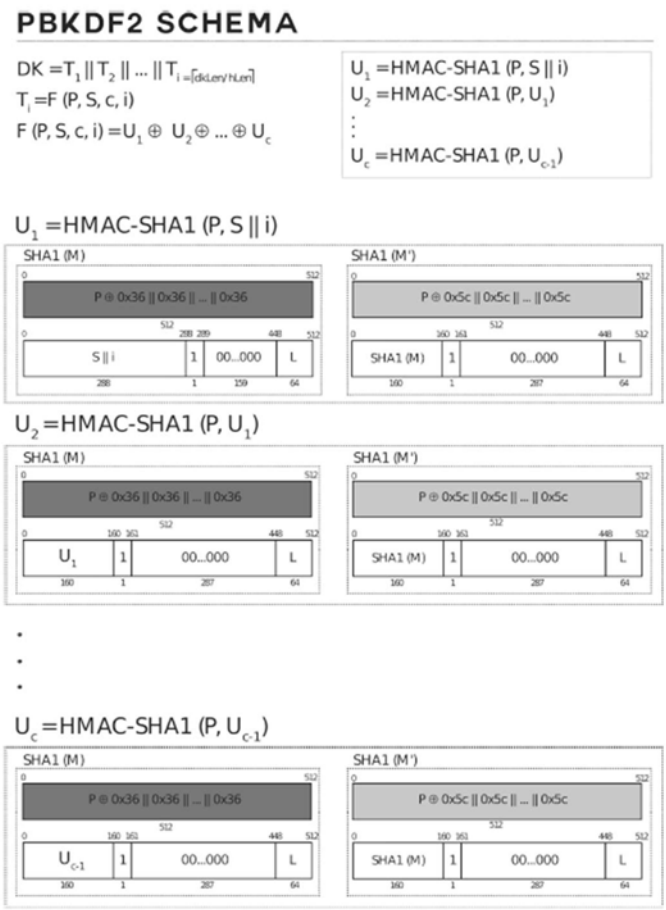

Abbildung 11: PBKDF2 Schema

6 Vergleich der Festplatten-Verschlüsselungssoftware

In diesem Kapitel werden die verschiedenen Verschlüsselungssoftwares miteinander verglichen und in einer Tabelle dargestellt.

6.1 Anwendbarkeit

Dieser Punkt betrachtet die Kompatibilität von Verschlüsselungssoftware mit verschiedenen Betriebssystemen und zeigt, welche Voraussetzungen oder Bedingungen erfüllt sein müssen.

6.1.1 Vergleich

Tabelle 2: Anwendbarkeit

Bitlocker	FileVault	LUKS	Data Protection	Android Encryption
Windows 10, 11 in den Versionen: Pro, Education, Enterprise, bzw. Pro for Workstations, Pro Education. [64] TPM (kann aber umgangen werden)	Ab MacOS X 10.7	Jedes Linux-System seit Kernel-Version 4.12 LUKS2	Ab iOS 3.0	Android 4.4 bis 9 Full-Disk-Encryption Ab Android 10 ausschließlich File-Based-Encryption

6.1.2 Ergebnis

Jede Verschlüsselungssoftware findet Anwendung auf den vorgesehenen Betriebssystemen. Microsoft ist in diesem Zusammenhang der einzige Hersteller, der sein Programm

auf die teureren Windows-Versionen beschränkt. Apple hat aufgrund der Kontrolle über Hard- und Software die Möglichkeit, ihre Verschlüsselungssoftware FileVault und Data Protection sowie ihre Produkte so zu entwickeln, dass sie mit allen neueren Geräten kompatibel sind.

6.2 Verschlüsselungsalgorithmen

Hierunter werden die verschiedenen Verschlüsselungsalgorithmen und weitere kryptographischen Prozesse dargestellt.

6.2.1 Vergleich

Tabelle 3: Verschlüsselungsalgorithmen

Bitlocker	FileVault	LUKS	Data Protection	Android Encryption
AES-XTS 128 oder 256 RSA zur Verschlüsselung des SRK mit der TPM	Bis 2013 AES-XTS 128 Seit 2013 AES-XTS 256	Chiffre: AES, Twofish [36] Blockmodus: CBC, ECB oder XTS[36] KDF: PBKDF2 oder Argon2	Chiffre: AES-CTR 256 KDF: PBKDF2	Chiffre: AES-CBC-128-ESSIV:SHA256 RSA zur Verschlüsselung des KEK KDF: PBKDF2 oder scrypt

6.2.2 Ergebnis

Im Allgemeinen werden in allen Verschlüsselungssoftwares dieselben Algorithmen verwendet. Weshalb sich die Sicherheit aufgrund dessen kaum unterscheidet. LUKS ist die einzige Software, die es dem Benutzer ermöglicht, die verwendete Verschlüsselungstechnik anzupassen. Das hat den Vorteil, dass er sein System in Richtung Performanz oder Sicherheit einrichten kann. Ein weiterer Unterschied liegt im Benutzerschlüssel. Während LUKS und BitLocker auf eine dedizierte Passphrase für die Datenentschlüsselung setzen, greifen die übrigen Hersteller auf das Benutzerkennwort zurück, welches ebenfalls zur Bildschirmentsperrung verwendet wird.

6.3 Wiederherstellungsoption

Die Vollverschlüsselung von Festplatten stellt ihre Sicherheit durch den Schlüssel da. Deshalb ist ein langes und komplexes Passwort empfehlenswert. Solch ein Passwort lässt sich allerdings nur schwierig merken. Deshalb ist es nicht unüblich, dass ein Benutzer seine Anmeldedaten vergisst. Im folgenden Abschnitt wird beschrieben, welche Vorkehrungen und Möglichkeiten die verschiedenen Verschlüsselungssysteme bieten, wenn der Benutzer sein Passwort nicht mehr weiß.

6.3.1 Vergleich

Tabelle 4: Wiederherstellungsoptionen

BitLocker	FileVault	LUKS	Data Protection	Android Encryption
Wiederherstellungsschlüssel bei der Einrichtung. Kann ausgedruckt, auf USB-Stick abgelegt oder in der Cloud abgespeichert werden	Wiederherstellungsschlüssel bei der Einrichtung. Kann ausgedruckt, auf USB-Stick abgelegt oder in der Cloud abgespeichert werden	Kein Wiederherstellungsschlüssel. Allerdings können mehrere Passwörter eine Festplatte schützen. [65]	Keine Wiederherstellungsmöglichkeit [66]	Standardmäßig keine Wiederherstellungsoption Manche Hersteller unterstützen Wiederherstellung mittels dem Google-Konto [67]

6.3.2 Ergebnis

Bei der Wiederherstellungsoption weisen BitLocker und FileVault identische Funktionalitäten auf und bieten dem Benutzer eine Einrichtungsmöglichkeit. LUKS hingegen macht den Benutzer nicht auf die Risiken bei Verlust des Passworts aufmerksam und eine Wiederherstellungsoption ist nur über Umwege möglich. Die Verschlüsselungssoftware bei den Smartphones besitzt keine Wiederherstellungsoption. Bei Verlust des Passworts sind die Daten auf dem Gerät dauerhaft unzugänglich. Manche Hersteller unter Android bieten allerdings die Möglichkeit an, Zugang zum System mittels dem Google-Konto zu erhalten.

6.4 Performanz

Festplattenverschlüsselung bietet Sicherheit im Fall eines Diebstahls. Allerdings ist nicht jeder Benutzer bereit, aufgrund von Sicherheitsaspekten, Abstriche in der Leistung des Systems in Kauf zu nehmen. In diesem Abschnitt wird die Thematik des Leistungsverlusts eingehend untersucht.

6.4.1 Vergleich

Tabelle 5: Leistungsverlust

BitLocker	FileVault	LUKS	Data Protection	Android Encryption
i7-6700HQ CPU SSD: -3,6 % bis 15,1 % Lesen -0,3 % bis 17,0 % Schreiben HDD: -14,9 % bis 16,9 % Lesen -4,8 % bis 27,6 % Schreiben [68]	MacBook Pro 2016: -20,2 % Lesen -8,4 % Schreiben MacBook Pro 2020: -0,09 % Lesen -0,12 % Schreiben [69]	Intel Core i7-5600 SSD: -50,24 % bis 62 % Lesen -57,15 % bis 57 % Schreiben HDD: -0,30 % bis 3,37 % Lesen -5,2 % bis 8,91 % Schreiben [70]	Keine Information, da Verschlüsselung nicht deaktiviert werden kann	Google Nexus 6 -62,86 % Lesen -50,52 % Schreiben [71]

6.4.2 Ergebnis

Im Tabelle 5 ist ein deutlicher Unterschied im Leistungsverlust zu erkennen. Die Hauptursache dafür ist allerdings der verwendeten Hardware zu verschulden.
Moderne, leistungsstärkere Prozessoren weisen eine deutliche Verbesserung in Bezug auf die Verschlüsselungsleistung auf. Dies zeigt sich besonders bei der Analyse von FileVault zwischen dem Mac Book Pro 2016 mit einem Intel Core i7 Prozessor im Vergleich zum Mac Book Pro 2020 mit Apples M1-Chip. Der neuere Prozessor im Mac Book Pro 2020 bietet im Allgemeinen eine bessere Gesamtleistung und resultiert somit in einem geringeren Leistungsverlust bei der Verschlüsselung [72].

6.5 Benutzerfreundlichkeit

Unter diesem Kriterium wird die Benutzerfreundlichkeit der verschiedenen Verschlüsselungssoftwares bei der Einrichtung betrachtet.

6.5.1 Vergleich

Tabelle 6: Benutzerfreundlichkeit

BitLocker	FileVault	LUKS	Data Protection	Android Encryption
Ausreichend: Bei der Einrichtung ist nur die TPM-Only Variante aktiv, wenn ein Microsoft-Konto verwendet wird. Alles andere muss kompliziert über die Group-Policy und die Powershell erfolgen	Sehr gut: Alle Vorkehrungen der Verschlüsselung direkt bei der Einrichtung vorgenommen. Nutzer muss nur ein Passwort wählen	Befriedigend: Nutzer muss die Verschlüsselung selbst über die Befehlszeile einrichten. Manche grafische Installationsprogramme sorgen für eine geführte Einrichtung [65]	Sehr gut: Alle Vorkehrungen der Verschlüsselung direkt bei der Einrichtung vorgenommen. Nutzer muss nur ein Passwort wählen	Sehr gut: Alle Vorkehrungen der Verschlüsselung direkt bei der Einrichtung vorgenommen. Nutzer muss nur ein Passwort wählen

6.5.2 Ergebnis

Die Einrichtung einer Festplattenverschlüsselung unter MacOS, Android und iOS ist am benutzerfreundlichsten, da das Betriebssystem beim ersten Start alle Vorkehrungen trifft.

Bei Windows und Linux kann die Einrichtung benutzerfreundlich sein, sofern BitLocker im TPM-Only Modus und LUKS mittels grafischem Installationsprogramm verwendet wird. Sollte dies allerdings nicht erwünscht sein, gestaltet sich die Erstellung einer Festplattenverschlüsselung als anspruchsvoll.

6.6 Schutz vor Angriffen

Dieser Punkt betrachtet die software- und hardwareseitigen Maßnahmen, die den Schutz vor Angriffen gewährleisten.

6.6.1 Vergleich

Tabelle 7: Schutzmaßnahmen

BitLocker	FileVault	LUKS	Data Protection	Android Encryption
Hardwareschlüssel isoliert durch TPM. Schutz vor Brute-Force durch rechen- und kostenintensive KDF. Anti-Hammering Schutz bei dTPM	Hardwareschlüssel isoliert durch Secure Enclave Schutz vor Fremdbooten und Anti-Hammering durch T2-Chip. Anmeldeversuche auf 30 beschränkt	Schutz vor Brute-Force durch rechen- und kostenintensive KDF	Hardwareschlüssel isoliert durch Secure Enclave. Schutz vor Brute-Force durch rechen- und kostenintensive KDF. Anmeldeversuche auf 10 beschränkt (kann deaktiviert werden)	Hardwareschlüssel isoliert durch TEE. Schutz vor Brute-Force durch rechen- und kostenintensive KDF.

6.6.2 Ergebnis

Jede Festplattenverschlüsselung, mit Ausnahme von FileVault, ist im Besitz einer KDF, die einen Brute-Force-Angriff einschränkt. Bei LUKS ist dies die einzige Schutzmaßnahme, da es keine Hardwarekomponenten voraussetzt. Alle anderen isolieren ihren Hardwareschlüssel über eine Systemkomponente. Die Software von Apple beschränkt zusätzlich die Anzahl der Anmeldeversuche. Bei BitLocker sorgt die dTPM und bei FileVault der T2-Chip für weiteren Anti-Hammering-Schutz.

7 Schluss

Im abschließenden Kapitel werden die gewonnenen Ergebnisse zusammengefasst und die Leistung aus Sicht des Autors auf seine Arbeit bewertet. Ein Ausblick zeigt Möglichkeiten der Weiterentwicklung in der Festplattenverschlüsselung.

7.1 Ergebnisse

Das Bachelorthema „Analyse und forensische Evaluation von Technologien zur Festplattenverschlüsselung“ ergab sich aus der weitverbreiteten Festplattenverschlüsselung bei mobilen Endgeräten. Diese Schützen die Daten auf einem System vor nicht autorisiertem Zugriff. Da bei einer kriminaltechnischen Untersuchung Laptops und Smartphones immer mehr in den Fokus rücken, schaffte die Analyse und forensische Evaluation der Festplattenverschlüsselung die Grundlage für diese Bachelorarbeit.

Nach der Findung des Themas wurden die Grundlagen der Verschlüsselungstechnik erläutert. Anschließend wurden die verschiedenen Verschlüsselungssoftwares analysiert und verglichen.

Dabei wurden zahlreiche Gemeinsamkeiten festgestellt. Im Allgemeinen ist die verwendete Schlüsselhierarchie identisch. Die Festplatte wird über einen Master-Key gesichert. Dieser ist auf der Festplatte abgelegt und verschlüsselt. Die Verschlüsselung findet über einen Benutzerschlüssel statt. Dieser wird in den meisten Fällen mit einem Hardwareschlüssel kombiniert, um zusätzliche Sicherheit zu bieten. Ein weiterer Vorteil eines Hardwareschlüssels, sofern dieser nicht extrahiert werden kann, ist, dass Angriffe an das System gebunden werden.
Zudem verwendeten alle AES als Verschlüsselungsalgorithmus. Dieser gilt als sicher und ist auch aufgrund seiner Geschwindigkeit weitverbreitet [73].

Abweichung gab es in der eingesetzten Hardware. LUKS ist die einzige Software, welche keine kryptographische Hardwarekomponente voraussetzt. Die Sicherheit hierbei wird ausschließlich über den Benutzerschlüssel gewährleistet.

Anschließend wurde eine forensische Analyse auf die Verschlüsselungssoftwares durchgeführt. Das Hauptziel dieser Analysen bestand darin, Methoden zu identifizieren, um die Verschlüsselung zu umgehen.

Im Zuge dieser Untersuchung wurden wirksame Angriffsmethoden auf die TPM, welche von BitLocker verwendet wird, gefunden. Diese Angriffe ermöglichen die Extraktion des Hardwareschlüssel. Sollte BitLocker die Festplatte ausschließlich durch den Hardwareschlüssel sichern, so kann die Verschlüsselung mit solch einer Angriffsmethode umgangen werden. Wenn BitLocker die TPM in Verbindung mit einem Benutzerpasswort verwendet, ermöglicht dies zumindest die Auslagerung eines Brute-Force-Angriffs auf ein alternatives System.

Es wurde zudem der Brute-Force-Angriff auf LUKS analysiert. Dabei stellte sich heraus, dass ein Angriff, der auf die Nutzung von GPU-Beschleunigung und die Verteilung über verschiedene Systeme setzt, eine valide Möglichkeit darstellt, die Verschlüsselung zu umgehen. Allerdings wurde diese Methode durch die Einführung von Argon2 in LUKS2 obsolet, da die neue KDF die Verwendung von Grafikkarten bei einem Brute-Force-Angriff beschränkt.

Weiterhin wurden keine erfolgreichen Angriffsmethoden auf die TEE von Android oder der Secure Enclave identifiziert. Somit erwies sich die forensische Analyse bei Macs, iPhones und Android-Smartphones als schwierig. Diese Systeme gelten als sicher und es gibt keine valide forensische Untersuchungsmethode bei verschlüsselten Festplatten mit starkem Passwort.

7.2 Bewertung der Arbeit

Im Rahmen der Bachelorarbeit ist es gelungen eine ausführliche Analyse der zu untersuchenden Festplattenverschlüsselungssoftwares durchzuführen. Des Weiteren konnten die Grundlagen der Verschlüsselungstechnik erläutert und ein Vergleich der Festplattenverschlüsselung der unterschiedlichen Betriebssysteme kreiert werden.

Dies wurde mithilfe verschiedener wissenschaftlicher Arbeiten ausgewählt und hinterher vorgestellt. Anschließend konnte darüber ein tabellarischer Vergleich erstellt und eine forensische Analyse beschrieben werden.

Zusammenfassend lässt sich sagen, dass die Verschlüsselungstechnologie aufgrund von kryptographischer Hardware und Algorithmen sicher ist und aufgrund von fehlendem Leistungsverlust als Standard in jedem mobilen Endgerät Verwendung finden kann.

7.3 Ausblick

Microsofts BitLocker zeigt unter den analysierten Verschlüsselungssoftwares die größten Schwachstellen. Der Grund dafür ist die Umgehung der TPM auf dem die Software beruht. Als Standard wird die TPM-Only-Variante eingerichtet. Diese bietet allerdings einen falschen Eindruck von Sicherheit. Microsoft sollte bei der Einrichtung des Systems dem Benutzer die Möglichkeit geben, über die verwendete TPM-Variante zu entscheiden.

Zudem sollte der Benutzer darauf aufmerksam gemacht werden, welche Bedeutung die jeweilige Verschlüsselungsvariante unter Bitlocker hat.

Die Schwachstelle von BitLocker liegt größtenteils an der Sicherheitslücke der TPM. Die Angriffsmethode darauf wurde im Jahre 2023 veröffentlicht. Davor galt die TPM, sowie alle aktuellen kryptographischen Hardwarekomponenten als sicher. Da 2020 eine angebliche Schwachstelle in Apples Secure Enclave [59] und 2016 eine Sicherheitslücke in Androids TEE gefunden wurde [74], ist es nicht unrealistisch anzunehmen, dass auch die Sicherheit anderer Hardwarekomponenten in absehbarer Zukunft umgangen werden könnte.

Ein weiterer Nachteil von BitLocker ist, dass die Festplattenentschlüsselung ein eigenes Benutzerpasswort unabhängig von dem des Betriebssystems nutzt. Nach der Authentifizierung in BitLocker läuft Windows unter Normalbedingungen. Wird das System in den Ruhezustand versetzt, bleibt es anschließend immer noch entschlüsselt. Im Gegensatz dazu verwendet die Festplattenverschlüsselung bei Android, iOS und MacOS kein dediziertes Benutzerpasswort. Anstelle dessen wird das Kennwort für die Bildschirmsperre genutzt. Wird das System in den Energiesparmodus versetzt, wird der Verschlüsselungsschlüssel aus dem Arbeitsspeicher gelöscht. Bei der Reaktivierung des Gerätes fungiert dann das Kennwort als Benutzerpasswort für die Entschlüsselung.

Literatur

[1] publisher, „Arten der Verschlüsselung“, *Bundesamt für Sicherheit in der Informationstechnik*. https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Onlinekommunikation/Verschluesselt-kommunizieren/Arten-der-Verschluesselung/arten-der-verschluesselung.html?nn=131790 (zugegriffen 21. September 2023).

[2] „What is the RSA algorithm? Definition from SearchSecurity“, *Security*. https://www.techtarget.com/searchsecurity/definition/RSA (zugegriffen 21. September 2023).

[3] „Difference Between AES and RSA Encryption“, *GeeksforGeeks*, 13. Mai 2023. https://www.geeksforgeeks.org/difference-between-aes-and-rsa-encryption/ (zugegriffen 21. September 2023).

[4] M. J. Dworkin, „Advanced Encryption Standard (AES)“, National Institute of Standards and Technology, Gaithersburg, MD, NIST FIPS 197-upd1, 2023. doi: 10.6028/NIST.FIPS.197-upd1.

[5] „RSA Verschlüsselung: Einfach erklärt mit Beispiel“, *Studyflix*. https://studyflix.de/informatik/rsa-verschlusselung-1608 (zugegriffen 23. September 2023).

[6] D. Suarez und D. Mayer, „Faux disk encryption: realities of secure storage on mobile devices“, in *Proceedings of the International Conference on Mobile Software Engineering and Systems*, Austin Texas: ACM, Mai 2016, S. 283–284. doi: 10.1145/2897073.2897711.

[7] S. Almuhammadi und I. Al-Hejri, „A comparative analysis of AES common modes of operation“, in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Apr. 2017, S. 1–4. doi: 10.1109/CCECE.2017.7946655.

[8] M. Witkowski, „Verschlüsselungsmodus im Detail / Empfehlung“, 26. Dezember 2013. https://itsecblog.de/verschluesselungsmodus-im-detail-empfehlung/ (zugegriffen 21. September 2023).

[9] „Der AES-XTS-Blockverschlüsselungsmodus wird in den am besten verschlüsselten USB-Sticks von Kingston verwendet - Kingston Technology“, *Kingston Technology Company*. https://www.kingston.com/de/blog/data-security/xts-encryption (zugegriffen 21. September 2023).

[10] K. Lipinski, „Initialisierungsvektor“, *ITWissen.info*. https://www.itwissen.info/Initialisierungsvektor-initialization-vector-security-IV.html (zugegriffen 21. September 2023).

[11] „Initialization vector | Initialisierungsvektor | IV“, 8. April 2009. https://www.storage-insider.de/initialization-vector-initialisierungsvektor-iv-a-181900/ (zugegriffen 21. September 2023).

[12] P. Schmitz und S. Luber, „Was ist Festplattenverschlüsselung?“, 11. März 2019. https://www.security-insider.de/was-ist-festplattenverschluesselung-a-808249/ (zugegriffen 21. September 2023).

[13] J. A. Halderman *u. a.*, „Lest we remember: cold-boot attacks on encryption keys“, *Commun. ACM*, Bd. 52, Nr. 5, S. 91–98, Mai 2009, doi: 10.1145/1506409.1506429.

[14] „Verschlüsselung mit Software & Hardware“, *Bundesamt für Sicherheit in der Informationstechnik*. https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Cyber-Sicherheitsempfehlungen/Daten-sichern-verschluesseln-und-loeschen/Datenverschluesselung/Soft-und-hardwaregestuetzte-Verschluesselung/soft-und-hardwaregestuetzte-verschluesselung.html?nn=131264 (zugegriffen 21. September 2023).

[15] S. Bossi und A. Visconti, „What users should know about Full Disk Encryption based on LUKS“. 2016. Zugegriffen: 21. September 2023. [Online]. Verfügbar unter: https://eprint.iacr.org/2016/274

[16] „File-Based Encryption“, *Android Open Source Project*. https://source.android.com/docs/security/features/encryption/file-based (zugegriffen 22. September 2023).

[17] „File-based encryption vs full-disk encryption“, *Hexnode Blogs*, 24. Januar 2022. https://www.hexnode.com/blogs/file-based-encryption-vs-full-disk-encryption/ (zugegriffen 23. September 2023).

[18] „dys2p“. https://dys2p.com/de/2023-05-luks-security.html#kdf (zugegriffen 22. September 2023).

[19] M. Reiter und D. Naccache, Hrsg., *Cryptology and Network Security: 14th International Conference, CANS 2015, Marrakesh, Morocco, December 10-12, 2015, Proceedings*, Bd. 9476. in Lecture Notes in Computer Science, vol. 9476. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-26823-1.

[20] „Argon2“. https://cryptobook.nakov.com/mac-and-key-derivation/argon2 (zugegriffen 22. September 2023).

[21] heise online, „Argon2 gewinnt Passwort-Hashing-Wettbewerb“, *Security*, 27. Juli 2015. https://www.heise.de/news/Argon2-gewinnt-Passwort-Hashing-Wettbewerb-2763389.html (zugegriffen 22. September 2023).

[22] „phc-winner-argon2/argon2-specs.pdf at master · P-H-C/phc-winner-argon2“, *GitHub*. https://github.com/P-H-C/phc-winner-argon2/blob/master/argon2-specs.pdf (zugegriffen 22. September 2023).

[23] „scrypt“, *Wikipedia*. 29. Juni 2023. Zugegriffen: 22. September 2023. [Online]. Verfügbar unter: https://de.wikipedia.org/w/index.php?title=Scrypt&oldid=235040211

[24] J.-D. Kranz, „Passwörter: Entropie berechnen“, *IT-Talents.de*, 23. Januar 2014. https://it-talents.de/it-wissen/passworter-entropie-berechnen/ (zugegriffen 23. September 2023).

[25] A. Czernik, „Hashwerte und Hashfunktionen einfach erklärt“, *Dr. Datenschutz*, 2. September 2016. https://www.dr-datenschutz.de/hashwerte-und-hashfunktionen-einfach-erklaert/ (zugegriffen 23. September 2023).

[26] „HMAC“, *Wikipedia*. 10. September 2023. Zugegriffen: 23. September 2023. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=HMAC&oldid=1174791885

[27] N. Pohlmann und devnpo, „Keyed-Hashing for Message Authentication - Prof. Pohlmann“, *Prof. Dr. Norbert Pohlmann*, 28. September 2019. https://norbert-pohlmann.com/glossar-cyber-sicherheit/keyed-hashing-for-message-authentication-hmac/ (zugegriffen 23. September 2023).

[28] „About TCG“, *Trusted Computing Group*. https://trustedcomputinggroup.org/about/ (zugegriffen 21. September 2023).

[29] A. Tomlinson, „Introduction to the TPM“, in *Smart Cards, Tokens, Security and Applications*, K. E. Mayes und K. Markantonakis, Hrsg., Boston, MA: Springer US, 2008, S. 155–172. doi: 10.1007/978-0-387-72198-9_7.

[30] H. N. Jacob, C. Werling, R. Buhren, und J.-P. Seifert, *faulTPM: Exposing AMD fTPMs' Deepest Secrets*. 2023.

[31] G. Proudler, L. Chen, und C. Dalton, *Trusted Computing Platforms: TPM2.0 in Context*. Cham: Springer International Publishing, 2014.

[32] „Was ist ein Logic Analyzer?“ https://evision-webshop.de/blog/was-ist-ein-logic-analyzer (zugegriffen 21. September 2023).

[33] „PSPTool“. PSPReverse, 19. September 2023. Zugegriffen: 21. September 2023. [Online]. Verfügbar unter: https://github.com/PSPReverse/PSPTool

[34] ~ Hucktech, „PSPTool: Display, extract and manipulate AMD PSP UEFI firmware“, *Firmware Security*, 30. Mai 2019. https://firmwaresecurity.com/2019/05/30/psptool-display-extract-and-manipulate-amd-psp-uefi-firmware/ (zugegriffen 21. September 2023).

[35] „Brute-Force-Attacke: Definition und Funktionsweise | Myra“, 16. Januar 2023. https://www.myrasecurity.com/de/brute-force-attacke/ (zugegriffen 24. September 2023).

[36] „Breaking LUKS Encryption“, *ElcomSoft blog*, 18. August 2020. https://blog.elcomsoft.com/2020/08/breaking-luks-encryption/ (zugegriffen 22. September 2023).

[37] „Brute-force attack“, *Wikipedia*. 20. September 2023. Zugegriffen: 24. September 2023. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=Brute-force_attack&oldid=1176217027

[38] „Understanding Rainbow Table Attack“, *GeeksforGeeks*, 10. Juni 2018. https://www.geeksforgeeks.org/understanding-rainbow-table-attack/ (zugegriffen 24. September 2023).

[39] H. May und R. Kienzler, „Grenzen von MOS-Speicherzellen“, *Arch. Für Elektrotechnik*, Bd. 63, Nr. 6, S. 327–336, Nov. 1981, doi: 10.1007/BF01574226.

[40] „memory-encryption-white-paper.pdf“. Zugegriffen: 22. September 2023. [Online]. Verfügbar unter: https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/memory-encryption-white-paper.pdf

[41] P. McGregor, T. Hollebeek, A. Volynkin, und M. White, „Braving the Cold: New Methods for Preventing Cold Boot Attacks on Encryption Keys“, 2008.

[42] „Was ist BitLocker? - Definition von WhatIs.com“, *ComputerWeekly.de*. https://www.computerweekly.com/de/definition/BitLocker (zugegriffen 22. September 2023).

[43] paolomatarazzo, „Übersicht über die BitLocker-Geräteverschlüsselung in Windows - Windows Security“, 6. Juni 2023. https://learn.microsoft.com/de-de/windows/security/operating-system-security/data-protection/bitlocker/bitlocker-device-encryption-overview-windows-10 (zugegriffen 22. September 2023).

[44] „Bitlocker – IT-Forensik Wiki“. https://it-forensik.fiw.hs-wismar.de/index.php/Bitlocker (zugegriffen 22. September 2023).

[45] „FileVault“, *Wikipedia*. 4. August 2023. Zugegriffen: 22. September 2023. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=FileVault&oldid=1168711512

[46] „Apple FileVault 2: Full disk encryption software overview | TechTarget“, *Security*. https://www.techtarget.com/searchsecurity/feature/Apple-FileVault-2-Full-disk-encryption-software-overview (zugegriffen 22. September 2023).

[47] „Volumeverschlüsselung mit FileVault bei macOS“, *Apple Support*. https://support.apple.com/de-de/guide/security/sec4c6dc1b6e/web (zugegriffen 22. September 2023).

[48] D. Sladović, D. Topolčić, und D. Delija, „Overview of Mac system security and its impact on digital forensics process“, in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, Sep. 2020, S. 1236–1241. doi: 10.23919/MIPRO48935.2020.9245397.

[49] hoakley, „Explainer: xART and nonces“, *The Eclectic Light Company*, 3. Juli 2021. https://eclecticlight.co/2021/07/03/explainer-xart-and-nonces/ (zugegriffen 22. September 2023).

[50] „Why Your Default iPhone Encryption Isn’t Enough“. https://www.virtru.com/blog/iphone-encryption (zugegriffen 22. September 2023).

[51] „Secure Enclave“, *Apple Support*. https://support.apple.com/de-de/guide/security/sec59b0b31ff/web (zugegriffen 22. September 2023).

[52] „Full-Disk Encryption“, *Android Open Source Project*. https://source.android.com/docs/security/features/encryption/full-disk (zugegriffen 22. September 2023).

[53] Aorimn, „Dislocker“. 22. September 2023. Zugegriffen: 22. September 2023. [Online]. Verfügbar unter: https://github.com/Aorimn/dislocker

[54] „Anleitung zur Verwendung von BitLocker mit PIN | Dell Deutschland“. https://www.dell.com/support/kbdoc/de-de/000142382/anleitung-zur-verwendung-von-bitlocker-mit-pin (zugegriffen 22. September 2023).

[55] B. Lovejoy, „T2 Mac security vulnerability: Passwords can now be cracked“, *9to5Mac*, 17. Februar 2022. https://9to5mac.com/2022/02/17/t2-mac-security-vulnerability-password/ (zugegriffen 22. September 2023).

[56]„Verschüsselte Volumes von LUKS1 auf LUKS2 konvertieren | [Mer]Curius“, 28. Januar 2021. https://curius.de/2021/01/verschuesselte-volumes-von-luks1-auf-luks2-konvertieren/ (zugegriffen 22. September 2023).

[57]„Probing Linux Disk Encryption: LUKS2, Argon 2 and GPU Acceleration“, *ElcomSoft blog*, 16. August 2022. https://blog.elcomsoft.com/2022/08/probing-linux-disk-encryption-luks2-argon-2-and-gpu-acceleration/ (zugegriffen 22. September 2023).

[58]M. Moreb, „Introduction to iOS Forensics“, in *Practical Forensic Analysis of Artifacts on iOS and Android Devices: Investigating Complex Mobile Devices*, M. Moreb, Hrsg., Berkeley, CA: Apress, 2022, S. 37–70. doi: 10.1007/978-1-4842-8026-3_2.

[59] heise online, „Secure Enclave im iPhone angeblich ‚unpatchbar‘ geknackt“, *Mac & i*, 4. August 2020. https://www.heise.de/news/Secure-Enclave-im-iPhone-angeblich-unpatchbar-geknackt-4862041.html (zugegriffen 22. September 2023).

[60]M. Moreb, „Introduction to Android Forensics“, in *Practical Forensic Analysis of Artifacts on iOS and Android Devices: Investigating Complex Mobile Devices*, M. Moreb, Hrsg., Berkeley, CA: Apress, 2022, S. 71–108. doi: 10.1007/978-1-4842-8026-3_3.

[61]„Sicherheitsmechanismen der Android-Plattform“, *Sicherheitsmechanismen der Android-Plattform*. https://movi.fokus.fraunhofer.de/androidSecurityFeatures/ (zugegriffen 24. September 2023).

[62]T. Müller und M. Spreitzenbarth, „FROST: Forensic Recovery of Scrambled Telephones“, in *Applied Cryptography and Network Security*, M. Jacobson, M. Locasto, P. Mohassel, und R. Safavi-Naini, Hrsg., in Lecture Notes in Computer Science, vol. 7954. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 373–388. doi: 10.1007/978-3-642-38980-1_23.

[63]R. Bali, „Cold Boot Attack On Cell Phones, Cryptographic Attacks“, Juli 2018. doi: 10.13140/RG.2.2.13560.14088.

[64]„Geräteverschlüsselung in Windows - Microsoft-Support“. https://support.microsoft.com/de-de/windows/ger%C3%A4teverschl%C3%BCsselung-in-windows-ad5dcf4b-dbe0-2331-228f-7925c2a3012d (zugegriffen 23. September 2023).

[65]„LUKS – Betriebssystem verschlüsseln | [Mer]Curius“. https://curius.de/verschluesselung-eine-uebersicht/luks-betriebssystem-verschluesseln/ (zugegriffen 23. September 2023).

[66]„Wenn du deinen iPhone-Code vergessen hast“, *Apple Support*, 20. Juni 2023. https://support.apple.com/de-de/HT204306 (zugegriffen 23. September 2023).

[67] heise online, „Android: Passwort vergessen“, *Tipps-Tricks*, 8. Mai 2020. https://www.heise.de/tipps-tricks/Android-Passwort-vergessen-4013228.html (zugegriffen 23. September 2023).

[68]„Bitlocker: Einfluss auf die Platten-Leistung | faq-o-matic.net“. https://www.faq-o-matic.net/2016/08/29/bitlocker-einfluss-auf-die-platten-leistung/ (zugegriffen 23. September 2023).

[69]Ujjwal, „Do I Really Need FileVault? (And Does It Affect Performance)“, *MacMyths*, 23. Mai 2019. https://macmyths.com/do-i-really-need-filevault/ (zugegriffen 23. September 2023).

[70]„Performance impact of disk encryption using LUKS | Sovereign Cloud Stack“. https://scs.community/2023/02/24/impact-of-disk-encryption/ (zugegriffen 23. September 2023).

[71]B. C. Ho Joshua, „Encryption and Storage Performance in Android 5.0 Lollipop“. https://www.anandtech.com/show/8725/encryption-and-storage-performance-in-android-50-lollipop (zugegriffen 23. September 2023).

[72]H. Brecher, „Der Intel Core i7-11700K landet hinter dem Apple M1 in der PassMark-Bestenliste“, *Notebookcheck*, 25. März 2021. https://www.notebookcheck.com/Der-Intel-Core-i7-11700K-landet-hinter-dem-Apple-M1-in-der-PassMark-Bestenliste.529225.0.html (zugegriffen 23. September 2023).

[73]„AES-256 (Glossar IT-Sicherheit)“, *Materna Virtual Solution*. https://www.virtual-solution.com/glossar/aes-256/ (zugegriffen 24. September 2023).

[74] heise online, „Heftiger Schlag für Android-Verschlüsselung“, *Security*, 4. Juli 2016. https://www.heise.de/news/Heftiger-Schlag-fuer-Android-Verschluesselung-3254136.html (zugegriffen 24. September 2023).

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 24.09.2023

David Eisele